

# Computation of Stochastic Nash Equilibrium via Variable Sample Distributed Methods

Dali Zhang

Sino-US Global Logistics Institute  
Shanghai Jiao Tong University, Shanghai, 200030, China

Lizhi Wang

Department of Industrial and Manufacturing Systems Engineering,  
Iowa State University, Ames, Iowa, 50011, USA

Ming Dong

Sino-US Global Logistics Institute,  
Antai College of Management and Economics,  
Shanghai Jiao Tong University, Shanghai, 200030, China

November 18, 2013

## Abstract

In this paper, we propose a variable sample distributed algorithm for the computation of stochastic Nash equilibrium in which the objective functions are replaced, at each iteration, by sample average approximations. We investigate the contraction mapping properties of the variable sample distributed algorithm and show that the accuracy of estimators yielded in the algorithms to their true counterparts are determined by both the sample size schedules and the contraction mapping parameters. We also investigate conditions on the sample size schedule under which the accumulation point generated by the algorithm asymptotically converges to the true Nash equilibrium. In the numerical tests, we comparatively analyze the accuracy and precision errors of estimators with different sample size schedules with respect to the sampling loads and the computational times. Finally, we present numerical results on the effectiveness of different cumulative sampling schemes for the algorithm.

**Key words.** Monte Carlo methods, distributed algorithm; stochastic Nash equilibrium; sample size schedule.

# 1 Introduction

Over the past decade, a considerable amount of research has been devoted to theoretical and practical aspects of combining simulation and optimization techniques to solve stochastic programming problem. This paper proposes a way to generalize this combination to the stochastic Nash equilibrium (SNE) problems where the objective functions in the game cannot be evaluated exactly but rather must be estimated by sample averaging or simulation.

*Distributed iterative scheme* (DIS) is important and highly relevant in the computation of Nash equilibria in noncooperative games. This is mainly because the scheme can be implemented in a real-world game situation where decision makers do not necessarily know each other's cost functionals and private information, and they only have to communicate to each other for their tentative decisions during each phase of computation [19]. One of the first algorithmic researches on DIS was explored by Papavassilopoulos [21] for quadratic adaptive Nash equilibrium problems. The first work related to one of the most important concepts in the algorithm, contraction mapping theorem, was developed by Bertsekas [3]. The algorithm was extended by Li and Basar [19] to a class of nonquadratic convex Nash games for the computation of the unique stable Nash equilibrium, and implemented by Bozma [6] into a parallel gradient descent framework for the computation of Nash equilibria. More recently, the DIS and parallel computation concept have been further broadened. Yuan [39] proposed a trust region framework for the iterations in the distributed algorithms for solving Nash equilibrium. The algorithm has also been applied to Stackelberg games by Zhang and Lin [42] where the decision makers distributively execute the algorithm at leader or follower positions.

Due to the intrinsic uncertainty in many real world problems, various SNE models have been proposed to study specific decision making problems which involve random data as well as multiple decision makers in a competitive relationship. For solving SNE problems, we face another challenging issue on how to estimate the equilibrium, or broadly stationary point, via a deterministic approximation. One approach to approximate the problems with expected objective functions is to resort to sample average approximation (SAA) or Monte Carlo sampling methods. The idea of the SAA is to replace the expected value function with its corresponding sample average and then solve the resulting deterministic problems. The SAA is a popular method in stochastic optimization, and it is also known as the Sample Path Optimization (SPO) method in [25]. There has been extensive literature [1, 8, 13, 20, 29, 31, 34] on the SAA and the SPO including a comprehensive review by Shapiro in [30]. To our best knowledge, recent progresses to SNE problems can be found in [32] and [35]. Vorobeychik [32] proposed a simulation-based approach for stochastic game theory with mixed strategy selection. Xu and Zhang [35] investigated several convergence properties on the Monte Carlo sampling approximation to general stochastic game theoretic models essentially by using the strong law of large numbers to underlying sample averaged first order equations which define the statistical estimator.

When applying iterative schemes, including pure search algorithm, gradient-based algorithm,

trust region method, direct search method, etc., to solve SAA approximation of stochastic programming problems, how to allocate the sample resource into different iterations of the algorithms becomes a key point influencing the efficiency of the algorithm. This issue has been widely studied for stochastic ranking and selection problems [5, 16, 17], stochastic constrained optimizations [14, 22], and stochastic root finding problems [23, 24]. For the ranking and selecting problems, Lee et al. [17] developed an optimal computing budget allocation (OCBA) approach and obtain an asymptotic closed-form allocation rule of correctly selection with a fixed sample budget. More recently, this method was extended in Brantley et al. [5] to achieve an efficient simulation budget allocation with regression. This type of methods were also used for solving stochastic constrained optimization problem in Hunter and Pasupathy [14] which characterizes the asymptotically optimal sample allocation with the minimization of the rate at which the probability of false selection tends to zero. Moreover, in Pasupathy [23], a sample-path technique, retrospective-approximation, was proposed for solving stochastic root finding problems by sequentially solving sample average approximations. This retrospective-approximation technique characterizes a class of error-tolerance and sample-size sequences optimally ensuring the convergence of the sequence of obtained SAA solutions. Recently, optimal budget allocation analysis was further developed in Royset and Szechtman [27] for solving SAA estimators via an iterative optimization algorithm and in Homem-de-Mello and Bayraksan [12] for balancing the effort between optimization and estimation in a Monte Carlo sampling-based method for stochastic optimization.

One of the main contributions of this paper is that we propose a variable sample distributed algorithm (VSDA) for the computation of SNE with the idea of integrating the Monte Carlo sampling approximation and the DIS. By doing so, the computation of the SNE can be extended to a broader and more empirical range where decision makers do not necessarily know or analytically have each other's cost functionals. Another merit of the method is the design of the sample size, where the algorithm's requirement on the finite growth rate of the samples size used at each iteration makes the application of the SAA method practical. As a result, how sample size is updated at each iteration turns out to be critical for achieving an unbiased convergence.

This type of variable sample schemes was explored by Homem-de-Mello [11] for stochastic optimization problems. In [11], the integration of the method to a pure random search algorithm presented two advantages: the first is that independent samples can be generated at each iteration which avoids getting biased in a single sample-path; the other is that the sample size can increase along with the algorithm which avoids the waste of sampling effort at the initial iterations. In this paper, we extend the framework in [11] to solve SNE problems. The most different point is that random search methods in [11] are performed under a stochastic comparison process with finite feasible points, where the gradient information are excluded. To our best knowledge, the VSDA frameworks for general stochastic optimization problem have been further extended. Deng and Ferris [10] and Kim and Zhang [18] extended the underlying iterative algorithms for variable sample schemes by using the trust region method and the direct search method respectively. Until now, how to use the variable sample scheme for solving SNE

still remains unsolved.

The rest of this paper is organized as follows. In the next section, some fundamental knowledge and preliminary results on the contraction mapping and distributed algorithms are presented for SNE problems. In Section 3, we apply the Monte Carlo sampling methods for the distributed algorithm with fixed sample size at iteration. The results show that the error bounds on the iterates yielded by the VSDA to its true counterparts are influenced by both the sample size and the contraction mapping parameter. In Section 4, we integrate the variable sample scheme to the distributed algorithm for solving a SNE problem, and establish a set of conditions on the sample size schedule which guarantees the convergence of the algorithms and the finite stopping of the algorithm. Moreover, in this section, we also present a numerical example to show that the well-known “bucket effect” also exists in solving SNE. Finally, in Section 5, we present some numerical results to illustrate the performance of variable sample schemes in saving the sampling and cumulative sampling effort and enhancing accuracy.

## 2 Stochastic Nash Equilibrium and Distributed Algorithm

In this section, we review the distributed iterative algorithm and relevant concepts for the computation of Nash equilibrium. To this end, we consider a game problem where a set of decision makers compete in a non-cooperative manner.

### 2.1 Description

Consider an  $M$ -player noncooperative game where each player wishes to minimize his cost function. In the problem, decision maker (player)  $i$  chooses his optimal strategy  $x_i$  in a compact and convex subset  $X_i$  of  $\mathbb{R}^{m_i}$  where we index decision makers by  $i \in I = \{1, 2, \dots, M\}$ . In the SNE problem, each decision maker chooses his strategy to minimize his expected cost function before knowing information of the realization  $\xi$ , which is

$$F_i(x_i, x_{-i}) := \mathbb{E} [f_i(x_i, x_{-i}, \xi(\omega))], \quad \forall i \in I. \quad (2.1)$$

Here we use a random variable  $\xi(\omega)$  to characterize these uncertain factors, where  $\xi : \Omega \rightarrow \Xi \subset \mathbb{R}^d$  is continuous with a known or unknown distribution. We define decision makers’ cost functions as  $f(x, \xi) := (f_1(x_1, x_{-1}, \xi), \dots, f_M(x_M, x_{-M}, \xi))$  to emphasize the dependence of decision makers’ costs on  $\xi$  which is the realization of  $\xi(\omega)$ . To ease the notation, we will write  $\xi(\omega)$  as  $\xi$ , and the context will make it clear when  $\xi$  should be interpreted as a deterministic vector. Moreover  $\mathbb{E}[\cdot]$  denotes the mathematical expectation with respect to the distribution of  $\xi$ , and  $x_{-i} := (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_M) \in X_{-i}$  where we define the joint feasible sets of decision maker  $i$ ’s rivals by  $X_{-i} := X_1 \times \dots \times X_{i-1} \times X_{i+1} \times \dots \times X_M$  and all decision makers by  $X = X_1 \times X_2 \times \dots \times X_M$  respectively. In this paper we focus our research on the algorithm aspects and investigate the problems with bounded feasible set  $X_i$  for  $i \in I$ .

We denote this SNE problem with expected objective functions by  $\Gamma = (I, X, F)$ . In the triplet,  $F := (F_1(x_1, x_{-1}), \dots, F_M(x_M, x_{-M}))$  is a vector-form of the decision makers' cost functions, which maps joint strategy  $(x_i, x_{-i}) \in X_i \times X_{-i}$  to a vector of the cost entries for all decision makers for  $i \in I$ . The SNE model  $\Gamma$  can be stated as: find  $x := (\tilde{x}_1, \dots, \tilde{x}_M) \in X$  such that for  $i \in I$ ,  $\tilde{x}_i$  solves the following problem:

$$\min_{x_i \in X_i} F_i(x_i, \tilde{x}_{-i}) = \mathbb{E}[f_i(x_i, \tilde{x}_{-i}, \xi)]. \quad (2.2)$$

To emphasize the computation process of the SNE, we focus our investigation on the cases with  $f_i : X_i \times X_{-i} \times \Xi \rightarrow \mathbb{R}$  being Lipschitz continuous and convex with respect to  $x_i \in X_i$  for  $i \in I$ .

## 2.2 The deterministic distributed algorithm

Let us first consider the case where the distribution function of  $\xi$  is known and the expected functions  $F_i(\cdot)$ , for  $i \in I$ , can be integrated out. In this case  $\Gamma$  can be treated as a deterministic problem. Here we refer  $\Gamma$  as the *true problem*. The distributed iterative algorithm has been proposed in [19] for solving it. We outline the algorithm as follows.

---

### Algorithm 1: Distributed Algorithm for Problem $\Gamma$ .

---

**Inputs:** Initial point  $x^0 := (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_M)$  and error tolerance  $\epsilon > 0$ .

**Outputs:** The terminal iteration  $K$ , the solution  $x^K := (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_M)$  and the values of objective functions  $F^K := (F_1(\tilde{x}_1, \tilde{x}_{-1}), \dots, F_M(\tilde{x}_M, \tilde{x}_{-M}))$ .

**Initialization:** Let  $k = 0$ . Set an initial point  $x[k] = x^0$  where  $x[k] := (x_1[k], \dots, x_M[k])$  and  $x_i[k] = \hat{x}_i$  for  $i \in I$ .

**Step 1:** Set iteration  $k := k + 1$ . For each decision maker  $i$ ,  $i \in I$ , at iteration  $k + 1$ , find  $x_i[k + 1]$  by solving the following optimization problem:

$$\min_{x_i \in X_i} F_i(x_i, x_{-i}[k]), \quad (2.3)$$

where  $x_{-i}[k] := (x_1[k], \dots, x_{i-1}[k], x_{i+1}[k], \dots, x_M[k])$ . Go to Step 2.

**Step 2:** Stopping criterion: If  $\|x[k + 1] - x[k]\|$  is sufficiently small (less than  $\epsilon$  and  $\|\cdot\|$  is the Euclidean norm) then terminate the algorithm, return  $x[k]$  and set the terminal iteration  $K = k$ . Otherwise, go to Step 1.

