LINEAR SPEEDUP IN SADDLE-POINT ESCAPE FOR DECENTRALIZED NON-CONVEX OPTIMIZATION

Stefan Vlaski and Ali H. Sayed

School of Engineering, École Polytechnique Fédérale de Lausanne

ABSTRACT

Under appropriate cooperation protocols and parameter choices, fully decentralized solutions for stochastic optimization have been shown to match the performance of centralized solutions and result in linear speedup (in the number of agents) relative to non-cooperative approaches in the strongly-convex setting. More recently, these results have been extended to the pursuit of first-order stationary points in non-convex environments. In this work, we examine in detail the dependence of second-order convergence guarantees on the spectral properties of the combination policy for non-convex multi agent optimization. We establish linear speedup in saddle-point escape time in the number of agents for symmetric combination policies and study the potential for further improvement by employing asymmetric combination weights. The results imply that a linear speedup can be expected in the pursuit of *second-order stationary* points, which exclude local maxima as well as strict saddle-points and correspond to local or even global minima in many important learning settings.

Index Terms— Non-convex optimization, saddle-point, second-order stationarity, minima, decentralized algorithm, centralized algorithm, diffusion strategy.

1. INTRODUCTION AND RELATED WORK

We consider a collection of K agents, where each agent k is equipped with a local stochastic cost function:

$$J_k(w) \triangleq \mathbb{E}\, Q_k(w; \boldsymbol{x}_k) \tag{1}$$

where $w \in \mathbb{R}^M$ denotes a parameter vector and $\boldsymbol{x}_k$ denotes the random data at agent k. We construct the global cost function:

$$J(w) \triangleq \sum_{k=1}^{K} p_k J_k(w) \tag{2}$$

where the $p_k \geq 0$ denote convex combination weights that add up to one, i.e. $\sum_{k=1}^{K} p_k = 1$. When data realizations for $\boldsymbol{x}_k$ can be aggregated at a central location, descent along the negative gradient of (2) can be approximated by means of a *centralized* stochastic gradient algorithm of the form [1–3]:

$$\boldsymbol{w}_i^{\text{cent}} = \boldsymbol{w}_{i-1} - \mu \widehat{\nabla J}(\boldsymbol{w}_{i-1}^{\text{cent}}) \tag{3}$$

where $\widehat{\nabla J}(\cdot)$ denotes a stochastic gradient approximation constructed at time $i-1$. One possible construction is to let:

$$\widehat{\nabla J}^{c,K}(\boldsymbol{w}_{i-1}^{\text{cent}}) \triangleq \sum_{k=1}^{K} p_k \nabla Q_k(\boldsymbol{w}_{i-1}^{\text{cent}}; \boldsymbol{x}_{k,i-1}) \tag{4}$$

This work was supported in part by NSF grant CCF-1524250. Emails:{stefan.vlaski, ali.sayed}@epfl.ch.

which is obtained by employing a weighted combination of instantaneous approximations using all K realizations available at time $i-1$. This construction requires the evaluation of K (stochastic) gradients per iteration. If computational constraints limit the number of gradient evaluations per iteration to one, we can instead randomly sample an agent location k from the available data and let:

$$\widehat{\nabla J}^{c,1}(\boldsymbol{w}_{i-1}^{\text{cent}}) = \nabla Q_k(\boldsymbol{w}_{i-1}^{\text{cent}}; \boldsymbol{x}_{k,i-1}), \text{ with prob. } p_k, \tag{5}$$

The evident drawback of such a simplified centralized strategy is that only one sample is processed and a large number of samples is discarded at every iteration, resulting in a higher-variance estimate than (4). We can hence expect the construction (4) to result in better performance relative to the simplified choice (5) for a given learning rate. When communication constraints limit the exchange of information among agents, we can instead appeal to decentralized strategies. For the purpose of this work, we shall focus on the standard diffusion strategy, which takes the form:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{w}_{k,i-1} - \mu \nabla Q_k(\boldsymbol{w}_{k,i-1}; \boldsymbol{x}_{k,i-1}) \tag{6a}$$

$$\boldsymbol{w}_{k,i} = \sum_{\ell=1}^{N} a_{\ell k} \boldsymbol{\phi}_{\ell,i} \tag{6b}$$

where $a_{\ell k}$ denote convex combination coefficients satisfying:

$$a_{\ell k} \geq 0, \quad \sum_{\ell \in \mathcal{N}_k} a_{\ell k} = 1, \quad a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k \tag{7}$$

