Distributed Online Non-convex Optimization with Composite Regret

[1]Zhanhong Jiang, [2]Aditya Balu, [2]Xian Yeow Lee, [1]Young M. Lee, [3]Chinmay Hegde, [2]Soumik Sarkar
[1]*Johnson Controls,* [2]*Iowa State University,* [3]*New York University*
{zhanhong.jiang, young.m.lee}@jci.com, {xylee, baditya, soumiks}@iastate.edu, chinmay.h@nyu.edu

***Abstract*—Regret has been widely adopted as the metric of choice for evaluating the performance of online optimization algorithms for distributed, multi-agent systems. However, variations in data or model associated with each agent can significantly impact decisions, and requires consensus among agents. Moreover, most existing works have focused on developing approaches for (strongly) convex losses, and very few results have been obtained in terms of regret bounds in distributed online optimization for general non-convex losses. To address these two issues, we propose a novel composite regret with a new network-based metric to evaluate distributed online optimization algorithms. We concretely define static and dynamic forms of the composite regret. By leveraging the dynamic form of our composite regret, we develop a consensus-based online normalized gradient (CONGD) approach for pseudo-convex losses and then show a sublinear behavior for CONGD relating to a regularity term for the path variation of the optimizer. For general non-convex losses, we first explore the regrets defined based on the recent advances such that no deterministic algorithm can achieve the sublinear regret. We then develop the distributed online non-convex optimization with composite regret (DINOCO) without access to the gradients, depending on an offline optimization oracle. We show that DINOCO can achieve sublinear regret; to our knowledge, this is the first regret bound for general distributed online non-convex learning.**

***Index Terms*—Composite regret, distributed optimization, online optimization, non-convex optimization**

I. Introduction

a) Setting: Online convex optimization (OCO) is well established in the algorithmic learning theory literature [27]. However, the emergence of complex tasks and massive datasets requires the investigation of *distributed* online optimization for multi-agent networked systems. While extensive studies have been done for distributed OCO [17, 18, 10], online non-convex optimization has rarely been explored. However, this now becomes more demanding due to the rapid development of various deep learning models. In this paper, we initially fill this gap.

b) Prior work: The performance of a distributed OCO algorithm is generally measured in terms of either the *static* [35] (or *dynamic* [6]) regret, which compares the accumulated losses suffered by each player with the loss suffered by the best fixed (or dynamic) policy. By leveraging static regret, quite a few previous works have been able to achieve sublinear regret bounds. For instance, Nedić et al. [22] proposed a form of Nesterov's primal-dual algorithm with dual averaging which achieved a sublinear growth $\mathcal{O}(\sqrt{K})$ when the step size was $1/\sqrt{K}$ and the losses were Lipschitz continuous convex functions with Lipschitz gradients. Akbari et al. [2] improved the order of regret to $\mathcal{O}(\log K)$ when the loss functions were strongly convex with the online subgradient push-sum algorithm. To cope with the presence of cost uncertainties and the switching topologies, Hosseini et al. [14] developed the distributed weighted dual averaging method and the distributed online allocation algorithm to achieve the sublinear regret of order $\mathcal{O}(\sqrt{K})$. Lee and Zavlanos [16] developed a distributed online primal-dual algorithm with a balanced communication graph and bounded Lagrangian multipliers, which obtained an analogous sublinear regret bound.

More recent work by Li et al. [17] extended the problem to unbalanced communication graphs while relaxing the assumption that the Lagrangian multipliers were bounded. They also studied the capability of handling coupled inequality constraints by proposing the distributed online primal-dual push-sum algorithm. To determine an effective strategy to handle constraints, a distributed online saddle point algorithm was proposed in [24] to achieve a sublinear regret $\mathcal{O}(\sqrt{K})$.

The definition of static regret correspondingly requires that the best policy remains unchanged during the time interval of interest. However, this restriction may not apply to all practical situations (particularly in the distributed setting), therefore motivating the notion of *dynamic* regret. Shahrampour and Jadbabaie [26] developed a distributed mirror descent method and established a sublinear regret bound $\mathcal{O}(\sqrt{K})$ that was inversely proportional to the spectral gap of the network. Nazari et al. [21] proposed an adaptive gradient method for distributed online optimization (DADAM) using dynamic regret and showed that the proposed method was able to outperform the centralized variant for certain classes of loss functions. However, they defined a local regret for dealing with the nonconvex case instead of using dynamic regret. Recently, Zhang et al. [34] extended the gradient tracking techniques motivated by the recent development [25] from the offline setting to the online setting with a gradient path variation [9] and proved the sublinear regret. A more recent work [4] showed that when the losses were strongly convex, using an inexact proximal gradient method enabled to converge R-linearly to a neighborhood of the optimal solution trajectory. With long-term constraints, Yuan et al. [31] investigated the regret and the cumulative constraint violation for strongly convex loss functions and obtained a similar regret bound as in the state-of-the-art centralized variants. Additionally, they considered the bandit feedback case and showed the sublinear growth for

the regret. In [28], the authors combined mirror descent and primal-dual approaches to achieve a sublinear dynamic regret and fits. To reduce the overhead of communication, the authors in [7] developed a decentralized online subgradient algorithm with only the signs of the states from the neighborhood of an agent and showed a standard sublinear regret $\mathcal{O}(\sqrt{K})$, which still provably holds true in a noisy environment. When the feedback delay was considered [8], if the delay was sublinear, the regret could still maintain the standard sublinear bound. To investigate the non-cooperative games in a dynamic environment, Meng et al. [19] proposed a distributed online primal-dual dynamic mirror descent algorithm, which was found to achieve a relatively worse sublinear regret $\mathcal{O}(K^{\frac{7}{8}})$, compared to the standard regret. When only with bandit feedback [20], they showed the regret of $\mathcal{O}(K^{\frac{13}{14}})$. While all the results only apply to convex losses.