**Return:** The solution  $\tilde{x}_i = x_i[K]$  for  $i \in I$  ( $x^K := x[K]$  in **Outputs**) and the values of objective function

$$F^K := (F_1(\tilde{x}_1, \tilde{x}_{-1}), \dots, F_M(\tilde{x}_M, \tilde{x}_{-M})).$$


---

### 2.3 Stable Nash equilibrium

The algorithm essentially provides a DIS for solving a Nash equilibrium problem. Here, we can rewrite the sequence  $\{x[k], k = 1, 2, \dots\}$  yielded in each iteration of Algorithm 1 as the solutions to the following set of iterative equations,

$$x_i[k+1] = \mathcal{Q}_i(x_{-i}[k]), \quad (2.4)$$

where operator  $\mathcal{Q}_i$  can be specified as

$$\mathcal{Q}_i(x_{-i}) = \arg \min_{x_i \in X_i} \mathbb{E}[f_i(x_i, x_{-i}, \xi)].$$

We define  $\mathcal{Q} \in X \rightarrow X$  via concatenating  $\mathcal{Q}_i$  for  $i \in I$ , which is a continuous operator [4]. We refer the reader to [19] for the existence of  $\mathcal{Q}$ . Now let us introduce a concept of stable Nash equilibrium, which was first proposed by Li and Basar in [19].

**Definition 2.1 (Stable Nash Equilibrium)** *A Nash equilibrium solution is stable if it can be obtained as the limit point of Algorithm 1 as iteration  $K \rightarrow \infty$  or equally  $\epsilon \rightarrow 0$  for any initial point  $x^0$  in the feasible set  $X$ . It is locally stable if the convergence is valid for all initial points  $x^0 \in X$  in some  $\delta$ -neighborhood of the equilibrium solution as  $K \rightarrow \infty$  or equally  $\epsilon \rightarrow 0$ .*

From Definition 2.1, we can see that the concept of stable Nash equilibrium is directly defined based on Algorithm 1, and hence the existence and uniqueness of the (locally) stable Nash equilibrium essentially depends on the mechanism of Algorithm 1 and the structures of functions  $F_i(\cdot)$  in ( $\delta$ -neighborhood) feasible sets for  $i \in I$ .

### 2.4 Contraction property

Here, we present some properties of the stable Nash equilibrium and the conditions under which a Nash equilibrium exists and is stable. Before proceeding to the contraction property, we have the following two conditions on  $\Gamma$ .

**(A1)** Feasible set  $X_i \subset \mathbb{R}^{m_i}$  is a closed, convex, and bounded neighborhood of the union of all the iterates  $x_i[k]$  for  $k = 1, 2, \dots$  for all  $i \in I$ ;

**(A2)** Objective function  $f_i : X_i \times X_{-i} \times \Xi \rightarrow \mathbb{R}$  is continuously second-order differentiable and convex with respect to  $x_i \in X_i$  for all  $i \in I$ .

We first have the following property on the existence of Nash equilibrium  $\Gamma$ .

**Property 1.** *Under Conditions (A1~2), game  $\Gamma$  admits a Nash equilibrium.*

The property can be seen as a repetition of Theorem 5.1 in [6] and is directly from [26]. Note that here we put a stricter condition on the differentiability of  $f_i$  than ‘Lipschitz continuity’ in

Theorem 5.1 in [6], which also automatically satisfied the condition in Property 3 below. By doing so, we can focus our investigation on the computation of Nash equilibrium and avoid theoretical investigations like “to what extent the existence results can be generalized”. In this paper, we focus our investigation on  $\Gamma$  satisfying Conditions (A1~2).

Now let us focus on  $\Gamma$  satisfying Conditions (A1~2) and step further to the properties of mapping  $\mathcal{Q}$ .

Let  $x[k] := (x_1[k], x_2[k], \dots, x_M[k])$  be the sequence generated by Algorithm 1 at iteration  $k$ , where the elements of the sequences are defined as,

$$\begin{aligned} x_1[k+1] &= \arg \min_{x_1 \in X_1} \mathbb{E} [f_1(x_1, x_{-1}[k], \xi)], \\ x_2[k+1] &= \arg \min_{x_2 \in X_2} \mathbb{E} [f_2(x_2, x_{-2}[k], \xi)], \\ &\vdots \\ x_M[k+1] &= \arg \min_{x_M \in X_M} \mathbb{E} [f_M(x_M, x_{-M}[k], \xi)], \end{aligned} \tag{2.5}$$

for iterations  $k = 0, 1, \dots$ , and all  $x_i[0] \in X_i$  for  $i \in I$ . Then, we have following results on (2.5).

**Property 2.** (Lemma 5.1 in [6]) *Under Condition (A1~2), there exist continuously differentiable mappings  $g_i : X_{-i} \rightarrow X_i$  such that equations (2.5) can be equivalently expressed as*

$$\begin{aligned} x_1[k+1] &= g_1(x_{-1}[k]), \\ x_2[k+1] &= g_2(x_{-2}[k]), \\ &\vdots \\ x_M[k+1] &= g_M(x_{-M}[k]), \end{aligned} \tag{2.6}$$

for  $k = 0, 1, 2, \dots$ , and all  $x_i[0] \in X_i$  with  $i \in I$ .

The proof of the above property for the case when  $f_i$  is continuously differentiable has been given in Lemma 5.1 in [6]. Note also that Property 2 can be extended, that is, when  $f_i(x_i, x_{-i}, \xi)$  is locally Lipschitz continuous, we have that  $g_i$  is locally Lipschitz continuous mapping for every  $i \in I$ . This can be proved from the Clarke generalized implicit function [7].

We can rewrite (2.6) as the following formulation,

$$x[k+1] = h_p(x[k]),$$

where  $h_p : X \rightarrow X$  is defined as

$$h_p(x) = \begin{bmatrix} g_1(x_{-1}) \\ g_2(x_{-2}) \\ \vdots \\ g_M(x_{-M}) \end{bmatrix}. \tag{2.7}$$

Here  $h_p(x)$  is continuously differentiable or Lipschitz continuous mappings when  $g_i$  is continuously differentiable or Lipschitz continuous for every  $i \in I$ . Now we specify the conditions under which the fixed point of algorithm,  $x[k+1] = h_p(x[k])$ , and hence the contraction property of  $h_p(\cdot)$  is guaranteed. The conditions were first proposed in Proposition 5.1 in [6], and we summarize them as the following property.

**Property 3.** *Suppose that for  $i \in I$ ,  $f_i$  is bounded and continuously second-order differentiable in  $\mathbb{R}^{m_i}$ , and locally strongly convex in  $X_i \subset \mathbb{R}^{m_i}$  for all  $x_j \in X_j$ ,  $j \neq i$ ; let  $\alpha_p$  be defined as*

$$\alpha_p = \|Dh_p\| = \left\| \begin{array}{ccc} \nabla_{x_1} g_1 & \cdots & \nabla_{x_M} g_1 \\ \vdots & \ddots & \vdots \\ \nabla_{x_1} g_M & \cdots & \nabla_{x_M} g_M \end{array} \right\|. \quad (2.8)$$

If  $0 < \alpha_p < 1$ , then a Nash equilibrium point in  $X$  is stable and there exists a Lipschitz modulus  $\kappa_h \in (0, 1)$  such that,

$$\|h_p(x) - h_p(x')\| \leq \kappa_h \|x - x'\| \quad (2.9)$$

for any  $x$  and  $x'$  in  $X$ .

(A3) Throughout this paper, we assume in the SNE problems that  $\alpha_p \in (0, 1)$ .

### 3 Variable Sample Distributed Algorithm

In most real-world applications, due to complexity from the randomness or the unavailability of the distribution functions, it is difficult to obtain the closed form of expected objective functions. This motivates the methods of incorporating the Monte Carlo sampling scheme for solving the SNE problems.

#### 3.1 Sample average approximation

First, let us consider a problem of applying the SAA framework directly to the DIS for solving the SNE where the close form of  $\mathbb{E}[f_i(x_i, x_{-i}, \xi)]$  in Algorithm 1 is not explicitly obtainable. Instead of integrating out the expected objective function in  $\Gamma$ , the basic idea of the SAA method is to generate an independently and identically distributed (i.i.d.) sample  $\{\xi^n, n = 1, 2, \dots, N\}$  of  $\xi$ , and then approximate expected values with the sample average. In this context, the objective functions in  $\Gamma$ , or equivalently (2.2), for decision makers  $i \in I$  are approximated by

$$F_i^N(x_i, x_{-i}) := \frac{1}{N} \sum_{n=1}^N f_i(x_i, x_{-i}, \xi^n). \quad (3.10)$$

Let us review the SAA Nash equilibrium proposed by Xu and Zhang in [35]. Without the knowledge of the expected objective functions or distribution of  $\xi$ , we investigate the following

Monte Carlo sampling approximation of the true Nash equilibrium problem  $\Gamma$ : find  $\hat{x}^N := (\hat{x}_i^N, \hat{x}_{-i}^N) \in X$  such that for  $i \in I$ ,  $\hat{x}_i^N$  solves the following problem:

$$\min_{x_i \in X_i} F_i^N(x_i, \hat{x}_{-i}^N) := \frac{1}{N} \sum_{n=1}^N f_i(x_i, \hat{x}_{-i}^N, \xi^n). \quad (3.11)$$

In [35], problem (3.11) is called *sample averaged Nash equilibrium problem* of the true problem  $\Gamma$ . Moreover, the solution to problem (3.11) is called *sample average approximation of the Nash equilibrium in true problem*  $\Gamma$  or shortly as “SAA Nash equilibrium of  $\Gamma$ ”. Note that for any finite sample size  $N$ ,  $F_i^N(x_i, \hat{x}_{-i}^N)$  is the arithmetic average of  $N$  values of  $f_i(x_i, x_{-i}, \xi^n)$  for  $n = 1, 2, \dots, N$  with  $\xi^n$  sampled from  $\xi(\omega)$ . The sampling result  $\xi^n$  and hence  $f_i(x_i, x_{-i}, \xi^n)$  are also random variable (function) of  $\omega$  for  $i \in I$ .

It is straightforward to see that we have two ways to incorporate the SAA framework: one way is to generate the sample before Algorithm 1. By doing so, we have the deterministic objective function  $F_i^N(x_i, \hat{x}_{-i}^N) = \frac{1}{N} \sum_{n=1}^N f_i(x_i, \hat{x}_{-i}^N, \xi^n)$  with the sample  $\{\xi^n\}_{n=1}^N$  being realized and fixed in the algorithm. However, generating the sample before running Algorithm 1 will cause a dilemma for the computation. On the one hand, Algorithm 1 cannot guarantee a non-biased approximation of the true SNE, since the samples are identical in each iteration of Algorithm 1 with finite size. On the other hand, if we increase the sample size to infinity, the sampling load will be accordingly infinite and prevent the execution of Algorithm 1. Motivated by the above fact, we propose an alternative way to incorporate the SAA framework for the computation of SNEs, where the variable sample will be used. We name the second way as *variable sample distributed algorithm* (VSDA). Before proceeding to the details in the VSDA, we first look at some properties of SAA Nash equilibrium of  $\Gamma$ .

### 3.2 Contraction property in SAA

We start to look at the contraction property of SAA Nash equilibrium problem instead the investigation of the stationary condition in [35]. First, we define operator  $\mathcal{Q}_i^N$  in the SAA of SNE problems as follows,

$$\mathcal{Q}_i^N(x_{-i}) = \arg \min_{x_i \in X_i} \frac{1}{N} \sum_{n=1}^N f_i(x_i, x_{-i}, \xi^n),$$

for  $i \in I$ . Under Conditions (A1~2), given the structure of  $f_i$  and the continuous distribution of  $\xi$ , there exist continuously differentiable mappings  $g_{i,N}$  for  $i \in I$  such that

$$x[k+1] = h_p^N(x[k]),$$

with its  $i$ th element being  $x_i[k+1] = g_{i,N}(x_{-i}[k])$ . Here,  $h_p^N : X \rightarrow X$  is defined as  $h_p^N(x) = (g_{1,N}^\top(x_{-1}), g_{2,N}^\top(x_{-2}), \dots, g_{M,N}^\top(x_{-M}))^\top$ . With these notation, we can describe the contraction property for the SAA Nash equilibrium problem of  $\Gamma$  as follows.

**Property 4.** Under Conditions in (A1~2) hold, if  $\alpha_p^N$  is positive and less than 1, then, with probability 1, there exists a stable Nash equilibrium of problem (3.11), and hence a SAA Nash equilibrium of  $\Gamma$  in set  $X$ . Here

$$\alpha_p^N := \|Dh_p^N\| = \left\| \begin{array}{ccc} \nabla_{x_1} g_{1,N} & \cdots & \nabla_{x_M} g_{1,N} \\ \vdots & \ddots & \vdots \\ \nabla_{x_1} g_{M,N} & \cdots & \nabla_{x_M} g_{M,N} \end{array} \right\| \quad (3.12)$$

Compared to Property 3, the most different points in Property 4 are the terms  $h_p^N$ ,  $\alpha_p^N$ ,  $Q_i^N$  and  $g_{i,N}$  for  $i \in I$ , which depend on the sampling results.

