XOR-SGD: Provable Convex Stochastic Optimization for Decision-making under Uncertainty

Fan Ding[1] **Yexiang Xue**[1]

[1]Computer Science Dept., Purdue University, West Lafayette, Indiana, USA

Abstract

Many decision-making problems under uncertainty can be formulated as convex stochastic optimization, which minimizes a convex objective in expectation across exponentially many probabilistic scenarios. Despite its convexity, evaluating the objective function is #P-hard. Previous approaches use samples from MCMC and its variants to approximate the objective function but have a slow mixing rate. We present XOR-SGD, a stochastic gradient descent (SGD) approach guaranteed to converge to solutions that are at most a constant away from the true optimum in linear number of iterations. XOR-SGD harnesses XOR-sampling, which reduces the sample approximation of the expectation into queries of NP oracles via hashing and projection. We evaluate XOR-SGD on two real-world applications. The first stochastic inventory management problem searches for a robust inventory management plan in preparation for the virus pandemics, natural disasters, etc. The second network design problem decides an optimal land conservation plan which promotes the free movement of wild-life animals. We show that our approach finds better solutions with drastically fewer samples needed compared to a couple of state-of-the-art solvers.

1 INTRODUCTION

Decision-making in an uncertain world requires solving stochastic optimization problems which optimizes the expectation of a stochastic outcome across multiple probabilistic scenarios. Indeed, stochastic optimization problems have attracted much research attention given its wide applicability in finance, control, robotics, management science, operational research, and conservation (Sodomka et al. [2007], Ziukov [2016], Gomes et al. [2019]). Advancements made to address this problem will have ramifications in many domains. In mathematical form, a stochastic optimization problem is:

$$\begin{aligned} &\min_{x} \ \mathbb{E}_{\theta\sim Pr(\theta)} \ f(x,\theta), \\ &s.t. \quad \forall i,\ h_i(x)=0 \text{ and } \forall j,\ g_j(x)\leq 0. \end{aligned} \tag{1}$$

In this paper, we focus on *convex* stochastic optimization problems. More precisely, we require the function $f(x,\theta)$ to be convex with respect to x. $g_j(x)$ are convex functions and $h_i(x)$ are linear. Variable θ is sampled from distribution $Pr(\theta)$, which is represented as a Markov random field (MRF) in this paper. Despite our limited scope, problems in equation 1 are still highly intractable (#P-hard). The main source of intractability comes from the computation of the expectation over a general probability distribution, a common operator in probabilistic inference. Computing such an expectation is #P-complete, where #P denotes the complexity class that counts the number of accepting paths of a polynomial-time non-deterministic Turing machine.

Theorem 1. *The convex stochastic optimization problem in Equation 1 is #P-hard.*

Many problems can be formulated as convex stochastic optimizations. In this paper, we focus on attacking two real-world problems. The first is the inventory management problem in operational research. In this problem, managers have to decide the amount of each material to buy at the beginning of the season to meet the production demand. He should buy neither too much because of the limited inventory space, nor too little since a back order later would cost more. The manager, therefore, has to place a purchase order x, which minimizes the expected cost $\mathbb{E}_\theta[f(x,\theta)]$, taking into account of various stochastic events θ, such as materials price fluctuations, supply chain complications, virus pandemic, etc. Our second problem is a network design problem, where we need to decide an optimal investment plan under a fixed budget to increase the landscape connectivity. In

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

biodiversity conservation, increasing landscape connectivity facilitates the free movement of wildlife animals, hence accelerating their gene flows. In disaster preparation, increased connectivity allows rescue teams to reach their sites faster. Interestingly, a commonly used connectivity measure, the commuting time of a random walk model, is convex (Ghosh et al. [2008]). Therefore, the network design problem can also be cast as a convex stochastic optimization problem.

We present XOR-SGD, a simple stochastic gradient descent (SGD) algorithm, which is ***guaranteed to converge to a solution that is within a vanishing constant away from the true optimum in linear number of SGD iterations.*** The linear convergence rate towards the optimum is rather surprising, considering the #P-hardness of the problem and the simpleness of the SGD. The key of XOR-SGD is to draw a representative set of samples from $Pr(\theta)$ which yield an accurate estimation of the gradient direction. Common sampling approaches, such as MCMC, cannot serve our purpose, because of the exponentially many steps to mix. Belief Propagation (BP) has difficulties in dealing with multi-modal distributions. Recently proposed BPChain (Fan and Xue [2020]) uses an inference chain to draw samples sequentially. Nevertheless, the errors tend to propagate.

Our XOR-SGD leverages XOR-Sampling, a recently proposed sampling scheme with a constant approximation guarantee, which reduces the sampling problem into queries of NP oracles via hashing and projection. Indeed, XOR-SGD requires accessing NP-oracle queries at each iteration. Nevertheless, our contribution is built on the recent success of solving NP-complete problems, where several industrial sized problems are successfully solved by latest constraint reasoning solvers. In another view, our contribution extends the success of solving NP-complete problems to problems with even higher complexity, namely, #P-hard problems. Our key contribution is the extension of classical convergence analysis of SGD on convex problems, where we show that a constant multiplicative bound on the expectation of the gradient direction is sufficient to bound the final result against the true optimum (Theorem 3). Our theoretic contribution does not depend on the unbiasness of the gradients, which was a necessary condition in previous analysis.

XOR-SGD was motivated by *Sample Average Approximation* (SAA) (Kleywegt et al. [2002], Verweij et al. [2003]), which is widely used to solve stochastic optimization problems. On learning probabilistic graphical models, stochastic optimization is related to *the Marginal Maximum-a-posterior (MMAP) problem* (Xue et al. [2016], Liu and Ihler [2013], Marinescu et al. [2014], Mauá and de Campos [2012], Marinescu et al. [2015], Domke [2013]). These problems can be formulated as (albeit non-convex) stochastic optimization problems. *Convergence analysis of gradient descent* has been studied for both convex and non-convex functions (Wang et al. [2013], Dubey et al. [2016], Agarwal et al. [2017], Lee et al. [2015], Ruder [2016], Jin et al. [2017], Ge et al. [2015]). Recently several algorithms (Duchi et al. [2011], Hinton et al. [2012], Kingma and Ba [2014], Duchi et al. [2018], Allen-Zhu [2017, 2018]) were proposed to accelerate the convergence rate of SGD. They require an unbiased estimation of either the gradient or the momentum. Our XOR-SGD was derived without this assumption. *Probabilistic inference via hashing and randomization* is proposed for both sampling (Ermon et al. [2013b], Ivrii et al. [2015]), counting (Gomes et al. [2007a], Ding et al. [2019]), and marginal inference problems (Ermon et al. [2013a], Kuck et al. [2019], Chakraborty et al. [2014, 2015], Belle et al. [2015]) with constant approximation guarantees. *Inventory management* is a classic problem for supply chain management in operational research (Ziukov [2016]) where SAA is often used (Shapiro and Philpott [2007]), nonetheless has no formal guarantees. Previously a few approaches have been proposed for *network design* (Sheldon et al. [2012], Wu et al. [2017]). We consider optimizing the commuting time, which is a more challenging objective and to our knowledge, no prior work derives algorithms with guarantees.

Experimental results reveal that XOR-SGD is effective in optimizing constrained convex stochastic functions. XOR-SGD outperforms competing solvers which run SGD with either MCMC, BP or BPChain samplers on both the inventory management and the network design problems on real-world data under various conditions. In particular, ***XOR-SGD accessing merely 60 XOR samples finds better solutions than SGD accessing 20,000 MCMC samples for the inventory management problem. XOR-SGD accessing 40 XOR samples outperforms SGD accessing 20,000 MCMC samples for the network design problem.*** Meanwhile, XOR-SGD ***runs faster*** than SGD with MCMC Sampling. XOR-SGD with 40 samples takes 1 minute 40 seconds per SGD iteration, while SGD with 20,000 MCMC samples needs 2.5 minutes for the network design problem. See the experiments section for more details.

2 PRELIMINARIES

2.1 PROBABILISTIC MODELS

In this paper, we mainly use Markov Random Field as the probability distribution $Pr(\theta)$. MRF is a general model for the joint distribution of multiple correlated random variables. In a MRF, the probability $Pr(\theta)$ is defined as:

$$Pr(\theta) = \frac{1}{Z} \prod_{\alpha \in \mathscr{I}} \phi_\alpha(\{\theta\}_\alpha). \tag{2}$$

where $\{\theta\}_\alpha$ is a subset of variables in θ that the function ϕ depends on. $\phi_\alpha : \{\theta\}_\alpha \to \mathbb{R}^+$ is a potential function, or commonly referred to as a clique. ϕ_α maps every assignment of variables in $\{\theta\}_\alpha$ to a non-negative real value. $\mathscr{I}$ is an index set and Z is a normalization constant, which ensures that the probability adds up to one: $Z = \sum_{\theta \in \{0,1\}^m} \prod_{\alpha \in \mathscr{I}} \phi_\alpha(\{\theta\}_\alpha)$.

A potential function $\phi_\alpha(\{\theta\}_\alpha)$ defines the correlation between all variables in the subset $\{\theta\}_\alpha$. The structure of the MRF or the set $\mathscr{I}$ can be built from domain knowledge and potential functions can be learned from real-world data. The focus of this paper is not on how to construct the MRF but is how we solve problem in Equation 1 in general when $Pr(\theta)$ is given in the form shown in Equation 2.