The symbol $\mathcal{N}_k$ denotes the set of neighbors of agent k. When the graph is strongly-connected, it follows from the Perron-Frobenius theorem that the combination matrix A has a spectral radius of one and a single eigenvalue at one with corresponding eigenvector [3]:

$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0 \tag{8}$$

We note that the vector p defines the weights appearing in (2), and can be designed in a decentralized manner by choosing appropriate combination weights $a_{\ell k}$ [3, Eq. (8.96)]. Comparing the diffusion strategy (6a)–(6b) to the centralized constructions (4) or (5), we observe that the adaptation step (6a) carries the same complexity *per agent* as the simplified construction (5). However, since these computations are performed at K agents in parallel, and the information is diffused over the network through the combination step (6b), we expect the diffusion strategy (6a)–(6b) to outperform the simplified centralized strategy (5) and more closely match the full construction (4). In fact, the spectral properties (8) of the combination weights (7) allow us to establish the following relation for the weighted network mean $\boldsymbol{w}_{c,i} \triangleq \sum_{k=1}^{K} p_k \boldsymbol{w}_{k,i}$ [3,4]:

$$\boldsymbol{w}_{c,i} = \boldsymbol{w}_{c,i-1} - \mu \sum_{k=1}^{K} p_k \nabla Q_k(\boldsymbol{w}_{k,i-1}; \boldsymbol{x}_{k,i-1}) \tag{9}$$

which *almost* corresponds to the centralized recursion (3)–(4) with the full gradient approximation (4) with the only difference being that the stochastic gradients are evaluated at the individual iterates $\boldsymbol{w}_{k,i-1}$ instead of the weighted network centroid $\boldsymbol{w}_{c,i-1}$. So long as the iterates $\boldsymbol{w}_{k,i-1}$ cluster around the network centroid, and under appropriate smoothness conditions on the (stochastic) gradients, it is hence to be expected that the network centroid (9) will match the performance of the full gradient approximation (4). This intuition has been studied in great detail and formalized for strongly convex cost functions, establishing that *all* iterates $\boldsymbol{w}_{k,i}$ in (6a)–(6b) will actually match the centralized full gradient approximation (4) both in terms of convergence rate [4] and steady-state error [5], which implies a linear improvement over the simplified construction (5) in terms of the number of agents [3] when employing a symmetric combination policy for which $p_k = \frac{1}{K}$.

More recently, these results have been extended to the pursuit of first-order stationary points in non-convex environments [6, 7] for consensus and the exact diffusion algorithm [8]. First-order stationary points can include saddle-points and even local maxima and can generate a bottleneck for many optimization algorithms and problem formulations [9]. Hence, the purpose of this work is is to establish that linear speedup can also be expected in the escape from saddle-points and pursuit of second-order stationary points for non-convex optimization problems. To this end, we refine and exploit recent results in [10, 11].

1.1. Related Works

Strategies for decentralized optimization include incremental strategies [12], and decentralized gradient descent (or consensus) [13], as well as the diffusion algorithm [3, 4, 14]. A second class of strategies is based on primal-dual arguments [8, 15–19]. While most of these algorithms are applicable to non-convex optimization problems, most performance guarantees in non-convex environments are limited to establishing convergence to first-order stationary points, i.e., points where the gradient is equal to zero [6, 7, 20–22].

Landscape analysis of commonly employed loss surfaces has uncovered that in many important settings such as tensor decomposition [23], matrix completion [24], low-rank recovery [25], as well as certain deep learning architectures [26], *all* local minima correspond to *global* minima and *all* other first-order stationary points have a *strict-saddle* property, which states that the Hessian matrix has at least one negative eigenvalue. These results have two implications. First, while first-order stationarity is a useful result in the sense that it ensures stability of the algorithm, even in non-convex environments, it is not sufficient to guarantee satisfactory performance, since first-order stationary points include strict saddle-points, which need not be globally or even locally optimal. On the other hand, establishing the escape from strict saddle-points, is sufficient to establish convergence to *global* optimality in all of these problems.