### 3.3 Variable sample distributed algorithm

Here, we establish a framework of variable sampling approximation for solving the stable SNE. In this framework, we use different samples  $\{\xi_1^k, \dots, \xi_{N_k}^k\}$  at each iteration  $k = 0, 1, \dots$ , where  $N_k$  is an integer number representing the size of the sample used at iteration  $k$  for  $k = 1, 2, \dots$ . To standardize the discussion, we try to accommodate the terminologies within the framework of pilot research in [11]. We call  $\{N_k\}_{k=1}^\infty$  the *schedule* of sample sizes associated with the algorithm under scrutiny.

*In Sections 4 and 5, we will propose a set of variations of this algorithm for the numerical experiments. In these variations, this sampling step is performed by each decision maker with different sample size or resampling scheme. Necessary descriptions on these variations will be presented before the corresponding numerical experiments.*

We start with the concepts used in the VSDA (Algorithm 2). First, we look into the definition on the sample used in each iteration of the algorithm – where the most different point from the approach in [35] is that in the VSDA the samples at different iterations are not necessarily drawn from the same distribution. Here we use notation  $\Xi^{N_k}$  and  $\mathbb{P}_k\{\cdot\}$  to emphasize the distribution functions used in iterations might not be identical in the VSDA, where  $\Xi^{N_k}$  denotes the  $N_k$ –fold Cartesian product of the sample space  $\Xi$  and  $\mathbb{P}_k$  denotes a probability measure on  $\Xi^{N_k}$  for  $k = 1, 2, \dots$ . By doing so, we define the sample used in iteration  $k$  of the VSDA as  $\xi^k(\omega) := \{\xi_n^k(\omega), n = 1, 2, \dots, N_k\}$  and reformulate it as  $(\xi_1^k(\omega), \dots, \xi_{N_k}^k(\omega)) \in \Xi^{N_k}$ , which can be simply written as  $\xi^k = (\xi_1^k, \dots, \xi_{N_k}^k)$  or equally  $\{\xi_n^k, n = 1, 2, \dots, N_k\}$ . Moreover, we can collectively write the sample as

$$\xi = (\xi^1; \dots, \xi^k; \dots) = (\xi_1^1, \dots, \xi_{N_1}^1; \xi_1^2, \dots, \xi_{N_2}^2; \dots) \in \tilde{\Xi}$$

where we define  $\tilde{\Xi} := \Xi^{N_1} \times \Xi^{N_2} \times \dots$  and let  $\tilde{\mathbb{P}}$  denote the corresponding probability distribution on  $\tilde{\Xi}$  generated by the  $\mathbb{P}_k$ s.

Consequently, we look into random variables and functions in the VSDA defined on the probability space  $(\tilde{\Xi}, \tilde{\mathbb{P}})$ : For each  $\xi \in \tilde{\Xi}$ , the SAA functions in each iteration  $k = 0, 1, \dots$  can

---

**Algorithm 2:** Variable Sample Distributed Algorithm (VSDA)

---

**Inputs:** The decision makers' initial positions  $x^0 := (\hat{x}_1, \hat{x}_2, \dots, \hat{x}_M)$ , error tolerance  $\epsilon_V > 0$ .

**Outputs:** The terminate iteration  $K$ , the solution  $x^K := (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_M)$ , and the objective functions  $F^K := (F_1(\tilde{x}_1, \tilde{x}_{-1}), \dots, F_M(\tilde{x}_M, \tilde{x}_{-M}))$ .

**Initialization:**

Let iteration  $k = 0$ . Set an initial point  $x^N[k] = x^0$  where  $x^N[k] = (x_1^N[k], \dots, x_M^N[k])$  and  $x_i^N[k] = \hat{x}_i$  for  $i \in I$ .

**Algorithm:**

**Step 1.** Generating a sample  $\xi^{k+1} = (\xi_1^{k+1}, \dots, \xi_{N_k}^{k+1})$  from probability distribution  $\mathbb{P}_{k+1}$ .

- For each decision maker  $i$  at iteration  $k + 1$ , find  $x_i^N[k + 1]$  for  $i \in I$  by solving an optimization problem:

$$\min_{x_i \in X_i} F_i^{k+1}(x_i, x_{-i}^N[k]) := \frac{1}{N_{k+1}} \sum_{n=1}^{N_{k+1}} f_i(x_i, x_{-i}^N[k], \xi_n^k), \quad (3.13)$$

where  $x_{-i}^N[k]$  is obtained in the previous iterations.

**Step 2.** Check the stopping criterion. If  $\|x^N[k + 1] - x^N[k]\|$  is sufficiently small (less than  $\epsilon_V$ ), then terminate the algorithm, return  $x^N[k]$  and set  $K = k$ . Otherwise, set  $k := k + 1$  and go to Step 1.

**Return:** The solution  $x^K := x^N[K]$  and the objective functions

$$F^K := (F_1^K(x_1^N[K], x_{-1}^N[K]), \dots, F_M^K(x_M^N[K], x_{-M}^N[K])).$$


---

be formulated as,

$$F_i^{N_k}(x_i^N[k], x_{-i}^N[k]) = \frac{1}{N_k} \sum_{n=1}^{N_k} f_i(x_i^N[k], x_{-i}^N[k], \xi_n^k), \quad \forall k = 0, 1, \dots \quad (3.14)$$

for decision makers  $i \in I$ , where we omit the dependence of function  $F_i^{N_k}(x_i, x_{-i})$  on  $\xi^k$  for brevity. Moreover, in the rest of the paper, we further rewrite  $F_i^{N_k}$  to  $F_i^N$ . Based on our definition, to show the convergence of the VSDA is to show that the algorithm converges for  $\tilde{\mathbb{P}}$ -almost all  $\xi \in \tilde{\Xi}$ . To this end, we need the *consistent* estimators (See Condition A3) and the following conditions for the convergence of the VSDA for almost all sample paths.

**(A4)** (Boundedness) For each  $(x_i, x_{-i}) \in X_i \times X_{-i}$ , there exists a positive constant  $\tilde{M}_i = \tilde{M}_i(x) > 0$  such that for all  $i \in I$

$$F_i^N(x_i, x_{-i}) \leq \tilde{M}_i$$

with probability one.

Now let us look at Condition (A4).  $F_i^N$  is not the arithmetic mean of  $N$  values when  $N$  is finite, but was defined as the average with finite sample, which is a random variable. Hence the concepts of “with probability one” and “the expectation” in Condition (A5) below are both well defined. This condition and Condition (A5) below are standard and have been imposed by Homem-de-Mello [11]. The statement in Condition (A4) can be obtained directly from the Lipschitz continuity of function  $f_i(x_i, x_{-i}, \xi)$  and the boundedness of  $X_i$  for  $i \in I$ . Condition (A4) says that all random variables are *uniformly bounded* with probability one. This is particularly true in a simulation model when all input distributions have supports which are contained in a finite region that does not depend on  $N$ . This is the case when all input distributions, for instant service times, market demands, stock prices, etc., have supports which are contained in bounded feasible sets.

**(A5)** (Unbiasedness) For each  $(x_i, x_{-i}) \in X_i \times X_{-i}$ , we have that for all  $i \in I$

$$\lim_{N \rightarrow +\infty} \mathbb{E}[F_i^N(x_i, x_{-i})] = F_i(x_i, x_{-i}).$$

Condition (A5) implies that the estimators  $F_i^N(x_i, x_{-i})$  are asymptotically unbiased, and obviously holds in case all  $\mathbb{E}[f_i(x_i, x_{-i}, \xi)]$  are unbiased estimators of  $F_i(x_i, x_{-i})$  for  $i \in I$  and the sample with size  $N_k$ .

Before proceeding to the analysis of the VSDA, let us review the definitions on operator  $\mathcal{Q}_i^N$ , the mappings  $h_p^N, g_i^N, \alpha_p^N$ , and Lipschitz modulus  $k_p^N$  in Section 3.2. Note that the definitions of the above operators in Section 3.2 essentially characterize the relationship between the iterates at step  $k + 1$  and step  $k$  for  $k = 1, 2, \dots$  with the sample size of objective function being  $N$ . From the statement in Section 3.2, we can easily see that the contraction property for the SAA Nash equilibrium problem is still valid for the cases where the sample varies at different steps such as the VSDA. To avoid the redundancy, here we omit the definition of  $\mathcal{Q}_i^{N_k}, g_i^{N_k}, h_p^{N_k}, \alpha_p^{N_k}$ , and Lipschitz modulus  $k_p^{N_k}$  for  $i \in I$  and present them in the appendix. Moreover, the

terms  $h_p^{N_k}$ ,  $\alpha_p^{N_k}$ ,  $Q_i^{N_k}$  and  $g_i^{N_k}$  for all  $i \in I$  depends on the sample used in each iteration, which consequently are functions of random variable  $\xi$ .

### 3.4 Convergence analysis

First, we look into the accumulation point yielded by the VSDA. It is well known that, under convex conditions and contraction property, the accumulation points generated by the distributed algorithm is a stable Nash equilibrium [4, 6]. Here, our aim is to establish a result that the accumulation point generated by the VSDA converges to the stable SNE of  $\Gamma$  for  $\tilde{\mathbb{P}}$ -almost.

From the aforementioned VSDA context, we can see that the convergence investigated in the paper is different from either the asymptotic convergence of SAA Nash equilibrium in [35] characterized by sample averaged Karush-Kuhn-Tucker (KKT) conditions or stationary conditions, or the convergence of accumulation points generated by distributed asynchronous computation algorithm in [3] identified via the “gradient/derivative” being iteratively updated to zero. Our investigation on the convergence of the VSDA targets at answering the following two questions by combining the analyses on probability convergence [35] and iterative convergence [3]:

- (a) Under what conditions of sample size schedule  $\{N_k, k = 1, 2, \dots\}$ , the sequence of the iterates  $\{x^N[k]\}$  yielded in the VSDA converges in probability (or may not to) the true Nash equilibrium?
- (b) If yes, does this accumulation point of iterates  $\{x^N[k], k = 1, 2, \dots\}$  converge to the true Nash equilibrium in  $\Gamma$ ?

In alignment with the above context, the definition of convergence for the VSDA is integrally contains two aspects: the convergence of the sequence  $\{x^N[k], k = 1, 2, \dots\}$  in the framework of the distributed algorithm scheme; and the asymptotic convergence of sampling scheme in probability along with the increase of  $N_k$ .

Before proceeding to the convergence analysis of the VSDA, we need the following lemma.

**Lemma 3.1** *Suppose that Conditions (A1~5) hold. If the gradient  $\|\nabla_{x_i} F_i(x_i, x_{-i})\| \geq \epsilon_g$  at a point  $x \in X$  and for some positive constant  $\epsilon_g$  for an  $i \in I$ , then there exists a constant  $\epsilon_\Delta > 0$  such that,*

$$\tilde{\mathbb{P}} \left\{ \left\| x - h_p^{N_{k+1}}(x) \right\| \leq \epsilon_\Delta \right\} \leq 2M e^{-\gamma(\epsilon_\Delta)N_{k+1}}. \quad (3.15)$$

We include the proof of Lemma 3.1 in the appendix. This lemma essentially implies that if a point  $x$  is not a stationary point of  $\Gamma$ , then the probability that it is a fix-point in SAA Nash equilibrium problem of  $\Gamma$  will upper bounded by a small positive value which is exponentially decreasing along with the increase of the sample size used in an iteration of the VSDA.

### 3.4.1 Accumulation point

Based on Lemma 3.1, we have the following result on the convergence of the VSDA,

**Theorem 3.1** *Suppose that the schedule of the sample size  $\{N_k\}$  satisfies the property that for all  $\alpha \in (0, 1)$ ,*

$$\sum_{k=1}^{\infty} \alpha^{N_k} < \infty. \quad (3.16)$$

*Suppose also that Conditions A1~5 hold, then,*

$$\tilde{\mathbb{P}} \left\{ \liminf_{k \rightarrow +\infty} \|\nabla_{x_i} F_i(x_i^N[k], x_{-i}^N[k])\| = 0 \right\} = 1, \quad (3.17)$$

for all  $i \in I$ .

