ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2021, Vol. 61, No. 2, pp. 297–328. © Pleiades Publishing, Ltd., 2021.
Russian Text © The Author(s), 2021, published in Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, 2021, Vol. 61, No. 2, pp. 312–344.

INFORMATION SCIENCE

Numerical Methods for the Resource Allocation Problem in a Computer Network

E. A. Vorontsova[a], A. V. Gasnikov[a,b], P. E. Dvurechensky[c,b], A. S. Ivanova[d,*], and D. A. Pasechnyuk[a]

[a] *Moscow Institute of Physics and Technology (National Research University), Dolgoprudnyi, Moscow oblast, 141701 Russia*

[b] *Kharkevich Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, 127051 Russia*

[c] *Weierstrass Institute for Applied Analysis and Stochastics, Berlin, 10117 Germany*

[d] *National Research University Higher School of Economics, Moscow, 109028 Russia*

**e-mail: asivanova@hse.ru*

Received November 29, 2019; revised September 10, 2020; accepted September 16, 2020

Abstract—The resource allocation problem in computer networks with a large number of links is considered. The links are used by consumers (users), whose number can also be very large. For the dual problem, numerical optimization methods are proposed, such as the fast gradient method, the stochastic projected subgradient method, the ellipsoid method, and the random gradient extrapolation method. A convergence rate estimate is obtained for each of the methods. Algorithms for distributed computation of steps in the considered methods as applied to computer networks are described. Special attention is given to the primal-dual property of the proposed algorithms.

Keywords: resource allocation, communication networks, network utility maximization, primal-dual property, fast gradient method, stochastic projected subgradient method, ellipsoid method, random gradient extrapolation method

DOI: 10.1134/S0965542521020135

1. INTRODUCTION

1.1. Motivation

In this paper, the problem of controlling modern communication networks is considered from the point of view of optimization and stochastic modeling. To solve problems of this type, we need to represent and analyze the mathematical model arising in the simulation of large-scale broadband networks. It is expected that, in future communication networks, there appear applications that will be able to change their data transmission rates according to the available network capacity. An example of such a network is TCP traffic through the Internet.

The key issue addressed in this paper is how the available capacity of the network is to be allocated among competing flows. The use of available capacities by consumers is controlled by correcting the link prices.

Thus, we consider the problem of optimizing resource allocation in computer networks with a large number of links. The links are used by consumers (users), whose number can also be very large. The goal of this study is to determine a resource allocation mechanism, where the resources are understood as available capacities of network links. Additionally, it is necessary to ensure stable performance of the system and to prevent overloads. As an optimality criterion, we use the sum of the utilities of all users of the computer network.

Originally, standard resource allocation problems reducing to the maximization of the aggregate utility of users in the case of shared use of available resources were considered in [1]. Resource allocation in computer networks was investigated in the recent work [2]. Proposed in [3], the mechanisms of decentralized resource allocation drew much attention in economic studies (see, e.g., [4–6] and references therein). In this paper, following [7, 8], we consider various mechanisms of price adjustment. The proposed approaches are of practical importance due to their decentralized nature, which means that the price of an individual link is established and adjusted relying only on the reactions of users employing this link,

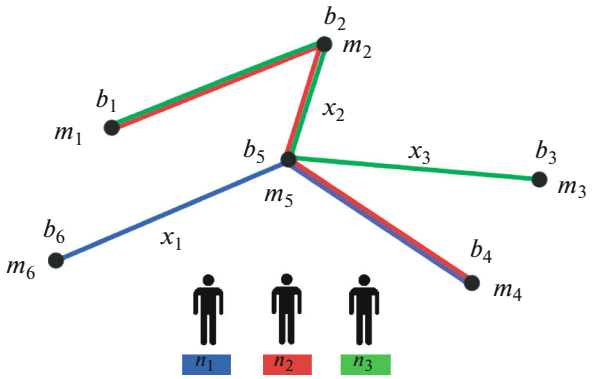

Fig. 1. Example of a computer network with $m = 6$ and $n = 3$.

rather than on the reactions of all network users. In the case of such an adjustment mechanism, all links perform independently.

Additionally, one of the approaches proposed in this paper relies on the stochastic projected subgradient method and overcomes the following difficulty arising in actual networks: data packets sent by users arrive at a link at different times, so the total traffic through the link is not known in practice. This difficulty is obviated by applying stochastic methods. They can do without the exact value of total traffic, managing only with its estimate, which can be obtained using the traffic of a single user. The idea of using the stochastic projected subgradient method for solution of this problem was proposed in [2].

1.2. Content of This Paper

This paper is organized as follows. The formulation of the problem and the construction of its dual are described in Section 2. Additionally, we state all necessary assumptions for the primal problem. In Section 3, the problem is solved by applying Nesterov's fast gradient method [9], whose complexity bound is found to be $O\left(\frac{1}{\sqrt{\varepsilon}}\right)$. In Section 4, this problem is solved using the stochastic projected subgradient method with $O\left(\frac{1}{\varepsilon^2}\right)$ complexity bounds.

In Section 5, the problem is solved by applying the ellipsoid method, which is well suited for low-dimensional problems, and an algorithm for constructing the accuracy certificate for this method is described. We present complexity bounds on the order of $O\left(m^2 \ln\frac{1}{\varepsilon}\right)$, where m is the number of links in the network. A regularization technique for recovering the solution of the primal problem from the solution of the dual one if the method is not primal-dual is described in Section 6. The regularized problem is solved using the random gradient extrapolation method in Section 7. Its complexity bounds are presented, which are on the order of $O\left(\frac{1}{\sqrt{\varepsilon}}\ln\frac{1}{\varepsilon}\right)$, where the logarithmic factor appears due to the regularization of the dual problem.

Numerical experiments supporting the theoretical results obtained in the preceding sections are presented in Section 8.

Additionally, for each algorithm, we describe its distributed computation in the context of the problem under consideration.

2. FORMULATION OF THE PROBLEM

Consider a computer network with m links and n users (or nodes), see Fig. 1.

The users exchange data packets through a fixed set of links. The network structure is specified by the routing matrix $C = (C_i^j) \in \mathbf{R}^{m\times n}$. The matrix columns $\mathbf{C}_i \neq 0$, $i = 1,\ldots,n$, are m-dimensional Boolean vec-

tors such that $C_i^j = 1$ if node i uses link j and $C_i^j = 0$ otherwise. The link capacities are described by a vector $\mathbf{b} \in \mathbf{R}^m$ with strictly positive components.

The users estimate the performance of the network with the help of utility functions $u_k(x_k)$, $k = 1,\ldots,n$, where $x_k \in \mathbf{R}_+$ is the rate of data transmission from the kth user. As an optimality criterion for the system, we use the sum of the utility functions for all users [1].

The problem of maximizing the aggregate utility of the network under constraints imposed on the link capacities is stated as follows:

$$\max_{\left\{C\mathbf{x}=\sum_{k=1}^{n}\mathbf{C}_k x_k\right\}\le\mathbf{b}} \left\{U(\mathbf{x}) = \sum_{k=1}^{n} u_k(x_k)\right\}, \tag{1}$$

where $\mathbf{x} = (x_1,\ldots,x_n) \in \mathbb{R}_+^n$. The solution of this problem is an optimal resource allocation $\mathbf{x}^*$.

Consider the standard transition to the dual problem for (1). Given a vector of dual multipliers $\boldsymbol{\lambda} = (\lambda_1,\ldots,\lambda_m) \in \mathbb{R}_+^m$, which can be interpreted as the price vector of the links, the dual objective function is defined as

$$\varphi(\boldsymbol{\lambda}) = \max_{\mathbf{x}\in\mathbf{R}_+^n}\left\{\sum_{k=1}^{n} u_k(x_k) + \left\langle \boldsymbol{\lambda}, \mathbf{b} - \sum_{k=1}^{n}\mathbf{C}_k x_k\right\rangle\right\} = \left\langle \boldsymbol{\lambda}, \mathbf{b}\right\rangle + \sum_{k=1}^{n}\left(u_k(x_k(\boldsymbol{\lambda})) - \left\langle \boldsymbol{\lambda}, \mathbf{C}_k x_k(\boldsymbol{\lambda})\right\rangle\right), \tag{2}$$

here, the users choose optimal data transmission rates x_k by solving the optimization problem

$$x_k(\boldsymbol{\lambda}) = \arg\max_{x_k\in\mathbf{R}_+}\left\{u_k(x_k) - x_k\left\langle \boldsymbol{\lambda}, \mathbf{C}_k\right\rangle\right\}. \tag{3}$$

Let $\mathbf{x}(\boldsymbol{\lambda})$ denote the vector with components $x_k(\boldsymbol{\lambda})$. Then, to find optimal prices $\boldsymbol{\lambda}^*$, we need to solve the problem

$$\min_{\boldsymbol{\lambda}\in\mathbf{R}_+^m} \varphi(\boldsymbol{\lambda}). \tag{4}$$

Assume that the Slater condition is satisfied for the primal problem. Then, by virtue of strong duality, both primal and dual problems have solutions. By using the Slater condition, it is possible to compactify the solution of the dual problem. Assume that the solution of the dual problem satisfies the estimate

$$\left\|\boldsymbol{\lambda}^*\right\|_2 \le R.$$

Here, R has no effect on the performance of the considered algorithms, but is only involved in their convergence rate estimates.

The basic idea of this paper is to apply various optimization methods for solving dual problem (4) with the addition of primal-dual analysis of these methods, which makes it possible to recover the solution of primal problem (1). In this sense, we develop the approach addressed in our previous works [10–21]. The basic difference is that we consider inequality constraints and analyze stochastic algorithms in the terms of estimates with high probability, rather than on average.

2.1. Strongly Concave Utility Functions

In some sections, we assume that the utility functions $u_k(x_k)$, $k = 1,\ldots,n$, are *strongly concave* with a constant μ. In this subsection, we describe the properties of the dual problem under this assumption.

Proposition 1 (Demyanov–Danskin–Rubinov theorem, see [22, 23]). *Suppose that, for any* $\boldsymbol{\lambda} \in \mathbb{R}_+$, it holds that $\varphi(\boldsymbol{\lambda}) = \max\limits_{\mathbf{x}\in X} F(\mathbf{x},\boldsymbol{\lambda})$, *where* $F(\mathbf{x},\boldsymbol{\lambda})$ *is a convex and smooth function of* $\boldsymbol{\lambda}$ with a maximum reached at the only point $x(\boldsymbol{\lambda})$. *Then* $\nabla\varphi(\boldsymbol{\lambda}) = \nabla_{\boldsymbol{\lambda}} F(x(\boldsymbol{\lambda}),\boldsymbol{\lambda})$.

Proposition 2 (see [24]). *Suppose that the functions* $u_k(x_k)$ are μ-strongly concave for all $k = 1,\ldots,n$. *Then function* (2), *where* $x_k(\boldsymbol{\lambda})$, $k = 1,\ldots,n$, solve problem (3), *is* nm^2/μ-smooth, *i.e.*, *the gradient of the function* $\varphi(\boldsymbol{\lambda})$ satisfies the Lipschitz condition with constant $L = nm^2/\mu$:

$$\left\|\nabla\varphi(\boldsymbol{\lambda}^2) - \nabla\varphi(\boldsymbol{\lambda}^1)\right\|_2 \le L\left\|\boldsymbol{\lambda}^2 - \boldsymbol{\lambda}^1\right\|_2.$$

The proof of the proposition can be found in the Appendix.

2.2. Concave Utility Functions

Now we assume that the utility functions $u_k(x_k)$, $k = 1,\ldots,n$, are *concave, but not strongly concave*. Then the dual problem is not smooth. In this subsection, we describe some properties of subgradients of the dual problem under these assumptions.

The subgradient of dual problem (4) is defined as

$$\nabla\varphi(\boldsymbol{\lambda}) = \mathbf{b} - C\mathbf{x}.$$

Since $\mathbf{x}$ is a bounded rate of data transmission and the vector $\mathbf{b}$ is also bounded, we see that the subgradients of the dual problem are bounded. Thus, there exists a positive constant M such that

$$\|\nabla\varphi(\boldsymbol{\lambda})\|_2 \le M. \tag{5}$$

As a rough estimate from above for the constant M in (5), we can use $O(n\sqrt{m})$. The multiplier n appears because there are n terms and $\sqrt{m}$ is used as an estimate for the dependence of the 2-norm on the vector dimension m.

3. FAST GRADIENT METHOD

In this section, we assume that the utility functions $u_k(x_k)$, $k = 1,\ldots,n$, are *strongly concave* with a constant μ; therefore, the dual problem is smooth.