These observations have sparked a number of works examining second-order guarantees of local descent algorithms. Strategies for the escape from saddle-points can generally be divided into one of two classes. First, since the Hessian at every strict-saddle point, by definition, contains at least one negative eigenvalue, the descent direction can be identified by directly employing the Hessian matrix [27] or through an intermediate search for the negative curvature direction [28, 29]. The second class of strategies leverages the fact that perturbations in the initialization [30] or the update direction [23, 31–33] cause iterates of first-order algorithms to not get "stuck" in strict saddle-points, which can be shown to be unstable. Recently these results have been extended to decentralized optimization with deterministic gradients and random initialization [34] as well as stochastic gradients with diminishing step-size and decaying additive noise [35] as well as constant step-sizes [10, 11]. We establish in this work, that the saddle-point escape time of the diffusion strategy (6a)–(6b) decays linearly with the number of agents in the network when symmetric combination policies are employed and show how asymmetric combination policies can result in further improvement when agents have access to estimates of varying quality.

2. MODELING CONDITIONS

We shall be employing the following common modeling conditions [3, 23, 32, 35]. See [10, 11] for a discussion.

Assumption 1 (Smoothness). *For each k, the gradient $\nabla J_k(\cdot)$ is Lipschitz, namely, for any $x, y \in \mathbb{R}^M$:*

$$\|\nabla J_k(x) - \nabla J_k(y)\| \leq \delta \|x - y\| \tag{10}$$

Furthermore, $J_k(\cdot)$ is twice-differentiable with Lipschitz Hessian:

$$\|\nabla^2 J_k(x) - \nabla^2 J_k(y)\| \leq \rho \|x - y\| \tag{11}$$

For each pair of agents k and ℓ, the gradient disagreement is bounded, namely, for any $x \in \mathbb{R}^M$:

$$\|\nabla J_k(x) - \nabla J_\ell(x)\| \leq G \tag{12}$$

□

Assumption 2 (Gradient noise process). *For each k, the gradient noise process is defined as*

$$\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) = \widehat{\nabla J}_k(\boldsymbol{w}_{k,i-1}) - \nabla J_k(\boldsymbol{w}_{k,i-1}) \tag{13}$$

and satisfies

$$\mathbb{E}\left\{\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) | \boldsymbol{\mathcal{F}}_{i-1}\right\} = 0 \tag{14a}$$

$$\mathbb{E}\left\{\|\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\|^4 | \boldsymbol{\mathcal{F}}_{i-1}\right\} \leq \sigma_k^4 \tag{14b}$$

where we denote by $\boldsymbol{\mathcal{F}}_i$ the filtration generated by the random processes $\boldsymbol{w}_{k,j}$ for all k and $j \leq i$ and for some non-negative constants σ_k^4. We also assume that the gradient noise processes are pairwise uncorrelated over the space conditioned on $\boldsymbol{\mathcal{F}}_{i-1}$. □

Assumption 3 (Lipschitz covariances). *The gradient noise process has a Lipschitz covariance matrix, i.e.,*

$$R_{s,k}(\boldsymbol{w}_{k,i-1}) \triangleq \mathbb{E}\left\{\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})\boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1})^{\mathsf{T}} | \boldsymbol{\mathcal{F}}_{i-1}\right\} \tag{15}$$

satisfies

$$\|R_{s,k}(x) - R_{s,k}(y)\| \leq \beta_R \|x - y\|^\gamma \tag{16}$$

for some β_R and $0 < \gamma \leq 4$. □

We shall also make the simplifying assumption.

Assumption 4 (Gradient noise lower bound). *The gradient noise covariance $R_{s,k}(x)$ at every agent is bounded from below:*

$$R_{s,k}(x) \geq \sigma_{\ell,k} I \tag{17}$$

□

This condition can be loosened significantly by requiring a gradient noise component to be present only in the vicinity of strict saddle-points and only in the local descent direction, see e.g. [11, 32]. Nevertheless, the simplified condition can always be ensured for example by adding a small amount of isotropic noise, similar to [23, 31] and will be sufficient for the purpose this work.