**Proof of Theorem 3.1.** We prove the statement (3.17) by contradiction. Suppose that (3.17) is not satisfied. We have there exists an  $\alpha \in [0, 1)$  and an  $i \in I$  such that

$$\tilde{\mathbb{P}} \left\{ \liminf_{k \rightarrow +\infty} \|\nabla_{x_i} F_i(x_i^N[k], x_{-i}^N[k])\| = 0 \right\} = \alpha,$$

which can be equivalently formulated as: there exists a positive value  $\epsilon_g$  and an  $i \in I$  such that

$$\tilde{\mathbb{P}} \left\{ \liminf_{k \rightarrow +\infty} \|\nabla_{x_i} F_i(x_i^N[k], x_{-i}^N[k])\| \geq \epsilon_g \right\} = \alpha',$$

for an  $\alpha' \in (0, 1]$ . Let us denote the vector

$$\left( \nabla_{x_1}^\top F_1(x_1^N[k], x_{-1}^N[k]), \dots, \nabla_{x_M}^\top F_M(x_M^N[k], x_{-M}^N[k]) \right)^\top$$

by  $\tilde{F}(x^N[k])$ .

This implies that there exists a subset  $\Xi_r \subset \tilde{\Xi}$  such that (a) the measure of  $\Xi_r$  is not zero; (b) for every  $\xi_r \in \Xi_r$ , there exists a positive integer number  $k(\xi_r) < \infty$  such that

$$\Delta(\xi_r) := \inf \left\{ \|\tilde{F}(x^N[k])\|, \forall k \geq k(\xi_r) \right\} \geq \epsilon_g$$

which also implies that for all  $k \geq k(\xi_r)$ ,

$$\|\tilde{F}(x^N[k])\| \geq \epsilon_g.$$

Moreover, from Lemma 3.1, we have the following inequality on the probability,

$$\begin{aligned} & \tilde{\mathbb{P}} \left\{ \|\tilde{F}(x_i, x_{-i})\| \geq \epsilon_g, \left\| x - h_p^{N_{k+1}}(x) \right\| \leq \epsilon_\Delta \right\} \\ & \leq \tilde{\mathbb{P}} \left\{ \left\| x - h_p^{N_{k+1}}(x) \right\| \leq \epsilon_\Delta \right\} \\ & \leq 2M e^{-\gamma(\epsilon_\Delta/M)N_k}. \end{aligned} \quad (3.18)$$

Define the event  $E_k := \left\{ \|\tilde{F}(x_i^N[k], x_{-i}^N[k])\| \geq \epsilon_g, \left\| x^N[k] - h_p^{N_{k+1}}(x^N[k]) \right\| \leq \epsilon_\Delta \right\}$ . According to the property (3.16) of the schedule of the sample sizes  $\{N_k\}$ , we have

$$\sum_{k=1}^{\infty} \tilde{\mathbb{P}}\{E_k\} < 2M \sum_{k=1}^{\infty} e^{-\gamma(\epsilon_\Delta/M)N_k} < +\infty. \quad (3.19)$$

By applying the Borel-Cantelli Lemma, the event  $E_k$  occurs only finitely many times w.p.1. Thus, for almost all  $\xi \in \tilde{\Xi}$ , we have that for all  $k \geq k(\xi_r)$ , the event

$$\left\{ \left\| x^N[k] - h_p^{N_{k+1}}(x^N[k]) \right\| \leq \epsilon_\Delta \right\} \equiv \left\{ \left\| x^N[k] - x^N[k+1] \right\| \leq \epsilon_\Delta \right\},$$

occurs finitely many times where we define this number by  $n(\xi_r) < +\infty$ . Consequently, summing up all  $\|x^N[k] - x^N[k+1]\|$  for  $k \geq k(\xi_r)$  yields that

$$\sum_{k=k(\xi_r)}^{\infty} \left\| x^N[k] - x^N[k+1] \right\| \geq (T - k(\xi_r) - n(\xi_r)) \epsilon_\Delta. \quad (3.20)$$

On the other hand, for every  $\xi \in \tilde{\Xi}$ , we have

$$\left\| x^N[k+2] - x^N[k+1] \right\| \leq \kappa_p^{N_{k+1}} \left\| x^N[k+1] - x^N[k] \right\|.$$

According to the Contraction Property,  $\kappa_p^{N_k}$  takes a value within  $[0, 1)$  for any  $k = 1, 2, \dots$ . Moreover, we have  $\kappa'_p = \sup_k \kappa_p^{N_k}$  and  $\kappa'_p \in [0, 1)$ . Then, the sum of  $\|x^N[k+1] - x^N[k]\|$  attains that

$$\begin{aligned} \sum_{k=1}^{\infty} \left\| x^N[k] - x^N[k+1] \right\| &= \left\| x^N[2] - x^N[1] \right\| + \left\| x^N[3] - x^N[2] \right\| + \dots \\ &\leq \frac{1}{1 - \kappa'_p} \left\| x^N[2] - x^N[1] \right\|. \end{aligned} \quad (3.21)$$

Combining the inequality (3.20), the above inequality implies that for all  $\xi_r \in \Xi^r$ ,

$$\begin{aligned} (T - t(\xi_r) - n(\xi_r)) \epsilon_\Delta &\leq \sum_{k=k(\xi_r)}^T \left\| x^N[k] - x^N[k+1] \right\| \\ &\leq \sum_{k=1}^{\infty} \left\| x^N[k] - x^N[k+1] \right\| \\ &\leq \frac{1}{1 - \kappa'_p} \left\| x^N[2] - x^N[1] \right\|. \end{aligned} \quad (3.22)$$

Note that the inequalities in (3.22) stands for all  $T \geq 0$ . Let  $T$  go to  $+\infty$ , we have that the term  $(T - s(\xi_r) - n(\xi_r)) \epsilon_\Delta$  goes to infinity, while at the other side, the term  $\frac{1}{1 - \kappa'_p} \left\| x^N[2] - x^N[1] \right\|$  is finitely fixed. This contradiction also prove the results.  $\blacksquare$

The results in Theorem 3.1 mean that for a given schedule of the sample size  $\{N_k, k = 1, 2, \dots\}$  satisfying the condition  $\sum_{k=1}^{\infty} \alpha^{N_k} < \infty$ , the accumulation point yielded by the VSDA with probability one is the stable SNE of the true problem  $\Gamma$ . Here, we can straightforwardly extend the result in Theorem 3.1 to the asymptotic convergence of the accumulation points yielded by the VSDA to the true SNE in  $\Gamma$ .

**Theorem 3.2** *Suppose that Conditions A1~5 and Condition (3.16) in Theorem 3.1 hold, then there exists an  $x^* \in X^*$ , such that,*

$$\tilde{\mathbb{P}} \left\{ \liminf_{k \rightarrow +\infty} \|x^N[k] - x^*\| = 0 \right\} = 1, \quad (3.23)$$

where  $X^*$  is the fix point set,  $\{x \in X \mid x = h_p(x)\}$ , in  $\Gamma$ .

We include the proof of Theorem 3.2 in the appendix.

### 3.4.2 Termination

For a given error tolerance in the VSDA, we use  $K$  to denote the number of the terminal iteration, which can be denoted as

$$K = \inf_{k \geq 1} \{k : \|x^N[k+1] - x^N[k]\| \leq \epsilon\}.$$

We now turn our attention to finite stopping and show that the sequential procedure stops with probability one. We state this formally in the proposition below.

**Proposition 3.1 (Finite Termination)** *Suppose that Conditions (A1~5) and Condition (3.16) in Theorem 3.1 hold. For a fixed positive error tolerance  $\epsilon$  in the VSDA, we have  $\tilde{\mathbb{P}}\{K < \infty\} = 1$ .*

**Proof of Proposition 3.3.** Note that

$$\begin{aligned} \tilde{\mathbb{P}}\{K = \infty\} &= \tilde{\mathbb{P}} \left\{ \inf_{k \geq 1} \{k : \|x^N[k+1] - x^N[k]\| \leq \epsilon\} = \infty \right\} \\ &= \tilde{\mathbb{P}} \left\{ \|x^N[k+1] - x^N[k]\| \geq \epsilon, \quad \text{for all } k = 1, 2, \dots \right\} \\ &\leq \tilde{\mathbb{P}} \left\{ \|x^N[k+1] - x^*\| + \|x^* - x^N[k]\| \geq \epsilon, \quad \text{for all } k = 1, 2, \dots \right\} = 0, \end{aligned}$$

where  $x^* \in X^*$  defined in Theorem 3.2 and the last equation is from Theorem 3.2. This contradiction also prove the results. ■

## 4 Computational Analysis

In this section, we comparatively perform numerical experiments to identify how the key factors in the VSDA affect its effectiveness. We also investigate a scenario where each decision maker has a different sample size schedule. By doing so, we reveal the existence of “buckets effect” in the framework of the VSDA.

## 4.1 Sampling schedules

Let us consider a Nash equilibrium problem with three decision makers in a competitive market. We index the decision makers by  $i = 1, 2, 3$ . Here  $\{x_i\}_{i=1}^3$  are decision variables denoting the quantities sold by each decision maker before knowing the realization of the maker uncertainty  $\xi$ . The market uncertainty  $\xi$  in the model is specified as  $(\xi_1, \xi_2, \xi_3) \in \Xi$ , where  $\xi_j$  for  $j = 1, 2, 3$  represent the market uncertainty's influences on decision makers' profit functions, including 1) on the coefficients of the market price function  $p(x, \xi) := a_0(\xi_1) - b_0(\xi_2)x$  with  $x = x_1 + x_2 + x_3$ ; 2) on the marginal cost  $c_i(\xi_3)$ . Here, we respectively set  $a_0(\xi_1) = 90 + 10\xi_1$ ,  $b_0(\xi_2) = 900 + 10\xi_2$ ,  $c_1(\xi_3) = 20 + 4\xi_3$ ,  $c_2(\xi_3) = 20 + 40\xi_3$  and  $c_3(\xi_3) = 20 + 20\xi_3$ .

We have decision maker  $i$ 's expected profit as

$$F_i(x_i, x_{-i}) = \mathbb{E} [x_i \{a_0(\xi_1) - b_0(\xi_2)(x_i + x_{-i})\} - c_i(\xi_3)x_i]. \quad (4.24)$$

After verifying Condition (A1~5), we have the existence of the stable SNE for this numerical problem.

### 4.1.1 Fixed and linear variable sample schedule

First, we apply Algorithm 1 for solving the SAA Nash equilibrium problem (4.24) with fixed sample size, where the realization of the sample is predetermined before the running of the algorithm. On the other hand, in the VDSA, the size of the sample used at each iteration for  $k = 1, 2, \dots$ , follows a linear (growth) schedule  $N_k = 200k$ . The evolutions of distances between iterates generated by algorithms with different schedule and the true SNE of  $\Gamma$  are presented in Figure 1.

From the results in Figure 1 we can observe that, both of the sequences of iterates converge to the true SNE of  $\Gamma$  in a very fast way (within around 0.1 minute). The algorithm with fixed sample size cannot achieve unbiased accumulation point, i.e. the deviations from the iterates to the true SNE cannot converge to zero. Compared to the fixed sample case, the iterates in the VSDA with linear schedule can achieve an approximation of the true SNE with very small deviation within 0.1 minute.

### 4.1.2 Variable sample schedules

Now let us analyze how different variable sample schedules affect the results in the VSDA. In this experiment it is not a necessary requirement that the sample size schedule  $\{N_k, k = 1, 2, \dots\}$  must obey Condition (3.16) in Theorem 3.1. By doing so, we investigate how the schedules satisfying or unsatisfying Condition (3.16) in Theorem 3.1 behave differently. In the trials, we first investigate the algorithms with sample schedules which do not satisfy Condition (3.16), where the sample size are respectively fixed at 100, 1000 and 10000 but with different sample

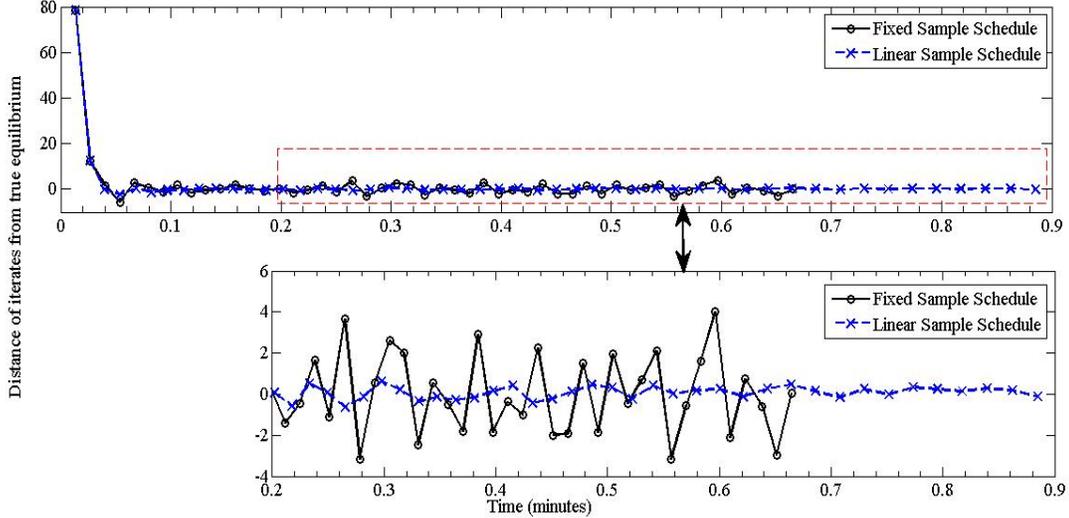


Figure 1: The fixed and linear sample schedule for solving SNE.

being used at each iteration. Secondly, we comparatively look at the VSDAs with linear and nonlinear growth sample size.