Dual problem (4) is solved by applying Nesterov's fast gradient method (FGM) in the following version (PDFGM method, see Algorithm 1).

Algorithm 1. Primal-Dual Fast Gradient Method (PDFGM)

Input: $u_k(\mathbf{x})$, $k = 1,\ldots,n$, are strongly concave utility functions for each user; $\boldsymbol{\lambda}^0$ is the initial price vector,

$$\alpha_t := \frac{t+1}{2},\ A_{-1} := 0,\ A_t := A_{t-1} + \alpha_t = \frac{(t+1)(t+2)}{4},\ \text{and } \tau_t := \frac{\alpha_{t+1}}{A_{t+1}} = \frac{2}{t+3},\ t = 0,1,\ldots,N-1.$$

1: **for** $t = 0,1,\ldots,N-1$

2: Compute $\varphi(\boldsymbol{\lambda}^t)$, $\nabla\varphi(\boldsymbol{\lambda}^t)$

3: $\mathbf{y}^t := \left[\boldsymbol{\lambda}^t - \frac{1}{L}\left(\mathbf{b} - \sum_{k=1}^n \mathbf{C}_k x_k(\boldsymbol{\lambda}^t)\right)\right]_+$

4: $\mathbf{z}^t := \left[\boldsymbol{\lambda}^0 - \frac{1}{L}\sum_{j=0}^t \alpha_j\left(\mathbf{b} - \sum_{k=1}^n \mathbf{C}_k x_k(\boldsymbol{\lambda}^j)\right)\right]_+$

5: $\boldsymbol{\lambda}^{t+1} := \tau_t\mathbf{z}^t + (1-\tau_t)\mathbf{y}^t$

6: $\hat{\mathbf{x}}^{t+1} := \frac{1}{A_{t+1}}\sum_{j=0}^{t+1}\alpha_j\mathbf{x}(\boldsymbol{\lambda}^j)$

7: **end for**

8: **return** $\boldsymbol{\lambda}^N$, $\hat{\mathbf{x}}^N$

3.1. Distributed Method

The problem under consideration can also be solved using the distributed version of FGM, which means that each link can compute an optimal data transmission rate only relying on the reactions of the users that employ this link without interacting with the other links.

The process occurring at the tth iteration for link j can be described as follows.

1. Given information received from the users after the preceding iteration with index $t-1$ (vector $\mathbf{x}^t = \mathbf{x}(\boldsymbol{\lambda}^t)$), the link j computes

$$y_j^t = \left[\lambda_j^t - \frac{1}{L}\left(b_j - \sum_{k=1}^n C_k^j x_k^t\right)\right]_+.$$

Here, $C_k^j \ne 0$ only for users employing the link j. Therefore, to compute this step, the link needs only information from the users employing this link.

2. Similarly, the link j computes

$$z_j^t = \left[\lambda_j^0 - \frac{\alpha_j}{L}\left(b_j - \sum_{k=1}^{n} C_k^j x_k^t\right)\right]_+.$$

3. After obtaining values at two preceding steps, link j computes the price for the next iteration $t+1$:

$$\lambda_j^{t+1} = \tau_t z_j^t + (1-\tau_t) y_j^t$$

and sends out this information to all users connected to it.

4. The users compute the optimal data transmission rates $\hat{\mathbf{x}}^{t+1}$; specifically, for user k, we obtain

$$x_k(\boldsymbol{\lambda}^{t+1}) = \arg\max_{x_k \in \mathbf{R}_+}\left(u_k(x_k) - x_k \sum_{j=1}^{m} \lambda_j^{t+1} C_k^j\right),$$

where, by the definition of the matrix C, the user needs only data from the links it employs. Next, the user computes the optimal rate

$$\hat{x}_k^{t+1} = \frac{A_t \hat{x}_k^t + x_k^{t+1}}{A_{t+1}}.$$

Remark 1. A disadvantage of this algorithm is that each link has to know the reactions of all users that employ it at every iteration step. Unfortunately, in actual networks, users do not transmit data simultaneously, so it is rather difficult to collect this information for the link. However, if complete information on the users is available, the link can establish an equilibrium price more quickly.

3.2. Estimation of the Convergence Rate of FGM

Before proving the convergence of FGM for the problem under consideration, we state the key lemma necessary for estimating the residuals in the constraints and the duality gap after running PDFGM.

Lemma 1. *Suppose that Algorithm 1 starts at an initial point* $\boldsymbol{\lambda}^0$ *lying in the Euclidean ball of radius* R *centered at the origin. Then, after performing* N *iterations of Algorithm* 1, *it holds that*

$$A_N \varphi(\mathbf{y}^N) - A_N U(\hat{\mathbf{x}}^N) + 2\hat{R} A_N \left\|(C\hat{\mathbf{x}}^N - \mathbf{b})_+\right\|_2 \le \frac{37 L \hat{R}^2}{9}, \tag{6}$$

where $\hat{\mathbf{x}}^N = \frac{1}{A_N}\sum_{t=0}^{N-1} \alpha_t \mathbf{x}(\boldsymbol{\lambda}^t)$ *and* $\hat{R} = 3R$.

The proof of the lemma can be found in the Appendix.

Now we formulate a theorem on the convergence rate estimate for Algorithm 1.

Theorem 1. *Suppose that Algorithm* 1 *starts at an initial point* $\boldsymbol{\lambda}^0$ *lying in the Euclidean ball of radius* R *centered at the origin. Then, after performing*

$$N = \left\lceil \frac{2\hat{R}}{3}\sqrt{\frac{37L}{\varepsilon}} \right\rceil$$

iterations of Algorithm 1, *it holds that*

$$U(\mathbf{x}^*) - U(\hat{\mathbf{x}}^N) \le \varepsilon, \quad \left\|(C\hat{\mathbf{x}}^N - \mathbf{b})_+\right\|_2 \le \frac{\varepsilon}{\hat{R}},$$

where $\hat{\mathbf{x}}^N = \frac{1}{A_N}\sum_{t=0}^{N-1} \alpha_t \mathbf{x}(\boldsymbol{\lambda}^t)$, $\mathbf{x}^*$ *is an optimal solution of problem* (1), *and* $\hat{R} = 3R$.

Proof. Let $\mathrm{Opt}[P]$ denote the optimal value in the original primal problem (1), and let $\mathrm{Opt}[D]$ denote the optimal value in the dual problem (4). By the weak duality, we have

$$\mathrm{Opt}[D] \ge \mathrm{Opt}[P].$$

Moreover, for all $\mathbf{x} \in \mathbb{R}^n_+$, the optimal solution $\boldsymbol{\lambda}^*$ of dual problem (4) satisfies

$$\mathrm{Opt}[P] \geq U(\mathbf{x}) - \left\langle \boldsymbol{\lambda}^*, \left(\sum_{k=1}^{n} \mathbf{C}_k x_k - \mathbf{b} \right)_+ \right\rangle \geq U(\mathbf{x}) - \hat{R} \left\| (C\mathbf{x} - \mathbf{b})_+ \right\|_2. \tag{7}$$

Then

$$\varphi(\mathbf{y}^N) - U(\hat{\mathbf{x}}^N) = \varphi(\mathbf{y}^N) - U(\hat{\mathbf{x}}^N) + \mathrm{Opt}[P] - \mathrm{Opt}[P] + \mathrm{Opt}[D] - \mathrm{Opt}[D] = \underbrace{(\mathrm{Opt}[D] - \mathrm{Opt}[P])}_{\geq 0}$$

$$+ (\mathrm{Opt}[P] - U(\hat{\mathbf{x}}^N)) + \underbrace{(\varphi(\mathbf{y}^N) - \mathrm{Opt}[D])}_{\geq 0} \overset{(7)}{\geq} -\left\langle \boldsymbol{\lambda}^*, (\mathbf{b} - C\hat{\mathbf{x}}^N)_+ \right\rangle \geq -\hat{R} \left\| (C\hat{\mathbf{x}}^N - \mathbf{b})_+ \right\|_2.$$

Substituting the last inequality into (6) yields the estimate

$$\hat{R} \left\| (C\hat{\mathbf{x}}^N - \mathbf{b})_+ \right\|_2 \leq \frac{37L\hat{R}^2}{9A_N}.$$

Consequently, $\varphi(\mathbf{y}^N) - U(\hat{\mathbf{x}}^N) \geq -\frac{37L\hat{R}^2}{9A_N}$. On the other hand, it follows from (6) that $\varphi(\mathbf{y}^N) - U(\hat{\mathbf{x}}^N) \leq \frac{37L\hat{R}^2}{9A_N}$. Therefore,

$$\left| \varphi(\mathbf{y}^N) - U(\hat{\mathbf{x}}^N) \right| \leq \frac{37L\hat{R}^2}{9A_N}.$$

Since $\varphi(\mathbf{y}^N) \geq \mathrm{Opt}[D] = \varphi(\mathbf{y}^*) \geq \mathrm{Opt}[P] = U(\mathbf{x}^*)$, we have

$$U(\mathbf{x}^*) - U(\hat{\mathbf{x}}^N) \leq \frac{37L\hat{R}^2}{9A_N} = \frac{148L\hat{R}^2}{9(N+1)(N+2)} \leq \frac{148L\hat{R}^2}{9N^2} \leq \varepsilon.$$

Expressing N from the last inequality gives the estimate from the condition of the theorem.

4. STOCHASTIC PROJECTED SUBGRADIENT METHOD

Consider the original problem (1), now assuming that the utility functions $u_k(x_k)$, $k = 1, \ldots, n$, are concave, but not strongly concave. In this case, dual problem (4) becomes nonsmooth. Accordingly, for its solution, we propose the stochastic projected subgradient method. For the first time, the idea of using this method for solving the given problem was proposed in [2].

Consider the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Suppose that a sequence of independent random variables $\{\xi^t\}_{t=0}^{\infty}$ uniformly distributed on $\{1, \ldots, n\}$ is defined on $(\Omega, \mathcal{F}, \mathbb{P})$, i.e.,

$$\mathbb{P}(\xi^t = i) = \frac{1}{n}, \quad i \in \{1, \ldots, n\}.$$

If there is an oracle producing the stochastic subgradient of the dual function $\nabla\varphi(\boldsymbol{\lambda}, \xi)$, i.e.,

$$\nabla\varphi(\boldsymbol{\lambda}, \xi) = \mathbf{b} - nC_\xi x_\xi(\boldsymbol{\lambda}),$$

then

$$\mathbb{E}[\mathbf{b} - nC_{\xi_t} x_{\xi_t}(\boldsymbol{\lambda}^t) | \xi^t] = \mathbf{b} - \sum_{k=1}^{n} \mathbf{C}_k x_k(\boldsymbol{\lambda}^t) = \nabla\varphi(\boldsymbol{\lambda}^t)$$

Algorithm 2. Primal-Dual Stochastic Projected Subgradient Method (PDSPSGM), Version 1

Input: $u_k(\mathbf{x})$, $k = 1, \ldots, n$, are concave utility functions for each user, and β is the step of the method.

1: $\boldsymbol{\lambda}^0 := 0$

2: **for** $t = 1, \ldots, N-1$

3: Compute $\nabla\varphi(\boldsymbol{\lambda}^{t-1}, \xi)$

4: $\lambda^t := [\lambda^{t-1} - \beta(\mathbf{b} - nC_{\xi^{t-1}} x_{\xi^{t-1}}(\lambda^{t-1}))]_+$

5: $\hat{\mathbf{x}}^{t+1} := \frac{1}{t+1}\sum_{j=0}^{t} \mathbf{x}(\lambda^j)$

6: $\hat{\lambda}^{t+1} := \frac{1}{t+1}\sum_{j=0}^{t} \lambda^j$

7: **end for**

8: **return** $\hat{\lambda}^N$, $\hat{\mathbf{x}}^N$

Therefore, the stochastic subgradient is an unbiased estimator of the subgradient.

An optimal solution of problem (2) is sought using PDSPSGM. We describe two versions of this method (see Algorithms 2 and 3). Algorithm 2 relies on a complete model of reconstructing the vector $\mathbf{x}(\lambda)$ at every iteration. However, the computation of $\mathbf{x}(\lambda)$ is nearly equivalent in complexity to the computation of a complete subgradient of $\varphi(\lambda)$. Therefore, the basic algorithm is Algorithm 3, in which the vector $\mathbf{x}(\lambda)$ is reconstructed using an incomplete stochastic model, which means that only one component of the vector $\mathbf{x}(\lambda)$ is updated at every iteration step, while the others remain unchanged. In the proof of the convergence theorem, we first establish the convergence estimate for Algorithm 2 and then show that the approximate solution of the primal problem produced by Algorithm 3 is close in accuracy to the solution obtained using Algorithm 2.

