2022 IEEE 61st Conference on Decision and Control (CDC)
December 6-9, 2022. Cancún, Mexico

Safe Dual Gradient Method for Network Utility Maximization Problems

Berkay Turan Mahnoosh Alizadeh

2022 IEEE 61st Conference on Decision and Control (CDC) | 978-1-6654-6761-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CDC51059.2022.9992972

***Abstract*— In this paper, we introduce a novel first-order dual gradient algorithm for solving network utility maximization problems that arise in resource allocation schemes over networks with safety-critical constraints. Inspired by applications where customers' demand can only be affected through posted prices and real-time two-way communication with customers is not available, we require an algorithm to generate *safe prices*. This means that at no iteration should the realized demand in response to the posted prices violate the safety constraints of the network. Thus, in contrast to existing first-order methods, our algorithm, called the safe dual gradient method (SDGM), is guaranteed to produce feasible primal iterates at all iterations. We ensure primal feasibility by 1) adding a diminishing safety margin to the constraints, and 2) using a sign-based dual update method with different step sizes for plus and minus directions. In addition, we prove that the primal iterates produced by the SDGM achieve a sublinear static regret of $\mathcal{O}(\sqrt{T})$.**

I. INTRODUCTION

Many applications falling within the scope of resource allocation over networks, e.g., power distribution systems [1], congestion control in data networks [2], and wireless cellular networks [3], deal with a multi-agent optimization problem that falls under the general umbrella of *network utility maximization* (NUM) problems. The shared goal in these problems is to allocate the resources to the users subject to a set of coupling constraints such that the total utility of the users is maximized.

In NUM problems, the user-specific utility functions are assumed to be private to the users and therefore a centralized solution is not possible. Accordingly, distributed optimization methods have become suitable tools thanks to the separable structure of NUM problems [4], [5]. The idea is to decompose the main problem into sub-problems that can be solved by the individual users and use the solutions of the sub-problems to solve the main problem [6], [7], which has been widely advocated for use in different applications, e.g., [2], [8]. Among the two main types of decomposition methods, primal decomposition methods correspond to a direct allocation of the resources by a central coordinator and solve the primal problem, whereas dual decomposition methods based on the Lagrangian dual problem [9] correspond to resource allocation via pricing and solve the dual problem [4]. Due to the structure of NUM problems, the latter approach has been widely adopted in the literature [4], [10], [11]. Additionally, it gives users the freedom of determining their own resource consumption based on prices.

The (dual) subgradient method is the most basic approach for solving the Lagrangian dual problem, whose convergence properties under various step-size rules have been well established [6], [9], [12]–[14]. Besides the performance of the subgradient method in the dual space, scholars have also focused on characterizing the suboptimality, the convergence rate, and the infeasibility amount of the primal iterates[1]. Under no strong concavity or smoothness assumptions on the objective function, the average of the primal sequence produced by the dual subgradient method is shown to achieve an $\mathcal{O}(1/\sqrt{t})$ primal suboptimality and infeasibility [10], where t is the iteration number. When the objective function is strongly concave, the dual problem is smooth and therefore it is possible to achieve a rate of $\mathcal{O}(1/t)$ for the primal suboptimality, convergence rate, and infeasibility of the last iterate [11] using accelerated methods (e.g., [15]). Using primal averaging schemes, rates of $\mathcal{O}(1/t^2)$ for the primal suboptimality and infeasibility can be obtained [16]–[18]. Under an additional smoothness assumption on the objective function, global linear convergence rates are achieved for linearly constrained [19] and unconstrained [20], [21] convex optimization problems over networks. It is worthwhile to highlight that none of these works guarantee feasible iterates throughout the optimization process, but only provide bounds on the infeasibility amount of the primal iterates. Therefore, solutions are only realizable after convergence to a near-feasible point for resource allocation systems with safety-critical constraints (even at a near-feasible point, the amount of infeasibility still needs to be accounted for).

This paper is motivated by network resource allocation applications, where 1) users determine their own resource consumption in response to the prices and the realized consumption is only observed afterward, and 2) the system has safety-critical hard constraints that should not be violated by the users' resource consumption at any time. For instance, in price-based demand response, users determine their own electricity consumption in response to prices, where the prices should be set such that the realized demand does not violate the capacity constraints of the electric grid [22]. This is to ensure the safe operation of the system because violating the capacity constraints could cause physical damage to the grid. This implies that the resource consumption of the users (i.e., primal variables) in response to the prices (i.e., the dual variables) should always satisfy the constraints of the system (i.e., be feasible). This allows users to have realizable demand all the time and gain some utility throughout the optimization process. Unlike existing methods which pro-

B. Turan and M. Alizadeh are with Dept. of ECE, UCSB, Santa Barbara, CA, USA. This work is supported by NSF grant #1847096. E-mails: bturan@ucsb.edu, alizadeh@ucsb.edu

[1]The suboptimality is measured by the difference in the optimal objective value and the objective value of the iterates. The convergence rate is measured by the distance of the iterates to the optimal solution. The infeasibility amount is measured by the norm of the constraint violation.

duce prices that are only implementable (i.e., safe) after convergence to a near-optimal point, our framework does not require convergence before prices can be posted. Hence, it removes the need for complex negotiations with users over what their potential demand would be in response to different prices in order to converge to the optimal price.