In the case of non-convex objective functions, we are only aware of a limited number of (very recent) works. Gao et al. [11] studied the pseudo-convex loss using the online normalized gradient descent and its bandit variant to show sublinear regret bounds. Hazan et al. [12] and Aydore et al. [3] respectively proposed different computationally tractable notions of local regret, motivated by offline non-convex optimization. However, the techniques studied do not guarantee a good sublinear regret for general non-convex losses in the online case. The recent work [1] utilized the follow-the-perturbed-leader approach, showing that with a good offline optimization oracle (instead of the gradient oracle), the regret can be upper bounded by $\mathcal{O}(K^{\frac{2}{3}})$ for general non-convex losses. Subsequently, Suggala and Netrapalli [29] improved this bound to $\mathcal{O}(\sqrt{K})$. Such analysis facilitates understanding the regret growth order in online non-convex learning, but unfortunately do not hold for the distributed setting. In a more recent work Lu and Wang [18], the authors leveraged a first-order optimality condition-based regret to achieve the subliear regret $\mathcal{O}(\sqrt{K})$ with a diminishing step size. Though they claimed the online distributed mirror algorithm performed well, the techniques presented in their work do not guarantee the vanishing regret for general distributed non-convex losses.

c) Our contributions.: To summarize, most previous works have investigated distributed online optimization using either the static or dynamic regret, which depends entirely on the original loss objective. However, in a distributed setting with multiple agents, the consensus objective among them can be moved into the loss as a soft constraint, which is a tradeoff between the original loss and the network loss. Moreover, with the popularity of dynamic regret, considering only the temporal change of the path in a distributed setting, involving multiple agents is not sufficient. Different agents may not benefit from a biased best strategy played by one agent. Additionally, there still exists a theoretical gap in the regret bound for distributed online non-convex learning. Motivated by recent works in the offline setting [15, 33, 5], we propose in this paper a composite regret which takes into account both of original loss and network loss. Specifically, the following contributions are made in this paper.

- We propose a composite regret for distributed online optimization for both static and dynamic forms corresponding to different classes of loss functions. We then leverage this to prove regret bounds with pseudo-convex and non-convex losses.
- For pseudo-convex loss functions, we propose consensus-based online normalized gradient descent (CONGD). To analyze the regret bound for general non-convex losses, we first extend the result from a recent work to the distributed setting and conclude that no deterministic algorithm can achieve $o(1)$. We then propose the Distributed Online Non-convex Optimization with Composite Regret (DINOCO) based on the follow-the-perturbed-leader, defining an offline optimization oracle.
- With mild assumptions, we prove for pseudo-convex losses the regret can be upper bounded by $\mathcal{O}(\sqrt{K+P_K K})$, which matches the best performance in the centralized (non-distributed) variant. Here, P_K is a parameter defined below in Eq 2. For general non-convex losses, DINOCO is provably able to achieve the sublinear regret $\mathcal{O}(\sqrt{K})$. Table I summarizes the comparison between proposed and existing methods.

II. Preliminaries and Composite Regret

A. Distributed Regret

Consider a networked system involving N agents, which can be described by a graph $\mathcal{G}=(\mathcal{V},\mathcal{E})$, where $\mathcal{V}=\{1,2,...,N\}$ and $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$. In the rest of analysis, the graph $\mathcal{G}$ is assumed to be *static and undirected*. In distributed online optimization, at each period $k\in\{1,2,...,K\}$, an online player $i\in\{1,2,...,N\}$ individually chooses a feasible policy x_k^i from its decision set known by itself, $\mathcal{X}^i\subset\mathbb{R}^n$, and subsequently incurs a loss $f_k^i(x_k^i):\mathbb{R}^n\to\mathbb{R}$, where $f_k^i(\cdot)$ is a differentiable loss function. In contrast with offline optimization, a remarkably different feature in online optimization is that the player i must choose a policy for the period k without knowing the specific form of loss function $f_k^i(\cdot)$. We next introduce the concepts of distributed regrets. Specifically, the static regret is defined as

$$\text{S-regret}_K(\mathcal{A})=\sum_{k=1}^{K}\sum_{i=1}^{N}f_k^i(x_k^i)-\sum_{k=1}^{K}\sum_{i=1}^{N}f_k^i(x_*)$$

where $\mathcal{A}$ is an algorithm, N is the number of agents, x_* is the best strategy learned collaboratively by all agents and $x_*=\underset{x\in\mathcal{X}\subset\mathbb{R}^n}{\text{argmin}}\sum_{k=1}^{K}\sum_{i=1}^{N}f_k^i(x)$, where $\mathcal{X}:=\cap_{i=1}^{N}\mathcal{X}^i$. However, to emphasize each agent's individual best strategy, we define x_*^i as the best strategy played by the i-th agent such that $x_*^1=x_*^2=...=x_*^N=x_*$, which corresponds to all agents communicating sufficiently well. Without loss of generality, we denote the compact form of all the best strategies by $\mathbf{x}_*$, which stacks each best strategy from each agent, i.e., $\mathbf{x}_*=[x_*^1;x_*^2;...;x_*^N]\in\mathbb{R}^{nN}$. Hence, it is immediately obtained that $\mathbf{x}_*=[x_*;x_*;...;x_*]_{nN\times1}$. For the compact form $\mathbf{x}$, it should be noted that the constraint set is expanded to a

TABLE I
COMPARISONS BETWEEN DIFFERENT ONLINE OPTIMIZATION METHODS.

Method	setting	$\mathcal{F}$	$\nabla\mathcal{F}$	**step size**	**regret**	**regret bound**
OGD [13]	centralized	Str	B	Di	S	$\mathcal{O}(\log K)$
	centralized	C	B	Di/F	S	$\mathcal{O}(\sqrt{K})$
OMD [13]	centralized	C	B	F	S	$\mathcal{O}(\sqrt{K})$
RFTL [13]	centralized	C	B	F	S	$\mathcal{O}(\sqrt{K})$
FTPL [13]	centralized	C	B	F	S	$\mathcal{O}(\sqrt{K})$
	centralized	NC	B	F	S	$\mathcal{O}(\sqrt{K})$
ONGD [11]	centralized	PC	B	F	Dy	$\mathcal{O}(\sqrt{K+P_K K})$
DPD-MD [28]	distributed	C	B	Di	Dy	$\mathcal{O}(K^{\max\{a,1-a+b\}}+K^a)$
DOPD-PS [17]	distributed	C	B	Di	S	$\mathcal{O}(K^{\frac{9}{10}})$
DOPD-DMD [30]	distributed	C	B	Di	Dy	$\mathcal{O}(K^{\max(1-\kappa,\kappa)}+K^{1-\kappa}P_K)$
DOGT [34]	distributed	Str	B & L	F	Dy	$\mathcal{O}(1+P_K+\mathcal{V}_K)$
ODCMD [32]	distributed	C	B	F	S	$\mathcal{O}(\sqrt{K})$
ODM [18]	distributed	NC	B	Di	FOC	$\mathcal{O}(\sqrt{K})$
DABMD [10]	distributed	C	B	F	Dy	$\mathcal{O}(\sqrt{KP_K+K})$
CONGD (ours)	distributed	PC	B	F	DC	$\mathcal{O}(\sqrt{K+P_K K})$
DINOCO (ours)	distributed	NC	B	F	SC	$\mathcal{O}(\sqrt{K}N^2)$