Algorithm 3. Primal-Dual Stochastic Projected Subgradient Method (PDSPSGM), Version 2

Input: $u_k(\mathbf{x})$, $k = 1,\ldots,n$, are concave utility functions for each user, and β is the step of the method.

1: $\lambda^0 := 0$

2: **for** $t = 1,\ldots,N-1$

3: Compute $\nabla\varphi(\lambda^{t-1}, \xi)$

4: $\lambda^t := [\lambda^{t-1} - \beta(\mathbf{b} - nC_{\xi^{t-1}} x_{\xi^{t-1}}(\lambda^{t-1}))]_+$

5: $\tilde{\mathbf{x}}_{\xi^t}^{t+1} := \frac{t}{t+1}\tilde{\mathbf{x}}_{\xi^t}^{t} + \frac{1}{t+1} n x_{\xi^t}(\lambda^t)$, $\tilde{\mathbf{x}}_j^{t+1} := \tilde{\mathbf{x}}_j^t$ for $j \neq \xi^t$

6: $\hat{\lambda}^{t+1} := \frac{1}{t+1}\sum_{j=0}^{t} \lambda^j$

7: **end for**

8: **return** $\hat{\lambda}^N$, $\tilde{\mathbf{x}}^N$

4.1. Distributed Method

Let us describe how the distributed version of the stochastic projected subgradient method can be applied for solving the problem under consideration.

The process occurring at the tth iteration for link j is as follows:

1. Given the information received from the links after the preceding iteration with index $t-1$, the random user ξ^t transmits data at the optimal rate

$$x_{\xi^t}(\lambda^{t+1}) = \arg\max_{x_{\xi^t} \in \mathbf{R}_+} \left(u_{\xi^t}(x_{\xi^t}) - x_{\xi^t} \sum_{j=1}^{m} \lambda_j^{t+1} C_{\xi^t}^j \right),$$

where, by the definition of the matrix C, the information required for the user is only from the links used by the user.

2. The link j computes the price for the next iteration based on the reaction of this user:

$$\lambda_j^{t+1} = [\lambda_j^t - \beta(b_j - nC_{\xi^t}^j x_{\xi^t}^t)]_+.$$

Here, $C_{\xi^t}^j \neq 0$ only for users employing link j. Therefore, the price changes only for actual links of the user transmitting data.

Remark 2. The main advantage of this method is that the link changes the price relying only on the reactions of a single user, which makes the problem formulation much closer to the situation occurring in actual networks, where users do not transmit data simultaneously.

4.2. Estimation of the Convergence Rate of the Stochastic Projected Subgradient Method

Before proving the main theorem on convergence rate estimates for the proposed methods, we state the necessary assumptions for the problem under study. Assume that there exists a positive constant $M = O(n\sqrt{m})$ such that

$$\|\nabla\varphi(\lambda, \xi)\|_2 \leq M. \tag{8}$$

This assumption holds, since the data transmission rate $\mathbf{x}$ is bounded and the capacity vector $\mathbf{b}$ is bounded as well in view of the physical considerations. Therefore, by its definition, the stochastic subgradient is also bounded.

Additionally, we assume that

$$\mathbb{E}\left[\exp\left(\frac{\|\nabla\varphi(\lambda, \xi) - \nabla\varphi(\lambda)\|_2^2}{\sigma^2}\right)\right] \leq \exp(1),$$

where σ is a positive numerical constant and the order of dependence on n and m is the same as for M.

To estimate the convergence rate of Algorithm 3, it is necessary to assume that $u_k(x_k), k = 1, \ldots, n,$ are Lipschitz continuous functions with constant M_{u_k}. Then $U(\mathbf{x})$ is a Lipschitz continuous function with a constant M_U:

$$\forall \mathbf{x}, \mathbf{y} \quad |U(\mathbf{x}) - U(\mathbf{y})| \leq M_U \|\mathbf{x} - \mathbf{y}\|_2,$$

where $M_U = O(\sqrt{n})$. It may happen that $u_k(x_k)$ is a Lipschitz continuous function everywhere, except, for instance, the point 0. An example of such a function is $u_k(x_k) = \ln x_k$, which is one of the most widespread utility functions. However, by the specific features of the problem, there always exist $\overline{\varepsilon} > 0$ and $\underline{\varepsilon} > 0$ such that $x_k^* \geq \underline{\varepsilon}$ and $x_k^* \leq \overline{\varepsilon}$. Then the problem can be solved on the compact set $Q = \{\mathbf{x} : \underline{\varepsilon} \leq x_k \leq \overline{\varepsilon},\ k = 1, \ldots, n\}$, and the considered function $u_k(x_k) = \ln x_k$ becomes Lipschitz continuous on Q. In the general case, a concave utility function $u(x)$ is Lipschitz continuous on a compact set lying in the relative interior of the domain of $u(x)$.

Suppose that

$$\mathbb{E}\left[\exp\left(\frac{\|\mathbf{x}(\boldsymbol{\lambda}\lambda, \xi) - \mathbf{x}(\boldsymbol{\lambda})\|_2^2}{\sigma_x^2}\right)\right] \leq \exp(1),$$

where $\sigma_x = O(\sqrt{n})$ is a positive numerical constant and

$$\mathbf{x}(\boldsymbol{\lambda}, \xi) = (0, \ldots, nx_\xi(\boldsymbol{\lambda}), \ldots, 0)^{\mathrm{T}}.$$

Below is the key lemma necessary for obtaining convergence rate estimates for the residual in the constraints and the duality gap after running PDSPSGM.

Lemma 2. *Suppose that Algorithm* 3 *starts at the initial point* $\boldsymbol{\lambda}^0 = 0$ *with a step* β. *Then, after performing* N *iterations of Algorithm* 3, *with probability* $1 - 4\delta$,

$$\varphi(\hat{\boldsymbol{\lambda}}^N) - U(\tilde{\mathbf{x}}^N) + 2R\left\|[C\tilde{\mathbf{x}}^N - \mathbf{b}]_+\right\|_2 \leq C_1\frac{R^2\sigma\sqrt{g(N)J}}{\sqrt{N}} + \frac{2R^2}{\beta N} + \frac{\beta M^2}{2}$$
$$+ \frac{\sqrt{2}\left(1 + \sqrt{3\ln\frac{1}{\delta}}\right)}{\sqrt{N}}\left(M_U\sigma_x + 2R\left(\sigma + \sigma_x\sqrt{\lambda_{\max}(C^{\mathrm{T}}C)}\right)\right),$$

where

$$\hat{\boldsymbol{\lambda}}^N = \frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\lambda}^t,$$

$$\tilde{\mathbf{x}}^N = \frac{1}{N}\sum_{t=0}^{N-1}\mathbf{x}(\boldsymbol{\lambda}^t, \xi^t),$$

C_1 is a positive numerical constant, $g(N) = \ln\left(\frac{N}{\delta}\right) + \ln\ln\left(\frac{F}{f}\right)$,

$$F = 2\sigma^2 N(2\beta)^N\left(2R^2 + 2\beta^2M^2 + \beta R^2 + 24\ln\frac{N}{\delta}\beta\sigma^2 N\right),$$

$f = \sigma^2R^2$,

$$J = \max\left\{1, \quad \frac{1}{R}\beta C_1\sqrt{\sigma^2 g(N)} + \sqrt{\frac{1}{R^2}\beta^2C_1^2\sigma^2 g(N) + \frac{2R^2 + 2\beta^2M^2}{R^2}}\right\},$$

and R is determined by the condition $\|\boldsymbol{\lambda}^*\|_2 \le R$.

The proof of the lemma can be found in the Appendix.

Now we formulate a theorem on the convergence rate estimate for Algorithm 3.

Theorem 2. *Suppose that Algorithm* 3 *starts at the initial point $\boldsymbol{\lambda}^0 = 0$ with step $\beta = \frac{R}{M\sqrt{N}}$. Define*

$$A = \sqrt{2}\left(1 + \sqrt{3\ln\frac{1}{\delta}}\right)\left(M_U\sigma_x + 2R\left(\sigma + \sigma_x\sqrt{\lambda_{\max}(C^{\mathrm{T}}C)}\right)\right) + 2.5RM.$$

Then, after performing

$$N = O\left(\left\lceil\frac{A^2}{\varepsilon^2}\ln\left(\frac{MR}{\varepsilon\delta}\right)\right\rceil\right)$$

iterations of Algorithm 3, *with probability* $1 - 4\delta$,

$$U(\mathbf{x}^*) - U(\tilde{\mathbf{x}}^N) \le \varepsilon, \quad \left\|(C\tilde{\mathbf{x}}^N - \mathbf{b})_+\right\|_2 \le \frac{\varepsilon}{R},$$

where $\tilde{\mathbf{x}}^N = \frac{1}{N}\sum_{t=0}^{N-1}\mathbf{x}(\boldsymbol{\lambda}^t, \xi^t)$ and $\mathbf{x}^*$ is an optimal solution of problem (1).

Proof. The beginning of the proof is the same as for Theorem 1, but we use the estimate from Lemma 2. As a result, for the step $\beta = \frac{R}{M\sqrt{N}}$, we obtain

$$\frac{\sqrt{2}\left(1 + \sqrt{3\ln\frac{1}{\delta}}\right)}{\sqrt{N}}\left(M_U\sigma_x + 2R\left(\sigma + \sigma_x\sqrt{\lambda_{\max}(C^{\mathrm{T}}C)}\right)\right) + \frac{5RM}{2\sqrt{N}} + C_1\frac{R^2\sigma\sqrt{g(N)J}}{\sqrt{N}},$$

moreover, up to constants, $g(N) \approx \ln\left(\frac{N}{\delta}\right)$ and $J \approx \max\left\{1, \beta\sqrt{g(N)}\right\}$. Next, we find N for which the estimate becomes less than ε.

We introduce the following notation:

$$A = \sqrt{2}\left(1 + \sqrt{3\ln\frac{1}{\delta}}\right)\left(M_U\sigma_x + 2R\left(\sigma + \sigma_x\sqrt{\lambda_{\max}(C^{\mathrm{T}}C)}\right)\right) + 2.5RM,$$

$$B = C_1R^2\sigma.$$

It is necessary to obtain the minimum estimate on the iteration number N required for achieving the prescribed accuracy ε. For $J = 1$, we obtain

$$\sqrt{N} = \left\lceil\frac{A + B\sqrt{\ln\left(\frac{N}{\delta}\right)}}{\varepsilon}\right\rceil. \tag{9}$$

Substituting N recursively, we derive from (9) the complexity bound

$$N = O\left(\left\lceil \frac{A^2}{\varepsilon^2} \ln\left(\frac{MR}{\varepsilon\delta}\right)\right\rceil\right).$$

For $J = \beta\sqrt{g(N)} = \dfrac{R\sqrt{g(N)}}{M\sqrt{N}}$, we assume that

$$\frac{A}{\sqrt{N}} + \frac{Bg(N)R}{MN} = \frac{A}{\sqrt{N}} + \frac{\bar{B}g(N)}{N} \le \varepsilon.$$

Since the minimum N is needed, replacing the last inequality with equality and solving the resulting equation, we obtain

$$\sqrt{N} = \left\lceil \frac{A + \sqrt{A^2 + 4\varepsilon\bar{B}\ln\left(\dfrac{N}{\delta}\right)}}{2\varepsilon} \right\rceil.$$

By analogy with the case $J = 1$, this equality yields the estimate

$$N = O\left(\left\lceil \frac{A^2}{\varepsilon} \ln\left(\frac{MR}{\varepsilon\delta}\right)\right\rceil\right).$$

The worst of the complexity bounds for $J = 1$ and $J = \beta\sqrt{g(N)}$ is the estimate from the condition of the theorem.

5. ELLIPSOID METHOD

In this section, the original problem (1) is solved by applying the ellipsoid method [25]. This method can be used when the dual problem has a low dimension (m) or when high accuracy of the solution is required. The method is primal-dual, i.e., the solution of the primal problem can be recovered from the solution of the dual problem.

Consider the original problem (1) and its dual (2). As in the preceding section, the functions $u_k(x_k)$, $k = 1,\ldots,n$, are assumed to be concave, but not strongly concave. Additionally, we assume that the solution of the dual problem lies in the Euclidean ball of radius R centered at the origin, i.e., $\|\lambda^*\|_2 \le R$. As an initial point of the method, we use the zero vector $\lambda^0 = 0$. The problem is solved on the set

$$\Lambda_{2R} = \left\{\lambda \in \mathbb{R}^m_+ : \|\lambda\|_2 \le 2R\right\}.$$

Let us describe the ellipsoid method (Algorithm 4), which is used to solve the dual problem.