To this end, in this paper, we develop a distributed algorithm for NUM based on the dual decomposition scheme, called the safe dual gradient method (SDGM), that produces feasible primal iterates at all iterations. Our method does not use any second-order information (except for a lower bound on the strong concavity constant) and the dual updates solely rely on the constraints evaluated at the current feasible primal iterate. Our contributions are as follows:

- We introduce a novel algorithm, the SDGM, for solving NUM problems in a distributed fashion. We characterize a principled way to choose algorithm parameters to guarantee feasible primal iterates at all iterations.
- We prove that the static regret incurred by the feasible primal iterates produced by the SDGM, i.e., the cumulative gap between the optimal objective value and the objective function evaluated at the primal iterates, up to time T is bounded by $\mathcal{O}(\sqrt{T})$.
- We numerically evaluate our algorithm to support our theoretical findings and compare its performance to existing first-order distributed methods for NUM problems.

The primal feasibility and the regret guarantees of the SDGM result from a combination of two ingredients: 1) by adding a safety margin to the constraints, we perturb the dual gradients and increase the dual variables before the constraint is violated (in contrast, the basic dual subgradient method [4], [11] only increases a dual variable after the corresponding constraint is violated), and 2) we only use the sign of the perturbed gradient and utilize different step-sizes for plus and minus directions. The latter allows us to have global control over the changes in the dual variables independent of the values of the constraints. This is done to ensure a sufficient amount of increase in a dual variable whenever the corresponding constraint is close to being tight, which is crucial for the feasibility of the primal iterates.

Related work: Besides dual (sub)gradient methods for solving the NUM problem, our work is closely related to interior point methods and safe learning/optimization literature.

1) *Interior point methods:* Interior point methods solve an inequality constrained problem by converting it into a sequence of equality constrained problems using barrier functions, and implementing Newton's method to solve the sequence of problems [23]. They produce feasible iterates, however, Newton's method is a second-order method that requires the Hessian, which is generally not available in the applications of interest to this paper, such as demand response with no two-way communications. Accordingly, in [24] a feasible interior point method is introduced by approximating the Hessian using the first-order information. However, the algorithm defines a primal update rule, whereas in practical applications we would like to allow users to freely determine their resource consumption in response to the posted prices. Similarly in [25], the authors propose a distributed Newton method for NUM problems, where the Hessian is approximated by second-order information exchange between the users and the primal updates follow a Newton direction update rule. Closest to the setup we study in this paper would be [26], [27]. In [26], a Newton-like dual update is proposed by approximating the Hessian using only the first-order information. However, only asymptotic convergence of the algorithm is proven and the feasibility of the primal iterates is not guaranteed. On the other hand, the authors of [27] propose an interior point method using Lagrangian dual decomposition with theoretical guarantees, however, it requires the exact Hessian for the dual update.

2) *Safe optimization/learning:* This line of work aims to develop safe algorithms that produce feasible iterates/actions for the optimization [28], [29]/bandit [30] frameworks, respectively. In [28] and [30], the feasible set is unknown and the approach is to conservatively estimate the feasible set and pick the primal iterates/actions accordingly. In [29], the gradient flow that directly optimizes the primal variables is augmented with a control barrier function to maintain safety. In contrast, although the feasible set is known, the dual decomposition architecture does not allow for direct control of the primal iterates, which differentiates our work from this literature.

Paper Organization: The remainder of the paper is organized as follows. In Section II, we formalize the problem setup. In Section III, we describe the SDGM (Algorithm 2) and in Section IV, we prove its feasibility and regret guarantees. In Section V, we provide a numerical study demonstrating the efficacy of the SDGM.

Notation and Basic Definitions: We denote the set of real numbers by $\mathbb{R}$ and the set of non-negative real numbers by $\mathbb{R}_+$. Unless otherwise specified, $\|\cdot\|$ denotes the standard Euclidean norm and $\|\cdot\|_p$ denotes the p-norm. Given a positive integer $n > 0$, $[n]$ denotes the set of integers $\{1, 2, \ldots, n\}$. Given a vector $x \in \mathbb{R}^n$, $x_i \in \mathbb{R}$ or $[x]_i \in \mathbb{R}$ denotes the i'th entry of x. Given a function $f : \mathbb{R} \to \mathbb{R}$, f' denotes the first derivative of f. For a vector $x \in \mathbb{R}^m$, $[x]_+$ is the component-wise maximum of the vector x and the zero vector. Given two vectors $x, y \in \mathbb{R}^m$, $x \leq y$ implies element-wise inequality. The vector $e_m \in \mathbb{R}^m$ denotes the m dimensional vector with all elements equal to 1.

Definition 1. *A differentiable function $f(\cdot)$ is said to be **μ-strongly concave** over the domain $\mathcal{X}$ if there exists $\mu > 0$ such that*

$$\langle \nabla f(x_2) - \nabla f(x_1), x_1 - x_2 \rangle \geq \mu \|x_1 - x_2\|^2 \qquad (1)$$

holds for all $x_1, x_2 \in \mathcal{X}$.

Definition 2. *A differentiable function $f(\cdot)$ is said to be **L-smooth** over the domain $\mathcal{X}$ if there exists $L > 0$ such that*

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leq L \|x_1 - x_2\| \qquad (2)$$

holds for all $x_1, x_2 \in \mathcal{X}$.