2.2 XOR-SAMPLING

In our method, we approximate the otherwise intractable objective in Equation 1 with the empirical mean of a finite number of samples from $Pr(\theta)$ and use SGD to minimize it. Although widely used, MCMC based samplers are known to have a notoriously slow mixing rate. Their variance cannot be controlled effectively and therefore does not lead to algorithms with provable guarantees.

Our XOR-SGD leverages recent advancements in sampling via hashing and randomization. In particular, we embed XOR-Sampling (Ermon et al. [2013b]) into SGD, a sampling scheme which guarantees that the probability of drawing a sample is sandwiched between a constant bound of the true probability. We only present the general idea of XOR-Sampling on unweighted functions here and refer the readers to the paper (Ermon et al. [2013b]) for the weighted case. For the unweighted case, assuming $w(\theta)$ takes binary values, we need to draw samples from the set $\mathscr{W} = \{\theta : w(\theta) = 1\}$ uniformly at random; i.e., suppose $|\mathscr{W}| = 2^l$, then each member in $\mathscr{W}$ should have 2^{-l} probability to be sampled. Following notations from the SAT community, we call one assignment θ_0 which makes $w(\theta_0) = 1$ a "satisfying assignment". XOR-Sampling obtains near-uniform samples by querying a NP oracle to find one satisfying assignment subject to additional randomly generated XOR constraints. Initially, the NP oracle will find one satisfying assignment subject to zero XOR constraints, albeit not at random. We then keep adding XOR constraints. We can prove that in expectation, each newly added XOR constraint rules out approximately half of the satisfying assignments at random. Therefore, if we start with 2^l satisfying assignments in $\mathscr{W}$, after adding l XOR constraints, we will be left with only one satisfying assignment in expectation. We return this assignment as our first sample. Because we can prove that the assignments are ruled out randomly, we can guarantee that the returned assignment must be a randomly chosen one from $\mathscr{W}$. Figure 1 (right) shows an intuitive picture of the unweighted case. See Gomes et al. [2007a,b] for details.

For the weighted case, the authors of Ermon et al. [2013b] present a sampler with a constant approximation guarantee. The idea is to discretize $w(\theta)$ using points that are geometrically far apart, transforming the weighted problem into an unweighted one by introducing additional variables. The discretization scheme is in the supplementary materials. XOR-Sampling draws a sample θ_0 with probability

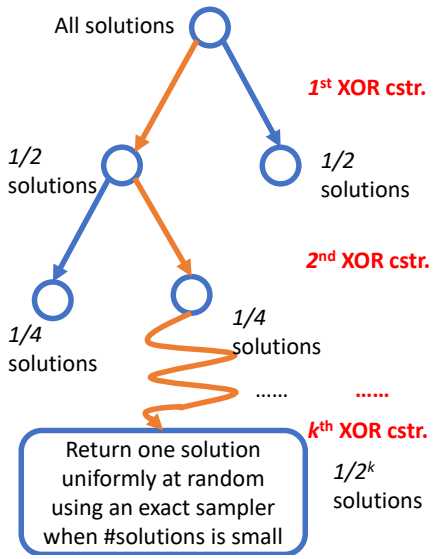

Figure 1: High-level idea of XOR-Sampling. When sampling uniformly at random from the set of solutions $\mathscr{W} = \{\theta : w(\theta) = 1\}$, XOR-Sampling repeatedly adds randomly generated XOR constraints, each of which randomly removes half of the solutions from $\mathscr{W}$. Finally XOR-Sampling returns one solution uniformly at random when the set is small enough after adding k XOR constraints.

proportional to $w(\theta)$, i.e., $Pr(\theta_0) \propto w(\theta)$. Notice that XOR-Sampling only needs unnormalized probability distribution. Our paper uses their results through the following theorem:

Theorem 2. *(Ermon et al. [2013b]) Let $\varepsilon > 0, b > 1$, $P \geq 2$, and $\gamma = \log((P + 2\sqrt{P+1} + 2)/P)$. For any $\alpha \in \mathbb{Z}$, $\alpha > \gamma$, let $c(\alpha, P) = 1 - 2^{\gamma-\alpha}/(1 - \frac{1}{P} - 2^{\gamma-\alpha})^2$. Let $r = 2^b/(2^b - 1), l = \lceil \log_r(2^n/\varepsilon) \rceil, \rho = r^2/(1-\varepsilon), \kappa = 1/c(\alpha, P)$ and bucket $\mathscr{B}_l$ as in Definition 1 in the supplementary materials. Denote $Pr'_s(\theta)$ as distribution of the samples generated by XOR-Sampling$(w, l, b, \delta, P, \alpha)$. Let $\phi : \{0,1\}^n \to \mathbb{R}^+$ be one non-negative function [1] satisfying $\eta_\phi = \max_{\theta \in \mathscr{B}_l} |\phi(\theta)| \leq ||\phi||_\infty$. Then, with probability at least $(1-\delta)c(\alpha, P)2^{-(\gamma+\alpha+1)}\frac{P}{P-1}$, XOR-Sampling succeeds and outputs a sample θ_0. Upon success, each θ_0 is output with probability $Pr'_s(\theta_0)$, which is within a constant factor of the true $Pr(\theta_0)$. Furthermore, the expectation $\mathbb{E}_{Pr(\theta)}[\phi(\theta)]$, can be bounded by the sample estimate:*

$$\frac{1}{\rho\kappa}\mathbb{E}_{Pr'_s(\theta)}[\phi(\theta)] - \varepsilon\eta_\phi \leq \mathbb{E}_{Pr(\theta)}[\phi(\theta)] \leq \rho\kappa\mathbb{E}_{Pr'_s(\theta)}[\phi(\theta)] + \varepsilon\eta_\phi. \tag{3}$$

[1] The theorem requires that ϕ is non-negative, which was held as an implicit assumption in paper Ermon et al. [2013b]. A mirrored result can be obtained when ϕ is non-positive, at which time $\rho\kappa\mathbb{E}_{Pr'_s(\theta)}[\phi(\theta)] - \varepsilon\eta_\phi \leq \mathbb{E}_{Pr(\theta)}[\phi(\theta)] \leq \frac{1}{\rho\kappa}\mathbb{E}_{Pr'_s(\theta)}[\phi(\theta)] + \varepsilon\eta_\phi$.

The value of Theorem 2 mainly comes from the fact that the expectation of function ϕ, $\mathbb{E}_{Pr(\theta)}[\phi(\theta)]$ can be estimated by the empirical mean of the samples generated by XOR-Sampling within a constant approximation bound (Equation 3). The tail $\varepsilon\eta_\phi$ is often negligible. Furthermore, there is a way to set the hyper-parameters of XOR-Sampling which makes $\varepsilon\eta_\phi$ zero (see the supplementary materials). Hence for the rest of the paper, we assume $\varepsilon\eta_\phi$ is zero for our derivations.

Notations For function $f:\mathbb{R}^d\to\mathbb{R}$, we call it L-smooth if for all x,y in the convex domain *dom* f, $f(y)\le f(x)+\nabla f(x)^T(y-x)+\frac{L}{2}||y-x||^2$. Denote $f^+(x)$ as the positive part of function $f(x)$. In other words, $f^+(x)=\max\{f(x),0\}$. $f^-(x)$ is defined similarly. For a random vector x, we define $\mathbb{E}[x]$ as the element-wise expectation and $Var(x)=\mathbb{E}[||x||_2^2]-||\mathbb{E}[x]||_2^2$ where $||\cdot||_2^2$ is the square of l_2 norm.

3 XOR-SGD

In this section we propose XOR-SGD, a new stochastic gradient descent method to solve convex stochastic optimization problems. XOR-SGD converges to solutions that are at most a constant away from the true optimum in linear number of SGD iterations. We first present XOR-SGD for unconstrained optimization (i.e., no constraints in Equation 1), and will extend our result for constrained optimization in the second subsection. The detailed procedure of XOR-SGD for unconstrained optimization is shown in Algorithm 1. To approximate the gradient $\nabla_x\mathbb{E}_\theta f(x_k,\theta)$ at step k, XOR-SGD draws N samples $\theta_1,\dots,\theta_N$ from $Pr(\theta)$ using XOR-Sampling. Because XOR-Sampling has a failure rate, XOR-SGD repeatedly call XOR-Sampling until all N samples are obtained successfully (line 4 – 10). Once $\theta_1,\dots,\theta_N$ are obtained, XOR-SGD uses the empirical mean $\overline{g_k}=\frac{1}{N}\sum_{i=1}^N\nabla_x f(x_k,\theta_i)$ as an approximation for $\nabla_x\mathbb{E}_\theta f(x_k,\theta)$.

Due to Theorem 2, we know $\overline{g_k}$ is bounded within a constant factor of $\nabla_x\mathbb{E}_\theta f(x_k,\theta)$. More precisely, we have $\frac{1}{\rho\kappa}[\nabla_x\mathbb{E}_\theta f(x_k,\theta)]^+\le[\overline{g_k}]^+\le\rho\kappa[\nabla_x\mathbb{E}_\theta f(x_k,\theta)]^+$ and $\frac{1}{\rho\kappa}[\nabla_x\mathbb{E}_\theta f(x_k,\theta)]^-\le[\overline{g_k}]^-\le\rho\kappa[\nabla_x\mathbb{E}_\theta f(x_k,\theta)]^-$. Using this constant approximation, we can prove that the output of XOR-SGD in expectation converges to the true optimum within a small constant distance at a linear speed w.r.t. the number of SGD iterations K (our main result is stated in Theorem 4). To prove Theorem 4, we first prove the bounds on two terms stated in Lemma 1. Notice the inequalities in Lemma 1 hold not only for XOR-SGD, but also for SGD algorithms applied on arbitrary L-smooth convex functions with constant approximate gradients.