Algorithm 4. Ellipsoid Method

Input: $u_k(x_k), k = 1,\ldots,n$, are concave utility functions

1: $B_0 := 2R \cdot I_n$, I_n is the identity matrix
2: **for** $t = 0,\ldots,N-1$
3: Compute $\nabla\varphi(\lambda^t)$
4: $\mathbf{q}_t := B_t^{\mathrm{T}}\nabla\varphi(\lambda^t)$
5: $\mathbf{p}_t := \dfrac{B_t^{\mathrm{T}}\mathbf{q}_t}{\sqrt{\mathbf{q}_t^{\mathrm{T}} B_t B_t^{\mathrm{T}} \mathbf{q}_t}}$
6: $B_{t+1} := \dfrac{m}{\sqrt{m^2-1}} B_t + \left(\dfrac{m}{m+1} - \dfrac{m}{\sqrt{m^2-1}}\right) B_t \mathbf{p}_t \mathbf{p}_t^{\mathrm{T}}$
7: $\lambda^{t+1} := \lambda^t - \dfrac{1}{m+1} B_t \mathbf{p}_t$
8: **end for**
9: **return** λ^N

To reconstruct the solution of the primal problem from the solution of the dual one, it is necessary to determine the *accuracy certificate* ξ for the ellipsoid method. Recall that the accuracy certificate is a sequence of weights $\xi = \{\xi_t\}_{t=0}^{N-1}$ such that

$$\xi_t \geq 0, \quad \sum_{t=0}^{N-1} \xi_t = 1.$$

In our case, the accuracy certificate is constructed in the course of running the ellipsoid method (see Algorithm 5); its general scheme can be described as follows [26].

1. Find the "narrowest strip" containing the ellipsoid Q_N remaining after iteration N, i.e., a vector $\mathbf{h}$ such that the following inequality holds on Q_N:

$$\max_{\boldsymbol{\lambda} \in Q_N} \langle \mathbf{h}, \boldsymbol{\lambda} \rangle - \min_{\boldsymbol{\lambda} \in Q_N} \langle \mathbf{h}, \boldsymbol{\lambda} \rangle \leq 1. \tag{10}$$

For the ellipsoid method, all Q_N are represented in the form

$$Q_N = \{B_N \mathbf{z} + \lambda^N : \mathbf{z}^{\mathrm{T}} \mathbf{z} \leq 1\}.$$

Then, to solve (10), we need to perform a singular value decomposition $B_N = UDV$, where U and V are orthogonal matrices and D is a diagonal matrix with positive diagonal elements. Next, the desired vector $\mathbf{h}$ is determined as $\mathbf{h} = 1/(2\sigma^{i_*}) \cdot U\mathbf{e}^{i_*}$, where i_* is the index of the smallest diagonal element of D, σ^{i_*} is the value of this element, and $\mathbf{e}^i$ are the vectors of the standard basis.

2. For the vectors $\mathbf{h}^+ = \left[\mathbf{h}, -\langle \mathbf{h}, \lambda^N \rangle\right]$ and $\mathbf{h}^- = -\mathbf{h}^+$, find expansions of the form

$$\mathbf{h}^+ = \sum_{t=0}^{N-1} \nu_t \left[\nabla\varphi(\boldsymbol{\lambda}^t), -\langle \nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda}^t \rangle\right] + \mathbf{y}^+,$$

$$\mathbf{h}^- = \sum_{t=0}^{N-1} \mu_t \left[\nabla\varphi(\boldsymbol{\lambda}^t), -\langle \nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda}^t \rangle\right] + \mathbf{z}^+;$$

their existence follows from Proposition 4.1 in [26]. This step is described by Steps 6–13 in Algorithm 5 (see below).

3. From the expansion coefficients ν_t and μ_t of the vectors $\mathbf{h}^+$ and $\mathbf{h}^-$, respectively, derive expressions for ξ_t, $t \in I_N$, where

$$I_N = \{t \leq N-1 : \boldsymbol{\lambda}^t \in \operatorname{int} \Lambda_{2R}\}.$$

Expansion coefficients are determined only for feasible points obtained in the course of running Algorithm 5.

Algorithm 5. Construction of the Accuracy Certificate for the Ellipsoid Method

---

Input: $N-1$ is the number of the iteration at which the accuracy certificate is computed, and $\left\{B_t, \boldsymbol{\lambda}^t, \nabla\varphi(\boldsymbol{\lambda}^t)\right\}_{t=0}^{N-1}$ is the work protocol of the ellipsoid method after N iterations

1: **if** $\nabla\varphi(\boldsymbol{\lambda}^{N-1}) = 0$, **then**
2: $\quad \xi_t := 0$ for all $t = 0, \ldots, N-2$
3: $\quad \xi_{N-1} := 1$
4: **otherwise**
5: $\quad \mathbf{h} := 1/(2\sigma^{i_*}) \cdot U\mathbf{e}^{i_*}$
6: $\quad \mathbf{g}_\nu := \mathbf{h}$, $\mathbf{g}_\mu := -\mathbf{h}$
7: $\quad$ **for** $t = 0, \ldots, N-1$
8: $\quad\quad \mathbf{q} := B_t^{\mathrm{T}} \nabla\varphi(\boldsymbol{\lambda}^t)$

$\nu_t := [\mathbf{g}_\nu^{\mathrm{T}} B_t \mathbf{q}]_+ / \|\mathbf{q}\|_2^2$

$\mathbf{g}_\nu := \mathbf{g}_\nu - \nu_t \nabla\varphi(\lambda^t)$

$\mu_t := [\mathbf{g}_\mu^{\mathrm{T}} B_t \mathbf{q}]_+ / \|\mathbf{q}\|_2^2$

$\mathbf{g}_\mu := \mathbf{g}_\mu - \mu_t \nabla\varphi(\lambda^t)$

end for

$\xi_t := (\nu_t + \mu_t) / \sum_{i \in I_N} (\nu_i + \mu_i)$, $t \in I_N$

end if

return $\{\xi_t\}_{t=0}^{N-1}$

Remark 3. In contrast to FGM and the stochastic projected subgradient method, the computation of Steps 4–6 of Algorithm 4 in the ellipsoid method requires information about all gradient components, i.e., information from all users. Accordingly, it is necessary to have a common center for all links that collects information from them and performs these computations.

An estimate for the convergence rate of the ellipsoid method for the problem under study is provided by the following result.

Theorem 3 (see [26]). *Suppose that Algorithm* 4 *starts from the initial ball* $B_0 = \{\lambda \in \mathbb{R}^m : \|\lambda\| \le 2R\}$ *and the accuracy certificate* ξ *is produced by Algorithm* 5. *Then, after performing*

$$N = 2m(m+1)\left\lceil \ln\left(\frac{32 \cdot 4MR}{\varepsilon}\right)\right\rceil \tag{11}$$

iterations, it is true that

$$U(\mathbf{x}^*) - U(\hat{\mathbf{x}}^N) \le \varepsilon, \quad \left\|[C\hat{\mathbf{x}}^N - \mathbf{b}]_+\right\|_2 \le \frac{\varepsilon}{R},$$

where

$$\hat{\mathbf{x}}^N = \sum_{t \in I_N} \xi_t \mathbf{x}(\lambda^t), \quad I_N = \left\{t \le N-1 : \lambda^t \in \operatorname{int} \Lambda_{2R}\right\}.$$

The proof of this theorem can be found in the Appendix.

6. REGULARIZATION OF THE DUAL PROBLEM

In previous sections, we considered primal-dual methods for solving the dual problem. However, there is a standard approach in which the solution of the primal problem can be recovered from the solution of the dual problem without using primal-dual methods. The key idea of this approach is a regularization of the dual problem such that the resulting regularized problem is strongly convex. In what follows, we describe this approach in detail and state lemmas relating the solutions of the primal and dual problems.

Functional (2) is regularized in the sense of Tikhonov:

$$\varphi_\delta(\lambda) = \varphi(\lambda) + \frac{\delta}{2}\|\lambda\|_2^2$$

and, instead of problem (4), we solve the regularized problem

$$\min_{\lambda \in \mathbf{R}_+^m} \varphi_\delta(\lambda).$$

An optimal parameter δ will be specified later. As in Section 5, we assume that the problem is solved on the set

$$\Lambda_{2R} = \{\lambda \in \mathbb{R}_+^m : \|\lambda\|_2 \le 2R\}.$$

For the resulting regularized function, we formulate the following lemma on the smoothness of the regularized problem.

Lemma 3. *Suppose that the function* $\varphi(\boldsymbol{\lambda})$ *is* L*-smooth. Then the regularized function* $\varphi_\delta(\boldsymbol{\lambda})$ *is* $(L+\delta)$*-smooth, i.e., for any* $\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2 \in \mathbf{R}^m_+$,

$$\left\|\nabla\varphi_\delta(\boldsymbol{\lambda}^1) - \nabla\varphi_\delta(\boldsymbol{\lambda}^2)\right\|_2 \le (L+\delta)\left\|\boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^2\right\|_2. \tag{12}$$

Proof. The gradient of the regularized function is given by

$$\nabla\varphi_\delta(\boldsymbol{\lambda}) = \nabla\varphi(\boldsymbol{\lambda}) + \delta\boldsymbol{\lambda}.$$

Therefore, we have

$$\left\|\nabla\varphi_\delta(\boldsymbol{\lambda}^1) - \nabla\varphi_\delta(\boldsymbol{\lambda}^2)\right\|_2 = \left\|\nabla\varphi(\boldsymbol{\lambda}^1) - \nabla\varphi(\boldsymbol{\lambda}^2) + \delta(\boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^2)\right\|_2 \le \left\|\nabla\varphi(\boldsymbol{\lambda}^1) - \nabla\varphi(\boldsymbol{\lambda}^2)\right\|_2 + \delta\left\|\boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^2\right\|_2.$$

By Proposition 2, this estimate implies (12).

Additionally, to estimate the convergence of the algorithm for the primal problem, we need the following auxiliary lemma concerning the relationship between the gradient estimate for the dual problem and convergence estimates with respect to the function and the residual in the constraint for the primal problem.

Lemma 4 (see [10]). *Let* $\mathbf{x}^*$ *be a solution of primal problem* (1). *Then*

$$\left\|C\mathbf{x}(\boldsymbol{\lambda}) - \mathbf{b}\right\|_2 \le \left\|\nabla\varphi_\delta(\boldsymbol{\lambda})\right\|_2 + \delta\left\|\boldsymbol{\lambda}\right\|_2, \tag{13}$$

$$U(\mathbf{x}^*) - U(\mathbf{x}(\boldsymbol{\lambda})) \le \left\|\nabla\varphi_\delta(\boldsymbol{\lambda})\right\|_2 \cdot \left\|\boldsymbol{\lambda}\right\|_2 + \delta\left\|\boldsymbol{\lambda}\right\|_2^2, \tag{14}$$

where $\mathbf{x}(\boldsymbol{\lambda})$ *is defined by* (3).

Proof. By virtue of (3), we have

$$U(\mathbf{x}(\lambda)) + \left\langle\lambda, \mathbf{b} - C\mathbf{x}(\lambda)\right\rangle \ge U(\mathbf{x}^*) + \left\langle\lambda, \mathbf{b} - C\mathbf{x}^*\right\rangle \ge U(\mathbf{x}^*),$$

whence

$$U(\mathbf{x}(\boldsymbol{\lambda})) \ge U(\mathbf{x}^*) - \left\langle\boldsymbol{\lambda}, \mathbf{b} - C\mathbf{x}(\boldsymbol{\lambda})\right\rangle = U(\mathbf{x}^*) - \left\langle\boldsymbol{\lambda}, \nabla\varphi(\boldsymbol{\lambda})\right\rangle.$$

Since $\varphi(\boldsymbol{\lambda}) = \varphi_\delta(\boldsymbol{\lambda}) - \dfrac{\delta}{2}\|\boldsymbol{\lambda}\|_2^2$, it is true that

$$\left\|\nabla\varphi(\boldsymbol{\lambda})\right\|_2 = \left\|\nabla\varphi_\delta(\boldsymbol{\lambda}) - \delta\boldsymbol{\lambda}\right\|_2 \le \left\|\nabla\varphi_\delta(\boldsymbol{\lambda})\right\| + \delta\left\|\boldsymbol{\lambda}\right\|_2.$$

Combining this inequality with the relation $\nabla\varphi(\boldsymbol{\lambda}) = \mathbf{b} - C\mathbf{x}(\boldsymbol{\lambda})$ yields (13).