For the comparison, we need to introduce two concepts *Accuracy Error* and *Precision Error*: In the test, the VSDA is repeated  $R$  times, where in each time the algorithm is started from the same initial guesses, with the samples from the same distribution  $\mathbb{P}_k$  and the same sample size schedule  $N_k$  at each iteration  $k$ . At step  $k$ , the accuracy error is defined as

$$r^A[k] = \sum_{i=1}^M \left| \frac{1}{R} \sum_{r=1}^R x_i^{r,N}[k] - x_i^* \right|,$$

where  $x_i^*$  is the true equilibrium solution and  $x_i^{r,N}[k]$  is the  $x_i^N[k]$  in the  $k$ th step of the  $r$ th repeat of running the VSDA for  $r = 1, 2, \dots, R$ . The accuracy error measures the average distance between the true SNE. In our computational experiments, we use the solution from a run with  $10^7$  samples as an estimate  $x^*$ . The *precision error* is defined as

$$r^P[k] = \frac{1}{R} \sum_{i=1}^M \sum_{r=1}^R (x_i^{r,N}[k])^2 - \frac{1}{R^2} \sum_{i=1}^M \left( \sum_{r=1}^R x_i^{r,N}[k] \right)^2.$$

The precision error measures the standard deviation of a solution over different runs.

First, we look into accuracy and precision errors generated in the algorithm with fixed sample size schedules, where the sample sizes range from 100 to 1000. Figure 2 presents the accuracy and precision errors in the algorithm with respect to the computation time. Both of the algorithms can provide a SAA Nash equilibrium with high accuracy within 0.2 minutes. However, along with the iterations, the evidence shown in Figure 2 denied the convergence in precision errors.

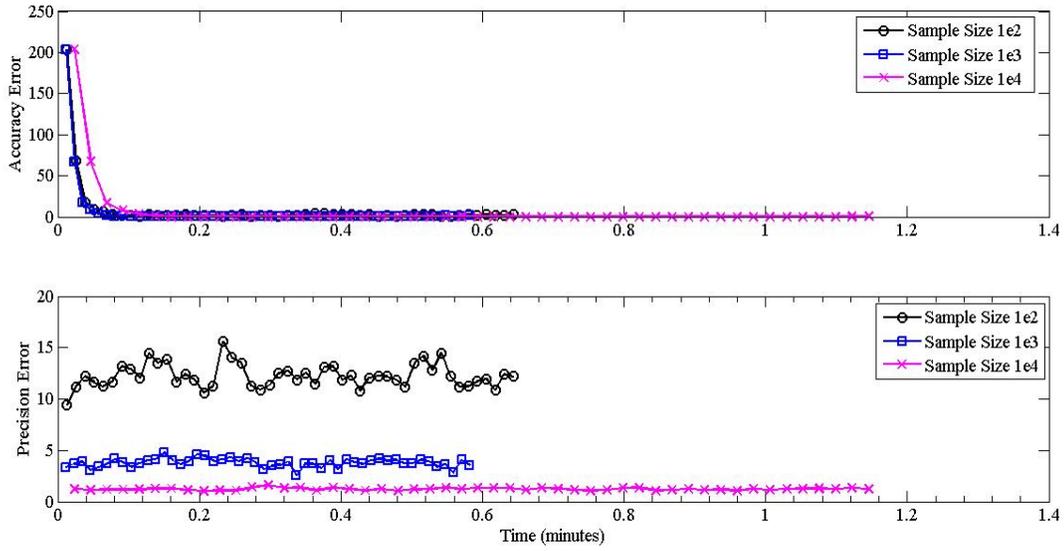


Figure 2: Variable sample distributed algorithm with fixed sample sizes I.

We now look into accuracy errors and precision errors with respect to the *sampling load*, where the sampling load is measured by the overall number of samples has been used until a certain iterate has been achieved. The results are presented in Figure 3 for the VSDA with fixed sample size schedule. Figure 3 presents a similar results as in Figure 2, which indicates

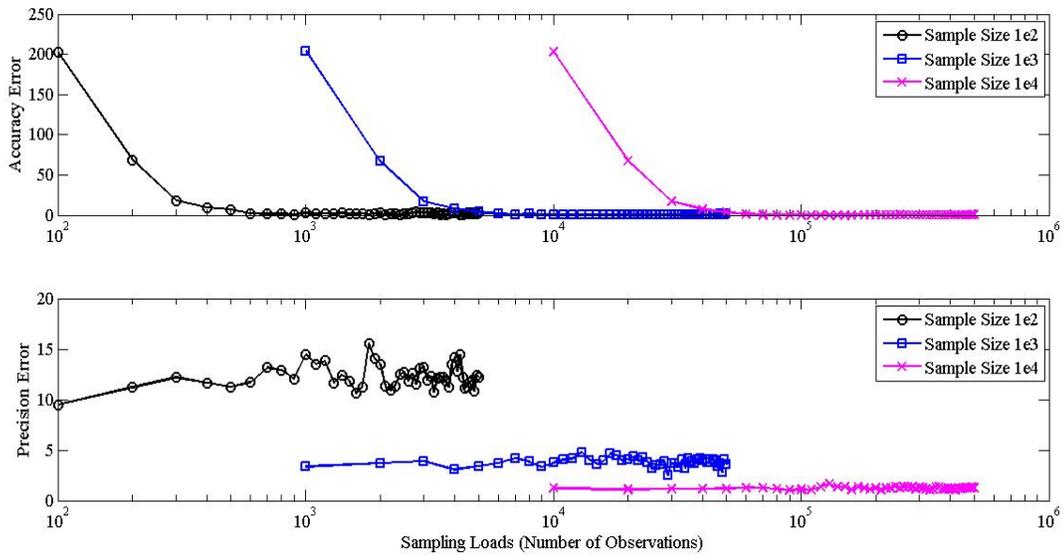


Figure 3: Variable sample distributed algorithm with fixed sample sizes II.

that without a well-designed sample size schedule, additional sampling inputs or the increase of iterations cannot improve the precision errors.

Now, we comparatively investigate the VSDA with linear/nonlinear sample size schedules. The following figures compare the evolutions of accuracy and precision errors in the cases with  $N_k = 200k$  and  $N_k = \lfloor 20k^{1.5} \rfloor$  to their counterparts with  $N_k \equiv 1000$  and  $N_k \equiv 10000$  in Figures 2 and 3. The comparisons are performed with respect to the computation time (Figure 4) and the sampling load (Figure 5) respectively.

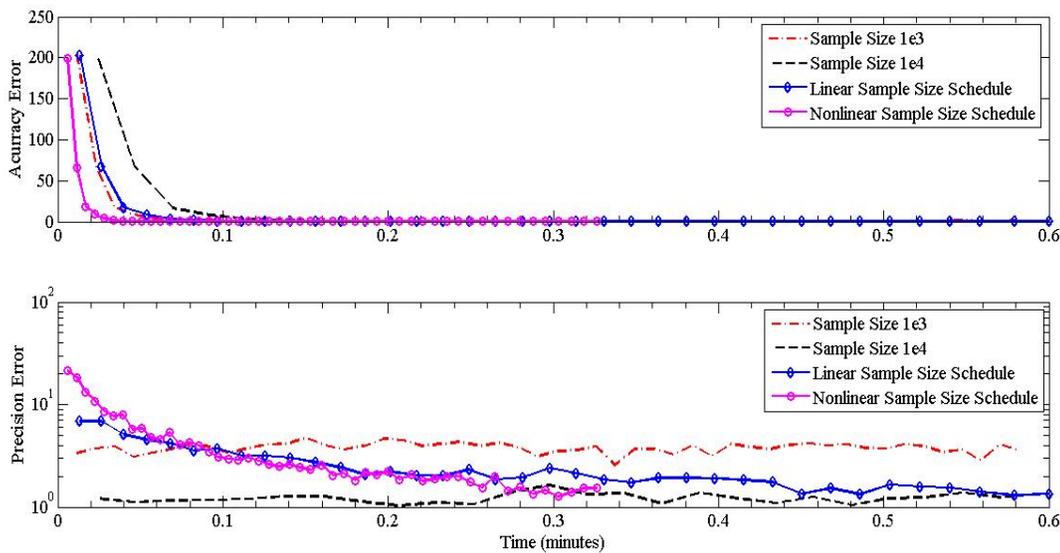


Figure 4: Variable sample distributed algorithm with variable sample sizes I.

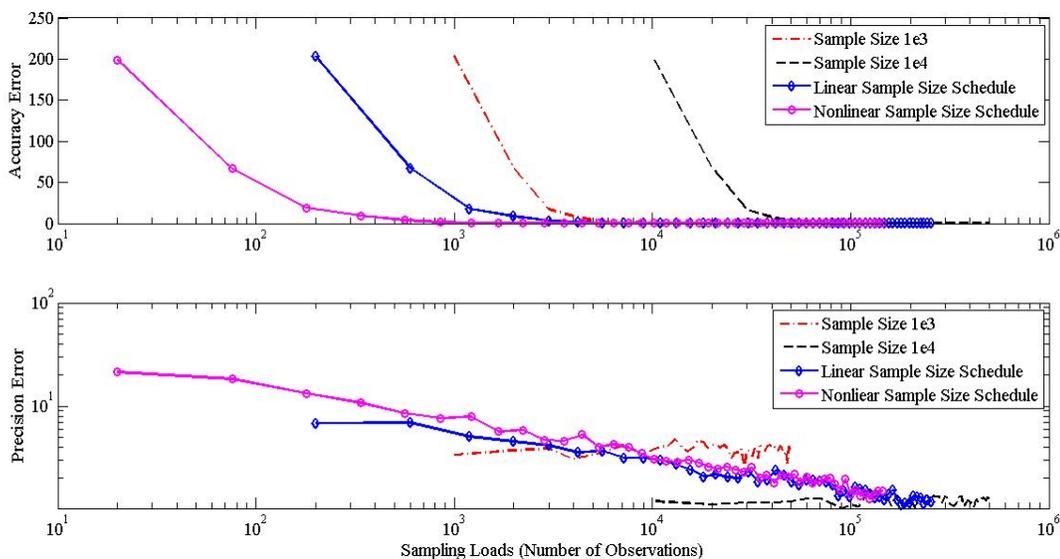


Figure 5: Variable sample distributed algorithm with variable sample sizes II.

Compared with the fixed sample size schedule, the VSDAs with linear and nonlinear sample size schedules can achieve a significant reduction in their precision errors. It is implied that the

VSDA with schedules satisfying condition (3.16) attains a sequence of iterates converges to the SNE with both accuracy and precision errors converging to zero. The same conclusion can be drawn from Figures 4 and 5 for the reduction of the precision errors along with the growth of sampling efforts.

### 4.1.3 Buckets effect in sampling

In real world problems, it is natural that different decision makers have different capabilities to obtain the samples even when they are competing in the same market and facing the same market uncertainty. Here we apply our algorithm to model this type of situation. For ease of comparison, we set that the only difference in the uncertainty’s impact onto each decision maker is through its own cost function. Let us redefine the cost functions for decision makers as

$$c_1(x_1, \xi_3) = 20 + 4\xi_3, \quad c_2(\xi_3) = 20 + 40\xi_3, \quad c_3(\xi_3) = 20 + 20\xi_3. \quad (4.25)$$

From the above cost formulation, we can observe that decision maker 2 is more sensitive to the uncertainty, i.e. random variable  $\xi_3$ . Through this experiment, we show that the well-known “bucket effect” also exists in the sampling methods for using SAA to solve the SNE, that is, the quality of the SAA Nash equilibrium is determined by decision maker(s) with least sampling capability. To this end, we design different sampling schedules to each decision maker to represent their capability of accessing the samples, which are  $N_k^1 = \lfloor 200k^{1.5} \rfloor$ ,  $N_k^2 = 200k$  and  $N_k^3 = 200k^2$ . To emphasize the “bucket effect” phenomena, we here intentionally set decision maker 2 being with least capability to access samples and hence make it as the “shortest piece of wood”.