Lemma 1. *Let $f:\mathbb{R}^d\to\mathbb{R}$ be a L-smooth convex function and $x^*=argmin_x f(x)$. In iteration k of SGD, g_k is the estimated gradient, i.e., $x_{k+1}=x_k-tg_k$. If there exists a constant $c\ge1$ s.t. $\frac{1}{c}[\nabla f(x_k)]^+\le\mathbb{E}[g_k^+]\le c[\nabla f(x_k)]^+$ and $c[\nabla f(x_k)]^-\le\mathbb{E}[g_k^-]\le\frac{1}{c}[\nabla f(x_k)]^-$, then we have*

$$\frac{1}{c}||\mathbb{E}[g_k]||_2^2\le\langle\nabla f(x_k),\mathbb{E}[g_k]\rangle\le c||\mathbb{E}[g_k]||_2^2.$$

$$\frac{1}{c}\langle\mathbb{E}[g_k],x_k-x^*\rangle\le\langle\nabla f(x_k),x_k-x^*\rangle\le c\langle\mathbb{E}[g_k],x_k-x^*\rangle.$$

Algorithm 1: XOR-SGD

Input: $f(x,\theta),w(\theta),K,N,t,l,b,\delta,P,\alpha$

1 Initialize x_0 for function $f(x,\theta)$
2 **for** $k=0$ *to* K **do**
3 $i\leftarrow1$
4 **while** $i\le N$ **do**
5 $s\leftarrow$ XOR-Sampling$(w(\theta),l,b,\delta,P,\alpha)$
6 **if** $s\ne$ *Failure* **then**
7 $\theta_i\leftarrow s$
8 $i\leftarrow i+1$
9 **end**
10 **end**
11 Compute $\overline{g_k}\leftarrow\frac{1}{N}\sum_{i=1}^N\nabla_x f(x_k,\theta_i)$
12 Compute $x_{k+1}\leftarrow x_k-t\overline{g_k}$
13 **end**
14 $\overline{x_K}\leftarrow\frac{1}{K}\sum_{k=1}^K x_k$
15 **return** $\overline{x_K}$

From Lemma 1 we can see both $\langle\nabla f(x_k),\mathbb{E}[g_k]\rangle$ and $\langle\nabla f(x_k),x_k-x^*\rangle$ can be bounded given the constant approximation bound of the gradient. We leave the proof of Lemma 1 to supplementary materials. Using this lemma, we can derive the following Theorem 3, which bounds the error of SGD on a convex optimization when the estimated gradient g_k in the k-th step resides in a constant bound of $\nabla f(x_k)$. Notice that previous convergence bounds on SGD usually need the gradient estimation to be unbiased, i.e., $\mathbb{E}[g_k]=\nabla f(x_k)$. We do not require this condition.

Theorem 3. *Let $f:\mathbb{R}^d\to\mathbb{R}$ be a L-smooth convex function and $x^*=argmin_x f(x)$. In iteration k of SGD, g_k is the estimated gradient, i.e., $x_{k+1}=x_k-tg_k$ where $Var(g_k)\le\sigma^2$. If there exists $1\le c\le\sqrt{2}$ s.t. $\frac{1}{c}[\nabla f(x_k)]^+\le\mathbb{E}[g_k^+]\le c[\nabla f(x_k)]^+$ and $c[\nabla f(x_k)]^-\le\mathbb{E}[g_k^-]\le\frac{1}{c}[\nabla f(x_k)]^-$, then for any $K>1$ and step size $t\le\frac{2-c^2}{Lc}$, let $\overline{x_K}=\frac{1}{K}\sum_{k=1}^K x_k$, we have*

$$\mathbb{E}[f(\overline{x_K})]-f(x^*)\le\frac{c||x_0-x^*||_2^2}{2tK}+\frac{t\sigma^2}{c}. \tag{4}$$

Proof. (Theorem 3) By L-smooth of f, for the k-th iteration,

$$f(x_{k+1})\le f(x_k)+\langle\nabla f(x_k),x_{k+1}-x_k\rangle+\frac{L}{2}||x_{k+1}-x_k||_2^2,$$
$$=f(x_k)-t\langle\nabla f(x_k),g_k\rangle+\frac{Lt^2}{2}||g_k||^2.$$

Because of the constant bound on gradient and $||\mathbb{E}[g_k]||_2^2=\mathbb{E}[||g_k||_2^2]-Var(g_k)$, by taking expectation on both sides

w.r.t g_k we get from Lemma 1 that

$$\begin{aligned}
\mathbb{E}[f(x_{k+1})] &\leq f(x_k) - \frac{t}{c}||\mathbb{E}[g_k]||_2^2 + \frac{Lt^2}{2}\mathbb{E}[||g_k||_2^2], \\
&= f(x_k) - \frac{t}{c}(\mathbb{E}[||g_k||_2^2] - Var(g_k)) + \frac{Lt^2}{2}\mathbb{E}[||g_k||_2^2], \\
&\leq f(x_k) - \frac{t(2-Ltc)}{2c}\mathbb{E}[||g_k||_2^2] + \frac{t}{c}\sigma^2, \\
&\leq f(x_k) - \frac{tc}{2}\mathbb{E}[||g_k||_2^2] + \frac{t}{c}\sigma^2,
\end{aligned}$$

where the last inequality follows as $Ltc \leq 2 - c^2$. Because f is convex, still from Lemma 1 we get

$$\begin{aligned}
\mathbb{E}[f(x_{k+1})] &\leq f(x^*) + \langle \nabla f(x_k), x_k - x^* \rangle - \frac{tc}{2}\mathbb{E}[||g_k||_2^2] + \frac{t}{c}\sigma^2, \\
&\leq f(x^*) + c\langle \mathbb{E}[g_k], x_k - x^* \rangle - \frac{tc}{2}\mathbb{E}[||g_k||_2^2] + \frac{t}{c}\sigma^2, \\
&= f(x^*) + c\mathbb{E}[\langle g_k, x_k - x^* \rangle - \frac{t}{2}||g_k||_2^2] + \frac{t}{c}\sigma^2.
\end{aligned}$$

We now repeat the calculations by completing the square for the middle two terms to get

$$\begin{aligned}
\mathbb{E}[f(x_{k+1})] &\leq f(x^*) + \frac{c}{2t}\mathbb{E}[2t\langle g_k, x_k - x^* \rangle - t^2||g_k||_2^2] + \frac{t}{c}\sigma^2, \\
&\leq f(x^*) + \frac{c}{2t}\mathbb{E}[||x_k - x^*||_2^2 - ||x_k - x^* - tg_k||_2^2] + \frac{t}{c}\sigma^2, \\
&= f(x^*) + \frac{c}{2t}\mathbb{E}[(||x_k - x^*||_2^2 - ||x_{k+1} - x^*||_2^2)] + \frac{t}{c}\sigma^2.
\end{aligned}$$

Summing the above equations for $k = 0, \ldots, K-1$, we get

$$\begin{aligned}
&\sum_{k=0}^{K-1} \mathbb{E}[f(x_{k+1}) - f(x^*)] \\
&\leq \frac{c}{2t}(||x_0 - x^*||_2^2 - \mathbb{E}[||x_K - x^*||_2^2]) + \frac{Kt}{c}\sigma^2 \\
&\leq \frac{c||x_0 - x^*||_2^2}{2t} + \frac{Kt}{c}\sigma^2.
\end{aligned}$$

Finally, by Jensen's inequality, $kf(\overline{x_K}) \leq \sum_{k=1}^{K} f(x_k)$,

$$\begin{aligned}
\sum_{k=0}^{K-1} \mathbb{E}[f(x_{k+1}) - f(x^*)] &= \mathbb{E}[\sum_{k=1}^{K} f(x_k)] - Kf(x^*) \\
&\geq K\mathbb{E}[f(\overline{x_K})] - Kf(x^*).
\end{aligned}$$

Combining the above equations we get

$$\mathbb{E}[f(\overline{x_K})] \leq f(x^*) + \frac{c||x_0 - x^*||_2^2}{2tK} + \frac{t}{c}\sigma^2.$$

This completes the proof. □

The bound of XOR sampling (Equation **??**) assumes a non-negative function ϕ. In XOR-SGD, the entries of vector $\overline{g_k}$ can be both positive or negative. Therefore, the bound from XOR-Sampling needs to be imposed on the positive and negative parts of $\overline{g_k}$ with a multiplicative factor of $\rho\kappa$. More precisely,

$$\begin{aligned}
\frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla_x f(x_k, \theta)]^+ &\leq \mathbb{E}[\overline{g_k}^+] \leq \rho\kappa\mathbb{E}_\theta[\nabla_x f(x_k, \theta)]^+, \\
\rho\kappa\mathbb{E}_\theta[\nabla_x f(x_k, \theta)]^- &\leq \mathbb{E}[\overline{g_k}^-] \leq \frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla_x f(x_k, \theta)]^-.
\end{aligned}$$

Leveraging this bound, our main result, Theorem 4 can be proved using Theorem 2 and 3, by replacing the objective $f(x)$ in Theorem 3 with $\mathbb{E}_{\theta\sim p(\theta)} f(x, \theta)$, while noticing $Var(\overline{g_k}) = Var_\theta(\nabla_x f(x, \theta))/N$ due to the sample size N.