Algorithm 1 Dual Subgradient Method [4]

Input: Initialize $\lambda^1 \geq 0$, step size γ

1: **for** $t = 1, 2, \ldots$ **do**

2: Each user $i \in [n]$ receives $p_i^t := [A^T \lambda^t]_i$ and solves

$$x_i^t = \arg\max_{x_i \in \mathcal{X}_i} f_i(x_i) - p_i^t x_i \tag{4}$$

3: The dual vector λ^t is updated as:

$$\lambda^{t+1} = \max\{0, \lambda^t + \gamma(Ax^t - c)\} \tag{5}$$

4: **end for**

II. PROBLEM SETUP

We study the standard NUM problem [2], where the goal is to allocate resources to n users subject to a set of linear coupling constraints such that the total utility of the users is maximized. It can be formulated as the following optimization problem:

$$\max_{x \in \mathcal{X} \subset \mathbb{R}^n} \; f(x) = \sum_{i=1}^{n} f_i(x_i) \tag{3a}$$

$$\text{s.t. } Ax \leq c, \tag{3b}$$

where $f_i(\cdot)$ is the strictly increasing and concave utility function of user i, $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \cdots \times \mathcal{X}_n$ with $\mathcal{X}_i = [\underline{x}_i, \overline{x}_i]$ (where $\underline{x}_i \geq 0$, and $\overline{x}_i = \infty$ is allowed), $c \in \mathbb{R}_+^m$, and $A \in \{0,1\}^{m \times n}$ is an $m \times n$ matrix[2].

Due to the constraint (3b), the feasible region of (3) $\overline{\mathcal{X}} = \mathcal{X} \cap \{x | Ax \leq c\}$ is compact, where $\overline{\mathcal{X}} \subseteq \prod_{i \in [n]} [\underline{x}_i, \max_{j \in [m]} c_j]$. Over this region $\overline{\mathcal{X}}$, we make the following assumption on the utility functions:

Assumption 1. *For all $i \in [n]$, the utility function $f_i(\cdot)$ is μ_i-strongly concave over $\overline{\mathcal{X}}$.*

Standard utility functions considered in the literature such as the α-fair utility functions (see [31]) satisfy the strong concavity assumption over the compact interval $\overline{\mathcal{X}}$. Note that the objective function (3a) is strongly concave with coefficient $\mu = \min_{i \in [n]} \mu_i$. Accordingly, the convex optimization problem (3), whenever feasible, has a unique solution denoted by $x^\star$ and an optimal objective value denoted by $f^\star$.

Since $f_i(\cdot)$ are private to the users, (3) can not be solved centrally. Therefore, dual decomposition methods have been proposed in order to decompose (3) into subproblems that can be solved in a distributed fashion [4]. In order to decompose (3), we let $\lambda \in \mathbb{R}_+^m$ be the dual vector and form the Lagrangian:

$$q(\lambda) = \max_{x \in \mathcal{X}} \; \sum_{i=1}^{n} f_i(x_i) - \lambda^T (Ax - c) \tag{6}$$

The Lagrangian formulation allows us to separate (3) into two levels of optimization. At the lower level, each user $i \in [n]$ solves the following subproblem using their own utility function and the pricing signal $p_i := [\lambda^T A]_i$:

$$q_i(\lambda) = \max_{x_i \in \mathcal{X}_i} f_i(x_i) - p_i x_i. \tag{7}$$

At the higher level, the master problem determines the dual variable by solving the dual problem:

$$\min_{\lambda \geq 0} \; q(\lambda) = \sum_{i=1}^{n} q_i(\lambda) + \lambda^T c. \tag{8}$$

We note that this approach solves the dual problem instead of (3). However, the optimal solutions to both problems coincide under strong duality, which is satisfied under the following assumption (Slater's condition):

Assumption 2. *There exists a vector $\tilde{x}$ in the relative interior of $\mathcal{X}$ such that $A\tilde{x} < c$.*

It is well-known (see [32]) that, under Assumption 2, strong duality holds for problem (3).

In this work, we measure the performance of an algorithm by the suboptimality, convergence rate, and infeasibility of the primal iterates it produces, which are defined as follows:

Definition 3. *Let $\{x^t\}$, $t \geq 1$, be a sequence of primal iterates produced by an algorithm. For an iterate x^t, we define the suboptimality as $f^\star - f(x^t)$ and the infeasibility as $\|[Ax^t - c]_+\|$. The sequence is said to have a convergence rate of $r(t)$, if $\|x^t - x^\star\| \leq r(t)$ for all $t \geq 1$ and $\lim_{t \to \infty} r(t) = 0$.*

Since the dual problem is convex, one approach for solving the dual problem (8) (and thus also (3)) is to employ the projected subgradient method with constant step-size γ outlined in Algorithm 1 [4], [11]. It has been shown that without a strong concavity assumption on $f(\cdot)$, after t iterations the objective function value evaluated at $\overline{x} = (1/t)\sum_{i=1}^{t} x^i$ achieves $f^\star - f(\overline{x}) \leq \mathcal{O}(\gamma + 1/(t\gamma))$ [10]. Additionally, it has been shown that the primal infeasibility $\|[A\overline{x} - c]_+\|$ is bounded by $\mathcal{O}(1/(t\gamma))$ [10]. These results however do not consider strongly concave objective functions and, thus, the results there remain within the domain of non-smooth convex optimization (a rate of $\mathcal{O}(1/\sqrt{t})$ is achieved with $\gamma = \mathcal{O}(1/\sqrt{t})$). When the objective function is strongly concave, the resulting dual objective is smooth, and therefore it is possible to achieve primal suboptimality, convergence rate, and infeasibility of $\mathcal{O}(1/t)$ for the last iterate [11] (or primal suboptimality and infeasibility of $\mathcal{O}(1/t^2)$ for the average iterate [16]–[18]). When the objective function is both smooth and strongly concave, global linear convergence rates can be achieved [19].

Although existing distributed first-order methods establish bounds on the infeasibility of the primal iterates, there is no obvious way to modify the algorithms such that the primal iterates are always feasible. In the next section, we propose a first-order algorithm based on the dual decomposition scheme that produces feasible primal solutions at all iterations. In addition, the algorithm should produce primal

[2]In this work, we study the setting where $A \in \{0,1\}^{m \times n}$ and leave the case where $A \in \mathbb{R}^{m \times n}$ as a future direction.