[1] OGD: online gradient descent; [2] OMD: online mirror descent; [3] RFTL: regularized follow the leader; [4] FTPL: follow the perturbed leader; [5] ONGD: online normalized gradient descent; [6] $a, b \in (0,1)$; [7] DPD-MD: distributed primal-dual mirror descent; [8] DOPD-PS: distributed online primal-dual push-sum; [9] DOPD-DMD: distributed online primal-dual dynamic mirror descent; [10] Str: strongly convex; [11] C: convex; [12] DOGT: distributed online optimization with gradient tracking; [13] PC: pseudo-convex; [14] ODCMD: online distributed composite mirror descent; [15] B: bounded; [16] B & L: bounded and Lipschitz continuous; [17] Di: diminishing; [18] Di/F: diminishing or fixed; [19] F: fixed; [20] S: static; [21] Dy: dynamic; [22] SC: static composite; [23] DC: dynamic composite; [24] $\kappa \in (0,1)$; [25] ODM: online distributed mirror algorithm; [26] FOC: first-order optimality condition; [27] DABMD: Distributed Any-Batch Mirror Descent; [28] $P_K, \mathcal{V}_K$: path lengths of the optimizer and gradient variations.

higher dimension in the form of $\mathcal{X}^N$. For simplicity we assume that $n = 1$ in this paper such that $\mathcal{X}^N \subset \mathbb{R}^N$.

Alternatively, when using the dynamic regret, one can obtain the following definition

$$\text{D-regret}_K(\mathcal{A}) = \sum_{k=1}^{K}\sum_{i=1}^{N} f_k^i(x_k^i) - \sum_{k=1}^{K}\sum_{i=1}^{N} f_k^i(x_{*,k}),$$

where $x_{*,k}$ can be obtained by $x_{*,k} = \underset{x\in\mathcal{X}\subset\mathbb{R}^n}{\text{argmin}} \sum_{i=1}^{N} f_k^i(x)$. Similarly, the compact form of the time-varying best strategy is denoted by $\mathbf{x}_{*,k} = [x_{*,k}^1; x_{*,k}^2; ...; x_{*,k}^N]$. Unlike the static regret, the dynamic regret implies that when there is no restriction on the changes of loss functions, the regret is at most linear in K irrespective of the policies [6]. Additionally, to gain meaningful bounds, the temporal change of the loss function sequence $\{f_k\}_1^K$ is assumed to be bounded. Specifically, the losses are assumed to be taken from the set [6, 11]:

$$\mathcal{M} := \left\{ \{f_1, f_2, ..., f_K\} : \sum_{k=1}^{K-1} \sup_{x\in\mathcal{X}} |f_k(x) - f_{k+1}(x)| \leq M_K, M_K > 0 \right\}. \tag{1}$$

For OCO, Besbes et al. [6] proved sublinear regret with the above definition. In our analysis we adopt a slightly different regularity parameter defined in terms of the bounded worst-case variation of the optimal solution $x_{*,k}$ of $f_k(\cdot)$:

$$\mathcal{P} := \left\{ \{f_1, f_2, ..., f_K\} : \max_{x_{*,k}\in\text{argmin}_{x\in\mathcal{X}} f_k(x)} \sum_{k=1}^{K-1} \|x_{*,k} - x_{*,k+1}\| \leq P_K, P_K > 0 \right\}, \tag{2}$$

where $\|\cdot\|$ is the ℓ_2 norm.

B. Composite Regret

Based on the definitions of both static and dynamic regrets, the composite regret is defined formally as follows.

Definition 1. *The static composite regret of a distributed online algorithm $\mathcal{A}$ is defined as*

$$\begin{aligned}\text{SC-regret}_K(\mathcal{A}) &= \sum_{k=1}^{K}\sum_{i=1}^{N} f_k^i(x_k^i) \\ &+ c\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j\in Nb(i)} \pi_{ij}\|x_k^i - x_k^j\|^2 \\ &- \left(\sum_{k=1}^{K}\sum_{i=1}^{N} f_k^i(x_*^i) + c\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j\in Nb(i)} \pi_{ij}\|x_*^i - x_*^j\|^2\right),\end{aligned} \tag{3}$$

where $\Pi = (\pi_{ij}) \in \mathbb{R}^N \times \mathbb{R}^N$, $Nb(i) = \{j \in \mathcal{V} | (i,j) \in \mathcal{E}\} \cup \{i\}$. *Similarly, the dynamic composite regret of a distributed online algorithm* $\mathcal{A}$ *is defined as*

$$\begin{aligned}\text{DC-regret}_K(\mathcal{A}) &= \sum_{k=1}^{K}\sum_{i=1}^{N} f_k^i(x_k^i) \\ &+ c\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j\in Nb(i)} \pi_{ij}\|x_k^i - x_k^j\|^2 \\ &- \left(\sum_{k=1}^{K}\sum_{i=1}^{N} f_k^i(x_{*,k}^i) + c\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j\in Nb(i)} \pi_{ij}\|x_{*,k}^i - x_{*,k}^j\|^2\right).\end{aligned} \tag{4}$$