Theorem 4. *(Main) Let $b, \varepsilon, l, \delta, P, \alpha, \rho, \kappa$ and $\mathscr{B}_l$ be as in Theorem 2, function $f(x, \theta) : \mathbb{R}^d \times \{0,1\}^n \to \mathbb{R}$ be a L-smooth convex function w.r.t. x. Denote $OPT = \min_x \mathbb{E}_{\theta\sim Pr(\theta)} f(x, \theta)$ as the global optimum. Let $\sigma^2 = \max_x\{Var(\nabla_x f(x, \theta))\}$ and $\varepsilon^2 = \max_x\{||\mathbb{E}[\nabla_x f(x, \theta)]||_2^2\}$. For any $1 \leq \rho\kappa \leq \sqrt{2}$, step size $t \leq \frac{2-\rho^2\kappa^2}{L\rho\kappa}$ and sample size $N \geq 1$, $\overline{x_K}$ is the output of XOR-SGD and $obj = \mathbb{E}_\theta[f(\overline{x_K}, \theta)]$ is the objective function value at $\overline{x_K}$. We have:*

$$\mathbb{E}_{\overline{x_K}}[obj] - OPT \leq \frac{\rho\kappa||x_0 - x^*||_2^2}{2tK} + \frac{t(\sigma^2 + \varepsilon^2)}{N}. \quad (5)$$

Theorem 4 states that in expectation, the difference in terms of the objective function values between the output of XOR-SGD algorithm $\overline{x_K}$ and the true optimum OPT is bounded by a term that scales inversely proportional to the number of SGD iterations K and a tail term $\frac{t(\sigma^2+\varepsilon^2)}{N}$. To tighten the bound with fixed number of steps K, we can either conduct more accurate XOR-Sampling scheme leading to smaller $\rho\kappa$ (still greater than 1), or generate more samples at each iteration to reduce the variance (increase N) in the tail term. It should be noticed that although hard to compute, σ^2 and ε^2 are from the input which do not depend on the algorithm.

While Theorem 4 provides a linear convergence rate guarantee, we expect XOR-SGD can be further accelerated if new schemes can be developed to estimate higher moments reliably. In such case, our method can be fit into accelerated SGD algorithms such as Adagrad (Duchi et al. [2011]), RMSprop (Hinton et al. [2012]) and Adam (Kingma and Ba [2014]). In addition, it should be noticed that the convergence rate of XOR-SGD is determined by the approximation constant $\rho\kappa$ from XOR-sampling. By setting proper parameter values in Theorem 2, we can get $\rho\kappa = \sqrt{2}$. As a consequence, we can collect N samples successfully by running XOR-Sampling around $40N$ times. The time complexity can be further reduced via parallel sampling. Samples can be obtained before each optimization step since $Pr(\theta)$ does not depend on x. Despite obtaining samples in XOR-SGD is more expensive, we show in experiment section that our algorithm achieves better results in less time compared to SGD with MCMC samples.

3.1 EXTENSION TO CONSTRAINED CONVEX STOCHASTIC OPTIMIZATION

Now consider the constrained case as in equation 1. Writing down the Lagrangian:

$$F(x,\lambda,\mu,\theta) = \mathbb{E}_{\theta\sim Pr(\theta)}\ f(x,\theta) + \sum_i \lambda_i h_i(x) + \sum_j \mu_j g_j(x).$$

where we require $\forall j,\ \mu_j \geq 0$. In this paper we only consider convex problems which satisfies the Slater's condition. As a consequence, strong duality holds. It implies the following optimization

$$\min_x \max_{\lambda_i,\mu_j}\ F(x,\lambda,\mu,\theta) \tag{6}$$

shares the same optimal solution with the optimization problem in equation 1. We modify algorithm 1 to its constrained version (Algorithm 2), where we use alternating min-max to solve the problem in Equation 6. In this algorithm, outer loop optimizes over λ and μ for K steps. Every time when they are updated, in the inner loop we update x along its approximate gradient direction for M steps. The approximate gradient direction is computed via XOR-Sampling.

Qualitatively, with big M and N, from Theorem 4 we know that the solution of the inner loop will be close to the optimal solution for any μ and λ fixed by the outer loop. Due to the Slater's condition, F is convex in x and concave in λ and μ. Suppose the solution from the inner loop is close to optimum, the outer loop will also converge to the optimal values of μ and λ. Hence the overall solution will be close to optimal. We leave the theoretic characterization of the convergence speed of the constrained algorithm as future work. Constraints introduce additional difficulties for theoretic analysis. To our knowledge, the convergence speed analysis involving inequality constraints is still an active research area even assuming having access to unbiased gradients. The stochastic optimization problem considered in this paper attacks an even more complicated case, where we do not have unbiased gradient estimation.

Algorithm 2: XOR-SGD (constrained version)

Input: $f(x,\theta), w(\theta), M, K, N, t, \eta, l, b, \delta, P, \alpha$ and constraints $h_i(x) = 0, g_j(x) \leq 0$ for all i, j

Define $\mathbb{E}_\theta[F(x,\lambda,\mu,\theta)] = \mathbb{E}_\theta[f(x,\theta)] + \sum \lambda_i h_i(x) + \sum \mu_j g_j(x)$
Initialize $x = x_{00}$, $\lambda = \lambda_0 = (\lambda_{i0})_{i=1,\ldots,li}$, $\mu = \mu_0 = (\mu_{j0})_{j=1,\ldots,lj}$ for function $F(x,\lambda,\mu,\theta)$
for $k = 0$ *to* $K-1$ **do**
  for $m = 0$ *to* $M-1$ **do**
    $s \leftarrow 1$
    while $s \leq N$ **do**
      $result \leftarrow$ XOR-Sampling$(w(\theta), l, b, \delta, P, \alpha)$
      if $result \neq Failure$ **then**
        $\theta_s \leftarrow result$
        $s \leftarrow s+1$
      end
    end
    Compute $\overline{g_m} \leftarrow \frac{1}{N}\sum_{s=1}^{N} \nabla f(x_{km}, \theta_s) + \sum \lambda_{ik} \nabla h_i(x_{km}) + \sum \mu_{jk} \nabla g_i(x_{km})$
    Update $x_{k,m+1} \leftarrow x_{k,m} - t\overline{g_m}$
  end
  Let $\overline{x_k} = \frac{1}{M}\sum_{r=1}^{M} x_{kr}$, and set $x_{k+1,0} = \overline{x_k}$
  Update $\lambda_{i,k+1} = \lambda_{ik} + \eta h_i(\overline{x_k})$
  Update $\mu_{j,k+1} = \min\{\mu_{jk} + \eta g_j(\overline{x_k}), 0\}$
end
return $\frac{1}{K}\sum_{k=1}^{K} \overline{x_k}$

4 EXPERIMENTS

We evaluate our XOR-SGD algorithm on the inventory management (Ziukov [2016], Shapiro and Philpott [2007]) and the network design problems (Sheldon et al. [2012], Wu et al. [2017, 2016]). For comparison, we consider a baseline which uses SGD while the gradients are estimated by either Gibbs Sampling, Belief Propagation (BP) (Yedidia et al. [2001], Murphy et al. [2013]), or Belief Propagation Chain (BPChain) (Fan and Xue [2020]). For each setting of both applications, to produce a sample, Gibbs sampling first takes 100 steps to burn in, and then draws one sample every 30 steps. We fix the number of iteration steps of both BP and BPChain as 20, which is enough for belief propagation to converge. We allow SGD with Gibbs sampling, BP and BPChain to draw more samples than XOR-SGD for a fair comparison. For both applications, we use MRF as probabilistic models for $Pr(\theta)$. All experiments were conducted using single core architectures on Intel Xeon Gold 6126 2.60GHz machines with 96GB RAM and a wall-time limit of 10 hours. Please see the supplementary materials for more details on the experiment setups.