Algorithm 2 Safe Dual Gradient Method

Input: Initialize $\lambda_j^1 = \overline{\lambda}$ for all $j \in [m]$, step sizes γ_-^t and γ_+^t, safety margin vector $\Delta^t \in \mathbb{R}_+^m$

1: **for** $t = 1, 2, \dots$ **do**

2: Each user $i \in [n]$ receives $p_i^t := [A^T \lambda^t]_i$ and solves

$$x_i^t = \arg\max_{x_i \in \mathcal{X}_i} f_i(x_i) - p_i^t x_i \tag{10}$$

3: The dual vector λ^t is updated as:

$$\lambda_j^{t+1} = \max\{0, \lambda_j^t - \gamma_-^t\}, \text{ if } [Ax^t + \Delta^t - c]_j < 0 \tag{11}$$

$$\lambda_j^{t+1} = \min\{\overline{\lambda}, \lambda_j^t + \gamma_+^t\}, \text{ if } [Ax^t + \Delta^t - c]_j \geq 0 \tag{12}$$

4: **end for**

iterates that result in a sublinear static regret, which is measured by

$$R(T) = \sum_{t=1}^{T} f^\star - f(x^t). \tag{9}$$

We note that the above definition of regret corresponds to the cumulative sum of suboptimalities of the primal iterates. When the primal iterates are feasible, the solutions are implementable, and therefore regret is a well-defined measure. On the other hand, although the above sum is computable for many of the existing works (e.g., [11], [16]), regret is not a well-defined metric since the primal iterates are not necessarily feasible and therefore realizable.

III. SAFE DUAL GRADIENT METHOD

In this section, we describe the dual update method we propose that produces feasible primal iterates satisfying a sublinear regret. The algorithm, called the safe dual gradient method (SDGM), is outlined in Algorithm 2. At the heart of the algorithm lie two key ideas:

1) The classical subgradient method only increases the dual variables after a constraint has been violated, which results in an infeasible primal solution. We add a safety margin Δ^t to the constraints as

$$Ax^t - c \to Ax^t - c + \Delta^t$$

so that the dual variable λ_j^t increases when the constraint is Δ_j^t close to being tight.

2) In the SDGM, the amounts of the dual updates (11)-(12) are independent of the values of the modified constraints (i.e., the perturbed gradient), but only dependent on their signs. This is to ensure that when the j'th constraint is Δ_j^t close to being tight, we increase λ_j^t sufficiently by an amount of γ_+^t while controlling the reduction of the other dual variables by γ_-^t so that the constraint is not violated at iteration $t+1$. If we were to use the actual values of the constraints, then for a constraint j that is Δ_j^t close to being tight, we could only ensure less than $\mathcal{O}(\Delta_j^t)$ (potentially very close to 0) increase in λ_j^t. Combined with a reduction in another constraint k, which can be as big as $\mathcal{O}(c_k)$, this might result in a large increase in x_i^t for which $A_{ji} = A_{ki} = 1$ (since $p_i^t - p_i^{t+1} = [A^T(\lambda^t - \lambda^{t+1})]_i$ can be big). A large increase in x_i^t could cause the constraint j to be violated at the next iteration. On the other hand, by using a normalized update rule we ensure that x_i^t does not increase at the next iteration.

We note that Algorithm 2 is similar to sign gradient methods [33], where the plus and the minus directions have different step sizes. Although convergence guarantees of sign-based gradient methods have been established for unconstrained non-convex optimization [34], we are not aware of explicit non-asymptotic converge rates for convex optimization with inequality constraints (even using the same step sizes for plus and minus updates).

It is necessary that the initial dual variables produce feasible primal solutions. Since this has to hold before getting any feedback from the users, we make the following assumption:

Assumption 3. *For all constraints $j \in [m]$, there exists a uniform bound $\overline{\lambda}$ such that if $\lambda_j^t = \overline{\lambda}$ then $[Ax^t - c]_j \leq 0$.*

Assumption 3 is not too restrictive and is satisfied in practice. For instance, if $f_i'(\cdot)$ is bounded by M in $\overline{\mathcal{X}}$ for all $i \in [n]$, then $\overline{\lambda} = M$ satisfies the assumption.

In the next section, we characterize a principled way to choose parameters Δ^t, γ_-^t, and γ_+^t in order to produce feasible primal iterates. Additionally, we prove that the regret incurred by the iterates produced by Algorithm 2 is $\mathcal{O}(\sqrt{T})$.

IV. FEASIBILITY AND REGRET ANALYSIS

We will first characterize the choice of algorithm parameters that guarantee primal feasibility at all iterations and then prove the regret of Algorithm 2 under this choice of parameters.

A. Feasibility Analysis

The following proposition characterizes a principled way to choose the parameters Δ^t and γ_+^t with respect to γ_-^t that ensures feasible primal iterates:

Proposition 1. *Let $\Delta_j^t = \frac{[AA^T e_m]_j}{\mu} \gamma_-^t$ for all $j \in [m]$ and $\gamma_+^t = (m-1)\gamma_-^t$. Then for all $t \geq 1$, the iterates x^t produced by Algorithm 2 are feasible, i.e.,*

$$Ax^t - c \leq 0, \ \forall t \geq 1. \tag{13}$$

Proof: We prove Proposition 1 by induction. Suppose that $Ax^t - c \leq 0$ holds at time t. We consider the following two cases:

1) Pick a constraint j for which $[Ax^t + \Delta^t - c]_j < 0$. If for all users, $x_i^{t+1} \leq x_i^t$, then $[Ax^{t+1} - c]_j \leq 0$ holds trivially. If for a user i, if $p_i^{t+1} > f_i'(\underline{x}_i)$, then $x_i^{t+1} = \underline{x}_i$ and $x_i^{t+1} \leq x_i^t$ holds. Furthermore, if $p_i^t < f_i'(\overline{x}_i)$, then $x_i^t = \overline{x}_i$ and $x_i^{t+1} \leq x_i^t$ holds. Therefore, in order to have $x_i^{t+1} > x_i^t$, it is necessary to have $f_i'(x_i^{t+1}) \geq p_i^{t+1}$, $f_i'(x_i^t) \leq p_i^t$, and

$p_i^{t+1} \leq p_i^t$. Using strong concavity, we have that:

$$x_i^{t+1} \leq x_i^t + \frac{f_i'(x_i^t) - f_i'(x_i^{t+1})}{\mu} \leq \frac{p_i^t - p_i^{t+1}}{\mu} \tag{14}$$

$$= \frac{[A^T(\lambda^t - \lambda^{t+1})]_i}{\mu}. \tag{15}$$

From the update rule, we know that for all $j \in [m]$, $\lambda_j^t - \lambda_j^{t+1} \leq \gamma_-^t$. Therefore, $x_i^{t+1} \leq x_i^t + \frac{\gamma_-^t [A^T e_m]_i}{\mu}$ holds for all $i \in [n]$. Finally, we write

$$\begin{aligned}[Ax^{t+1} - c]_j &= [Ax^t - c]_j + [A(x^{t+1} - x^t)]_j \\ &\leq -\Delta_j^t + \frac{\gamma_-^t [AA^T e_m]_j}{\mu} = 0,\end{aligned} \tag{16}$$

which concludes the first case.

2) Now, pick a constraint j for which $[Ax^t + \Delta^t - c]_j > 0$. If $\lambda_j^{t+1} = \overline{\lambda}$, then by definition $[Ax^{t+1} - c]_j \leq 0$ holds. Therefore we assume that $\lambda_j^{t+1} < \overline{\lambda}$ and show that for all users i for which $A_{ji} = 1$, $p_i^t - p_i^{t+1} = [A^T\lambda^t]_i - [A^T\lambda^{t+1}]_i \leq 0$ (otherwise if $A_{ji} = 0$, x_i^{t+1} does not affect constraint j). This implies that $x_i^{t+1} \leq x_i^t$.

$$[A^T\lambda^t]_i - [A^T\lambda^{t+1}]_i = \sum_k A_{ki}(\lambda_k^t - \lambda_k^{t+1}) \tag{17}$$

$$\leq (m-1)\gamma_-^t - \gamma_+^t = 0. \tag{18}$$

Since $[Ax^t - c]_j \leq 0$ is given, $[Ax^{t+1} - c]_j \leq 0$ holds when $x_i^{t+1} \leq x_i^t$, which concludes the second case.

We have proven that $Ax^{t+1} - c \leq 0$ holds if $Ax^t - c \leq 0$. Since by definition $Ax^1 - c \leq 0$ is ensured, $Ax^t - c \leq 0$ holds for all $t \geq 1$. ∎

Given that under Proposition 1, x^t for all $t \geq 1$ are feasible and therefore implementable, the static regret (9) is a valid choice of performance metric. Next, we prove that the regret of Algorithm 2 is $\mathcal{O}(\sqrt{T})$.

B. Regret Analysis

The following theorem establishes an upper bound on the regret incurred by the primal iterates produced by Algorithm 2:

Theorem 1. *Let $\gamma_-^t = \gamma/\sqrt{t}$ for some $\gamma > 0$ and set Δ^t and γ_+^t as in Proposition 1. Then for all $t \geq 1$, the iterates produced by Algorithm 2 are feasible. Furthermore, the regret $R(T)$ for all $T \geq 1$ satisfies*

$$R(T) \leq \frac{\overline{\lambda}^2 \|c\|_1 \sqrt{T}}{\gamma} + 2C\gamma\sqrt{T}, \tag{19}$$

where

$$C = \|c\|_1 + \frac{\overline{\lambda} m(\|A^T e_m\|^2 + \rho(A^TA)(m-1)^2/\mu)}{\mu} \tag{20}$$

and $\rho(A^TA)$ is the spectral radius of the matrix A^TA.

Proof: According to Proposition 1, x^t is feasible and $Ax^t - c \leq 0$ holds for all t. Since x^t is the maximizer of the Lagrangian $f(x) - {\lambda^t}^T(Ax - c)$, strong duality implies that

$$q(\lambda^t) := f(x^t) - {\lambda^t}^T(Ax^t - c) \geq f^\star. \tag{21}$$

Accordingly, we can write

$$R(T) = \sum_{t=1}^T f^\star - f(x_t) \leq \sum_{t=1}^T {\lambda^t}^T(c - Ax^t). \tag{22}$$

Since $f_i(\cdot)$ are μ-strongly convex, the dual function denoted as $q(\lambda)$ is $L := \frac{\rho(A^TA)}{\mu}$-smooth [11, Lemma II.2]. The descent lemma for smooth functions implies (e.g., [6]):

$$q(\lambda^{t+1}) \leq q(\lambda^t) + \langle \nabla q(\lambda^t), \lambda^{t+1} - \lambda^t \rangle + \frac{L}{2}\|\lambda^{t+1} - \lambda^t\|^2 \tag{23}$$

Next, we decompose the $\langle \nabla q(\lambda^t), \lambda^{t+1} - \lambda^t \rangle$ term into three according to the following sets:

- Let $\mathcal{A}_1^t = \{j \in [m] | \lambda_j^t \geq \gamma_-^t, [Ax^t + \Delta^t - c]_j < 0\}$.
- Let $\mathcal{A}_2^t = \{j \in [m] | \lambda_j^t < \gamma_-^t, [Ax^t + \Delta^t - c]_j < 0\}$.
- Let $\mathcal{A}_3^t = \{j \in [m] | [Ax^t + \Delta^t - c]_j \geq 0\}$.