3. CONVERGENCE ANALYSIS

3.1. Noise Variance Relations

The performance guarantees established in [10, 11] depend on the statistical properties of the weighted gradient noise term:

$$\boldsymbol{s}_i \triangleq \sum_{k=1}^{K} p_k \, \boldsymbol{s}_{k,i}(\boldsymbol{w}_{k,i-1}) \tag{18}$$

Under assumptions 1–4, we can refine the bounds from [10]:

Lemma 1 (**Variance Bounds**). *Under assumptions 1–4 we have:*

$$\mathbb{E}\left\{\|\boldsymbol{s}_i\|^2 | \boldsymbol{\mathcal{F}}_{i-1}\right\} \leq \sum_{k=1}^{K} p_k^2 \sigma_k^2 \tag{19}$$

$$\left(\sum_{k=1}^{K} p_k^2 \sigma_{k,\ell}^2\right) I \leq \mathbb{E}\, \boldsymbol{s}_i \, \boldsymbol{s}_i^{\mathsf{T}} \leq \left(\sum_{k=1}^{K} p_k^2 \sigma_k^2\right) I \tag{20}$$

Proof. Relations (19) and (20) follow from the pairwise uncorrelatedness condition in assumption 2 after cross-multiplying. □

From (19) we observe that the average noise term (18) driving the network centroid experiences a variance reduction. Specifically, in the case when $p_k = 1/K$ and $\sigma_k = \sigma$ we would obtain $\mathbb{E}\left\{\|\boldsymbol{s}_i\|^2 | \boldsymbol{\mathcal{F}}_{i-1}\right\} \leq \sigma^2/K$. This K-fold reduction in gradient noise variance is at the heart of the improved performance established for strongly-convex costs [3] and in the pursuit of first-order stationary points [6]. We shall establish in the sequel that this improvement also holds in the time required to escape from undesired saddle-points.

3.2. Space Decomposition

Definition 1 (Sets). *The parameter space $\mathbb{R}^M$ is decomposed into:*

$$\mathcal{G} \triangleq \left\{ w : \|\nabla J(w)\|^2 \geq \mu \frac{c_2}{c_1} \left(1 + \frac{1}{\pi}\right) \right\} \tag{21}$$

$$\mathcal{H} \triangleq \left\{ w : w \in \mathcal{G}^C, \lambda_{\min}\left(\nabla^2 J(w)\right) \leq -\tau \right\} \tag{22}$$

$$\mathcal{M} \triangleq \left\{ w : w \in \mathcal{G}^C, \lambda_{\min}\left(\nabla^2 J(w)\right) > -\tau \right\} \tag{23}$$

where τ is a small positive parameter, $0 < \pi < 1$ is a parameter to be chosen, $c_1 = \frac{1}{2}(1 - 2\mu\delta) = O(1)$ and $c_2 = \frac{\delta}{2}\left(\sum_{k=1}^{K} p_k^2 \sigma_k^2\right) = O\left(\sum_{k=1}^{K} p_k^2 \sigma_k^2\right)$. Note that $\mathcal{G}^C = \mathcal{H} \cup \mathcal{M}$. We also define the probabilities $\pi_i^{\mathcal{G}} \triangleq \Pr\{\boldsymbol{w}_{c,i} \in \mathcal{G}\}$, $\pi_i^{\mathcal{H}} \triangleq \Pr\{\boldsymbol{w}_{c,i} \in \mathcal{H}\}$ and $\pi_i^{\mathcal{M}} \triangleq \Pr\{\boldsymbol{w}_{c,i} \in \mathcal{M}\}$. Then for all i, we have $\pi_i^{\mathcal{G}} + \pi_i^{\mathcal{H}} + \pi_i^{\mathcal{M}} = 1$. □

Points in the complement of $\mathcal{G}$ have small gradient norm and hence correspond to approximately first-order stationary points. These points are further classified into strict-saddle points $\mathcal{H}$, where the Hessian has a significant negative eigenvalue, and second-order stationary points $\mathcal{M}$. Pursuit of second-order stationary points requires descent for points in $\mathcal{G}$ as well as $\mathcal{H}$.

3.3. Performance Guarantees

Due to space limitations, we forego a detailed discussion on the derivation of the second-order guarantees of the diffusion algorithm (6a)–(6b) and refer the reader to [10, 11]. We instead briefly list the guarantees resulting from the variance bounds (19)–(20) and will focus on the dependence on the combination policy further below. Adjusting the theorems in [10, 11] to account for the variance bounds (19)–(20), we obtain:

Theorem 1 (**Network disagreement (4th order)**). *Under assumptions 1 2, the network disagreement is bounded after sufficient iterations $i \geq i_o$ by:*

$$\begin{aligned} &\mathbb{E}\left\|\boldsymbol{\mathcal{W}}_i - \left(\mathbb{1}p^{\mathsf{T}} \otimes I\right)\boldsymbol{\mathcal{W}}_i\right\|^4 \\ &\leq \mu^4 \frac{\|\mathcal{V}_L\|^4 \left\|J_\epsilon^{\mathsf{T}}\right\|^4}{\left(1 - \left\|J_\epsilon^{\mathsf{T}}\right\|\right)^4} \|\mathcal{V}_R^{\mathsf{T}}\|^4 K^2 \left(G^4 + \max_k \sigma_k^4\right) + o(\mu^4) \end{aligned} \tag{24}$$

where $\left\|J_\epsilon^{\mathsf{T}}\right\| = \lambda_2(A) + \epsilon \approx \lambda_2(A)$ denotes the mixing rate of the adjacency matrix, $\mathcal{A} = \mathcal{V}_\epsilon \mathcal{J} \mathcal{V}_\epsilon^{-1}$ with $\mathcal{V}_\epsilon = \mathrm{row}\{p \otimes I, \mathcal{V}_R\}$ and $\mathcal{V}_\epsilon^{-1} = \mathrm{col}\{\mathbb{1}^{\mathsf{T}}, \mathcal{V}_L^{\mathsf{T}}\}$, $i_o = \log\left(o(\mu^4)\right)/\log\left(\left\|J_\epsilon^{\mathsf{T}}\right\|\right)$ and $o(\mu^4)$ denotes a term that is higher in order than μ^4.

Proof. The argument is an adjustment of [10, Theorem 1]. □

This result ensures that the entire network clusters around the network centroid $\boldsymbol{w}_{c,i}$ after sufficient iterations, allowing us to leverage it as a proxy for all agents.

Theorem 2 (**Descent relation**). *Beginning at $\boldsymbol{w}_{c,i-1}$ in the large gradient regime $\mathcal{G}$, we can bound:*

$$\begin{aligned} &\mathbb{E}\left\{J(\boldsymbol{w}_{c,i}) | \, \boldsymbol{w}_{c,i-1} \in \mathcal{G}\right\} \\ &\leq \mathbb{E}\left\{J(\boldsymbol{w}_{c,i-1}) | \, \boldsymbol{w}_{c,i-1} \in \mathcal{G}\right\} - \mu^2 \frac{c_2}{\pi} + \frac{O(\mu^3)}{\pi_{i-1}^{\mathcal{G}}} \end{aligned} \tag{25}$$

as long as $\pi_{i-1}^{\mathcal{G}} = \Pr\{\boldsymbol{w}_{c,i-1} \in \mathcal{G}\} \neq 0$ where the relevant constants are listed in definition 1.

Proof. The argument is an adjustment of [10, Theorem 2]. □

Theorem 3 (**Descent through strict saddle-points**). *Suppose $\pi_{i^\star}^{\mathcal{H}} \neq 0$, i.e., $\boldsymbol{w}_{c,i^\star}$ is approximately stationary with significant negative eigenvalue. Then, iterating for i^s iterations after $i^\star$ with*

$$i^s = \frac{\log\left(2M \frac{\left(\sum_{k=1}^{K} p_k^2 \sigma_k^2\right)}{\left(\sum_{k=1}^{K} p_k^2 \sigma_{k,\ell}^2\right)} + 1\right)}{O(\mu\tau)} \tag{26}$$

guarantees

$$\begin{aligned} &\mathbb{E}\left\{J(\boldsymbol{w}_{c,i^\star+i^s}) | \, \boldsymbol{w}_{c,i^\star} \in \mathcal{H}\right\} \\ &\leq \mathbb{E}\left\{J(\boldsymbol{w}_{c,i^\star}) | \, \boldsymbol{w}_{c,i^\star} \in \mathcal{H}\right\} - \frac{\mu}{2} M \left(\sum_{k=1}^{K} p_k^2 \sigma_k^2\right) + \frac{o(\mu)}{\pi_{i^\star}^{\mathcal{H}}} \end{aligned} \tag{27}$$

Proof. The argument is an adjustment of [11, Theorem 1]. □

Theorem 2 ensures descent in one iteration as long as the gradient norm is sufficiently large, while 3 ensures descent even for first-order stationary points, as long as the Hessian has a negative eigenvalue in a number of iterations i^s that can be bounded. This ensures efficient escape from strict saddle-points. We conclude:

Theorem 4. *For sufficiently small step-sizes μ, we have with probability $1 - \pi$, that $\boldsymbol{w}_{c,i^o} \in \mathcal{M}$, i.e.,*

$$\|\nabla J(\boldsymbol{w}_{c,i^o})\|^2 \leq O\left(\mu \left(\sum_{k=1}^{K} p_k^2 \sigma_k^2\right)\right) \tag{28}$$

and $\lambda_{\min}\left(\nabla^2 J(\boldsymbol{w}_{c,i^o})\right) \geq -\tau$ *in at most* i^o *iterations, where*

$$i^o \leq \frac{2\left(J(w_{c,0}) - J^o\right)}{\mu^2\delta\left(\sum_{k=1}^K p_k^2\sigma_k^2\right)\pi} i^s \tag{29}$$

Proof. The argument is an adjustment of [11, Theorem 2]. □

4. COMPARATIVE ANALYSIS

4.1. Step-Size Normalization

Note that in Theorem 4, both the limiting accuracy (28) and convergence rate (29) depend on the combination policy and network size through $\sum_{k=1}^K p_k^2\sigma_k^2$. To facilitate comparison, we shall normalize the step-size in (6a):

$$\mu' \triangleq \frac{\mu}{\sum_{k=1}^K p_k^2\sigma_k^2} \tag{30}$$

Under this setting, Theorem 4 ensures a point i^o satisfying

$$\|\nabla J(\boldsymbol{w}_{c,i^o})\|^2 \leq O(\mu) \tag{31}$$

and $\lambda_{\min}\left(\nabla^2 J(\boldsymbol{w}_{c,i^o})\right) \geq -\tau$ in at most:

$$i^o \leq \frac{2\left(J(w_{c,0}) - J^o\right)}{\mu^2\delta\pi}\left(\sum_{k=1}^K p_k^2\sigma_k^2\right) i^s \tag{32}$$

iterations with

$$i^s = \frac{\log\left(2M\frac{\left(\sum_{k=1}^K p_k^2\sigma_k^2\right)}{\left(\sum_{k=1}^K p_k^2\sigma_{k,\ell}^2\right)} + 1\right)}{O(\mu\tau)}\left(\sum_{k=1}^K p_k^2\sigma_k^2\right) \tag{33}$$

Note that the normalization of the step-size causes (31) to become independent of $\left(\sum_{k=1}^K p_k^2\sigma_k^2\right)$, allowing for the fair evaluation of (32) and (33) as a function of the number of agents.

4.2. Linear Speedup Using Symmetric Combination Weights

When the combination matrix A is symmetric, i.e., $A = A^{\mathsf{T}}$, it follows that $p_k = \frac{1}{K}$ [3]. For simplicity, in this section, we shall also assume a uniform data profile for all agents, i.e., that $\sigma_k = \sigma$ and $\sigma_{\ell,k} = \sigma_\ell$ for all k. We obtain:

Theorem 5 (Linear Speedup for Symmetric Policies). *Under the step-size normalization* (30)*, and for symmetric combination policies* $A = A^{\mathsf{T}}$ *with the uniform data profile* $\sigma_k^2 = \sigma^2$ *and* $\sigma_{\ell,k}^2 = \sigma_\ell^2$ *for all* k*, the escape time simplifies to:*

$$i^s = \frac{\log\left(2M\frac{\sigma^2}{\sigma_\ell^2} + 1\right)}{O(\mu\tau)}\frac{\sigma^2}{K} = O\left(\frac{1}{\mu\tau K}\right) \tag{34}$$

Proof. The result follows immediately after cancellations. □

4.3. Benefit of Employing Asymmetric Combination Weights

In this subsection, we show how employing asymmetric combination weights can be beneficial in terms of the time required to escape saddle-points when the data profile across agents is no longer uniform. In particular, we will no longer require the upper and lower bounds σ_k^2 and $\sigma_{\ell,k}^2$ to be common for all agents, and no longer require the combination policy to be symmetric. Instead, to simplify the derivation, we assume that the gradient noise is approximately isotropic, i.e., $\sigma_k^2 \approx \sigma_{k,\ell}^2$ so that (33) can be simplified to:

$$i^s \approx O\left(\frac{\sum_{k=1}^K p_k^2\sigma_k^2}{\mu\tau}\right) \tag{35}$$