4.1 STOCHASTIC INVENTORY MANAGEMENT

We first investigate our algorithm on the stochastic inventory management problem studied in Shapiro and Philpott [2007]. A company manager has to decide, at the beginning of each season, how much of each materials to purchase to meet his demand later in the production season. Assuming there are n materials. The demand of material i is d_i. Let $d = (d_1, \ldots, d_n)^T$ be the demand vector. At the beginning of the season, only the distribution $Pr(d)$ is known due to the stochasticity down the supply chain. The demands of multiple materials can be correlated because one product typically needs many types of materials. In other words, $Pr(d)$ cannot be decomposed into the product of probabilities of individual demands. The manager stocks x_i amount of material i at the beginning of the season. Each unit of material i takes storage space w_i, and the total amount of pre-order is limited by the available storage space X. At the end of the produc-

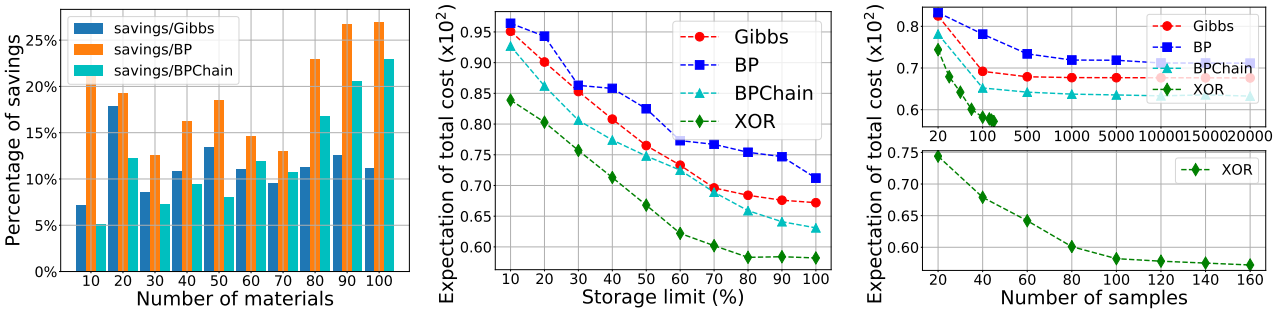

Figure 2: Experimental results on the inventory management problem. XOR-SGD is better than baselines in all cases. **(Left)** The percentage of savings of the solutions found by XOR-SGD against other methods on 100% storage limit varying the number of materials. XOR-SGD on average saves 10% cost. **(Middle)** The objective values found by all methods with 50 materials varying storage limits. **(Right)** The objective values found by all methods with different number of samples for approximation in SGD. 100% storage limit and 50 materials. XOR-SGD with 60 samples outperforms other methods with 20,000 samples.

tion season, demand d will be revealed to the the manager. We assume the cost of ordering the i-th material is c_i per unit. If the demand $d_i > x_i$, then a back order is needed, of which one unit costs $b_i \geq c_i$. Overall, the cost for back order is $b_i(d_i - x_i)$ if $d_i > x_i$, and is zero otherwise. On the other hand, if $d_i < x_i$, then a holding cost of h_i per unit is incurred, leading to an additional total cost $h_i(x_i - d_i)$. Summing it up, the cost for material i is $G_i = c_i x_i + b_i[d_i - x_i]^+ + h_i[x_i - d_i]^+$ where $[a]^+$ denotes the maximum of a and 0. Then, the total cost will be $G(x,d) = \sum_{i=1}^n G_i$. The manager want to minimize his operational cost, which translates to this problem:

$$\min_{x \geq 0} \ \mathbb{E}_{d \sim Pr(d)}[G(x,d)], \quad s.t. \quad w^T x \leq X. \tag{7}$$

We can show $G(x,d)$ is convex w.r.t. x. Hence the inventory management problem is a constrained convex stochastic optimization problem. We run the experiments varying the number of materials n, the storage limit, and the number of samples we use in XOR-SGD and other methods. Details on the experimental setup are in supplementary materials.

Figure 2 shows that our algorithm XOR-SGD outperforms the other methods on multiple experimental setups. The left figure shows the percentage reduction of the objective values of the solutions found by XOR-SGD against SGD with other sampling methods. In math form, for example for Gibbs Sampling, the metric is $(obj(\text{Gibbs}) - obj(\text{XOR-SGD}))/obj(\text{Gibbs})$ (metrics for other approaches are similar). We vary the number of materials from 10 to 100. The middle figure shows the objective values of solutions varying the storage limit. The right figure shows the objective values varying the number of samples. The green line in the upper picture is re-plotted in the bottom for clarity. For the left and the middle figures, we let XOR-SGD take 100 samples for approximation while SGD with other sampling methods take 10,000. The experiments in the right figure is with 100% storage limit and 50 different materials. We can see from the left figure that objective optimized by XOR-SGD is on average 10 percent better than that optimized by the baselines. With the storage limit increasing, the middle figure shows that XOR-SGD is always better than all baselines. From the right figure, XOR-SGD found better solutions with 60 samples compared to SGD with Gibbs sampling which uses 20,000 samples. In XOR-SGD, we set hyper-parameters to guarantee $\rho\kappa = \sqrt{2}$. Note XOR-SGD (5.5 minutes for 60 samples) runs even faster than SGD with Gibbs (17 minutes for 20,000 samples), even though it needs to solve NP-complete problems to get the samples. Notice the running times of both BP and BPChain are longer than Gibbs Sampling. Therefore XOR-SGD is both faster and better than competing methods.

4.2 STOCHASTIC NETWORK DESIGN

Network optimization searches for the optimal plan to increase the network connectivity under a given budget in preparation of stochastic events, such as natural disasters (Israeli and Wood [2002], Dilkina and Gomes [2010]). We consider the expected commuting time of a random walk defined over the network, which is studied in Ghosh et al. [2008] as the connectivity measure, and which is argued to be realistic among field experts (McClure et al. [2016], Inman et al. [2013]). Given an undirected graph $G = (V,E)$, where $|V| = m, |E| = n$. Each edge e is associated with a non-negative weight g_e, known as the *conductance* value of edge e, which indicates the degree of easiness to travel along edge e. Let $g = (g_1, \ldots, g_n)^T$. Natural disasters such as earthquakes and floods typically strike one region and can paralyze the connectivity of the road network in the given region. Each edge $e \in E$ is associated with a binary random variable θ_e that describes the state of the edge during disasters. $\theta_e = 0$ means that the edge is destroyed, and 1 otherwise. Let $\theta = (\theta_1, \ldots, \theta_n)$. Notice that the states of θ are correlated. The probability of θ is given by $Pr(\theta)$, which may be constructed from domain knowledge or learned from

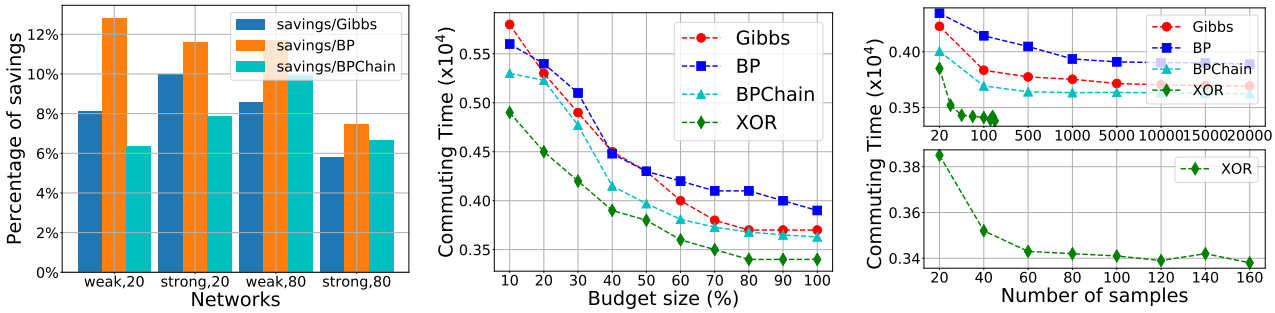

Figure 3: Experimental results on the network design problem. XOR-SGD is better than baselines in all cases. **(Left)** The percentage of savings of XOR-SGD against other methods. XOR-SGD saves on average $> 5\%$ commuting time on all 4 benchmarks with 100% budget. **(Middle)** Commuting time of the solutions found by XOR-SGD and baselines varying budgets on network "weak 20". **(Right)** Commuting time found by XOR-SGD and other methods with different number of samples used for approximation in SGD (100% budget, "weak 20"). XOR-SGD with 40 samples outperforms others with 20,000 samples.

real world data. We can make investments to improve the conductance g_e of edge e. It will take money c_e to increase one unit of g_e and we have a total budget of B. Denote $A \in \mathbb{R}^{m\times n}$ as an incidence matrix of graph G where each item $A_{ij} = 1$ if the vertex v_i and edge e_j are incident and 0 otherwise. $\text{diag}(g)$ is a diagonal matrix which has g on the diagonal. Then, the weighted Laplacian matrix L of graph G is defined as $L = A\text{diag}(g)A^T$. From the work of Ghosh et al. [2008] the commuting time $\overline{C}(g,\theta)$ can be calculated as $\frac{4(\mathbf{1}^T g)}{(m-1)}\left(\mathbf{Tr}(L+\mathbf{1}\mathbf{1}^T/m)^{-1}-1\right)$ which is convex w.r.t. g. Here L is calculated only with edges not destroyed. We would like to find the best network improvement plan under the given budget, which minimizes the expected commuting time averaged over all stochastic events to maximizes the network connectivity. Let Δg_e be the improvement of the conductance value at edge e and $\Delta g = (\Delta g_1, \ldots, \Delta g_n)^T$. Mathematically, our problem can be formulated as the following convex stochastic optimization:

$$\min_{\Delta g \geq 0} \mathbb{E}_{\theta\sim Pr(\theta)}[\overline{C}(g+\Delta g,\theta)],\ s.t.\ \sum_{e\in E} c_e \Delta g_e \leq B. \qquad (8)$$

We evaluate our algorithms on a real-world problem, the Flood Preparation problem for the emergency medical services (EMS) on road networks studied in Wu et al. [2016]. Edges of the graph represent road segments while nodes represent either road intersections or EMS centers or locations need to be accessible in case of emergencies. Some road segments are above the same river, which can be jointly destroyed by e.g., floods of the river. We test our algorithm on four benchmarks involving the weak and the strong network originally evaluated in Wu et al. [2016]. The weak network consists of 502 edges and 169 nodes. The strong network consist of 1,562 edges and 526 nodes. The number of vulnerable edges (i.e., $\theta_i = 0$) can be either 20 or 80 for both weak and strong network, resulting in 4 benchmarks.