Noting that $\nabla q(\lambda^t) = c - Ax^t$, we write

$$\begin{aligned}\langle \nabla q(\lambda^t), \lambda^{t+1} - \lambda^t \rangle &= -\sum_{j \in \mathcal{A}_1^t} [c - Ax^t]_j \gamma_-^t \\ &+ \sum_{j \in \mathcal{A}_2^t} [c - Ax^t]_j (\lambda_j^{t+1} - \lambda_j^t) + \sum_{j \in \mathcal{A}_3} [c - Ax^t]_j (\lambda_j^{t+1} - \lambda_j^t)\end{aligned} \tag{24}$$

$$\leq -\sum_{j \in \mathcal{A}_1^t} [c - Ax^t]_j \gamma_-^t + \sum_{j \in \mathcal{A}_3^t} \Delta_j^t \gamma_+^t \tag{25}$$

$$\leq -\sum_{j \in \mathcal{A}_1^t} [c - Ax^t]_j \gamma_-^t + \|\Delta^t\|_1 \gamma_+^t \tag{26}$$

where the first inequality follows from the fact that for $j \in \mathcal{A}_2^t$, $(\lambda_j^{t+1} - \lambda_j^t) \leq 0$, and for $j \in \mathcal{A}_3^t$, $(c - Ax^t)_j \leq \Delta_j^t$ and $(\lambda_j^{t+1} - \lambda_j^t) \leq \gamma_+^t$. We plug (26) into (23), rearrange, and use $\|\lambda^{t+1} - \lambda^t\|^2 \leq m(\gamma_+^t)^2$:

$$\sum_{j \in \mathcal{A}_1^t} [c - Ax^t]_j \leq \frac{q(\lambda^t) - q(\lambda^{t+1})}{\gamma_-^t} + \frac{\|\Delta^t\|_1 \gamma_+^t}{\gamma_-^t} + \frac{Lm(\gamma_+^t)^2}{\gamma_-^t}. \tag{27}$$

Since $\lambda_j^t \leq \overline{\lambda}$ and $c - Ax^t \geq 0$:

$$\begin{aligned}&\sum_{j \in \mathcal{A}_1^t} [c - Ax^t]_j \lambda_j^t \leq \overline{\lambda} \sum_{j \in \mathcal{A}_1^t} (c - Ax^t)_j \\ &\leq \overline{\lambda}\left(\frac{q(\lambda^t) - q(\lambda^{t+1})}{\gamma_-^t} + \frac{\|\Delta^t\|_1 \gamma_+^t}{\gamma_-^t} + \frac{Lm(\gamma_+^t)^2}{\gamma_-^t}\right)\end{aligned} \tag{28}$$

For $j \in \mathcal{A}_2^t$ by definition:

$$\sum_{j \in \mathcal{A}_2^t} (c - Ax^t)_j \lambda_j^t \leq \|c\|_1 \gamma_-^t \tag{29}$$

For $j \in \mathcal{A}_3^t$ by definition:

$$\sum_{j \in \mathcal{A}_3^t} (c - Ax^t)_j \lambda_j^t \leq \|\Delta^t\|_1 \overline{\lambda} \tag{30}$$

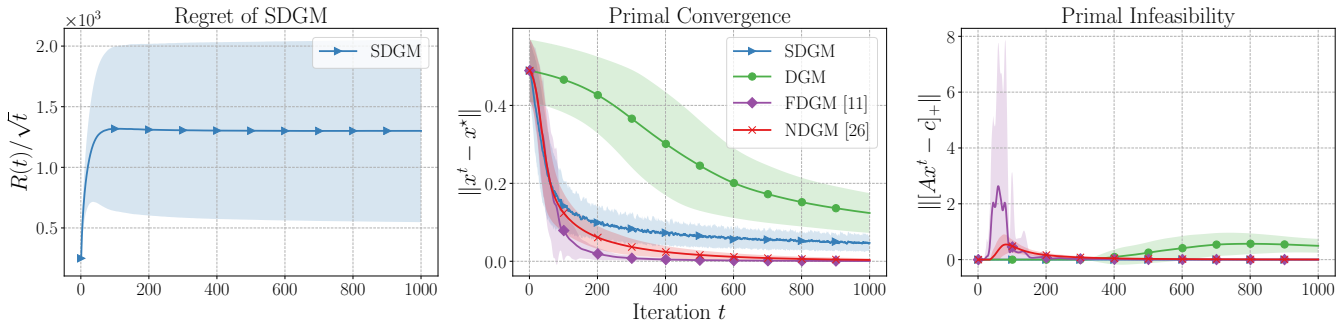

Fig. 1. Results comparing the performances of all four methods for the numerical study described in Section V: Regret of the SDGM (left), the convergence rate of the primal iterates to the optimal solution for all four methods (middle), and the infeasibility amount of the primal iterates for all four methods (right). The solid lines correspond to the means of the 100 experiments, while shaded areas correspond to the standard deviations.