Based on Definition 1, one extra term of

$$c\left(\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j\in Nb(i)} \pi_{ij}\|x^i - x^j\|^2 - \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j\in Nb(i)} \pi_{ij}\|x_*^i - x_*^j\|^2\right)$$

for SC-regret$_K$ or

$$\begin{aligned}c\Bigg(&\sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j\in Nb(i)} \pi_{ij}\|x_k^i - x_k^j\|^2 \\ &- \sum_{k=1}^{K}\sum_{i=1}^{N}\sum_{j\in Nb(i)} \pi_{ij}\|x_{*,k}^i - x_{*,k}^j\|^2\Bigg)\end{aligned}$$

for DC-regret$_K$ is added accounting for the network loss caused by the disagreement among agents. c is a constant parameter that may vary depending on the different methods adopted. For simplicity, c can be set equal to 1, while in this paper, we will show that c can be expressed using the step size. To obtain clean analytical results, we rewrite Eqs. 3 and 4 in a more compact form. Defining

$$V(\mathbf{x}) = \sum_{i=1}^{N} f^i(x^i) + c\mathbf{x}^\top(\mathbf{I}_N - \Pi)\mathbf{x} = \mathcal{F}(\mathbf{x}) + c\mathbf{x}^\top(\mathbf{I}_N - \Pi)\mathbf{x},$$

we have

$$\text{SC-regret}_K(\mathcal{A}) = \sum_{k=1}^{K} V_k(\mathbf{x}_k) - \sum_{k=1}^{K} V_k(\mathbf{x}_*) \tag{5}$$

Likewise, we can obtain the compact form for the dynamic regret as follows.

$$\text{DC-regret}_K(\mathcal{A}) = \sum_{k=1}^{K} V_k(\mathbf{x}_k) - \sum_{k=1}^{K} V_k(\mathbf{x}_{*,k}) \tag{6}$$

The mixing matrix Π is assumed to be doubly stochastic, symmetric, with diagonal elements $\pi_{ii} > 0$ and off-diagonal elements $\pi_{ij} > 0$ if and only if i and j are neighbors in the communication network. Under these conditions, Π only has one eigenvalue exactly equal to 1 while the rest of eigenvalues are strictly less than 1. We use λ to represent the second-largest eigenvalue where $0 < \lambda < 1$. The spectral gap, $1-\lambda$, will play a critical role in relating network topology to regret bounds.

The composite regret proposed in this context is an extension of the distributed empirical risk minimization in the offline setting. Although the regular static and dynamic regrets have been extended straightforward from centralized to distributed online learning, such regrets do not imply good performance, particularly when the agent-wise loss functions are quite different. This scenario could lead to even divergence for some online algorithms. The composite regret enables either the regular static or dynamic regret to have an extra regularization term in the loss functions for coping with the *agent variations* corresponding to the spatial change, compared to the path variation of only one agent shown in Eq. 2, corresponding to the temporal change. Therefore, the distributed online learning with the composite regret can simultaneously keep track of both spatial and temporal (hence spatiotemporal) changes along the optimizer path, shown in the regret bound in the rest of the analysis. Another advantage of having the composite regret will be illustrated for the distributed online non-convex learning where the gradient oracle is not used.

III. Proposed Algorithm for Pseudo-convex Losses

We present the proposed algorithm for pseudo-convex losses in this section as well as the associated results. Before that, some necessary assumptions and preliminaries are given.

Assumption 1. *There exists a constant* $G > 0$ *such that* $\mathcal{F}$ *is Lipschitz continuous for all* $\mathbf{x}, \mathbf{y} \in \mathcal{X}^N$*, i.e.,*

$$|\mathcal{F}(\mathbf{y}) - \mathcal{F}(\mathbf{x})| \leq G\|\mathbf{y} - \mathbf{x}\|. \tag{7}$$

Assumption 2. *The set* $\mathcal{X}^N \subset \mathbb{R}^N$ *is a compact set and* $\forall \mathbf{x}, \mathbf{y} \in \mathcal{X}^N, \|\mathbf{x} - \mathbf{y}\|_\infty \leq D$.

Assumption 2 implies immediately $\|\mathbf{x} - \mathbf{y}\| \leq \sqrt{N}\|\mathbf{x} - \mathbf{y}\|_\infty \leq \sqrt{N}D$. In Zhang et al. [34], the authors imposed one additional assumption that each local loss function is smooth, while in this paper, we relax such an assumption as in Li et al. [17]. Moreover, in most previous works [17, 28], they had explicit assumptions for the boundedness of $\mathcal{F}$. Similarly, in this paper, we have that $|\mathcal{F}| < \infty$. Assumption 2 does not indicate whether the set is convex or not, which allows the analysis throughout the paper to use this assumption. However, when analyzing (strongly) convex and pseudo-convex losses, it defaults to a convex set. At the same time, it is not assumed to be so for non-convex analysis. To simplify the analysis in the next section, we also assume that $\mathcal{X}^N$ contains the origin, while in practice, this may not necessarily be true. This simplification allows us to initialize $\mathbf{x}$ at the origin.

A. CONGD for Pseudo-convex Losses

In literature, numerous existing works have focused on convex optimization, and non-convex problems in the distributed online setting have rarely been explored and investigated with any established methods. In this context, we propose the CONGD for studying the scenario where the loss functions satisfy the pseudo-convexity and show that a sublinear dynamic regret can be achieved with the properly set step size. The online normalized gradient descent

(ONGD) has been adopted in Gao et al. [11], while Nesterov [23] proposed the normalized gradient descent. The CONGD utilizes the first-order information $\nabla f_k^i(x_k^i)$ for each agent to compute the normalized vector $\frac{\nabla f_k^i(x_k^i)}{\|\nabla f_k^i(x_k^i)\|}$ as the search direction. The algorithmic framework is presented as follows. $\mathbf{P}_{\mathcal{X}^i}(y) = \arg\min_{x\in\mathcal{X}^i}\|y-x\|$ is the Euclidean projection.