Figure 3 shows that XOR-SGD outperforms other methods. The results are similar to those for the inventory management problem. Additional experiment details and discussions are in the supplementary materials. We would like to emphasize that XOR-SGD with 40 samples already outperforms other methods in Figure 3 (right). In particular, XOR-SGD with 40 samples take 1 minutes 40 seconds, while SGD with 20,000 Gibbs samples needs 2.5 minutes. Results clearly show that XOR-SGD outperforms other methods both in efficiency and in the quality of solutions.

5 CONCLUSION

We proposed XOR-SGD, a novel algorithm based on stochastic gradient descent and XOR-Sampling, to attack constrained convex stochastic optimization problems, which are crucial for many decision-making applications with uncertainty. We showed theoretically that our algorithm has a linear convergence rate to the global optimum. Empirically, we demonstrated the superior performance of XOR-SGD on both the stochastic inventory management and the stochastic network design problems. In particular, XOR-SGD accessing 60 XOR samples runs faster and finds better solutions than SGD accessing 20,000 MCMC samples for the inventory management problem. XOR-SGD accessing 40 XOR samples outperforms SGD accessing 20,000 MCMC samples both in running speed and in solution quality for the network design problem. Overall, our paper demonstrates the power of integrating cutting-edge computer science technology with real-world problems. Our paper will also stimulate further academic progress in stochastic gradient descent, probabilistic inference with hashing and randomization, and more broadly, convex and non-convex optimizations with insights from real-world applications. Future work includes tightening the constant bound and accelerating the convergence rate with modifications to the SGD procedure. We will also investigate if our approach can motivate new algorithms for non-convex stochastic optimization problems.

References

Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1195–1199, 2017.

Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *The Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than sgd. In *Advances in neural information processing systems*, pages 2675–2686, 2018.

John P Barton, Eleonora De Leonardis, Alice Coucke, and Simona Cocco. Ace: adaptive cluster expansion for maximum entropy graphical model inference. *Bioinformatics*, 32(20):3089–3097, 2016.

Vaishak Belle, Guy Van den Broeck, and Andrea Passerini. Hashing-based approximate probabilistic inference in hybrid domains. In *Proceedings of the 31st UAI Conference*, 2015.

Supratik Chakraborty, Daniel J. Fremont, Kuldeep S. Meel, Sanjit A. Seshia, and Moshe Y. Vardi. Distribution-aware sampling and weighted model counting for sat. In *AAAI*, 2014.

Supratik Chakraborty, Dror Fried, Kuldeep S. Meel, and Moshe Y. Vardi. From weighted to unweighted model counting. In *Proceedings of the 24th International Joint Conference on AI (IJCAI)*, 2015.

Bistra Dilkina and Carla P Gomes. Solving connected subgraph problems in wildlife conservation. In *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, pages 102–116. Springer, 2010.

Fan Ding, Hanjing Wang, Ashish Sabharwal, and Yexiang Xue. Towards efficient discrete integration via adaptive quantile queries. *arXiv preprint arXiv:1910.05811*, 2019.

Justin Domke. Learning graphical model parameters with approximate marginal inference. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2454–2467, 2013.

Kumar Avinava Dubey, Sashank J Reddi, Sinead A Williamson, Barnabas Poczos, Alexander J Smola, and Eric P Xing. Variance reduction in stochastic gradient langevin dynamics. In *NIPS*, pages 1154–1162, 2016.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(Jul): 2121–2159, 2011.

John Duchi, Feng Ruan, and Chulhee Yun. Minimax bounds on stochastic batched convex optimization. In *Conference On Learning Theory*, pages 3065–3162, 2018.

Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Taming the curse of dimensionality: Discrete integration by hashing and optimization. In *Proceedings of the 30th ICML*, 2013a.

Stefano Ermon, Carla P. Gomes, Ashish Sabharwal, and Bart Selman. Embed and project: Discrete sampling with universal hashing. In *Advances in Neural Information Processing Systems (NIPS)*, 2013b.

Ding Fan and Yexiang Xue. Contrastive divergence learning with chained belief propagation. In *International Conference on Probabilistic Graphical Models*, 2020.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on Learning Theory*, pages 797–842, 2015.

Arpita Ghosh, Stephen Boyd, and Amin Saberi. Minimizing effective resistance of a graph. *SIAM review*, 50(1):37–66, 2008.

Carla Gomes, Thomas Dietterich, Christopher Barrett, Jon Conrad, Bistra Dilkina, Stefano Ermon, Fei Fang, Andrew Farnsworth, Alan Fern, Xiaoli Fern, et al. Computational sustainability: Computing for a better world and a sustainable future. *Communications of the ACM*, 62(9):56–65, 2019.

Carla P Gomes, Ashish Sabharwal, and Bart Selman. Near-uniform sampling of combinatorial spaces using xor constraints. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 481–488. MIT Press, 2007a.

Carla P. Gomes, Willem-Jan Van Hoeve, Ashish Sabharwal, and Bart Selman. Counting csp solutions using generalized xor constraints. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, 2007b.

Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. 2012.

Robert M. Inman, Brent L. Brock, Kristine H. Inman, Shawn S. Sartorius, Bryan C. Aber, Brian Giddings, Steven L. Cain, Mark L. Orme, Jay A. Fredrick, Bob J. Oakleaf, Kurt Alt, Eric A. Odell, and Guillaume Chapron. Developing priorities for metapopulation conservation at the landscape scale: Wolverines in the western united states. 2013.

Eitan Israeli and R Kevin Wood. Shortest-path network interdiction. *Networks: An International Journal*, 40(2): 97–111, 2002.

Alexander Ivrii, Sharad Malik, Kuldeep S Meel, and Moshe Y Vardi. On computing minimal independent support and its applications to sampling and counting. *Constraints*, pages 1–18, 2015.

Chi Jin, Praneeth Netrapalli, and Michael I Jordan. Accelerated gradient descent escapes saddle points faster than gradient descent. *arXiv preprint arXiv:1711.10456*, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

Jonathan Kuck, Tri Dao, Shengjia Zhao, Burak Bartan, Ashish Sabharwal, and Stefano Ermon. Adaptive hashing for model counting. In *Conference on Uncertainty in Artificial Intelligence*, 2019.

Jason D Lee, Qihang Lin, Tengyu Ma, and Tianbao Yang. Distributed stochastic variance reduced gradient methods and a lower bound for communication complexity. *arXiv preprint arXiv:1507.07595*, 2015.

Qiang Liu and Alexander T. Ihler. Variational algorithms for marginal MAP. *Journal of Machine Learning Research*, 14, 2013.

Radu Marinescu, Rina Dechter, and Alexander T. Ihler. AND/OR search for marginal MAP. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI*, 2014.

Radu Marinescu, Rina Dechter, and Alexander Ihler. Pushing forward marginal map with best-first search. In *Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI)*, 2015.

Denis Deratani Mauá and Cassio Polpo de Campos. Anytime marginal MAP inference. In *Proceedings of the 29th ICML*, 2012.

Meredith L McClure, Andrew J. Hansen, and Robert M. Inman. Connecting models to movements: testing connectivity model predictions against empirical migration and dispersal data. *Landscape Ecology*, 31:1419–1432, 2016.

Kevin Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. *arXiv preprint arXiv:1301.6725*, 2013.

Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

Alexander Shapiro and Andy Philpott. A tutorial on stochastic programming. 2007.

Daniel Sheldon, Bistra Dilkina, Adam N Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla P Gomes, David Shmoys, William Allen, Ole Amundsen, et al. Maximizing the spread of cascades using network design. *arXiv preprint arXiv:1203.3514*, 2012.

Eric Sodomka, John Collins, and Maria Gini. Efficient statistical methods for evaluating trading agent performance. 2007.

Bram Verweij, Shabbir Ahmed, Anton J. Kleywegt, George Nemhauser, and Alexander Shapiro. The sample average approximation method applied to stochastic routing problems: A computational study. *Computational Optimization and Applications*, 24(2-3):289–333, 2003.

Chong Wang, Xi Chen, Alexander J Smola, and Eric P Xing. Variance reduction for stochastic gradient optimization. In *Advances in Neural Information Processing Systems*, pages 181–189, 2013.

Xiaojian Wu, Daniel R Sheldon, and Shlomo Zilberstein. Optimizing resilience in large scale networks. In *Proceedings of the 30th Conference of AAAI*, 2016.

Xiaojian Wu, Yexiang Xue, Bart Selman, and Carla P. Gomes. Xor-sampling for network design with correlated stochastic events. In *Proceedings of the 26th IJCAI*, pages 4640–4647, 2017.

Yexiang Xue, Zhiyuan Li, Stefano Ermon, Carla P. Gomes, and Bart Selman. Solving marginal map problems with np oracles and parity constraints. In *Proceedings of the 29th Annual Conference on NIPS*, 2016.

Jonathan S Yedidia, William T Freeman, and Yair Weiss. Generalized belief propagation. In *Advances in neural information processing systems*, pages 689–695, 2001.