We sum (28), (29), and (30) and plug Δ^t and γ_+^t:

$$\lambda^{t^T}(c - Ax^t) \leq \frac{\overline{\lambda}(q(\lambda^t) - q(\lambda^{t+1}))}{\gamma_-^t} + \frac{\overline{\lambda}\gamma_-^t \|A^T e_m\|^2 (m-1)}{\mu}$$
$$+ \frac{\overline{\lambda} L m (m-1)^2 \gamma_-^t}{\mu} + \|c\|_1 \gamma_-^t + \frac{\overline{\lambda}\|A^T e_m\|^2 \gamma_-^t}{\mu} \tag{31}$$

Summing the above inequality from $t = 1$ to T with definitions of C and $\gamma_-^t = \gamma/\sqrt{t}$ (with $\gamma_-^0 := 0$):

$$R(T) \leq \overline{\lambda} \sum_{t=1}^{T} q(\lambda^t) \left(\frac{1}{\gamma_-^t} - \frac{1}{\gamma_-^{t-1}} \right) - \frac{\overline{\lambda} q(\lambda^{T+1})}{\gamma_-^T} + C \sum_{t=1}^{T} \gamma_-^t \tag{32}$$

$$\leq \overline{\lambda}(f^\star + \overline{\lambda}\|c\|_1) \sum_{t=1}^{T} \left(\frac{1}{\gamma_-^t} - \frac{1}{\gamma_-^{t-1}} \right) - \frac{\overline{\lambda} f^\star}{\gamma_-^T} + C \sum_{t=1}^{T} \gamma_-^t \tag{33}$$

$$\leq \overline{\lambda}(f^\star + \overline{\lambda}\|c\|_1) \frac{1}{\gamma_-^T} - \frac{\overline{\lambda} f^\star}{\gamma_-^T} + C \sum_{t=1}^{T} \gamma_-^t \tag{34}$$

$$\leq \overline{\lambda}^2 \|c\|_1 \frac{\sqrt{T}}{\gamma} + 2C\gamma\sqrt{T}, \tag{35}$$

where the first inequality uses (21) and

$$q(\lambda^t) = f(x^t) - \lambda^{t^T}(c - Ax^t) \leq f^\star + \overline{\lambda}\|c\|_1 \tag{36}$$

for any feasible x^t. ∎

According to Theorem 1, Algorithm 2 produces feasible solutions for all $t \geq 1$ that achieve a sublinear regret of $\mathcal{O}(\sqrt{T})$. To minimize the upper bound on the RHS of (19), we let $\gamma = \sqrt{\overline{\lambda}^2 \|c\|_1/(2C)}$ to get

$$R(T) \leq 2\overline{\lambda}\sqrt{2C\|c\|_1 T}. \tag{37}$$

It is worthwhile to highlight a trade-off between feasibility and performance: It is known that with strongly concave utility functions $f_i(\cdot)$, it is possible to produce primal iterates such that the last iterate (or the average iterate) has suboptimality of $\mathcal{O}(1/T)$ [11] (or $\mathcal{O}(1/T^2)$ [16]) after T iterations, where the primal iterates are not necessarily feasible for all $t \geq 1$. On the other hand, a regret of $\mathcal{O}(\sqrt{T})$ implies a suboptimality of $\mathcal{O}(1/\sqrt{T})$ (of the average iterate $\overline{x}$), while producing feasible iterates for all $t \geq 1$. Accordingly, the feasibility of the primal iterates comes at the cost of a slower reduction rate of suboptimality.

In the next section, we numerically demonstrate the feasibility and the convergence rate of, and the regret incurred by the primal iterates produced by Algorithm 2.

V. NUMERICAL STUDY

In this section, we compare the performance of the safe dual gradient method developed in Section III with three other distributed algorithms commonly used in the literature for solving the NUM problem: 1) the dual (sub)gradient method (DGM) explained in Section II, the fast weighted dual gradient method (FDGM) [11], and 3) the Newton-type diagonally scaled (dual) gradient method (NDGM) introduced in [26].

Inspired by [11], we have implemented all algorithms on a randomly generated collection of 100 networks with a random number of users n taking (integer) values in range $[10, 40]$, and a random number of constraints m taking values in the interval $[5, 25]$ (generated independently). For each configuration, we randomly generated the matrix A by sampling $m \times n$ Bernoulli random variables (when a row or a column of A is zero vector, we generate another one). For all users $i \in [n]$, we let the utility function be $\theta_i \log(x_i + 0.1)$, where θ_i is sampled from the range $[10, 30]$ uniformly at random for each network configuration. We added 0.1 to the log function to prevent numerical instability when x_i is close to 0. We let $\mathcal{X}_i = [0, \infty)$ for all $i \in [n]$. Finally, we assumed that for all the constraints $j \in [m]$, $c_j = 1$. For each configuration, we ran all four methods for $T = 1000$ and demonstrate the results in Figure 1.

In the left figure, we plot the regret incurred by the SDGM. We note that because the other three algorithms do not guarantee primal feasibility for all iterations, regret is not a well-defined metric for their performances. We show that the mean of $R(t)/\sqrt{t}$ is bounded by $\mathcal{O}(1)$, which implies that $R(t)$ is bounded by $\mathcal{O}(\sqrt{t})$.

In the middle figure, we plot the convergence of the primal iterates, i.e., $\|x^t - x^\star\|$, for all four methods. The figure shows that although the SDGM is not a fast method as the FDGM and the NDGM, it still performs closely to those

fast algorithms and is much better than DGM in terms of convergence rate.

In the right figure, we plot the primal infeasibility, i.e., $\|[Ax^t - c]_+\|$, for all four methods. The figure shows that although FDGM and NDGM are fast, they do not produce feasible iterates for all $t \geq 1$. On the other hand, the SDGM is guaranteed to produce feasible primal iterates.