Algorithm 1 Consensus-based Online Normalized Gradient Descent (CONGD)

1: **Input:** convex sets $\mathcal{X}^i$, K, $x_1^i = 0 \in \mathcal{X}^i$, step size $\{\alpha_k\}$, Π, $i \in \{1, ..., N\}$
2: **for** $k = 1$ to K **do**
3: **for** each agent i **do**
4: Play the strategy to form x_k^i and observe the loss $f_k^i(x_k^i)$
5: Calculate the feedback $\nabla f_k^i(x_k^i)$
6: **if** $\|\nabla f_k^i(x_k^i)\| > 0$ **then**
7: Update: $y_{k+1}^i = \sum_{i=1}^N \pi_{ij} x_k^j - \alpha_k \frac{\nabla f_k^i(x_k^i)}{\|\nabla f_k^i(x_k^i)\|}$
8: Project: $x_{k+1}^i = \mathbf{P}_{\mathcal{X}^i}(y_{k+1}^i)$
9: **else**
10: $x_{k+1}^i = x_k^i$
11: **end if**
12: **end for**
13: **end for**

We denote by $\|\nabla\mathcal{F}_k(\mathbf{x}_k)\| = \|[\nabla f_k^1(x_k^1), ..., f_k^N(x_k^N)]^\top\|$. For each agent, they play their own strategies and then observe the corresponding losses, subsequently followed by calculating the feedback. To update the strategy, if the norm of the feedback is not equal to 0, a consensus step is taken for an auxilibary variable y_k^i with the movement in the direction of negative normalized gradient. Then, a projection is taken for the updated auxiliary variable. Otherwise, the current strategy is taken as the next strategy. Algorithm 1 shows the agent-wise update for the implementation, while for analysis, we will use a compact form for the ease of the understanding. However, in Line 7, the compact form of the normalized vector is $\left[\frac{\nabla f_k^1(x_k^1)}{\|\nabla f_k^1(x_k^1)\|}, ..., \frac{\nabla f_k^N(x_k^N)}{\|\nabla f_k^N(x_k^N)\|}\right]^\top$, which is not exactly equivalent to $\frac{\nabla\mathcal{F}_k(\mathbf{x}_k)}{\|\nabla\mathcal{F}_k(\mathbf{x}_k)\|}$. Hence, in this context, a slight modification is applied for the analysis, i.e., $\left[\frac{\nabla f_k^1(x_k^1)}{\|\nabla\mathcal{F}_k(\mathbf{x}_k)\|}, ..., \frac{\nabla f_k^N(x_k^N)}{\|\nabla\mathcal{F}_k(\mathbf{x}_k)\|}\right]^\top$. Due to Assumption 1, such a replacement in the analysis will not affect the ultimate convergence rate, while it may lead to slight different constants in the error bound. Note that such a modification is *only* for the theoretical analysis as in practice, each agent cannot get access to other agents' local gradients.

In order to apply the composite regret to CONGD, such a form helps characterize the analysis for pseudo-convex losses. Recall $\mathbf{y}_{k+1} = \Pi\mathbf{x}_k - \alpha_k\frac{\nabla\mathcal{F}_k(\mathbf{x}_k)}{\|\nabla\mathcal{F}_k(\mathbf{x}_k)\|}$, which can be rewritten as

$$\mathbf{y}_{k+1} = \mathbf{x}_k - \alpha_k\left(\frac{\nabla\mathcal{F}_k(\mathbf{x}_k)}{\|\nabla\mathcal{F}_k(\mathbf{x}_k)\|} + \frac{1}{\alpha_k}(\mathbf{I}_N - \Pi)\mathbf{x}_k\right).$$

One can observe that the second term on the right-hand side of the last equation is close to a composite gradient, including the normalized gradient and the consensus term. For keeping consistency inside the composite gradient, we also derive the *normalized consensus* term, which will be found to be useful for the subsequent analysis. Thus, we have

$$\mathbf{y}_{k+1} = \mathbf{x}_k - \alpha_k\left(\frac{\nabla\mathcal{F}_k(\mathbf{x}_k)}{\|\nabla\mathcal{F}_k(\mathbf{x}_k)\|} + \frac{\|(\mathbf{I}_N - \Pi)\mathbf{x}_k\|}{\alpha_k}\frac{(\mathbf{I}_N - \Pi)\mathbf{x}_k}{\|(\mathbf{I}_N - \Pi)\mathbf{x}_k\|}\right),$$

which is

$$\mathbf{y}_{k+1} = \mathbf{x}_k - \alpha_k\left(\frac{\nabla\mathcal{F}_k(\mathbf{x}_k)}{\|\nabla\mathcal{F}_k(\mathbf{x}_k)\|} + \hat{\alpha}_k\frac{(\mathbf{I}_N - \Pi)\mathbf{x}_k}{\|(\mathbf{I}_N - \Pi)\mathbf{x}_k\|}\right), \quad (8)$$

with $\hat{\alpha}_k = \frac{\|(\mathbf{I}_N - \Pi)\mathbf{x}_k\|}{\alpha_k}$. When α_k is set constant for all k, $\hat{\alpha}_k$ can be bounded above by a constant. Such a property will help with the analysis when adopting the pseudo-convexity to the loss functions. However, we still have to define the constant c for the CONGD. Motivated by the distributed empirical risk minimization, $c = \frac{1}{2\alpha_k}, \forall k = 1, 2, ..., K$ is employed in the CONGD. Thus, the following theorem is shown to claim a sublinear non-stationary regret bound for CONGD.

Theorem 1. *Let all assumptions hold. CONGD with* DC-regret$_K$ *and the step size* $\alpha_k = \alpha = \sqrt{\frac{\sqrt{N}D(\sqrt{N}D+3P_K)}{K}}$ *guarantees the sublinear regret* $\mathcal{O}\left(\left(1 + \frac{1}{1-\lambda}\right)\sqrt{K(ND^2 + 3\sqrt{N}DP_K)}\right)$ *for all* $K \geq 1$ *when the loss functions are pseudo-convex.*

Theorem 1 implies that even if the loss function is pseudo-convex, which is weaker than convex, the regret minimization does not differ significantly. It is noted that we omit some constants in the expression of $\mathcal{O}(\cdot)$. The regret bound also implies the impact of network topology such that when the network is dense, which corresponds a smaller λ value, the regret bound approaches to $\mathcal{O}\left(2\sqrt{K(ND^2 + 3\sqrt{N}DP_K)}\right)$. However, a sparse network results in a larger regret bound.

IV. Proposed Algorithm for Non-convex Losses

As mentioned above, non-convex online learning remains an active research area in the algorithmic learning community. In the centralized setting, recent works [1, 29] have focused on the FTPL algorithm to leverage the perturbation for stabilizing the learning. However, in the distributed setting, no results have been reported, to the best of our knowledge. In this paper, we shed light on the availability of the regret bound on top of the existing works.