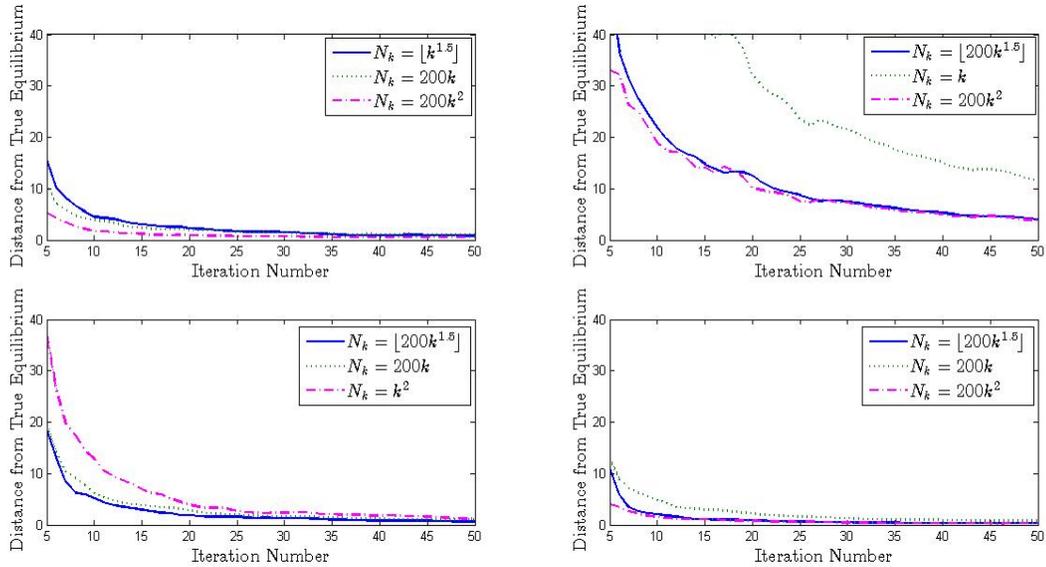


Figure 6: Bucket effects in the sampling scheme for the computation of SNE.

The results of implementing the joint schedule of  $\{N_k^1, N_k^2, N_k^3\}$  to the VSDA is shown in the fourth part (right-bottom) in Figure 6. After that, we perform the other three trials with their results also presented in Figure 6 where the sample size schedules for each trial are  $\{N_k^{1'}, N_k^2, N_k^3\}$ ,  $\{N_k^1, N_k^{2'}, N_k^3\}$  and  $\{N_k^1, N_k^2, N_k^{3'}\}$  with  $N_k^{1'} = \lfloor k^{1.5} \rfloor$ ,  $N_k^{2'} = k$  and  $N_k^{3'} = k^2$ . For these three schedules, the corresponding results are presented in the first part (left-top), second part (right-top) and third part (left-bottom).

The results in Figure 6 indicate that if we improve the “shortest piece of wood in the bucket”, decision maker 2’s sampling capability, the quality of the SAA Nash equilibrium (including decision makers 1 and 3’s estimations on their SNE position) can be significantly enhanced from results in the second part to results in the fourth part. Comparatively, the improvement of the “other pieces of wood in the bucket” cannot achieve the similar improvement rate.

## 5 Cumulative Sampling Schemes

In this section, we present numerical tests to compare the cumulative sampling (resampling) schemes proposed in [11] and [2]. The common point between these two cumulative sampling schemes is that at each iteration a new sample can be appended to the previous. In other words, using the notation defined earlier, if the sample used at some iteration, say  $k$ , is  $\xi_1^k, \dots, \xi_{N_k}^k$ , then under some conditions, the sample used at the  $k + 1$ st iteration can be  $\xi_1^{k+1} = \xi_1^k, \dots, \xi_{\tilde{N}_k}^{k+1} = \xi_{\tilde{N}_k}^k, \xi_{\tilde{N}_k+1}^{k+1}, \dots, \xi_{N_{k+1}}^{k+1}$  with some  $\tilde{N}_k \leq N_k$ . Our analysis is focused on how different cumulative sampling schemes affect the finite stopping of the VSDA algorithm numerically. Hence, we refer to the theoretical discussion on the cumulative sampling scheme to Section 3.3 in [11].

**Resampling Scheme** by Bayraksan and Morton in [2]. In [2], a resampling frequency  $k_f$  is proposed to control the sampling of new independent observations.

---

**Algorithm 3:** Variation of the VSDA Algorithm: Resampling.

---

**Variation of Step 1 of the VSDA:** For each decision maker  $i$  at iteration  $k + 1$

- If  $k_f$  divides  $k + 1$ , then sample observation  $\xi_1^{k+1}, \dots, \xi_{N_{k+1}}^{k+1}$  independently of samples generated in previous iterations. Else, set first  $N_k$  observations by  $\xi_1^{k+1} = \xi_1^k, \dots, \xi_{N_k}^{k+1} = \xi_{N_k}^k$  and sample  $N_{k+1} - N_k$  new observations  $\xi_{N_k+1}^{k+1}, \dots, \xi_{N_{k+1}}^{k+1}$ ;
- Find  $x_i^N[k + 1]$  for  $i \in I$  by solving an optimization problem:

$$\min_{x_i \in X_i} F_i^{k+1}(x_i, x_{-i}^N[k]) = \frac{1}{N_{k+1}} \sum_{n=1}^{N_{k+1}} f_i(x_i, x_{-i}^N[k], \xi_n^k), \quad (5.26)$$

where  $x_{-i}^N[k]$  is obtained in the previous iterations.

---

Note that  $k_f$  can be seen as the frequency of how fast the algorithm entirely resamples the observations. If the resampling frequency  $k_f = 1$ , then at every iteration we sample observations

$\xi_1^k, \dots, \xi_{N_k}^k$  independently of previously generated observations for  $k = 1, 2, \dots$ . At the other extreme, if  $k_f$  is sufficiently large, then we continuously augment the existing set of observations throughout execution of the algorithm.

**Cumulative Sampling** by Homem-de-Mello in [11]. In [11], a cumulative sampling methodology is proposed where at each iteration  $k + 1$ , a new sample with  $N_{k+1} - N_k$  observations are *appended* to the sample used in the previous iteration, i.e.  $\xi_1^{k+1} = \xi_1^k, \dots, \xi_{\tilde{n}_k}^{k+1} = \xi_{\tilde{n}_k}^k, \dots, \xi_{N_{k+1}}^{k+1} = \xi_{N_{k+1}}^k$ . Here, we introduce a parameter  $\beta \in [0, 1]$  to control the proportion of the observations from the previous iteration.

---

**Algorithm 4:** Variation of the VSDA Algorithm: Cumulative sampling.

---

**Variation of Step 1 of the VSDA:** For each decision maker  $i$  at iteration  $k + 1$

- Set first  $\tilde{n}_k := \lfloor (1 - \beta)N_k \rfloor$  observations as  $\xi_1^{k+1} = \xi_1^k, \dots, \xi_{\tilde{n}_k}^{k+1} = \xi_{\tilde{n}_k}^k$  and sample  $N_{k+1} - \tilde{n}_k$  observations  $\xi_{\tilde{n}_k+1}^{k+1}, \dots, \xi_{N_{k+1}}^{k+1}$ ;
- Find  $x_i^N[k + 1]$  for  $i \in I$  by solving an optimization problem:

$$\min_{x_i \in X_i} F_i^{k+1}(x_i, x_{-i}^N[k]) = \frac{1}{N_{k+1}} \sum_{n=1}^{N_{k+1}} f_i(x_i, x_{-i}^N[k], \xi_n^k), \quad (5.27)$$

where  $x_{-i}^N[k]$  is obtained in the previous iterations.

---

Note that  $\beta$  represents the proportion at which the algorithm resamples the observations and uses in the succeeding iterations. If the cumulative sample percentage  $\beta = 1$ , then at every iteration we sample observations  $\xi_1^k, \dots, \xi_{N_k}^k$  independently of previously generated observations for  $k = 1, 2, \dots$ . If  $\beta$  is 0, then we continuously augment the existing set of observations throughout execution of the algorithm.

We present the following example to comparatively analyse the impact on finite stopping and solution times of the VSDA from each resampling schemes and their resample parameters  $k_f$  and  $\beta$ . Here, we intentionally investigate a more complex competition than the problem in the previous section to illustrate the effects in solution time, iteration number, and quality of approximations.

**Example 5.1** We investigate a competitive market model (as in Section 4.1) with ten decision makers indexed by  $i = 1, \dots, 10$ , and the corresponding selling quantities  $x_i$  for  $i = 1, 2, \dots, 10$ . Here  $\{x_i, i = 1, 2, \dots, 10\}$  are deterministic which implies that the decision makers need to make their decisions before the realization of the market uncertainty  $\xi := (\xi_1, \xi_2, \dots, \xi_{12})^\top \in \Xi$ , where  $\xi_j$  for  $j = 1, \dots, 12$  represents the market uncertainty's influences on the market prices and each decision maker's production cost function.

Here, the market price function is piecewise linear and can be presented as

$$p(x, \xi) = \max_{t=0,1,\dots,5} \{p_t(x, \xi)\},$$

where  $p_0(x, \xi) \equiv 0$  and  $p_t(x, \xi) = a_t(\xi_1) - b_t(\xi_2)x$  for  $t = 1, 2, \dots, 5$  and  $x = \sum_{i=1}^{10} x_i$ , where the pairs  $(a_t(\xi_1), b_t(\xi_2))$  is of a linear form<sup>1</sup>  $(a'_t + a''_t \xi_1, b'_t + b''_t \xi_2)$  with the values of  $a'_t$  ranging from 90 to 900,  $a''_t$  ranging from 6 to 10,  $b'_t$  ranging from 30 to 400 and  $b''_t$  ranging from 10 to 14. In the example  $p(x, \xi)$  is still a decreasing and concave function of  $x$  for each  $\xi$ . With the notation, we have that by determining to sell  $x_i$  units of product into the market, decision maker  $i$ 's expected profit before knowing the realization of the market demand can be written as

$$F_i(x_i, x_{-i}) = \mathbb{E}[x_i p(x, \xi) - c_i(\xi_{i+2})x_i], \quad (5.28)$$

wherein cost co-efficient  $c_i(\xi_{i+2}) = c'_i + c''_i \xi_{i+2}$  for  $i = 1, 2, \dots, 10$  with the parameters  $c'_i$  ranging from 5 to 50 and  $c''_i$  ranging from 15 to 24 with even intervals.

## 5.1 Resampling

We set the sample size schedule in the VSDA as linear with the form  $N_k = n_0 k$  for  $k = 1, 2, \dots$ . We perform the static analysis on incorporating resampling scheme in [2] into the VSDA by varying the resampling frequency  $k_f$  from 1 to 50 and the schedule parameter  $n_0$  from 50 to 500. We also set the error tolerance in the VSDA being  $\epsilon = 0.001$ . In Table 1, we list the numerical results on the overall number of iterations and computation times used in the VSDA before achieving the stop criteria. In addition, we also list the error of the solution yielded by the algorithm as the *true SNE* denoted by  $x^*$ <sup>2</sup>. For each set  $k_f$ , we report the average solution time, average iterations and average solution errors over all 30 runs (in seconds).

From Table 1, we have the following observations: First, the VSDA, with a more frequent resampling procedure (i.e. smaller  $k_f$ ), will return a solution closer to  $x^*$ . For example, for the case with  $n_0 = 50$ , the deviation of  $x^N[K]$  to  $x^*$  varies from 1.0678 to 0.0748 when  $k_f$  reduces from 50 to 1. The underlying reason can be easily attained, that is, a more frequent resampling procedure (i.e. smaller  $k_f$ ) can help the VSDA more easily move to a different sample path. Note that the error tolerance  $\epsilon$  set in the VSDA only control the stop of the algorithm by measuring the difference between two consecutive iterates, i.e.  $x^N[k]$  and  $x^N[k+1]$ , but doesn't guarantee that the accuracy of the yielded iterate  $x^N[K]$  to  $x^*$  is better than  $\epsilon$ . Theoretically speaking, any sample with a finite size can be deemed as a *bad* sample path, since it will eventually result in the algorithm generating a biased solution. The motivation of a resampling scheme is to avoid the algorithm being trapped by these bad sample paths. We need to emphasize that the performance is averaged from 30 runs and is not realistic to show the strictly monotonicity in  $\|x^N[K] - x^*\|$  particularly in the range where  $\|x^N[K] - x^*\|$  is not sensitive to the change of  $k_f$ , but we still can observe a clear trend on this point.

Secondly, for fixed  $n_0$ , there exists a trend that smaller  $k_f$  makes the number of iterations and

---

<sup>1</sup>In the numerical test, we use the quadratic smoothing approximate function in Example 1 in [36] with the smoothing parameter being 0.001.

<sup>2</sup>The true equilibrium is obtained by using the VSDA with a sample with size being  $10^7$  at each iteration for 50th iterations.

Table 1: Average performance of the resampling scheme in the VSDA.