Furthermore, estimate (14) follows from

$$U(\mathbf{x}_*) - U(\mathbf{x}(\boldsymbol{\lambda})) \le \left\langle\boldsymbol{\lambda}, \nabla\varphi(\boldsymbol{\lambda})\right\rangle \le \left\|\nabla\varphi(\boldsymbol{\lambda})\right\|_2 \cdot \left\|\boldsymbol{\lambda}\right\|_2$$
$$\le \left\|\boldsymbol{\lambda}\right\|_2 \cdot \left(\left\|\nabla\varphi_\delta(\boldsymbol{\lambda})\right\|_2 + \delta\left\|\boldsymbol{\lambda}\right\|_2\right) \le \left\|\nabla\varphi_\delta(\boldsymbol{\lambda})\right\|_2 \cdot \left\|\boldsymbol{\lambda}\right\|_2 + \delta\left\|\boldsymbol{\lambda}\right\|_2^2.$$

Additionally, we need the following result concerning convergence with respect to the gradient of the regularized function.

Lemma 5. *Let* $\boldsymbol{\lambda}^*_\delta$ *be a solution of the regularized dual problem. Then*

$$\left\|\nabla\varphi_\delta(\boldsymbol{\lambda}^N)\right\|_2 \le (L+\delta)\left\|\boldsymbol{\lambda}^N - \boldsymbol{\lambda}^*_\delta\right\|_2.$$

The proof follows immediately from Lemma 3 and the relation

$$\nabla\varphi_\delta(\boldsymbol{\lambda}^*_\delta) = 0.$$

We have formulated the lemmas necessary for the regularized problem. An example of applying this approach is considered in the next section.

7. RANDOM GRADIENT EXTRAPOLATION METHOD

Consider the random gradient extrapolation method [27]. Note that this method does not require updating the gradient at every iteration step. It is necessary to update only one of its components at every iteration, which considerably reduces the computations, especially for large-scale problems. Since this method is not primal-dual, Algorithm 6 has to be applied to the regularized problem.

The parameters α, η, τ, and θ_t are specified as

$$\bar{\alpha} = 1 - \frac{1}{n + \sqrt{n^2 + 16nL/\delta}}, \tag{15}$$

$$\alpha = n\bar{\alpha}, \quad \eta = \frac{\delta\bar{\alpha}}{1-\bar{\alpha}}, \quad \tau = \frac{1}{n(1-\bar{\alpha})} - 1, \quad \theta_t = \bar{\alpha}^{-t}. \tag{16}$$

7.1. Distributed Method

This section presents a distributed version of the considered method. By way of introduction, we note that the vectors $\underline{\boldsymbol{\lambda}}_1^0, \ldots, \underline{\boldsymbol{\lambda}}_n^0$ are stored by the corresponding users and influence the formation of optimal data traffic for the corresponding user. As was noted in the description of the distributed FGM, the optimal traffic for a user is influenced only by the prices of the links through which this user exchanges packets. Therefore, we can assume that the only nonzero components in the vector $\underline{\boldsymbol{\lambda}}_k^t$ are those whose indices coincide with the indices of the used links.

Algorithm 6. Random Gradient Extrapolation Method (RGEM)

Input: Parameters α, η, τ, $\{\theta_t\}_{t=1}^N$

$\boldsymbol{\lambda}^0 := \mathbf{0}$

$\underline{\boldsymbol{\lambda}}_i^0 := \boldsymbol{\lambda}^0$, $i = 1, \ldots, n$

$y_{-1} = y_0 = \mathbf{0}$

for $t = 1, \ldots, N$

  Choose k_t at random from the set $\{1, \ldots, n\}$ uniformly over all values

  $\tilde{\mathbf{y}}_k^t := \mathbf{y}_k^{t-1} + \alpha(\mathbf{y}_k^{t-1} - \mathbf{y}_k^{t-2})$, $k = 1, \ldots, n$

  $\boldsymbol{\lambda}^t := \left[\eta\boldsymbol{\lambda}^{t-1} - \frac{1}{n}\sum_{k=1}^n \tilde{\mathbf{y}}_k^t\right]_+ /(\delta + \eta)$

  $\underline{\boldsymbol{\lambda}}_{k_t}^t := (\boldsymbol{\lambda}^t + \tau\underline{\boldsymbol{\lambda}}_{k_t}^{t-1})/(1+\tau)$

  $\underline{\boldsymbol{\lambda}}_k^t := \underline{\boldsymbol{\lambda}}_k^{t-1}$, $k \in \{1, \ldots, n\}\backslash\{k_t\}$

  $\mathbf{y}_{k_t}^t := \mathbf{b} - n\mathbf{C}_{k_t} x_{k_t}(\underline{\boldsymbol{\lambda}}_{k_t}^t)$

  $\mathbf{y}_k^t := \mathbf{y}_k^{t-1}$, $k \in \{1, \ldots, n\}\backslash\{k_t\}$

end for

$\bar{\boldsymbol{\lambda}}^N := \left(\sum_{t=0}^{N-1} \theta_t \boldsymbol{\lambda}^t\right) / \sum_{t=1}^N \theta_t$

return $\bar{\boldsymbol{\lambda}}^N$

Let us describe the distributed algorithm at the tth iteration.

1. Using information collected from the users at the preceding iteration, link j computes

$$\tilde{y}_{k,j}^t := y_{k,j}^{t-1} + \alpha(y_{k,j}^{t-1} - y_{k,j}^{t-2}) = b_j - nC_k^j x_k(\underline{\boldsymbol{\lambda}}_k^{t-1}) + \alpha(nC_k^j x_k(\underline{\boldsymbol{\lambda}}_k^{t-2}) - nC_k^j x_k(\underline{\boldsymbol{\lambda}}_k^{t-1})).$$

Note that, by the definition of the matrix C, link j needs information only from the users exchanging packets through this link.

2. The price of link j changes according to the rule

$$\lambda_j^t = \left[\eta\lambda_j^{t-1} - \frac{1}{n}\sum_{k=1}^n \tilde{y}_{k,j}^t\right]_+ /(\delta + \eta).$$

3. One of the users, k_t, reacts to the price change and stores the local price vector

$$\underline{\boldsymbol{\lambda}}_{k_t}^{t} = (\boldsymbol{\lambda}^{t} + \tau\underline{\boldsymbol{\lambda}}_{k_t}^{t-1})/(1+\tau),$$

while the local prices for the other users remain unchanged, i.e., $\underline{\boldsymbol{\lambda}}_{k_t}^{t} = \underline{\boldsymbol{\lambda}}_{k_t}^{t-1}$.

4. The user k_t computes

$$x_{k_t}^{t}(\underline{\lambda}_{k_t,j}^{t}) = \arg\max_{x_k \in \mathbf{R}_+}\left(u_k(x_k) - x_k\sum_{j=1}^{m}\underline{\lambda}_{k_t,j}^{t}C_k^j\right)$$

and transmits this information to the used links.

5. Link j updates the information for the user k_t:

$$y_{k_t,j}^{t} = b_j - nC_{k_t}^{j}x_{k_t}(\underline{\boldsymbol{\lambda}}_{k_t}^{t}).$$

This information is updated by the link only if the user k_t exchanges packets through it.

7.2. Estimation of the Convergence Rate of RGEM

Following Section 3, we consider problem (2) with μ-strongly concave cost functions $u_k(x_k), k = 1,\ldots,n$. Recall that, since the cost functions are strongly concave, the dual problem (4) is smooth with Lipschitz constant $L = \frac{nm^2}{\mu}$.

To estimate the convergence rate of the method, we need the estimate for the residual with respect to the argument from Theorem 2.1 in [27], namely,

$$\mathbb{E}\left[\left\|\boldsymbol{\lambda}_{\delta}^{*} - \boldsymbol{\lambda}^{N}\right\|_2^2\right] \le \frac{4\Delta(\bar{\alpha})^N}{\delta}, \tag{17}$$

where $\Delta = \delta\left\|\boldsymbol{\lambda}_{\delta}^{*} - \boldsymbol{\lambda}^{0}\right\|_2^2 + \frac{B}{n\delta} + \varphi_\delta(\lambda^0) - \varphi_\delta(\lambda_\delta^*)$ and $B = \|\mathbf{b}\|_2^2$.

By using (17), it is possible to prove the following convergence estimate theorem for the method as applied to problem (11).

Theorem 4. *Suppose that the regularized dual problem* (11) *is solved by applying RGEM with parameters* (15), (16), *and* $\delta = \frac{\varepsilon}{8R^2}$ *and with*

$$N = \left\lceil 2\left(n + \sqrt{n^2 + \frac{128nLR^2}{\varepsilon}}\right)\ln\left(\frac{4RA}{\varepsilon}\right)\right\rceil$$

iterations, where $A = 2\left(LR + \frac{\varepsilon}{8R}\right)\sqrt{6 + \frac{16LR^2n + 8B}{n\varepsilon}}$. *Then*

$$\mathbb{E}[U(\mathbf{x}^*) - U(\mathbf{x}(\boldsymbol{\lambda}^N))] \le \varepsilon, \quad \mathbb{E}\left[\left\|C\mathbf{x}(\boldsymbol{\lambda}^N) - \mathbf{b}\right\|_2\right] \le \frac{\varepsilon}{2R}.$$

Proof. Lemma 4 implies estimate (13) for the residual with respect to the constraints and estimate (14) for the residual with respect to the objective function. By the assumption $\boldsymbol{\lambda} \in \Lambda_{2R}$, we have

$$\left\|C\mathbf{x}^N - \mathbf{b}\right\|_2 \le \left\|\nabla\varphi_\delta(\boldsymbol{\lambda}^N)\right\|_2 + 2\delta R, \tag{18}$$

$$U(\mathbf{x}^*) - U(\mathbf{x}^N) \le 2R\left\|\nabla\varphi_\delta(\boldsymbol{\lambda}^N)\right\|_2 + 4\delta R^2, \tag{19}$$

where $\mathbf{x}^N = \mathbf{x}(\boldsymbol{\lambda}^N)$. Combining Lemma 5 with inequality (17) yields the following estimate for $\left\|\nabla\varphi_\delta(\boldsymbol{\lambda}^N)\right\|_2$:

$$\mathbb{E}\left[\left\|\nabla\varphi_\delta(\boldsymbol{\lambda}^N)\right\|_2\right] \le 2(L+\delta)\sqrt{\frac{\Delta}{\delta}}(\bar{\alpha})^{N/2}.$$

Let us estimate Δ. The function φ_δ with a Lipschitz continuous gradient satisfies the inequality

$$\varphi_\delta(\boldsymbol{\lambda}^0) - \varphi_\delta(\boldsymbol{\lambda}_\delta^*) \le \left\langle \nabla\varphi_\delta(\boldsymbol{\lambda}_\delta^*), \boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^* \right\rangle + \frac{L+\delta}{2}\left\|\boldsymbol{\lambda}_\delta^* - \boldsymbol{\lambda}^0\right\|_2^2.$$

Since $\nabla\varphi_\delta(\boldsymbol{\lambda}_\delta^*) = 0$, we obtain

$$\Delta \le \delta\left\|\boldsymbol{\lambda}_\delta^* - \boldsymbol{\lambda}^0\right\|_2^2 + \frac{B}{n\delta} + \frac{L+\delta}{2}\left\|\boldsymbol{\lambda}_\delta^* - \boldsymbol{\lambda}^0\right\|_2^2 \le (6\delta + 2L)R^2 + \frac{B}{n\delta}.$$

Suppose that δ is chosen so that $4\delta R^2 = \frac{\varepsilon}{2}$. Then $\delta = \frac{\varepsilon}{8R^2}$. It follows that

$$4\delta R^2 = \frac{\varepsilon}{2}, \quad 2\delta R = \frac{\varepsilon}{4R}.$$

Assume that $U(\mathbf{x}^*) - U(\mathbf{x}(\boldsymbol{\lambda}^N)) \le \varepsilon$. Then, by virtue of (18) and (19), it is true that

$$\left\|\nabla\varphi_\delta(\boldsymbol{\lambda}^N)\right\|_2 \le \frac{\varepsilon}{4R},$$

whence

$$2(L+\delta)\sqrt{\frac{\Delta}{\delta}}(\bar{\alpha})^{N/2} \le \frac{\varepsilon}{4R}.$$

Taking into account

$$2(L+\delta)\sqrt{\frac{\Delta}{\delta}} \le 2\left(LR + \frac{\varepsilon}{8R}\right)\sqrt{6 + \frac{16LR^2n + 8B}{n\varepsilon}},$$

we obtain the following estimate for the number of iterations:

$$N = \left\lceil 2\left(n + \sqrt{n^2 + \frac{128nLR^2}{\varepsilon}}\right)\ln\left(\frac{4RA}{\varepsilon}\right)\right\rceil,$$

where $A = 2\left(LR + \frac{\varepsilon}{8R}\right)\sqrt{6 + \frac{16LR^2n + 8B}{n\varepsilon}}$.