VI. CONCLUSIONS

In this work, we introduced a novel algorithm, called the safe dual gradient method (SDGM), for solving NUM problems in a distributed fashion. In contrast to the literature on first-order distributed methods, where bounds on the feasibility violation of the primal iterates are established, the SDGM is guaranteed to produce feasible primal iterates. This is done by: 1) adding a diminishing safety margin to the constraints, and 2) using a sign-based dual update method with different step sizes for plus and minus directions. Furthermore, we have proven that the regret incurred by the SDGM is $\mathcal{O}(\sqrt{T})$.

An immediate trade-off is that although the SDGM produces feasible iterates, it converges slower than the state-of-the-art methods. It would be an interesting future direction to study an accelerated version of the SDGM (e.g., similar to [15]) that still produces feasible iterates. Additionally, future work should include more general constraints (e.g., $A \in \mathbb{R}^{m \times n}$) as well as the case when there is uncertainty about constraints.

REFERENCES

[1] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V. W. Wong, and J. Jatskevich, "Optimal real-time pricing algorithm based on utility maximization for smart grid," in *2010 First IEEE International Conference on Smart Grid Communications*. IEEE, 2010, pp. 415–420.

[2] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[3] M. Chiang and J. Bell, "Balancing supply and demand of bandwidth in wireless cellular networks: utility maximization over powers and rates," in *IEEE INFOCOM 2004*, vol. 4. IEEE, 2004, pp. 2800–2811.

[4] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[5] I. Necoara, V. Nedelcu, and I. Dumitrache, "Parallel and distributed optimization methods for estimation and control in networks," *Journal of Process Control*, vol. 21, no. 5, pp. 756–766, 2011.

[6] D. P. Bertsekas, "Nonlinear programming," *Journal of the Operational Research Society*, vol. 48, no. 3, pp. 334–334, 1997.

[7] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

[8] N. Li, L. Chen, and S. H. Low, "Optimal demand response based on utility maximization in power networks," in *2011 IEEE power and energy society general meeting*. IEEE, 2011, pp. 1–8.

[9] N. Z. Shor, *Minimization methods for non-differentiable functions*. Springer Science & Business Media, 2012, vol. 3.

[10] A. Nedić and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods," *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1757–1780, 2009.

[11] A. Beck, A. Nedić, A. Ozdaglar, and M. Teboulle, "An $o(1/k)$ gradient method for network resource allocation problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 64–73, 2014.

[12] B. T. Polyak, "Introduction to optimization. optimization software," *Inc., Publications Division, New York*, vol. 1, p. 32, 1987.

[13] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical programming*, vol. 120, no. 1, pp. 221–259, 2009.

[14] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex analysis and optimization*. Athena Scientific, 2003, vol. 1.

[15] Y. E. Nesterov, "A method for solving the convex programming problem with convergence rate o (1/kˆ 2)," in *Dokl. akad. nauk Sssr*, vol. 269, 1983, pp. 543–547.

[16] I. Necoara and V. Nedelcu, "Rate analysis of inexact dual first-order methods application to dual decomposition," *IEEE Transactions on Automatic Control*, vol. 59, no. 5, pp. 1232–1243, 2013.

[17] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 18–33, 2013.

[18] A. Chernov, P. Dvurechensky, and A. Gasnikov, "Fast primal-dual gradient method for strongly convex minimization problems with linear constraints," in *International Conference on Discrete Optimization and Operations Research*. Springer, 2016, pp. 391–403.

[19] I. Necoara and V. Nedelcu, "On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems," *Automatica*, vol. 55, pp. 209–216, 2015.

[20] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *international conference on machine learning*. PMLR, 2017, pp. 3027–3036.

[21] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić, "A dual approach for optimal algorithms in distributed optimization over networks," in *2020 Information Theory and Applications Workshop (ITA)*. IEEE, 2020, pp. 1–37.

[22] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A survey on demand response programs in smart grids: Pricing methods and optimization algorithms," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 152–178, 2014.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] P. Armand, J. C. Gilbert, and S. Jan-Jégou, "A feasible bfgs interior point algorithm for solving convex minimization problems," *SIAM Journal on Optimization*, vol. 11, no. 1, pp. 199–222, 2000.

[25] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization," in *49th IEEE Conference on Decision and Control (CDC)*. IEEE, 2010, pp. 1816–1821.

[26] S. Athuraliya and S. H. Low, "Optimization flow control with newton-like algorithm," *Telecommunication Systems*, vol. 15, no. 3, pp. 345–358, 2000.

[27] I. Necoara and J. Suykens, "Interior-point lagrangian decomposition method for separable convex optimization," *Journal of Optimization Theory and Applications*, vol. 143, no. 3, pp. 567–588, 2009.

[28] I. Usmanova, A. Krause, and M. Kamgarpour, "Safe convex learning under uncertain constraints," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 2106–2114.

[29] A. Allibhoy and J. Cortés, "Anytime solution of constrained nonlinear programs via control barrier functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 6527–6532.

[30] S. Amani, M. Alizadeh, and C. Thrampoulidis, "Linear stochastic bandits under safety constraints," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[31] J. Mo and J. Walrand, "Fair end-to-end window-based congestion control," *IEEE/ACM Transactions on networking*, vol. 8, no. 5, pp. 556–567, 2000.

[32] R. T. Rockafellar, *Convex analysis*. Princeton university press, 2015.

[33] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in *IEEE international conference on neural networks*. IEEE, 1993, pp. 586–591.

[34] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "signsgd: Compressed optimisation for non-convex problems," in *International Conference on Machine Learning*. PMLR, 2018, pp. 560–569.