$n_0$	$k_f$	# of Iterations	Solution Time (secs)	$\ x^N[K] - x^*\ $
50	1	1691.1	130.05	0.0748
	2	827.2	29.38	0.2004
	5	269.5	7.32	0.4293
	10	161.4	4.43	0.8315
	20	141.8	3.12	0.6848
	50	132.7	2.68	1.0678
100	1	1207.8	95.69	0.1089
	2	566.7	23.31	0.1883
	5	218.2	5.89	0.3469
	10	135.7	3.31	0.6475
	20	112.4	3.14	0.5815
	50	117.0	4.99	0.6448
200	1	648.2	44.36	0.0885
	2	377.9	14.69	0.1347
	5	127.4	4.07	0.3007
	10	93.4	2.69	0.5000
	20	95.9	3.37	0.5105
	50	77.5	2.25	0.4904
500	1	475.8	68.19	0.0624
	2	272.7	22.47	0.1110
	5	102.4	3.99	0.1787
	10	70.4	2.64	0.3278
	20	58.6	1.98	0.3630
	50	64.1	2.05	0.3288

the solution time increase. For example, for  $n_0 = 50$ , the number of iterations was increased from an average of 132.7 to more than 1600, and the solution time was increased from an average of 2.68 seconds to 130.05 seconds, when the  $k_f$  was decreased from 50 to 1. The underlying reason for the phenomena is twofold: a) a smaller  $k_f$  introduces more sampling efforts in performing the algorithm, and b) more importantly, every time the resampling procedure occurs, the VSDA moves to a significantly different sample path, and hence a certain period is needed for the iterate  $x^N[k]$  to converge. We use the following figure to illustrate this point, where the sequence  $\{x^N[k], k = 1, 2, \dots\}$  of the VSDA is generated with parameter  $k_f$  being 1, 2, 20 and 50 respectively. From Figure 7, we can clearly see that for the case with  $k_f$  being 1 and 2, there exist fluctuations in the sequence  $\{x^N[k], k = 1, 2, \dots\}$  which is caused by resampling. Moreover, from the results in Table 1, we can also observe that, despite this trend, the number of iterations and the solution time don't change much when the value of  $k_f$  is increased beyond 10. Therefore, for this problem, we can make  $k_f$  increased from 1 to 10 to trade off the accuracy of the solution and a quicker solution time, where a further increase of  $k_f$  from 10 can not promote

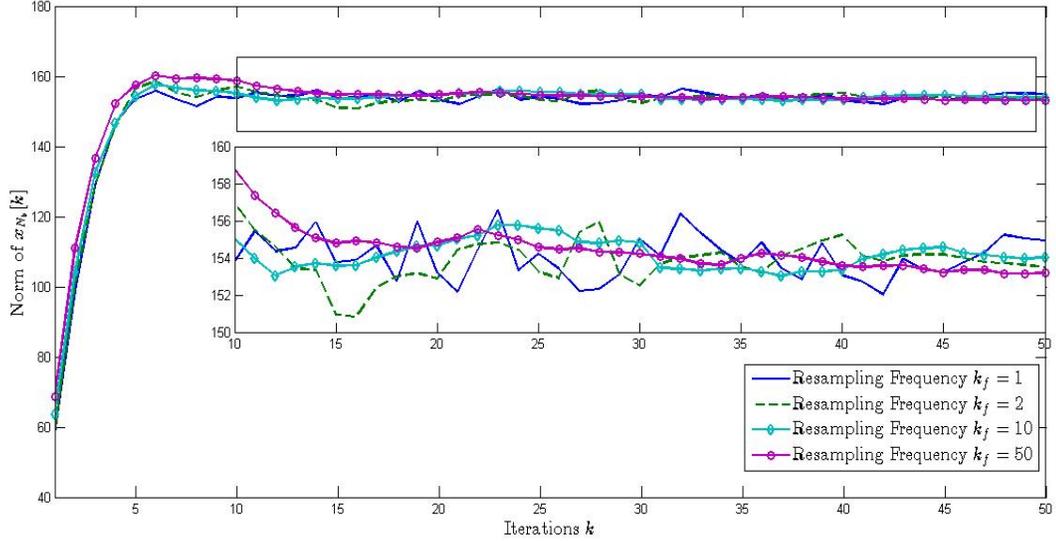


Figure 7: Sequence of  $x_{N_k}[k]$  yielded in the VSDA with  $k_f$  ranging from 1 to 100.

the solution time prominently.

Now, let us look at how the VSDA performance changes along with the sample size schedule. In this example, we can use the coefficient  $n_0$  to characterize how the sample size increases. When the value of  $n_0$  is increased from 50 to 200, we can observe that the solution times are accordingly decreased. This fact implies that though the increase of  $n_0$  introduces additional time for sampling, the number of iterations and hence the solution time will be correspondingly reduced due to more accurate approximation. However, when  $n_0$  is increased from 200 to 500, despite the decreased number of iterations, the solution time can not be further reduced due to the increasing requirement of the sampling effort.

## 5.2 Cumulative sampling

We set the sample size schedule in the VSDA as linear with the form  $N_k = n_0 k$  for  $k = 1, 2, \dots$ . We perform the static analysis on integrating the resampling scheme in [11] into the VSDA by varying the cumulative sampling parameter  $\beta$  from 0 to 1 and the schedule parameter  $n_0$  from 50 to 500. We also set the tolerance  $\epsilon = 0.001$ . In Table 1, we list the numerical results on the overall number of iterations and computation times used in the VSDA for reaching the tolerance  $\epsilon$ . In addition, we also list the error of the solution yielded by the algorithm to  $x^*$ . For each set  $\beta$ , we run the algorithm for 30 times and record average performances for the aforementioned parameters.

From the results in Table 2 we make the following observations of the cumulative sampling scheme. First, along with the decrease of  $\beta$ , the proportion of observations differing from the

Table 2: Average performance of the cumulative sampling scheme in the VSDA.

$n_0$	$\beta$	# of Iterations	Solution Time (secs)	$\ x^N[K] - x^*\ $
50	1.0	1691.0	130.05	0.0748
	0.8	1425.5	64.67	0.1412
	0.6	1244.7	50.70	0.1426
	0.4	1068.6	44.50	0.2481
	0.2	631.7	21.57	0.5322
	0.0	133.6	2.75	0.9502
100	1.0	1207.8	95.69	0.1089
	0.8	1008.8	69.86	0.1572
	0.6	779.6	54.70	0.1348
	0.4	640.1	27.55	0.2274
	0.2	457.0	23.61	0.2926
	0.0	112.5	4.59	0.5997
200	1.0	648.2	44.36	0.0855
	0.8	608.5	42.26	0.1209
	0.6	604.9	37.89	0.0855
	0.4	463.3	28.43	0.2299
	0.2	242.0	9.90	0.2837
	0.0	77.9	2.46	0.4912
500	1.0	475.8	68.19	0.0624
	0.8	377.6	40.04	0.1227
	0.6	364.9	35.14	0.1151
	0.4	230.6	18.08	0.1837
	0.2	175.7	9.40	0.2334
	0.0	69.1	2.27	0.3461

preceding iterations, the error of solutions  $\|x^N[K] - x^*\|$  increases. This fact implies that a higher parameter  $\beta$  can help the algorithm more easily move to a different sample path, and hence avoid being trapped in a “bad” sample path. For example, when  $n_0$  is 50, the error of final solution yielded by the VSDA is increased from 0.0748 to 0.9502 when the parameter  $\beta$  is increased from 0 to 1. Note that we must emphasize that the performance is averaged from 30 runs and is not realistic to show the strict monotonicity in  $\|x^N[K] - x^*\|$  particularly in the range where  $\|x^N[K] - x^*\|$  is not sensitive to the change of  $\beta$ .

Secondly, the increase of  $\beta$  enables less computation effort in sampling new observations, and thus results in fewer numbers of iterations and solution time until the error tolerance condition is satisfied. For example, when  $n_0$  is 100, the number of iterations and the solution time are decreased from an average of 1207.8 to 112.5 and 95.69 to 4.59 seconds when the parameter  $\beta$  is increased from 0 to 1.

Now, let us look at how the VSDA performance changes along with the sample size schedule.

In this example, we can use the coefficient  $n_0$  to characterize how the sample size increases. When the value of  $n_0$  is increased from 50 to 200, we can observe that the solution times are accordingly decreased. This fact implies that though the increase of  $n_0$  introduces additional time for sampling, the number of iterations and hence the solution time will be correspondingly reduced due to more accurate approximation. However, when  $n_0$  is increased from 200 to 500, despite the decreased number of iterations, the solution time can not be further reduced due to the increasing requirement of the sampling effort.

## 6 Conclusion

The variable sample distributed algorithm proposed in this paper solves the stochastic Nash equilibrium problem with stability conditions. By exploiting the contraction mapping structure inherent in the sequential sampling procedure, we have presented a general framework to show convergence of a certain class of variable sample distributed algorithm. In particular, we have established conditions on the schedule of sample size  $\{N_k\}$  that ensure convergence of the estimators. We have also showed the finite termination for the variable sample distributed algorithm with a positive error tolerance parameter. To illustrate the practice of the algorithm, in Sections 4 and 5, we have extended our analysis to the cases where different decision makers have their own samples in the procedure of variable sample distributed algorithm, and show the existence of the “bucket effect” in solving the stochastic Nash equilibrium. This result provided a guideline for the direction of the effort to be made when decision makers in a market have different capabilities in sampling observations. In addition, we have also exploited variations of the algorithm, where different schemes of resampling or cumulative sampling are implemented, and showed that how the algorithm performance varies with respect to different resampling frequency or cumulation sampling proportion.

There also exist some open questions. First of all, the study on how the statistics of the accumulate points yielded by the variable sample distributed algorithm, such as mean, variance and higher-order moments, is very important to provide intuitions for the behavior of the algorithm. Unfortunately, the calculations of these statistics are not inherent with the evolution of the algorithm, so some further study is required. Secondly, on the practical side, the implementation of the algorithm require the development of more effective schedules on the sample sizes used in each iteration of the algorithm, which might be relevant to the distribution or at least some statistics of the underlying uncertainty.

## Appendix

**Proof of Lemma 3.1.** According to the statement of Lemma 4.1, at the beginning of an iteration  $k$  in Algorithm 1, we set  $x[k] = x$ . Then from the contraction property for the VSDA,

we have the iterate  $x_i[k+1]$  for  $i \in I$  yielded by Algorithm 1 can be obtained by solving a set of optimization problems as follows,

$$x_i[k+1] = \arg \min_{x'_i} F_i(x'_i, x_{-i}^N[k]) = \arg \min_{x'_i} \mathbb{E} [f_i(x'_i, x_{-i}, \xi)]. \quad (6.29)$$

From the statement in Lemma 3.1, we have there exists at least one  $i$  satisfying

$$\nabla_{x_i} \mathbb{E} [f_i(x_i, x_{-i}, \xi)] > \epsilon_g,$$

which implies that  $x_i$  is not the solution of problem (6.29), that is

$$\|\mathbb{E} [f_i(x_i, x_{-i}, \xi)] - \mathbb{E} [f_i(x_i[k+1], x_{-i}, \xi)]\| \geq \hat{\epsilon}, \quad (6.30)$$

for some positive value  $\hat{\epsilon}$ . Due to the Lipschitz property of  $\mathbb{E} [f_i(x_i, x_{-i}, \xi)]$ , there exists a positive constant  $\epsilon_\Delta$  such that

$$\|x - h_p(x)\| = \|x - x[k+1]\| > 2\epsilon_\Delta.$$

From (6.30) we have the value of  $\epsilon_\Delta$  is determined by  $\epsilon_g$ ,  $M$  and  $x$  where are all independent of the value  $\xi$  and the number of iteration  $k$ .