Remark 4. The complexity bound for Algorithm 6 can also be represented in the form $O\left(\max\left\{n, \sqrt{nLR^2/\varepsilon}\right\}\ln\left(\frac{1}{\varepsilon}\right)\right)$, where the logarithmic factor appears due to the necessity of regularization of the dual problem. At every iteration, only one component of the user reaction vector to changed prices is computed; accordingly, the arithmetic complexity of the operation is better than that in the case of computing all components of these vectors. For FGM, the assumptions made about the objective function are similar, but, since the complete gradient has to be computed at every iteration step, the complexity bound for the algorithm is $O\left(n\sqrt{LR^2/\varepsilon}\right)$. Thus, although the theoretical convergence estimate for RGEM has the same order as for FGM, in practice the gain is obtained due to the cheaper computations within a single iteration.

8. NUMERICAL EXPERIMENTS

The software code for numerical experiments was written in Python 3.6 and C++14. The source code for experiments and the methods considered in this paper is available at https://github.com/dmivilensky/network-resource-allocation. The running time was measured on a computer with a 2-core Intel Core i5-5250U 1.6 GHz processor and 8 GB RAM.

8.1. Strongly Convex (Quadratic) Utility Functions

Consider problem (1) for utility functions of the form

$$u_k(x_k) = a_k x_k - \frac{\sigma n}{2}x_k^2, \quad a_k \sim \mathcal{U}(0,100), \quad \sigma = 0.1,$$

Table 1. Comparison of the number of iterations and the running time of FGM and RGEM for strongly convex (quadratic) utility functions

Network	FGM		RGEM	
	iterations	time	iterations	time
$m = 2$, $n = 1500$, $\varepsilon = 10^{-2}$	350	24.5 s	3000	21.1 s
$m = 5$, $n = 1500$, $\varepsilon = 10^{-2}$	380	42.7 s	6700	36.9 s
$m = 70$, $n = 5000$, $\varepsilon = 10^{-2}$	400	150.0 s	7800	132.6 s
$m = 70$, $n = 5000$, $\varepsilon = 10^{-3}$	1070	374.5 s	9180	283.7 s
$m = 100$, $n = 5000$, $\varepsilon = 10^{-2}$	417	175.1 s	8200	164.0 s
$m = 70$, $n = 7000$, $\varepsilon = 10^{-2}$	421	218.9 s	8600	206.4 s
$m = 100$, $n = 7000$, $\varepsilon = 10^{-2}$	427	290.3 s	9200	276.0 s
$m = 100$, $n = 7000$, $\varepsilon = 10^{-3}$	1120	761.6 s	10130	638.2 s

where a_k are independent identically distributed random variables. Then problem (3) can be solved explicitly:

$$\mathbf{x}(\lambda) = \frac{[\mathbf{a} - C\lambda]_+}{n\sigma}.$$

For a small number of users ($n = 1500$), the link capacities are chosen identical (in this case, $\mathbf{b} = (5,\ldots,5)^{\mathrm{T}}$), and the demand for data transmission is uniform ($c_{ij} = 1$ for any i, j). For a larger number of users, the capacity vector is generated at random, so that $b_i \sim \mathcal{U}(1,6)$. The elements of the demand matrix are also chosen randomly and independently, so that $c_{ij} = 1$ with probability $p = 0.5$ and $c_{ij} = 0$ with probability $q = 0.5$.

Table 1 presents the number of iterations and the running times of the fast gradient method (FGM) and the random gradient extrapolation method (RGEM) for various network configurations (with m links), various numbers of users n, and various values of the required accuracy ε. The cases in which RGEM converges to the solution faster than FGM, despite the larger number of iterations than in FGM, are highlighted in the table. Indeed, for $n \gg 0$, RGEM requires fewer queries for the optimal solution $x_k(\lambda)$ from users than in other algorithms, since a query at one RGEM iteration is sent to only one random user.

8.2. Convex (Logarithmic) Utility Functions

Consider the performance of the stochastic subgradient method (Algorithm 2) and the ellipsoid method (Algorithm 4) for the utility function

$$u_k(x_k) = \ln x_k.$$

In this case, an explicit solution of problem (3) is given by

$$\mathbf{x}(\lambda) = \frac{1}{C\lambda}$$

(the operation $1/\cdot$ as applied to a vector is understood elementwise). For a small number of users ($n = 1500$), the link capacities are chosen identical (in this case, $\mathbf{b} = (5,\ldots,5)^{\mathrm{T}}$) and the demand for data transmission is uniform ($c_{ij} = 1$ for any i, j). For a larger number of users, the capacity vector is randomly generated, so that $b_i \sim \mathcal{U}(1,6)$. The elements of the demand matrix are also chosen randomly and independently, so that $c_{ij} = 1$ with probability $p = 0.5$ and $c_{ij} = 0$ with probability $q = 0.5$.

Table 2 presents the number of iterations and the running times of the stochastic subgradient method (SGM) and the ellipsoid method for various network configurations, various numbers of users, and various values of the required accuracy. The cases in which SGM converges to the solution faster than the ellipsoid method are highlighted in the table.

Table 2. Comparison of the number of iterations and the running time of the stochastic subgradient method and the ellipsoid method for convex (logarithmic) utility functions

Network	Ellipsoid method		SGM	
	iterations	time	iterations	time
$m = 2$, $n = 1500$, $\varepsilon = 10^{-2}$	40	0.02 s	2000	0.2 s
$m = 5$, $n = 1500$, $\varepsilon = 10^{-2}$	85	0.06 s	2500	0.3 s
$m = 70$, $n = 5000$, $\varepsilon = 10^{-2}$	120	1.9 s	4000	1.3 s
$m = 70$, $n = 5000$, $\varepsilon = 10^{-3}$	800	5.4 s	9020	2.4 s
$m = 100$, $n = 5000$, $\varepsilon = 10^{-2}$	300	9.0 s	5000	3.1 s
$m = 70$, $n = 7000$, $\varepsilon = 10^{-2}$	250	8.7 s	5590	5.5 s
$m = 100$, $n = 7000$, $\varepsilon = 10^{-2}$	380	19.0 s	6480	10.8 s
$m = 100$, $n = 7000$, $\varepsilon = 10^{-3}$	1830	91.5 s	17970	30.6 s

Note that, as in RGEM, only one component of the user reaction vector $\mathbf{x}(\boldsymbol{\lambda})$ to established prices has to be computed at every iteration in SGM. Thus, when the number of iterations of the method is large, the number of computed components $x_k(\boldsymbol{\lambda}^t)$ is smaller than in other algorithms, for example, in the ellipsoid method, and the same is true of the communication complexity in the case of distributed implementation.

9. CONCLUSIONS

To conclude, we note some possible directions of development of this work and briefly describe suitable methods without detailed analysis of their convergence estimates.

In Section 5, as applied to low-dimensional problems, we considered the ellipsoid method, which is primal-dual. There are other methods that are highly accurate and well suited for low-dimensional problems. An example is Vaidya's cutting plane method [28]. However, to recover the solution of the primal problem when the dual one is solved using Vaidya's method, we need convergence in the gradient norm for the dual problem. For this purpose, the dual problem has to be smooth, which is ensured by the strong convexity of the objective function in the primal problem (Proposition 2). If the primal problem is not strongly convex, it can be regularized as described in Section 6, but the convergence estimate will then degrade logarithmically.

Additionally, if the dual problem is sufficiently smooth, it can be solved by applying high-order methods [29, 30]. The steps of these methods can be computed on a distributed basis, since the given problem makes use of a centralized architecture in terms of the interaction of a link and the users using it. Note, however, that high-order optimal methods that require linesearch and do not have the primal-dual property apply to only preliminarily regularized dual problems.

Another direction is represented by variance reduced methods (see, e.g., [31, 32]), which are intermediate between the stochastic gradient method and FGM. However, these methods are not primal-dual either, so they apply to preliminarily regularized dual problems.

Of special interest are the Hogwild! method [33] and minibatching techniques. In this case, data are sent out not by all users simultaneously, but by more than one of them, in contrast to stochastic methods. By setting the size of the batch equal to the number of users transmitting data at a time, one can take into account the specific features of actual networks.

APPENDIX

Auxiliary Results

Below are some lemmas from other works that are used in the proofs. Additionally, we prove assertions concerning the properties of the dual function that are used in the proof of the main theorems.

Lemma 6 (see [34], Lemma 2). *For a random vector $\xi \in \mathbf{R}^n$, the following assertions are equivalent up to a constant multiplying σ:*

1. *Tails*: $\mathbb{P}\{\|\xi\|_2 \geq \gamma\} \leq 2\exp\left(-\dfrac{\gamma^2}{2\sigma^2}\right) \ \forall \gamma \geq 0$.

2. *Moments*: $(\mathbb{E}[\xi^p])^{1/p} \le \sigma\sqrt{p}$ *for any positive integer* p.

3. *Light-tail assumption*: $\mathbb{E}\left[\exp\left(\frac{\|\xi\|_2^2}{\sigma^2}\right)\right] \le \exp(1)$.

Lemma 7 (see [34, Corollary 8]). *Let* $\{\xi^k\}_{k=1}^N$ *be a sequence of random vectors from* $\mathbf{R}^n$ *such that for* $k = 1,\ldots,N$ *and any* $\gamma \ge 0$,

$$\mathbb{E}[\xi^k \,|\, \xi^1,\ldots,\xi^{k-1}] = 0, \quad \mathbb{E}\left[\left\|\xi^k\right\|_2 \ge \gamma | \xi^1,\ldots,\xi^{k-1}\right] \le \exp\left(-\frac{\gamma^2}{2\sigma_k^2}\right) \quad \textit{almost surely},$$

where σ_k^2 *belongs to* $\sigma(\xi^1,\ldots,\xi^{k-1})$ *for all* $k = 1,\ldots,N$. *Let* $S_N = \sum_{k=1}^N \xi^k$. *Then there exists a constant* C_1 *such that, for any fixed* $\delta > 0$ *and* $B > b > 0$, *with probability* $1 - \delta$

$$\textit{either} \quad \sum_{k=1}^N \sigma_k^2 \ge B$$

$$\textit{or} \quad \|S_N\|_2 \le C_1\sqrt{\max\left\{\sum_{k=1}^N \sigma_k^2, b\right\}\left(\ln\frac{2n}{\delta} + \ln\ln\frac{B}{b}\right)}.$$

Lemma 8 (see [35, corollary to Theorem 2.1, case (ii)]). *Suppose that a sequence* $\{\xi^k\}_{k=1}^N$ *of random vectors from* $\mathbf{R}^n$ *satisfies the condition*

$$\mathbb{E}[\xi^k \,|\, \xi^1,\ldots,\xi^{k-1}] = 0 \quad \textit{almost surely}, \quad k = 1,\ldots,N,$$

and let $S_N = \sum_{k=1}^N \xi^k$. *Assume that the sequence*$\{\xi^k\}_{k=1}^N$ *satisfies the light-tail assumption*

$$\mathbb{E}\left[\exp\left(\frac{\left\|\xi^k\right\|_2^2}{\sigma_k^2}\right) | \xi^1,\ldots,\xi^{k-1}\right] \le \exp(1) \quad \textit{almost surely}, \quad k = 1,\ldots,N,$$

where $\sigma_1,\ldots,\sigma_N$ *are positive numbers. Then, for all* $\gamma \ge 0$,

$$\mathbb{P}\left\{\|S_N\| \ge (\sqrt{2} + \sqrt{2\gamma})\sqrt{\sum_{k=1}^N \sigma_k^2}\right\} \le \exp\left(-\frac{\gamma^2}{3}\right).$$

Proof of Proposition 2. The dual function is represented in the form

$$\varphi(\boldsymbol{\lambda}) = \sum_{k=1}^n \left\{u_k(x_k(\boldsymbol{\lambda})) - \langle \boldsymbol{\lambda}, \mathbf{C}_k\rangle x_k(\boldsymbol{\lambda}) + \frac{1}{n}\langle \boldsymbol{\lambda}, \mathbf{b}\rangle\right\} = \sum_{k=1}^n \varphi_k(\boldsymbol{\lambda}).$$