Then, we can formulate the following optimization problem to minimize the escape time i^s over the space of valid combination policies:

$$\min_A \sum_{k=1}^K p_k^2\sigma_k^2 \ \text{ s.t. } a_{\ell k} \geq 0,\ \sum_{\ell\in\mathcal{N}_k} a_{\ell k} = 1,\ a_{\ell k} = 0 \text{ if } \ell \notin \mathcal{N}_k,$$
$$Ap = p, \quad \mathbb{1}^{\mathsf{T}} p = 1, \quad p_k > 0. \tag{36}$$

This precise optimization problem has appeared before in the pursuit of asymmetric combination policies that minimize the steady-state error of the diffusion strategy (6a)–(6b) in *strongly-convex* environments [3]. Its solution is available in closed form and can even be pursued in a decentralized manner, requiring only exchanges among neighbors [3].

Theorem 6 (Metropolis-Hastings Combination Policy [3]). *Under the step-size normalization* (30)*, the asymmetric Metropolis-Hastings combination policy minimizes the approximate saddle-point escape time* (35)*. It takes the form:*

$$a_{\ell k}^o = \begin{cases} \frac{\sigma_k^2}{\max\{n_k\sigma_k^2, n_\ell\sigma_\ell^2\}}, & \ell \in \mathcal{N}_k, \\ 1 - \sum_{m\in\mathcal{N}_k\setminus\{k\}} a_{mk}^o, & \ell = k. \end{cases} \tag{37}$$

where $n_k = |\mathcal{N}_k|$ *denotes the size of the neighborhood of agent* k*.*

5. SIMULATIONS

We construct a sample landscape to verify the linear speedup in the size of the network indicated by the analysis in this work. The loss function is constructed from a single-layer neural network with a linear hidden layer and a logistic activation function for the output layer. Penalizing this architecture with the cross-entropy loss gives:

$$J(w_1, W_2) = \mathbb{E}\log\left(1 + e^{-\boldsymbol{\gamma} w_1^{\mathsf{T}} W_2 \boldsymbol{h}}\right) + \frac{\rho}{2}\|w_1\|^2 + \frac{\rho}{2}\|W_2\|_F^2 \tag{38}$$

where w_1 and W_2 denote the weights of the individual layers, $\boldsymbol{h} \in \mathbb{R}^M$ denotes the feature vector, and $\boldsymbol{\gamma} \in \{\pm 1\}$ is the class variable. It can be verified that this loss has a single strict saddle-point at $w_1 = W_2 = 0$ and global minima in the positive and negative quadrant, respectively [11]. We show the evolution of the function value at the network centroid under the step-size normalization rule (30) and observe a linear speedup in K, consistent with (34) while noting no significant differences in steady-state performance, which is consistent with (31).

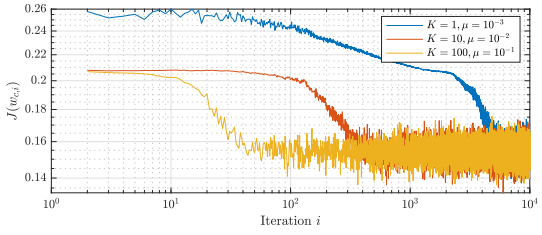

Fig. 1: Linear speedup in saddle-point escape time.

6. REFERENCES

[1] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, Sep 1951.

[2] B. T. Polyak, *Introduction to Optimization*, Optimization Software, 1997.

[3] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, July 2014.

[4] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks - Part I: Transient analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3487–3517, June 2015.

[5] J. Chen and A. H. Sayed, "On the learning behavior of adaptive networks – Part II: Performance analysis," *IEEE Transactions on Information Theory*, vol. 61, no. 6, pp. 3518–3548, June 2015.

[6] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? A case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems 30*, pp. 5330–5340. 2017.

[7] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "d^2: Decentralized training over decentralized data," in *Proceedings of the 35th International Conference on Machine Learning*, 2018, vol. 80, pp. 4848–4856.

[8] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, "Exact diffusion for distributed optimization and learning—Part I: Algorithm development," *IEEE Transactions on Signal Processing*, vol. 67, no. 3, pp. 708–723, Feb 2019.

[9] S. S. Du, C. Jin, J. D. Lee, M. I. Jordan, B. Póczos, and A. Singh, "Gradient descent can take exponential time to escape saddle points," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1067–1077.