On the other hand, let us consider the Algorithm VSDA. In the statement of the lemma, the value  $h_p^{N_{k+1}}(x)$  is obtained from

$$x_i^N[k+1] = \arg \min_{x'_i} \frac{1}{N_{k+1}} \sum_{n=1}^{N_{k+1}} f_i(x'_i, x_{-i}, \xi_n^k).$$

for every  $i \in I$ . By checking the conditions A1 and A2 in the statement of [30, Theorem 10] or [37], we have

$$\tilde{\mathbb{P}} \left\{ \|x_i^N[k+1] - x_i[k+1]\| \geq \epsilon_\Delta \right\} \leq 2e^{-\gamma(\epsilon_\Delta)N_{k+1}},$$

which shows that

$$\tilde{\mathbb{P}} \left\{ \|h_p^{N_{k+1}}(x) - h_p(x)\| \geq \epsilon_\Delta \right\} \leq 2Me^{-\gamma(\epsilon_\Delta)N_{k+1}}.$$

Consequently, we can establish the following inequality

$$\begin{aligned} & \tilde{\mathbb{P}} \left\{ \|h_p^{N_{k+1}}(x) - x\| > \epsilon_\Delta \right\} \\ & \geq \tilde{\mathbb{P}} \left\{ \|h_p(x) - x\| - \|h_p^{N_{k+1}}(x) - h_p(x)\| > \epsilon_\Delta \right\} \\ & \geq \tilde{\mathbb{P}} \left\{ 2\epsilon_\Delta - \|h_p^{N_{k+1}}(x) - h_p(x)\| > \epsilon_\Delta \right\} \\ & = \tilde{\mathbb{P}} \left\{ \|h_p^{N_{k+1}}(x) - h_p(x)\| < \epsilon_\Delta \right\} \\ & \geq 1 - 2Me^{-\gamma(\epsilon_\Delta)N_k}. \end{aligned}$$

We complete the proof. ■

**Proof for Theorem 3.2.** We prove the statement (3.23) in Theorem 3.2 by contradiction. Suppose that (3.23) is not satisfied, then we have there exists an  $\alpha \in (0, 1]$  and a positive value  $\epsilon_g$  such that

$$\tilde{\mathbb{P}} \left\{ \liminf_{k \rightarrow \infty} \|x^N[k] - x^*\| \geq \epsilon_g \right\} = \alpha,$$

where  $x^* \in X^*$ . This implies that there exists a subset  $\Xi_r \subset \tilde{\Xi}_r$  such that (a) the measure of  $\Xi_r$  is non-zero; (b) for every  $\xi_r \in \Xi_r$ , there exists a positive integer number  $k(\xi_r) < \infty$  such that for all  $k \geq k(\xi_r)$

$$\|x^N[k] - x^*\| \geq \epsilon_g.$$

On the other hand, from results in Theorem 3.1 and Lemma 3.1, we have that there exists a  $k_1(\xi_r) < \infty$  such that for all  $k \geq k_1(\xi_r)$

$$\tilde{\mathbb{P}} \left\{ \left\| x^N[k] - h_p^{N_{k+1}}(x^N[k]) \right\| \geq \epsilon_\Delta \right\} \leq 2M e^{-\gamma(\epsilon_\Delta)N_{k+1}},$$

where  $\epsilon_\Delta := (1 - \kappa_h)\epsilon_g/2$ . From the condition (3.16) in Theorem 3.1, we have the event

$$\left\{ \left\| x^N[k] - h_p^{N_{k+1}}(x^N[k]) \right\| \geq \epsilon_\Delta \right\}$$

occurs only finitely times.

Moreover,

$$\begin{aligned} \|x^N[k] - x^*\| &\leq \left\| x^N[k] - h_p^{N_{k+1}}(x^N[k]) \right\| + \left\| h_p(x^N[k]) - h_p^{N_{k+1}}(x^N[k]) \right\| \\ &\quad + \left\| h_p(x^N[k]) - h_p(x^*) \right\| + \|x^* - h_p(x^*)\|. \end{aligned}$$

where the last term in the right hand side of the above inequality is zero. Consequently, we have

$$\begin{aligned} &\left\| x^N[k] - h_p^{N_{k+1}}(x^N[k]) \right\| + \left\| h_p(x^N[k]) - h_p^{N_{k+1}}(x^N[k]) \right\| \\ &\geq \|x^N[k] - x^*\| - \left\| h_p(x^N[k]) - h_p(x^*) \right\| \geq (1 - \kappa_h) \|x^N[k] - x^*\|. \end{aligned}$$

Note that the second term in the left hand side of the above inequality is controlled by  $N_{k+1}$ . Due to the condition (3.16), we have that there exists a  $k^0$  such that  $\{N_k, k = k^1, k^1 + 1, \dots\}$  is a strictly increasing sequence without upper bound. Hence, there exists a  $k^1 \geq k^0$ , such that, (a)  $\tilde{\mathbb{P}}\{\Xi'\} \geq 1 - \tilde{\mathbb{P}}\{\Xi_r\}/2$ ; (b) for every  $\xi_r \in \Xi'$ , there exists a positive integer number  $k'(\xi_r) < \infty$  such that for all  $k \geq k'(\xi_r)$

$$\left\| h_p(x^N[k]) - h_p^{N_{k+1}}(x^N[k]) \right\| \leq (1 - \kappa_h)\epsilon_g/2.$$

We thus have for  $k \geq k'(\xi_r)$

$$\begin{aligned} &\left\| x^N[k] - h_p^{N_{k+1}}(x^N[k]) \right\| \\ &\geq (1 - \kappa_h) \|x^N[k] - x^*\| - \left\| h_p(x^N[k]) - h_p^{N_{k+1}}(x^N[k]) \right\| \\ &\geq (1 - \kappa_h)\epsilon_g/2. \end{aligned}$$

Therefore, for all  $\xi \in \Xi_r \cap \Xi'$ ,

$$\left\| x^N[k] - h_p^{N_{k+1}}(x^N[k]) \right\| \geq (1 - \kappa_h)\epsilon_g/2 = \epsilon_\Delta,$$

for  $k$  being greater than  $\max\{k^0, k'(\xi'), k_r(\xi_r)\}$ . Since,  $\tilde{\mathbb{P}}\{\Xi'\} \geq 1 - \tilde{\mathbb{P}}\{\Xi_r\}/2$ , we have  $\tilde{\mathbb{P}}\{\Xi'\} + \tilde{\mathbb{P}}\{\Xi_r\} > 1$ , the measure of  $\Xi_r \cap \Xi'$  is non-zero, which conflicts with the fact that the event

$$\left\{ \left\| x^N[k] - h_p^{N_{k+1}}(x^N[k]) \right\| \geq \epsilon_\Delta \right\}$$

occurs only finitely times for some  $\xi \in \Xi_r$ . We complete the proof. ■

## References

- [1] Z. Artstein and R. J-B Wets. 1995. Consistency of minimizers and the SLLN for stochastic programs, *Journal of Convex Analysis* **2** 1–17.
- [2] G. Bayraksan and D. P. Morton. 2011. A sequential sampling procedure for stochastic programming, *Operations Research* **11** 898–913.
- [3] Bertsekas, D. P., 1983. Distributed asynchronous computation of fixed points, *Mathematical Programming* **27** 107–120.
- [4] Bertsekas, D. P., and Tsitsiklis, J. N., 1991. Some aspects of parallel and distributed iterative algorithms: A survey. *Automatica* **27** 3-21.
- [5] M.W. Brantley, L.H. Lee, C.H. Chen and A. Chen, Efficient simulation budget allocation with regression, 2013. *IIE Transactions*, **45**, 291-308.
- [6] Bozma, H. I., 1996. Computation of Nash equilibria: Admissibility of parallel gradient descent. *Journal of Optimization Theory and Applications* **90** (1) 45-61.
- [7] Clarke, F. H., 1983. *Optimization and nonsmooth analysis* Wiley, New York.
- [8] Dai, L., Chen, C. H., and Birge, J. R., 2000. Convergence properties of two stage stochastic programming. *Journal of Optimization Theory and Applications* **106** 489-509.
- [9] DeMiguel, V., and Xu, H., 2009. A stochastic multiple leader Stackelberg model: Analysis, computation, and application. *Operations Research* **57** 1220-1235.
- [10] Deng, G., and Ferris M. C., Variable-number sample-path optimization. *Mathematical Programming* **117** 81-109.
- [11] T. Homem-de-Mello. Variable-sample methods for stochastic optimization, 2003. *ACM Transactions on Modelling and Computer Simulation* **13** 108-133.
- [12] T. Homem-de-Mello and G. Bayraksan, Monte Carlo sampling based methods for stochastic optimization. *Manuscript Optimization-Online*, 2013.

- [13] Z. Hu, L. J. Hong, L. Zhang. A smooth Monte Carlo approach to joint chance constrained programs, 2013. *IIE Transactions* **45** 716-735.
- [14] S. R. Hunter, and R. Pasupathy. Optimal sampling laws for stochastically constrained simulation optimization, 2012. *INFORMS Journal on Computing*, In press.
- [15] Jiang, H., and Pang, Z., Network capacity management under competition, 2010. *Computational Optimization and Applications*, in press, published online, July.
- [16] L.H. Lee, E.P. Chew, S.Y. Teng, and D. Goldsman, Finding the non-dominated Pareto set for multi-objective simulation models, 2010. *IIE Transactions*, **42**, 656-674.
- [17] L.H. Lee, N.A. Pujowidianto, L. Li, C.H. Chen and C.M. Yap, Approximate simulation budget allocation for selecting the best design in the presence of stochastic constraints, 2012. *IEEE Transactions on Automatic Control*, **57**, 2940-2945.
- [18] Kim, S., and Zhang, D., 2010. Convergence properties of direct search methods for stochastic optimization. *Winter Simulation Conference 2010* 1003-1011.
- [19] Li, S., and Basar, T., 1987. Distributed algorithms for the computation of noncooperative equilibria. *Automatica* **23** 523-533.
- [20] Linderoth, J., Shapiro, A., and Wright, S., 2006. The empirical behavior of sampling methods for stochastic programming. *Annals of Operations Research* **142** 215-241.
- [21] Papavassilopoulos, G. P., 1986. Adaptive repeated Nash games. *SIAM Journal on Control and Optimization* **24** 821–834.
- [22] R. Pasupathy, S. R. Hunter, N. Pujowidianto, , L. H. Lee and C.-H. Chen. SCORE: Stochastic Constrained Optimization using Rate Estimators, 2013. Manuscript, Virginia Tech.
- [23] R. Pasupathy. On choosing parameters in retrospective approximation algorithms for stochastic root finding and simulation-optimization, 2010. *Operations Research* **58**, 889–901.
- [24] R. Pasupathy and B.W. Schmeiser. Retrospective-approximation algorithms for the multi-dimensional stochastic root-finding problem, 2008. *ACM Transactions on Modeling and Computer Simulation* **19**, 1–35.
- [25] Robinson, S. M., 1996. Analysis of sample-path optimization. *Mathematics of Operations Research* **21** 513–528.
- [26] Rosen, J. B., 1965. Existence and uniqueness of equilibrium points for concave N-person games. *Econometrica* **33** 520-534.
- [27] J. O. Royset and R. Szechtman, Optimal budget allocation for sample average approximation, 2013. *Operations Research*, **61** 762-776.

- [28] J. O. Royset, On sample size control in sample average approximations for Solving Smooth Stochastic Programs, 2013. *Computational Optimization and Applications*, **55**, 265-309.
- [29] Shapiro, A., and T. Homem-de-Mello, 2000. On rate of convergence of Monte Carlo approximations of stochastic programs. *SIAM Journal on Optimization* **11** 70-86.
- [30] Shapiro, A., 2003. Monte Carlo sampling methods, Stochastic Programming, *Handbooks in Operations Research and Management Science* **10** Elsevier.
- [31] Shapiro, A., and Xu, H., 2008. Stochastic mathematical programs with equilibrium constraints, modeling and sample average approximation. *Optimization* **57** 395-418.
- [32] Vorobeychik, Y., 2010. Probabilistic analysis of simulation-based games, *ACM Transaction on Modeling and Computer Simulation* **20** 16-40.
- [33] Wang, J., Chiu, D., and Lui, C., 2008. A game theoretic analysis of the implications of overlay network traffic on ISP peering. *Computer Networks* **52** 2961-2974.
- [34] Xu, H., and Meng, F., 2007. Convergence analysis of sample average approximation methods for a class of stochastic mathematical programs with equality constraints. *Mathematics of Operations Research* **32** 648-668.
- [35] Xu, H., and Zhang, D., 2012. Stochastic Nash equilibrium problems: Sample average approximation and applications. *Computational Optimization and Applications*, in press 2012.
- [36] Xu, H., and Zhang, D., 2009. Smooth sample average approximation of stationary points in nonsmooth stochastic optimization and applications, *Mathematical Programming* **119** 371-401.
- [37] Xu, H., 2010. Uniform exponential convergence of sample average random functions under general sampling with applications in stochastic programming. *Journal of Mathematical Analysis and Applications* **368** 692-710.
- [38] Yao, J., Oren, S. S., and Adler, I., 2007. Two-settlement electricity markets with price caps and Cournot generation firms. *European Journal of Operational Research* **181** 1279-1296.
- [39] Yuan, Y., 2011. A trust region algorithm for Nash equilibrium problems. *Pacific Journal of Optimization* **7** 125-138.
- [40] Zhang, D., and Xu, H., 2012. Two stage stochastic equilibrium problems with equilibrium constraints: Modeling and numerical schemes. *Optimization*, in press, 2012.
- [41] Zhang, D., Xu, H., and Wu, Y., 2010. A stochastic two stage equilibrium model for electricity markets with two way contracts. *Mathematical Methods of Operations Research* **71** 1-45.
- [42] Zhang, D., and Lin, G., 2012. A bilevel direct search method for leader-follower optimization problems and applications. *Computers and Operations Research* Revision.

- [43] Zhou, J., Lam, WHK, and Heydecker, B, G, 2005. The generalized Nash equilibrium model for oligopolistic transit market with elastic demand. *Transportation Research Part B* **39** 519-544.