Proposition 1 implies that

$$\nabla\varphi(\boldsymbol{\lambda}) = \sum_{k=1}^n \nabla\varphi_k(\boldsymbol{\lambda}) = \sum_{k=1}^n \left(\frac{1}{n}\mathbf{b} - \mathbf{C}_k x_k(\boldsymbol{\lambda})\right).$$

Define

$$x_k(\boldsymbol{\lambda}^1) = \arg\max_{x_k \in \mathbf{R}_+} \left\{u_k(x_k) - x_k\left\langle \boldsymbol{\lambda}^1, \mathbf{C}_k\right\rangle\right\},$$
$$x_k(\boldsymbol{\lambda}^2) = \arg\max_{x_k \in \mathbf{R}_+} \left\{u_k(x_k) - x_k\left\langle \boldsymbol{\lambda}^2, \mathbf{C}_k\right\rangle\right\}.$$

The necessary maximum conditions of the first order are written as

$$\left\langle \nabla u_k(x_k(\boldsymbol{\lambda}^1)) - \left\langle \boldsymbol{\lambda}^1, \mathbf{C}_k\right\rangle, x_k(\boldsymbol{\lambda}^1) - x_k(\boldsymbol{\lambda}^2)\right\rangle \ge 0,$$
$$\left\langle \nabla u_k(x_k(\boldsymbol{\lambda}^2)) - \left\langle \boldsymbol{\lambda}^2, \mathbf{C}_k\right\rangle, x_k(\boldsymbol{\lambda}^2) - x_k(\boldsymbol{\lambda}^1)\right\rangle \ge 0.$$

Adding these inequalities yields

$$\left\langle \nabla u_k(x_k(\boldsymbol{\lambda}^2)) - \nabla u_k(x_k(\boldsymbol{\lambda}^1)), x_k(\boldsymbol{\lambda}^1) - x_k(\boldsymbol{\lambda}^2) \right\rangle \le \left\langle \left\langle \boldsymbol{\lambda}^2, \mathbf{C}_k \right\rangle - \left\langle \boldsymbol{\lambda}^1, \mathbf{C}_k \right\rangle, x_k(\boldsymbol{\lambda}^1) - x_k(\boldsymbol{\lambda}^2) \right\rangle.$$

Since $u_k(x_k)$ is strongly concave, for any x_k^1 and x_k^2, $k = 1,\ldots,n$, we have

$$\left\langle \nabla u_k(x_k^2) - \nabla u_k(x_k^1), x_k^1 - x_k^2 \right\rangle \ge \mu \left\| x_k^1 - x_k^2 \right\|_2^2,$$

whence

$$\mu \left\| x_k(\boldsymbol{\lambda}^1) - x_k(\boldsymbol{\lambda}^2) \right\|_2^2 \le \left\langle \left\langle \boldsymbol{\lambda}^2, \mathbf{C}_k \right\rangle - \left\langle \boldsymbol{\lambda}^1, \mathbf{C}_k \right\rangle, x_k(\boldsymbol{\lambda}^1) - x_k(\boldsymbol{\lambda}^2) \right\rangle \le \left\| \mathbf{C}_k \right\|_2 \cdot \left\| \boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^2 \right\|_2 \cdot \left\| x_k(\boldsymbol{\lambda}^1) - x_k(\boldsymbol{\lambda}^2) \right\|_2.$$

Then the following estimate can be obtained for all gradient components $\nabla \varphi_k$:

$$\left\| \nabla \varphi_k(\boldsymbol{\lambda}^1) - \nabla \varphi_k(\boldsymbol{\lambda}^2) \right\|_2 \le \left\| \mathbf{C}_k \right\|_2 \cdot \left\| x_k(\boldsymbol{\lambda}^1) - x_k(\boldsymbol{\lambda}^2) \right\|_2 \le \frac{1}{\mu} \left\| \mathbf{C}_k \right\|_2^2 \cdot \left\| \boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^2 \right\|_2.$$

The matrix C, in view of its structure, satisfies the estimate $\left\| \mathbf{C}_k \right\|_2 \le m$. Then the gradient of the dual function satisfies

$$\left\| \nabla \varphi(\boldsymbol{\lambda}^1) - \nabla \varphi(\boldsymbol{\lambda}^2) \right\|_2 \le \sum_{k=1}^{n} \left\| \nabla \varphi_k(\boldsymbol{\lambda}^1) - \nabla \varphi_k(\boldsymbol{\lambda}^2) \right\|_2 \le \frac{m^2 n}{\mu} \left\| \boldsymbol{\lambda}^1 - \boldsymbol{\lambda}^2 \right\|_2.$$

Proof of Lemma 1. First, we state and prove a technical lemma.

Define $d_L(\boldsymbol{\lambda}) = \frac{L}{2} \left\| \boldsymbol{\lambda} - \boldsymbol{\lambda}^0 \right\|_2^2$ and consider the sequences

$$l_t(\boldsymbol{\lambda}) = \sum_{j=0}^{t} \alpha_j \left[\varphi(\boldsymbol{\lambda}^j) + \left\langle \nabla \varphi(\boldsymbol{\lambda}^j), \boldsymbol{\lambda} - \boldsymbol{\lambda}^j \right\rangle \right]$$

and

$$\psi_t(\boldsymbol{\lambda}) = l_t(\boldsymbol{\lambda}) + d_L(\boldsymbol{\lambda}), \quad t = 0,1,\ldots,$$

where $\{\boldsymbol{\lambda}^j\}_{j \ge 0}$ is the sequence of points generated by Algorithm 1.

Lemma 9. *After executing N steps of Algorithm* 1, *it is true that*

$$A_N \varphi(\mathbf{y}^N) \le \min_{\boldsymbol{\lambda} \in \mathbf{R}_+^m} \psi_N(\boldsymbol{\lambda}) = \psi_N(\mathbf{z}^N). \tag{A.20}$$

Proof. Inequality (A.20) is proved by induction. At $t = 0$, (A.20) is true. Indeed,

$$\psi_0 = \min_{\boldsymbol{\lambda} \in \mathbf{R}_+^m} \left\{ \alpha_0 \left[\varphi(\boldsymbol{\lambda}^0) + \left\langle \nabla \varphi(\boldsymbol{\lambda}^0), \boldsymbol{\lambda} - \boldsymbol{\lambda}^0 \right\rangle \right] + \frac{L}{2} \left\| \boldsymbol{\lambda} - \boldsymbol{\lambda}^0 \right\|_2^2 \right\} \overset{①}{\ge}$$

$$\overset{①}{\ge} \alpha_0 \min_{\boldsymbol{\lambda} \in \mathbf{R}_+^m} \left\{ \varphi(\boldsymbol{\lambda}^0) + \left\langle \nabla \varphi(\boldsymbol{\lambda}^0), \boldsymbol{\lambda} - \boldsymbol{\lambda}^0 \right\rangle + \frac{L}{2} \left\| \boldsymbol{\lambda} - \boldsymbol{\lambda}^0 \right\|_2^2 \right\} \overset{②}{\ge} \alpha_0 \varphi(y_0),$$

where ① holds, since $\alpha_0 = 1/2 \le 1$, while ② holds, since the function $\varphi(\boldsymbol{\lambda})$ has a Lipschitz continuous gradient (see Proposition 2 and [36, Lemma 1.2.3]). Thus, $A_0 \varphi(\mathbf{y}^0) = \frac{1}{2} \varphi(\mathbf{y}^0) \le \psi_0$.

Assume that (A.20) holds for t:

$$A_t \varphi(\mathbf{y}^t) \le \psi_t(\mathbf{z}^t). \tag{A.21}$$

Let us prove that (A.20) holds for $t+1$. Indeed, we have

$$\psi_{t+1}(\mathbf{z}^{t+1}) = \min_{\boldsymbol{\lambda} \in \mathbf{R}_+^m} \left\{ \psi_t(\boldsymbol{\lambda}) + \alpha_{t+1} \left[\varphi(\boldsymbol{\lambda}^{t+1}) + \left\langle \nabla \varphi(\boldsymbol{\lambda}^{t+1}), \boldsymbol{\lambda} - \boldsymbol{\lambda}^{t+1} \right\rangle \right] \right\}$$

$$\overset{①}{\ge} \min_{\boldsymbol{\lambda} \in \mathbf{R}_+^m} \left\{ \psi_t(\mathbf{z}^t) + \frac{L}{2} \left\| \boldsymbol{\lambda} - \mathbf{z}^t \right\|_2^2 + \alpha_{t+1} \left[\varphi(\boldsymbol{\lambda}^{t+1}) + \left\langle \nabla \varphi(\boldsymbol{\lambda}^{t+1}), \boldsymbol{\lambda} - \boldsymbol{\lambda}^{t+1} \right\rangle \right] \right\}$$

$$\overset{②}{\geq} \min_{\lambda\in\mathbf{R}^m_+}\left\{A_t\varphi(\mathbf{y}^t)+\frac{L}{2}\left\|\lambda-\mathbf{z}^t\right\|_2^2+\alpha_{t+1}\left[\varphi(\lambda^{t+1})+\left\langle\nabla\varphi(\lambda^{t+1}),\lambda-\lambda^{t+1}\right\rangle\right]\right\}$$

$$\overset{③}{\geq} \min_{\lambda\in\mathbf{R}^m_+}\left\{A_t\left(\varphi(\lambda^{t+1})+\left\langle\nabla\varphi(\lambda^{t+1}),\lambda-\lambda^{t+1}\right\rangle\right)+\frac{L}{2}\left\|\lambda-\mathbf{z}^t\right\|_2^2+\alpha_{t+1}\left[\varphi(\lambda^{t+1})+\left\langle\nabla\varphi(\lambda^{t+1}),\lambda-\lambda^{t+1}\right\rangle\right]\right\},$$

where ① holds, since the prox-function $\frac{1}{2}\left\|\lambda-\lambda^0\right\|_2^2$ is strongly convex and in view of the properties of the extremum at the point $\mathbf{z}^t$; ② follows from (A.21); and ③ holds in view of the convexity of the function $\varphi(\lambda)$.

Since the FGM coefficients A_t and α_t are related by the equalities $A_{t+1}=\sum_{j=0}^{t+1}\alpha_j=A_t+\alpha_{t+1}$ and $\tau_t=\alpha_{t+1}/A_{t+1}$, the relation $\lambda^{t+1}=\tau_t\mathbf{z}^t+(1-\tau_t)\mathbf{y}^t$ from Algorithm 1 can be rewritten as

$$A_{t+1}\lambda^{t+1}=\alpha_{t+1}\mathbf{z}^t+A_t\mathbf{y}^t.$$

Using the last relations, we can make the following transformations:

$$A_t\left\langle\nabla\varphi(\lambda^{t+1}),\mathbf{y}^t-\lambda^{t+1}\right\rangle+\alpha_{t+1}\left\langle\nabla\varphi(\lambda^{t+1}),\lambda-\lambda^{t+1}\right\rangle=-A_{t+1}\left\langle\nabla\varphi(\lambda^{t+1}),\lambda^{t+1}\right\rangle$$
$$+\alpha_{t+1}\left\langle\nabla\varphi(\lambda^{t+1}),\lambda\right\rangle+A_t\left\langle\nabla\varphi(\lambda^{t+1}),\mathbf{y}^t\right\rangle=\alpha_{t+1}\left\langle\nabla\varphi(\lambda^{t+1}),\lambda-\mathbf{z}^t\right\rangle.$$

Then

$$\begin{aligned}A_t\left(\varphi(\lambda^{t+1})+\left\langle\nabla\varphi(\lambda^{t+1}),\mathbf{y}^t-\lambda^{t+1}\right\rangle\right)+\frac{L}{2}\left\|\lambda-\mathbf{z}^t\right\|_2^2+\alpha_{t+1}\left[\varphi(\lambda^{t+1})+\left\langle\nabla\varphi(\lambda^{t+1}),\lambda-\lambda^{t+1}\right\rangle\right]\\ =A_{t+1}\varphi(\lambda^{t+1})+\frac{L}{2}\left\|\lambda-\mathbf{z}^t\right\|_2^2+\alpha_{t+1}\left\langle\nabla\varphi(\lambda^{t+1}),\lambda-\mathbf{z}^t\right\rangle.\end{aligned}\tag{A.23}$$

After replacing the last expression in (A.22) by (A.23), we can use an extended version of the Fenchel inequality for conjugate functions [37], namely,

$$\langle\mathbf{g},\mathbf{s}\rangle+\frac{\xi}{2}\|\mathbf{s}\|^2\geq-\frac{1}{2\xi}\|\mathbf{g}\|_*^2,\quad \mathbf{g}\in\mathbb{E}^*,\quad \mathbf{s}\in\mathbb{E},$$

where $\mathbb{E}$ is a finite-dimensional real vector space, $\mathbb{E}^*$ is the space of linear functions on $\mathbb{E}$ (dual space), and the norm in the dual space is given by $\|\mathbf{g}\|_*=\max_{\mathbf{x}}\{\langle\mathbf{g},\mathbf{x}\rangle\,|\,\|\mathbf{x}\|_E=1\}$. In our case, $\mathbf{g}=\nabla\varphi(\lambda^{t+1})$, $\mathbf{s}=\lambda-\mathbf{z}^t$, $\xi=\frac{L}{\alpha_{t+1}}$. Therefore,

$$\psi_{t+1}(\mathbf{z}^{t+1})\geq A_{t+1}\varphi(\lambda^{t+1})-\frac{\alpha_{t+1}^2}{2L}\left\|\nabla\varphi(\lambda^{t+1})\right\|_2^2.\tag{A.24}$$

To complete the proof of the lemma, we need to show that $A_{t+1}\varphi(\mathbf{y}^{t+1})$ is smaller than the right-hand side of inequality (A.24).