A. Proposed Algorithm: DINOCO

Some preliminaries are presented before presenting the algorithmic framework of DINOCO for the setting of distributed non-convex online learning. The perturbation is attained by sampling a random vector from an exponential distribution. Therefore, we denote by σ an N dimension random vector from the distribution $\exp(\eta)$, where $\eta > 0$ is the rate parameter

for an exponential distribution. Intuitively, η plays a similar role as the step size, α, and it will be defined as a function of K, which significantly affects the growth order of the regret. Also, in Assumption 1, we use l_2 norm to define the Lipschitz continuity for $\mathcal{F}$. While we adjust the assumption slightly to replace the l_2 norm with the l_1 norm. Such an adjustment helps characterize the subsequent analysis for the regret bound as the random perturbation vector is sampled coordinate-wise. We will see that in the regret bound, the dimension of such a random perturbation vector also has an impact. By re-defining a constant $\hat{G} > 0$, the conclusion from Assumption 1 becomes $|\mathcal{F}(\mathbf{y}) - \mathcal{F}(\mathbf{x})| \leq \hat{G}\|\mathbf{y} - \mathbf{x}\|_1$. We also correspondingly modify the norm of network loss in Eqs. 3 and 4, changing the l_2 norm to l_1 norm. Such a change does not hurt the concept of composite regret due to the property of the equivalence of norms. Also, this change will not affect the regret bound as the difference is only related to some constants that would be omitted when presenting the regret bound. To obtain the DINOCO, we adapt an offline optimization oracle with an approximate form [29] and define it formally as follows.

Definition 2. *For each agent i, a (ρ_i, β_i)-approximate optimization oracle is an oracle that takes as input a function $V^i : \mathcal{X}^i \to \mathbb{R}$ and a random perturbation σ^i sampled from an exponential distribution such that*

$$V^i(x_*) - \sigma^i x_* \leq \inf_{x \in \mathcal{X}^i} V^i(x) - \sigma^i x + (\rho_i + \beta_i |\sigma^i|), \quad (9)$$

where $x_* \in \mathcal{X}^i$, $\rho_i, \beta_i > 0$.

For convenience, we denote by $\mathcal{T}_{\rho_i,\beta_i}(V^i, \sigma^i)$ the (ρ, β)-approximate optimization oracle for agent i. Given access to $\mathcal{T}_{\rho_i,\beta_i}(V^i, \sigma^i)$, the DINOCO algorithm utilizes the following core prediction rule

$$x^i_{k+1} = \mathcal{T}_{\rho_i,\beta_i}\left(\sum_{l=1}^{k} f^i_l + r^i_l, \sigma^i_{k+1}\right) \quad (10)$$

where σ^i_{k+1} is a random perturbation at the time step $k+1$ sampled from $\exp(\eta)$. η will be specified when determining the regret bound. We now formally present the DINOCO in Algorithm 2.

Algorithm 2 DINOCO

Input: rate parameter of exponential distribution η, (ρ_i, β_i)-approximate optimization oracle $\mathcal{T}_{\rho_i,\beta_i}$, Π
for $k = 1$ to K **do**
  for each agent i **do**
    Agent i plays the strategy to form x^i_k and observe the losses $f^i_k(x^i_k)$ and $r^i_k(x^i_k) = \frac{1}{2\eta}\sum_{j \in Nb(i)} \pi_{ij}\|x^i_k - x^j_k\|^2$
    Generate random perturbation $\sigma^i_{k+1} \overset{i.i.d}{\sim} \exp(\eta)$
    Update: $x^i_{k+1} = \mathcal{T}_{\rho_i,\beta_i}\left(\sum_{l=1}^{k}(f^i_l + r^i_l), \sigma^i_{k+1}\right)$
  end for
end for

$f^i_l + r^i_l$ is replaced by V^i_l throughout the analysis when presenting the update law. It can be observed from Algorithm 2 that each σ^i_{k+1} is drawn i.i.d from the same exponential distribution. $V^i_k(x^i_k)$ includes the network loss r^i_k among diverse agents in the neighborhood of agent i, which signifies the communication among them for each iteration. For the update at the iteration k, $\mathcal{T}_{\rho_i,\beta_i}$ takes as input the cumulative loss up to k and the random perturbation σ^i_{k+1} and then predicts x^i_{k+1}. Intuitively, the next approximate best decision is attained in hindsight by perturbing the cumulative loss. Next, we will show how such a decision-making process benefits from the perturbation for stability.

B. Sublinear Regret for DINOCO

In this section, we analyze the regret bound for DINOCO and shed light on how the composite regret is leveraged for the setting of distributed online non-convex learning. To ease the complication of the analysis, we introduce the (ρ, β)-*aggregate approximate optimization oracle* for the network such that

$$V(\mathbf{x}_*) - \sigma^\top \mathbf{x}_* \leq \inf_{\mathbf{x} \in \mathcal{X}^N} V(\mathbf{x}) - \sigma^\top \mathbf{x} + (\rho + \beta\|\sigma\|_1), \quad (11)$$

where $\rho = \max\{\rho_1, \rho_2, ..., \rho_N\}$ and $\beta = \max\{\beta_1, \beta_2, ..., \beta_N\}$. Each coordinate of σ corresponds to the random perturbation of an agent, which is sample from the same exponential distribution in the i.i.d. manner. The aggregate in this context is due to the agents in the network. A key lemma for upper bounding the expected composite regret is first given.