Serhii Ziukov. A literature review on models of inventory management under uncertainty. 2016.

SUPPLEMENTARY MATERIALS

A XOR-SAMPLING FOR THE WEIGHTED CASE

The text here provides a synopsis for the approach in Ermon et al. [2013b]. We still encourage the readers to read the original text for a better explanation. Let $w(\theta), p(\theta)$ and Z as defined before, the high-level idea of XOR-Sampling is to first dicretize $w(\theta)$ to $w'(\theta)$ as in Definition 1, followed by embedding the weighted $w'(\theta)$ to the unweighted space Δ_w. Finally, XOR-sampling uses counting based on hashing and randomization to sample uniformly from Δ_w.

Definition 1. *Assume $w(\theta)$ has both upper and lower bound, namely, $M = \max_\theta w(\theta)$ and $m = \min_\theta w(\theta)$. Let $b \geq 1, \varepsilon > 0, r = 2^b/(2^b-1)$ and $l = \lceil \log_r(2^n/\varepsilon) \rceil$. Partition the configurations into the following weight based disjoint buckets: $\mathscr{B}_i = \{\theta | w(\theta) \in (\frac{M}{r^{i+1}}, \frac{M}{r^i}]\}, i = 0, \ldots, l-1$ and $\mathscr{B}_l = \{\theta | w(\theta) \in (0, \frac{M}{r^l}]\}$. The discretized weight function $w' : \{0,1\}^n \to \mathbb{R}^+$ is defined as follows: $w'(\theta) = \frac{M}{r^{i+1}}$ if $\theta \in \mathscr{B}_i, i = 0, \ldots, l-1$ and $w'(\theta) = 0$ if $\theta \in \mathscr{B}_l$. This leads to the corresponding discretized probability distribution $p'(\theta) = w'(\theta)/Z'$ where Z' is the normalization constant of $w'(\theta)$.*

For the weighted case, the goal of XOR-sampling is to guarantee that the probability of sampling one θ is proportional to the unnormalized density (up to a multiplicative constant). By Definition 1, we obtain a distribution $p'(x)$ which satisfying $\frac{1}{\rho} p(x) \leq p'(x) \leq \rho\, p(x)$ where $\rho = \frac{r^2}{1-\varepsilon}$. Then, XOR-sampling implements a horizontal slice technique to transform a weighted problem into an unweighted one. For the easiness of illustration, we denote $M' = \max_\theta w'(\theta)$ and m' as the smallest non-zero value of $w'(\theta)$. Then consider the simple case where $b = 1$ and $r = 2$, where we have $M' = 2^{l-1} m'$. Let $\delta = (\delta_0, \ldots, \delta_{l-2})^T \in \{0,1\}^{l-1}$ be a binary vector of length $l-1$, XOR-sampling samples (θ, δ) uniformly at random from the following set Δ_w using the unweighted version of sampling based on hashing and randomization:

$$\Delta_w = \{(\theta, \delta) : w'(\theta) \leq 2^{i+1} m' \Rightarrow \delta_i = 0\}. \tag{9}$$

If we sample (θ, δ) uniformly at random from Δ_w and then only return θ, it can be proved that the probability of sampling θ from $w'(\theta)$ is proportional to $m' 2^{i-1}$ when $w(\theta)$ is sandwiched between $m' 2^{i-1}$ and $m' 2^i$. Therefore, this technique leads to the constant approximation guarantee of XOR-Sampling. The precise statement of the guarantee is in Theorem 2. For general case of b and r, please refer to Ermon et al. [2013b].

Setting $\varepsilon\eta_\phi$ to Zero In Definition 1 we can make b larger and ε smaller enough, then there will be a possibly large but finite value of l such that $\frac{M}{r^l}$ is smaller than m , which leads $\mathscr{B}_l$ to be empty and $\varepsilon\eta_\phi$ to be zero.

B PROOFS

B.1 PROOF OF LEMMA 1

We define two functions $g_k^+ = \max\{g_k, \mathbf{0}\}$ and $g_k^- = \min\{g_k, \mathbf{0}\}$ where $\mathbf{0}$ is a vector of all 0 which has the same dimension as g_k. We have $g_k = g_k^+ + g_k^-$. We define both $\nabla f(x_k)^+$ and $\nabla f(x_k)^-$ in the similar way. Then Lemma 1 gives the new bounds of two terms assuming the constant bound on the gradient, which are essential to the proof of convergence rate. The proof of Lemma 1 is as follows:

Proof. (Lemma 1) Since we have the constant bound that

$$\frac{1}{c}\nabla f(x_k)^+ \leq \mathbb{E}[g_k^+] \leq c\nabla f(x_k)^+. \tag{10}$$

$$c\nabla f(x_k)^- \leq \mathbb{E}[g_k^-] \leq \frac{1}{c}\nabla f(x_k)^-. \tag{11}$$

and because of $g_k^+ \geq \mathbf{0}$ and $g_k^- \leq \mathbf{0}$ we can obtain

$$\begin{aligned}
\frac{1}{c}||\mathbb{E}[g_k^+]||_2^2 &= \frac{1}{c}\langle \mathbb{E}[g_k^+], \mathbb{E}[g_k^+]\rangle \leq \langle \nabla f(x_k)^+, \mathbb{E}[g_k^+]\rangle \\
&\leq c\langle \mathbb{E}[g_k^+], \mathbb{E}[g_k^+]\rangle = c||\mathbb{E}[g_k^+]||_2^2. \\
\frac{1}{c}||\mathbb{E}[g_k^-]||_2^2 &= \frac{1}{c}\langle \mathbb{E}[g_k^-], \mathbb{E}[g_k^-]\rangle \leq \langle \nabla f(x_k)^-, \mathbb{E}[g_k^-]\rangle \\
&\leq c\langle \mathbb{E}[g_k^-], \mathbb{E}[g_k^-]\rangle = c||\mathbb{E}[g_k^-]||_2^2.
\end{aligned}$$

which exactly means

$$\frac{1}{c}||\mathbb{E}[g_k]||_2^2 \leq \langle \nabla f(x_k), \mathbb{E}[g_k]\rangle \leq c||\mathbb{E}[g_k]||_2^2.$$

To prove the second inequality, we need to take advantage of the convexity of f. Denote $[x_k - x^*]^+ = \max\{x_k - x^*, \mathbf{0}\}$ and $[x_k - x^*]^- = \min\{x_k - x^*, \mathbf{0}\}$, we know $x_k - x^* = [x_k - x^*]^+ + [x_k - x^*]^-$. In addition, because f is convex, the index set of non-zero entries of $[x_k - x^*]^+$ and $\nabla f(x_k)^+$ is the same. The index set of non-zero entries of $[x_k - x^*]^-$ and $\nabla f(x_k)^-$ is also the same. In addition, because of Equation 10 and 11, the index set of non-zero entries of $\mathbb{E}[g_k^+]$ ($\mathbb{E}[g_k^-]$) is the same with $\nabla f(x_k)^+$ ($\nabla f(x_k)^-$). Combining these facts with Equations 10 and 11, we have

$$\begin{aligned}
\frac{1}{c}\langle \mathbb{E}[g_k^+], [x_k - x^*]^+\rangle &\leq \langle \nabla f(x_k)^+, [x_k - x^*]^+\rangle \\
&\leq c\langle \mathbb{E}[g_k^+], [x_k - x^*]^+\rangle. \\
\frac{1}{c}\langle \mathbb{E}[g_k^-], [x_k - x^*]^-\rangle &\leq \langle \nabla f(x_k)^-, [x_k - x^*]^-\rangle \\
&\leq c\langle \mathbb{E}[g_k^-], [x_k - x^*]^-\rangle.
\end{aligned}$$

Combining these two equations, we have

$$\frac{1}{c}\langle \mathbb{E}[g_k], x_k - x^*\rangle \leq \langle \nabla f(x_k), x_k - x^*\rangle \leq c\langle \mathbb{E}[g_k], x_k - x^*\rangle.$$

This completes the proof. □

B.2 PROOF OF THEOREM 4

Proof. (Theorem 4) Since we use N samples at each iteration, we have $\overline{g_k} = \frac{1}{N}\sum_{i=1}^{N} g_k^i$ and $\mathbb{E}[\overline{g_k}] = \mathbb{E}[g_k^i]$. In each iteration k we can adjust the parameters in XOR-Sampling to make the tail $\varepsilon\eta_\phi$ zero, then for each sample g_k^i we can obtain from Theorem 2 that

$$\frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla f(x_k,\theta)]^+ \le \mathbb{E}[g_k^{i+}] \le \rho\kappa\mathbb{E}_\theta[\nabla f(x_k,\theta)]^+. \quad (12)$$

$$\rho\kappa\mathbb{E}_\theta[\nabla f(x_k,\theta)]^- \le \mathbb{E}[g_k^{i-}] \le \frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla f(x_k,\theta)]^-. \quad (13)$$