[10] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments – Part I: Agreement at a Linear rate," *available as arXiv:1907.01848*, July 2019.

[11] S. Vlaski and A. H. Sayed, "Distributed learning in non-convex environments – Part II: Polynomial escape from saddle-points," *available as arXiv:1907.01849*, July 2019.

[12] D. P. Bertsekas, "A new class of incremental gradient methods for least squares problems," *SIAM J. Optim.*, vol. 7, no. 4, pp. 913–926, April 1997.

[13] S. Sundhar Ram, A. Nedic, and V. V. Veeravalli, "Distributed stochastic subgradient projection algorithms for convex optimization," *Journal of Optimization Theory and Applications*, vol. 147, no. 3, pp. 516–545, Dec 2010.

[14] S. Vlaski, L. Vandenberghe, and A. H. Sayed, "Regularized diffusion adaptation via conjugate smoothing," *available as arXiv:1909.09417*, September 2019.

[15] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Cooperative convex optimization in networked systems: Augmented lagrangian algorithms with directed gossip communication," *IEEE Transactions on Signal Processing*, vol. 59, no. 8, pp. 3889–3902, Aug 2011.

[16] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic Control*, vol. 57, no. 3, pp. 592–606, March 2012.

[17] M. Jaggi, V. Smith, M. Takáč, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, "Communication-efficient distributed dual coordinate ascent," in *Proc. International Conference on Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 3068–3076.

[18] K. I. Tsianos and M. G. Rabbat, "Distributed dual averaging for convex optimization under communication delays," in *Proc. American Control Conference (ACC)*, Montreal, Canada, June 2012, pp. 1067–1072.

[19] W. Shi, Q. Ling, G. Wu, and W. Yin, "EXTRA: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[20] P. Di Lorenzo and G. Scutari, "NEXT: in-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, June 2016.

[21] T. Tatarenko and B. Touri, "Non-convex distributed optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3744–3757, Aug. 2017.

[22] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, Jan. 2019.

[23] R. Ge, F. Huang, C. Jin, and Y. Yuan, "Escaping from saddle points—online stochastic gradient for tensor decomposition," in *Proc. of Conference on Learning Theory*, Paris, France, 2015, pp. 797–842.

[24] R. Ge, J. D. Lee, and T. Ma, "Matrix completion has no spurious local minimum," in *Advances in Neural Information Processing Systems*, 2016, pp. 2973–2981.

[25] R. Ge, C. Jin, and Y. Zheng, "No spurious local minima in nonconvex low rank problems: A unified geometric analysis," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 1233–1242.

[26] K. Kawaguchi, "Deep learning without poor local minima," in *Advances in Neural Information Processing Systems*, pp. 586–594. 2016.

[27] Y. Nesterov and B.T. Polyak, "Cubic regularization of newton method and its global performance," *Mathematical Programming*, vol. 108, no. 1, pp. 177–205, Aug 2006.

[28] C. Fang, C. J. Li, Z. Lin, and T. Zhang, "SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator," in *Proc. of NIPS*, pp. 689–699. Montreal, Canada, 2018.

[29] Z. Allen-Zhu and Y. Li, "NEON2: Finding local minima via first-order oracles," in *Proc. of NIPS*, pp. 3716–3726. Montreal, Canada, Dec. 2018.

[30] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, "Gradient descent only converges to minimizers," in *29th Annual Conference on Learning Theory*, New York, 2016, pp. 1246–1257.

[31] C. Jin, P. Netrapalli, R. Ge, S. M. Kakade and M. I. Jordan, "Stochastic gradient descent escapes saddle points efficiently," *available as arXiv:1902.04811*, Feb. 2019.

[32] H. Daneshmand, J. Kohler, A. Lucchi and T. Hofmann, "Escaping saddles with stochastic gradients," *available as arXiv:1803.05999*, March 2018.

[33] S. Vlaski and A. H. Sayed, "Second-order guarantees of stochastic gradient descent in non-convex optimization," *available as arXiv:1908.07023*, August 2019.

[34] A. Daneshmand, G. Scutari and V. Kungurtsev, "Second-order guarantees of distributed gradient algorithms," *available as arXiv:1809.08694*, Sep. 2018.

[35] B. Swenson, S. Kar, H. V. Poor and J. M. F. Moura, "Annealing for distributed global optimization," *available as arXiv:1903.07258*, March 2019.