Lemma 1. *Let all assumptions hold. The expected composite regret of the DINOCO is upper bounded as*

$$\mathbb{E}\left[\sum_{k=1}^{K} V_k(\mathbf{x}_k) - \sum_{k=1}^{K} V_k(\mathbf{x}_*)\right] \leq L\sum_{k=1}^{K}\mathbb{E}[\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_1] + \frac{N(\beta K + \sqrt{N}D)}{\eta} + \rho K, \text{ where } \mathbf{x}_* = \inf_{\mathbf{x} \in \mathcal{X}^N}\sum_{k=1}^{K} V_k(\mathbf{x}). \quad (12)$$

As Lemma 1 shows, the upper bound is determined by the stability of decisions, the approximation error by the aggregate approximate optimization oracle, and the diameter of the decision set. Another suggestion from Lemma 1 is that the number of agents can impact the upper bound, which is evident in the second term on the right-hand side of Eq. 12. However, the impact could be reduced by choosing appropriately η and β. We defer the analysis after presenting the main theorem. This term also shows the tradeoff between the stability and the quality of predictions. Although more randomness can better stabilize the decision-making process, the prediction may become worse to cause the worse regret when η decreases. Additionally, if substituting $L \approx \hat{G}(1 + \frac{\gamma}{1-\lambda}), \gamma > 0$ into the upper bound, we can obtain the quantitative relationship between the regret bound and the network topology, which provides us a way to investigate regrets with different topologies. To get the main theorem, we have to show the first term on the right-hand side in Eq. 12 to be bounded above. Due to the limit of the space, we omit a formal statement for the bound while the

detailed analysis will be presented in the full version of this paper. In this context, we have adopted SC-regret$_K$ to show the regret bound, and the application of DC-regret$_K$ for non-convex losses is left as future work. We now state the main theorem for DINOCO.

Theorem 2. *Suppose that the offline optimization oracle used by DINOCO is a (ρ, β)-aggregate approximate. Also, the random perturbation σ is assumed to be sampled from an exponential distribution* $\exp(\eta)$ *with a constant rate parameter η. Then DINOCO can achieve the expected regret bound where $\mathbf{x}_* = \inf_{\mathbf{x}\in\mathcal{X}^N} \sum_{k=1}^K V_k(\mathbf{x})$:*

$$\mathbb{E}\left[\sum_{k=1}^K V_k(\mathbf{x}_k) - \sum_{k=1}^K V_k(\mathbf{x}_*)\right] \leq \mathcal{O}\left(K\eta N^{\frac{5}{2}} D\hat{G}^2\left(1+\frac{\gamma}{1-\lambda}\right)^2 + \frac{N(\beta K + \sqrt{N}D)}{\eta} + \rho K + K\beta N\hat{G}\left(1+\frac{\gamma}{1-\lambda}\right)\right), \tag{13}$$

Due to the limit of space, we skip the detailed proof in the paper. We omit some constants in the regret bound for the expected composite regret in Theorem 2. Unlike the previous analysis given for pseudo-convex losses, the regret bound for non-convex losses is more sophisticated. Its growth order relies on some parameters, including the rate parameter η and the parameters associated with the offline optimization oracle. Critical constants such as the number of agents and the spectral gap of the network topology also affect the regret bound. The regret bound increases as the number of agents N increases. When N has a similar value as the number of iterations K, the regret bound becomes trivial, but in practice, N remains finite and a smaller number compared to K. We now look into the impact of topology. With a dense topology, the spectral gap $1-\lambda$ is relatively larger, leading to a smaller regret bound, while a sparse topology yields a larger regret bound. Such a property has been well studied in distributed offline non-convex optimization and still holds for the distributed online non-convex optimization. Next, we present a corollary to show the sublinear regret bound by explicitly setting η, ρ, and β.

Corollary 1. *Suppose that the offline optimization oracle used by DINOCO is (ρ, β)-aggregate approximate, where $\rho = \mathcal{O}(\frac{1}{\sqrt{K}})$ and $\beta = \mathcal{O}(\frac{1}{K})$. Also, the random perturbation σ is assumed to be sampled from an exponential distribution* $\exp(\eta)$ *with a constant rate parameter $\eta = \mathcal{O}(\frac{1}{\sqrt{K}})$. Then DINOCO can achieve the expected regret bound*

$$\mathbb{E}\left[\sum_{k=1}^K V_k(\mathbf{x}_k) - \sum_{k=1}^K V_k(\mathbf{x}_*)\right] \leq \mathcal{O}\left(\sqrt{K}N^{\frac{5}{2}}\left(1+\frac{\gamma}{1-\lambda}\right)^2 + \sqrt{K}N^{\frac{3}{2}} + \sqrt{K}\right), \text{ where } \mathbf{x}_* = \inf_{\mathbf{x}\in\mathcal{X}^N} \sum_{k=1}^K V_k(\mathbf{x}). \tag{14}$$

The proof of Corollary 1 is an immediate consequence of Theorem 2 when substituting η, ρ, and β into the regret bound. It clearly shows that the regret bound for the DINOCO with the expected composite regret is sublinear and closely related to the number of agents and the topology of the network. When $N = 1$ such that the term $\frac{\gamma}{1-\lambda}$ disappears and the setting is centralized, the regret bound maintains $\mathcal{O}(\sqrt{K})$. Hence we can recover the regret bound shown in Suggala and Netrapalli [29]. When N is large, one can observe that the *first* term on the right-hand side of Eq. 14 dominates the regret bound. By specifically defining η and β as a function of N, the order of N in the regret bound can be reduced. Let $\eta = \mathcal{O}(\frac{1}{\sqrt{NK}})$ and $\beta = \mathcal{O}(\frac{1}{NK})$. We then have

$$\mathbb{E}\left[\sum_{k=1}^K V_k(\mathbf{x}_k) - \sum_{k=1}^K V_k(\mathbf{x}_*)\right] \leq \mathcal{O}\left(\sqrt{K}N^2\left(1+\frac{\gamma}{1-\lambda}\right)^2 + \sqrt{K}N^2 + \sqrt{K}\right), \tag{15}$$

which suggests that the regret bound is approximately reduced by an order of $\sqrt{N}$. In summary, though the extension of online non-convex optimization algorithm from centralized to distributed setting is straightforward, such an extension is still critical as rare results have been reported and the analysis involving diverse topologies is non-trivial.

V. Conclusions

This paper develops a novel regret that extends the regular static and dynamic regrets to account for the network loss in distributed online learning. With the composite regret, we propose algorithms for different classes of losses. For pseudo-convex losses, the proposed algorithm allows achieving the sublinear regret, which is the first time for the distributed setting. For general non-convex losses, based on the recent development, we have found that no deterministic algorithm can achieve sublinear regret. By introducing an offline optimization oracle, the proposed DINOCO using the composite regret can achieve the best sublinear regret, which sheds light on the regret bound in distributed online non-convex learning.