The variance of each sample g_k^i can also be bounded by

$$\begin{aligned}
&Var(g_k^i)\\
&= \mathbb{E}_{\theta'\sim p'(\theta')}[||\nabla f(x_k,\theta')||_2^2] - ||\mathbb{E}_{\theta'\sim p'(\theta')}[\nabla f(x_k,\theta')]||_2^2,\\
&\le \rho\kappa\mathbb{E}_{\theta\sim p(\theta)}[||\nabla f(x_k,\theta)||_2^2],\\
&= \rho\kappa(Var(\nabla f(x_k,\theta)) + ||\mathbb{E}_{\theta\sim p(\theta)}[\nabla f(x_k,\theta)]||_2^2),\\
&\le \rho\kappa(\sigma^2+\varepsilon^2).
\end{aligned}$$

Denote $\overline{g_k}^+ = \max\{\overline{g_k}, \mathbf{0}\}$ and $\overline{g_k}^- = \min\{\overline{g_k}, \mathbf{0}\}$. Clearly, $g_k^{i+} \ge 0$ and $g_k^{i-} \le 0$. Moreover, for a given dimension, either $g_k^{i+} = 0$ for that dimension or $g_k^{i-} = 0$. Evaluating $\overline{g_k}$ dimension by dimension, we can see that $\overline{g_k}^+ = \frac{1}{N}\sum_{i=1}^{N} g_k^{i+}$ and $\overline{g_k}^- = \frac{1}{N}\sum_{i=1}^{N} g_k^{i-}$. Combined with Equation 12 and 13, we know

$$\frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla f(x_k,\theta)]^+ \le \mathbb{E}[\overline{g_k}^+] \le \rho\kappa\mathbb{E}_\theta[\nabla f(x_k,\theta)]^+.$$

$$\rho\kappa\mathbb{E}_\theta[\nabla f(x_k,\theta)]^- \le \mathbb{E}[\overline{g_k}^-] \le \frac{1}{\rho\kappa}\mathbb{E}_\theta[\nabla f(x_k,\theta)]^-.$$

Because $\mathbb{E}[\overline{g_k}] = \mathbb{E}[g_k^i]$, we also have

$$Var(\overline{g_k}) = \frac{1}{N^2}Var(\sum_{i=1}^{N} g_k^i) = \frac{Var(g_k^i)}{N}.$$

Then the variance of $\overline{g_k}$ can be bounded as

$$Var(\overline{g_k}) \le \frac{\rho\kappa(\sigma^2+\varepsilon^2)}{N}.$$

Therefore, we can then apply Theorem 3 to get the result in equation 5.

$$\begin{aligned}
&\mathbb{E}_{\overline{x_K}}[\mathbb{E}_\theta[f(\overline{x_K},\theta)]] - \mathbb{E}_\theta[f(x^*,\theta)]\\
&\le \frac{\rho\kappa||x_0-x^*||_2^2}{2tK} + \frac{t\max_k\{Var(\overline{g_k})\}}{\rho\kappa},\\
&\le \frac{\rho\kappa||x_0-x^*||_2^2}{2tK} + \frac{t(\sigma^2+\varepsilon^2)}{N}.
\end{aligned}$$

which can also be written as

$$\mathbb{E}_{\overline{x_K}}[\text{obj}] - OPT \le \frac{\rho\kappa||x_0-x^*||_2^2}{2tK} + \frac{t(\sigma^2+\varepsilon^2)}{N}. \quad (14)$$

This completes the proof. □

C EXPERIMENTS

We evaluate our XOR-SGD algorithm on the inventory management Ziukov [2016], Shapiro and Philpott [2007] and the network design problems Sheldon et al. [2012], Wu et al. [2017, 2016]. For each setting of both applications, to produce a sample, Gibbs sampling first takes 100 steps to burn in, and then draws samples every 30 steps. We fix the iteration step of both BP and BPChain as 20, which is enough for BP to converge. We allow SGD with Gibbs sampling, BP and BPChain to draw more samples than XOR-SGD for a fair comparison. All experiments were conducted using single core architectures on Intel Xeon Gold 6126 2.60GHz machines with 96GB RAM and a wall-time limit of 10 hours. For both applications, we use MRF as probabilistic models for $Pr(\theta)$, which can be seen in the next section. For a fair comparison, once a solution x is generated by either algorithm, we use an exact weighted counter ACE Barton et al. [2016] to evaluate $\mathbb{E}_{\theta\sim Pr(\theta)}f(x,\theta)$ exactly. All objective values reported here are from ACE.

C.1 SETTINGS OF STOCHASTIC INVENTORY MANAGEMENT

Taking into account of the storage constraint, the original problem is equivalent to the following problem:

$$\min_{x\ge 0}\max_{\mu\ge 0}\mathbb{E}_{d\sim Pr(d)}[G(x,d)] + \mu(w^T x - X). \quad (15)$$

For inventory management problem, we assume each d_i can take two different values, one corresponding to the high demand one corresponding to the low demand. Then, we introduce a new vector θ where $\theta_i = 1$ means d_i is the high value while $\theta_i = 0$ otherwise. In the experiment we range n from 10 to 100 increased by a step size of 10 and draw 10 instances for each setting. Under each setting, we draw every c_i uniformly from $(0,5]$, h_i uniformly from $(0,10]$, sample s_i uniformly drawn from $(0,10]$ and let $b_i = c_i + s_i$. The two values of each d_i are also uniformly drawn from $(0,10]$. We model $Pr(\theta)$ as a MRF with several cliques. The variables in each clique are highly correlated with each other. For a problem with n products, we draw the number of cliques uniformly from $[n,2n]$. The domain size of each clique ϕ_α is chosen from the range of $[1,6]$ at random. The potential function of a clique involving l variables is in the form of a table of size 2^l. The i-th entry of this table, denoted as v_i, is modeled as $v_i = v_{i1} + v_{i2}v_{i3}$, where v_{i1} is uniformly drawn from $(0,1)$, v_3 uniformly from $(10,1000)$ and binary variable v_{i2} uniformly randomly drawn from $\{0,1\}$. Each storage requirement w_i is drawn from $(0,10]$ uniformly at random. The largest storage limit X is set to be $5n$. We also evaluate our method given different percentages of the largest storage limit, which is shown in Figure 2 (middle). In the SGD algorithm, x is initialized with the absolute value

of a Gaussian random variable from $\mathcal{N}(5,3)$ to ensure it is non-negative.

In terms of the parameters in XOR-Sampling we fix $P = 100, b = 7, \varepsilon = 0.01$ and the others the same as in Ermon et al. [2013b] to guarantee $\rho\kappa = \sqrt{2}$. Learning rate t is 0.1 at first and divided by 10 after 50 iterations, then further divided by 10 after 100 iterations. η is 10 at first and divided by 10 after 50 iterations, then further divided by 10 after 100 iterations. The total number of both K and M are set to be 200. However, since we run each algorithm on one single core with a wall-time limit of 10 hours for a fair comparison, not all algorithms can complete all iterations. The plots are based on the best results found by each algorithm within the time limit.

C.2 SETTINGS OF STOCHASTIC NETWORK DESIGN

The task in equation 8 is equivalent to solving the following problem:

$$\min_{\Delta g \geq 0} \max_{\mu \geq 0} \mathbb{E}_{\theta \sim Pr(\theta)}[\overline{C}(g + \Delta g, \theta)] + \mu(\sum_{e \in E} c_e \Delta g_e - B). \tag{16}$$

Because of the convexity of $\overline{C}(g + \Delta g, \theta)$ and strong duality, both problems have the same optimal solution.

We test our algorithm on a real-world problem, the so-called Flood Preparation problem for the emergency medical services (EMS) on road networks Wu et al. [2016]. The problem setup, including the graph structure and the definition of $Pr(\theta)$, are the same as that in Wu et al. [2016]. The original network is unweighted, hence we set the initial conductance value for each edge as 1. c_e is initialized uniformly from the range $(0, 10)$. The largest budget size B is 1000. We evaluate our method varying the percentage allowed of the largest budget size, which is shown in Figure 3 (middle). In the experiment, each entry of Δg is initialized with the absolute value of a Gaussian random variable from $\mathcal{N}(0,1)$. Total number of SGD iterations is 2000, while not all algorithms can complete all 2000 iterations within the time limit of 10 hours. The experimental results reported in the plots are based on the best solutions found by each algorithm within the time limit. Learning rate t is 1 at first and divided by 10 after 20 iterations, further by 10 after 100 iterations. Parameters in XOR-Sampling are set to be the same as in the inventory management problem.

The left figure in Figure 3 shows the percentage of savings between SGD with other sampling methods and XOR-SGD among all of the 4 different networks, while the middle and the right figures show the averaged commuting time with regard to different budget sizes and different number of samples, respectively. For the left and the middle figures, we let XOR-SGD take 100 samples in each iteration while SGD with other methods take 10,000. We can see from the left figure that objective optimized by XOR-SGD is at least 5% better than that optimized by other methods for all the 4 different networks. In addition, from the middle and the right figures we know that with the increase of either budget size or the number of samples, our method can find consistently better solutions than the compared methods. In particular, from the right figure we can see even 40 samples in each iteration are enough for XOR-SGD to compete with the result from Gibbs with 20,000 samples. Meanwhile, XOR-SGD also runs faster than the compared method under this situation. In this experiment, XOR-SGD with 40 samples take 1 minutes 40 seconds per SGD iteration, while SGD with 20,000 Gibbs samples need 2.5 minutes per iteration. Since sampling time of both BP and BPChain is no shorter than Gibbs Sampling, we thus conclude that XOR-SGD outperforms other methods both in efficiency and in the quality of solutions found.