References

[1] Naman Agarwal, Alon Gonen, and Elad Hazan. Learning in non-convex games with an optimization oracle. *arXiv preprint arXiv:1810.07362*, 2018.

[2] Mohammad Akbari, Bahman Gharesifard, and Tamás Linder. Distributed online convex optimization on time-varying directed graphs. *IEEE Transactions on Control of Network Systems*, 4(3):417–428, 2015.

[3] Sergul Aydore, Lee Dicker, and Dean Foster. A local regret in nonconvex online learning. *arXiv preprint arXiv:1811.05095*, 2018.

[4] Nicola Bastianello, Amirhossein Ajalloeian, and Emiliano Dall'Anese. Distributed and inexact proximal gradient method for online convex optimization. *arXiv preprint arXiv:2001.00870*, 2020.

[5] Albert S Berahas, Raghu Bollapragada, Nitish Shirish Keskar, and Ermin Wei. Balancing communication and

computation in distributed optimization. *IEEE Transactions on Automatic Control*, 64(8):3141–3155, 2018.

[6] Omar Besbes, Yonatan Gur, and Assaf Zeevi. Non-stationary stochastic optimization. *Operations research*, 63(5):1227–1244, 2015.

[7] Xuanyu Cao and Tamer Başar. Decentralized online convex optimization based on signs of relative states. *Automatica*, 129:109676, 2021.

[8] Xuanyu Cao and Tamer Basar. Decentralized online convex optimization with feedback delays. *IEEE Transactions on Automatic Control*, 2021.

[9] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Conference on Learning Theory*, pages 6–1, 2012.

[10] Nima Eshraghi and Ben Liang. Distributed online optimization over a heterogeneous network with any-batch mirror descent. In *International Conference on Machine Learning*, pages 2933–2942. PMLR, 2020.

[11] Xiand Gao, Xiaobo Li, and Shuzhong Zhang. Online learning with non-convex losses and non-stationary regret. In *International Conference on Artificial Intelligence and Statistics*, pages 235–243, 2018.

[12] Elad Hazan, Karan Singh, and Cyril Zhang. Efficient regret minimization in non-convex games. *arXiv preprint arXiv:1708.00075*, 2017.

[13] Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2 (3-4):157–325, 2016.

[14] Saghar Hosseini, Airlie Chapman, and Mehran Mesbahi. Online distributed convex optimization on dynamic networks. *IEEE Transactions on Automatic Control*, 61(11): 3545–3550, 2016.

[15] Zhanhong Jiang, Aditya Balu, Chinmay Hegde, and Soumik Sarkar. Collaborative deep learning in fixed topology networks. In *Advances in Neural Information Processing Systems*, pages 5904–5914, 2017.

[16] Soomin Lee and Michael M Zavlanos. On the sublinear regret of distributed primal-dual algorithms for online constrained optimization. *arXiv preprint arXiv:1705.11128*, 2017.

[17] Xiuxian Li, Xinlei Yi, and Lihua Xie. Distributed online optimization for multi-agent networks with coupled inequality constraints. *arXiv preprint arXiv:1805.05573*, 2018.

[18] Kaihong Lu and Long Wang. Online distributed optimization with nonconvex objective functions: Sublinearity of first-order optimality condition-based regret. *IEEE Transactions on Automatic Control*, 2021.

[19] Min Meng, Xiuxian Li, Yiguang Hong, Jie Chen, and Long Wang. Decentralized online learning for noncooperative games in dynamic environments. *arXiv preprint arXiv:2105.06200*, 2021.

[20] Min Meng, Xiuxian Li, and Jie Chen. Decentralized nash equilibria learning for online game with bandit feedback. *arXiv preprint arXiv:2204.09467*, 2022.

[21] Parvin Nazari, Davoud Ataee Tarzanagh, and George Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *arXiv preprint arXiv:1901.09109*, 2019.

[22] Angelia Nedić, Soomin Lee, and Maxim Raginsky. Decentralized online optimization with global objectives and local communication. In *2015 American Control Conference (ACC)*, pages 4497–4503. IEEE, 2015.

[23] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[24] Santiago Paternain, Soomin Lee, Michael M Zavlanos, and Alejandro Ribeiro. Distributed constrained online learning. *arXiv preprint arXiv:1903.06310*, 2019.

[25] Shi Pu and Angelia Nedić. A distributed stochastic gradient tracking method. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 963–968. IEEE, 2018.

[26] Shahin Shahrampour and Ali Jadbabaie. Distributed online optimization in dynamic environments using mirror descent. *IEEE Transactions on Automatic Control*, 63 (3):714–725, 2017.

[27] Shai Shalev-Shwartz et al. Online learning and online convex optimization. *Foundations and Trends® in Machine Learning*, 4(2):107–194, 2012.

[28] Pranay Sharma, Prashant Khanduri, Lixin Shen, Donald J Bucci Jr, and Pramod K Varshney. On distributed online convex optimization with sublinear dynamic regret and fit. *arXiv preprint arXiv:2001.03166*, 2020.

[29] Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. *arXiv preprint arXiv:1903.08110*, 2019.

[30] Xinlei Yi, Xiuxian Li, Lihua Xie, and Karl H Johansson. Distributed online convex optimization with time-varying coupled inequality constraints. *IEEE Transactions on Signal Processing*, 68:731–746, 2020.

[31] Deming Yuan, Alexandre Proutiere, and Guodong Shi. Distributed online optimization with long-term constraints. *arXiv preprint arXiv:1912.09705*, 2019.

[32] Deming Yuan, Yiguang Hong, Daniel WC Ho, and Shengyuan Xu. Distributed mirror descent for online composite optimization. *IEEE Transactions on Automatic Control*, 2020.

[33] Jinshan Zeng and Wotao Yin. On nonconvex decentralized gradient descent. *IEEE Transactions on signal processing*, 66(11):2834–2848, 2018.

[34] Yan Zhang, Robert J Ravier, Michael M Zavlanos, and Vahid Tarokh. A distributed online convex optimization algorithm with improved dynamic regret. *arXiv preprint arXiv:1911.05050*, 2019.

[35] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936, 2003.