Since the function $\varphi(\lambda)$ is L-smooth (see Proposition 2),

$$\varphi(\mathbf{y}^{t+1})\leq\varphi(\lambda^{t+1})+\left\langle\nabla\varphi(\lambda^{t+1}),\mathbf{y}^{t+1}-\lambda^{t+1}\right\rangle+\frac{L}{2}\left\|\mathbf{y}^{t+1}-\lambda^{t+1}\right\|_2^2$$
$$=\min_{\lambda}\left\{\varphi(\lambda^{t+1})+\left\langle\nabla\varphi(\lambda^{t+1}),\lambda-\lambda^{t+1}\right\rangle+\frac{L}{2}\left\|\lambda-\lambda^{t+1}\right\|_2^2\right\}=\varphi(\lambda^{t+1})-\frac{1}{2L}\left\|\nabla\varphi(\lambda^{t+1})\right\|_2^2.$$

Multiplying both sides of the resulting inequality by A_{t+1} yields

$$A_{t+1}\varphi(\mathbf{y}^{t+1})\leq A_{t+1}\varphi(\lambda^{t+1})-\frac{A_{t+1}}{2L}\left\|\nabla\varphi(\lambda^{t+1})\right\|_2^2.$$

Since the FGM coefficients satisfy $\alpha_{t+1}^2\leq A_{t+1}$, we obtain

$$A_{t+1}\varphi(\mathbf{y}^{t+1})\leq A_{t+1}\varphi(\lambda^{t+1})-\frac{\alpha_{t+1}^2}{2L}\left\|\nabla\varphi(\lambda^{t+1})\right\|_2^2.\tag{A.25}$$

Therefore, by virtue of (A.24) and (A.25), $A_{t+1}\varphi(\mathbf{y}^{t+1}) \le \psi_{t+1}(\mathbf{z}^{t+1})$, as required.

Proof of Lemma 1. Define the set

$$\Lambda_{2\hat{R}} = \left\{\boldsymbol{\lambda} \in \mathbb{R}^m_+ : \|\boldsymbol{\lambda}\|_2 \le 2\hat{R}\right\},$$

where $\hat{R}$ is defined by the inequalities

$$\left\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\right\|_2 + \left\|\boldsymbol{\lambda}^0\right\|_2 \le \|\boldsymbol{\lambda}^*\|_2 + 2\left\|\boldsymbol{\lambda}^0\right\|_2 \le 3R = \hat{R}.$$

All $\boldsymbol{\lambda}^t$ belong to $\Lambda_{2\hat{R}}$, since

$$\left\|\boldsymbol{\lambda}^t\right\|_2 \le \left\|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^*\right\|_2 + \left\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\right\|_2 + \left\|\boldsymbol{\lambda}^0\right\|_2 \le 2\left\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\right\|_2 + \left\|\boldsymbol{\lambda}^0\right\|_2 \le 2\|\boldsymbol{\lambda}^*\|_2 + 3\left\|\boldsymbol{\lambda}^0\right\|_2 \le 5R \le 2\hat{R},$$

where the second inequality was obtained taking into account that $\left\|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^*\right\|_2 \le \left\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\right\|_2$ for $t = 0, 1, \ldots$.

The last inequality can be proved as follows. For any $\boldsymbol{\lambda} \in \mathbf{R}^m_+$, by Lemma 9 and the strong convexity of the function $\psi_t(\boldsymbol{\lambda})$ with a constant L, it is true that

$$\begin{aligned} &A_t\varphi(\mathbf{y}^t) + \frac{L}{2}\left\|\boldsymbol{\lambda} - \mathbf{z}^t\right\|_2^2 \le \psi_t(\mathbf{z}^t) + \frac{L}{2}\left\|\boldsymbol{\lambda} - \mathbf{z}^t\right\|_2^2 \le \psi_t(\boldsymbol{\lambda}) \\ &= \sum_{j=0}^{t} \alpha_j\left[\varphi(\boldsymbol{\lambda}^j) + \left\langle\nabla\varphi(\boldsymbol{\lambda}^j), \boldsymbol{\lambda} - \boldsymbol{\lambda}^j\right\rangle\right] + \frac{L}{2}\left\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^0\right\|_2^2. \end{aligned} \tag{A.26}$$

Since the function $\varphi(\boldsymbol{\lambda})$ is convex, the last expression in (A.26) can be estimated from above as $A_t\varphi(\boldsymbol{\lambda}) + \frac{L}{2}\left\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^0\right\|_2^2$. Then, for $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$,

$$\frac{L}{2}\left\|\boldsymbol{\lambda}^* - \mathbf{z}^t\right\|_2^2 \le A_t(\varphi(\mathbf{y}^t) - \varphi(\boldsymbol{\lambda}^*)) + \frac{L}{2}\left\|\boldsymbol{\lambda}^* - \mathbf{z}^t\right\|_2^2 \le \frac{L}{2}\left\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\right\|_2^2.$$

Therefore,

$$\left\|\boldsymbol{\lambda}^* - \mathbf{z}^t\right\|_2 \le \left\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\right\|_2. \tag{A.27}$$

Since $\mathbf{y}^t$ in Algorithm 1 is determined by a gradient projection step for a convex function $\varphi(\boldsymbol{\lambda})$, the sequence of points $\mathbf{y}^t$, $t = 0, 1, \ldots$, generated by the algorithm is bounded (the proof of this fact can be found, e.g., in [38, Lemma 9.17, p. 183] or in [28, p. 265]):

$$\left\|\boldsymbol{\lambda}^* - \mathbf{y}^t\right\|_2 \le \left\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\right\|_2. \tag{A.28}$$

Furthermore,

$$\left\|\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^*\right\|_2 = \left\|\tau_t(\mathbf{z}^t - \boldsymbol{\lambda}^*) + (1-\tau_t)(\mathbf{y}^t - \boldsymbol{\lambda}^*)\right\|_2 \le \tau_t\left\|\mathbf{z}^t - \boldsymbol{\lambda}^*\right\|_2 + (1-\tau_t)\left\|\mathbf{y}^t - \boldsymbol{\lambda}^*\right\|_2.$$

Combining this inequality with (A.27) and (A.28) yields the required result:

$$\left\|\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^*\right\|_2 \le \left\|\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^0\right\|_2, \quad t = -1, 0, 1, \ldots .$$

By Lemma 9,

$$\begin{aligned} A_N\varphi(\mathbf{y}^N) &\le \min_{\boldsymbol{\lambda}\in\mathbf{R}^m_+}\left\{\frac{L}{2}\left\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^0\right\|_2^2 + \sum_{t=0}^{N}\alpha_t\left[\varphi(\boldsymbol{\lambda}^t) + \left\langle\nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t\right\rangle\right]\right\} \\ &\le \min_{\boldsymbol{\lambda}\in\Lambda_{2\hat{R}}}\left\{\frac{L}{2}\left\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^0\right\|_2^2 + \sum_{t=0}^{N}\alpha_t\left[\varphi(\boldsymbol{\lambda}^t) + \left\langle\nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t\right\rangle\right]\right\} \\ &\overset{①}{\le} \min_{\boldsymbol{\lambda}\in\Lambda_{2\hat{R}}}\left\{\sum_{t=0}^{N}\alpha_t\left[\varphi(\boldsymbol{\lambda}^t) + \left\langle\nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t\right\rangle\right]\right\} + \frac{37L\hat{R}^2}{9}, \end{aligned}$$

where ① holds, since

$$\left\|\boldsymbol{\lambda} - \boldsymbol{\lambda}^0\right\|^2 \le 2\left\|\boldsymbol{\lambda}\right\|^2 + 2\left\|\boldsymbol{\lambda}^0\right\|^2 \le 8\hat{R}^2 + \frac{2}{9}\hat{R}^2 = \frac{74}{9}\hat{R}^2. \tag{A.29}$$

Applying the definitions of the dual objective function $\varphi(\boldsymbol{\lambda}^t)$ (see (2)) and of its gradient $\nabla\varphi(\boldsymbol{\lambda}^t)$ (see Proposition 1) yields

$$\begin{aligned}
&\sum_{t=0}^{N}\alpha_t\left[\varphi(\boldsymbol{\lambda}^t) + \left\langle\nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t\right\rangle\right]\\
&= \sum_{t=0}^{N}\alpha_t\left(\left\langle\boldsymbol{\lambda}^t, \mathbf{b}\right\rangle + \sum_{k=1}^{n}\left(u_k(x_k^t(\boldsymbol{\lambda}^t)) - \left\langle\boldsymbol{\lambda}^t, \mathbf{C}_k x_k^t(\boldsymbol{\lambda}^t)\right\rangle\right) + \left\langle\mathbf{b} - \sum_{k=1}^{n}\mathbf{C}_k x_k^t(\boldsymbol{\lambda}^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t\right\rangle\right)\\
&= \sum_{t=0}^{N}\alpha_t\left(\sum_{k=1}^{n}u_k(x_k^t(\boldsymbol{\lambda}^t)) + \left\langle\boldsymbol{\lambda}, \mathbf{b} - \sum_{k=1}^{n}\mathbf{C}_k x_k^t(\boldsymbol{\lambda}^t)\right\rangle\right) \le A_N\left(U(\hat{\mathbf{x}}^N) + \left\langle\boldsymbol{\lambda}, \mathbf{b} - C\hat{\mathbf{x}}^N\right\rangle\right),
\end{aligned}$$

where the last the inequality holds, since the utility functions are concave.

Thus,

$$\begin{aligned}
A_N\varphi(\mathbf{y}^N) &\le A_N U(\hat{\mathbf{x}}^N) + \frac{37L\hat{R}^2}{9} + A_N\min_{\boldsymbol{\lambda}\in\Lambda_{2\hat{R}}}\left\{\left\langle\boldsymbol{\lambda}, \mathbf{b} - C\hat{\mathbf{x}}^N\right\rangle\right\} = A_N U(\hat{\mathbf{x}}^N) + \frac{37L\hat{R}^2}{9}\\
&- A_N\max_{\boldsymbol{\lambda}\in\Lambda_{2\hat{R}}}\left\{\left\langle\boldsymbol{\lambda}, C\hat{\mathbf{x}}^N - \mathbf{b}\right\rangle\right\} = A_N U(\hat{\mathbf{x}}^N) + \frac{37L\hat{R}^2}{9} - 2\hat{R}A_N\left\|\left(C\hat{\mathbf{x}}^N - \mathbf{b}\right)_+\right\|_2,
\end{aligned}$$

which yields estimate (6).

Proof of Lemma 2. First, we prove several auxiliary technical lemmas.

Lemma 10. *Let A, B, and $\{r_t\}_{t=0}^N$ be nonnegative numbers such that, for any $l = 1,\ldots,N$,*

$$\frac{1}{2}r_l^2 \le Ar_0^2 + Br_0\sqrt{\sum_{t=0}^{l-1}r_t^2}. \tag{A.30}$$

Then

$$r_l \le Cr_0, \tag{A.31}$$

where C is a positive constant satisfying $C^2 \ge \max\{1, 2A + 2BC\sqrt{N}\}$, i.e., for example, it is possible to use

$$C = \max\left\{1, B\sqrt{N} + \sqrt{B^2N + 2A}\right\}.$$

Proof. Relation (A.31) is proved by induction. For $l = 0$, this inequality holds, since $C \ge 1$. Assuming that (A.31) holds for all $l < N$, we prove that it holds for $l + 1$ as well. Indeed,

$$r_{l+1} \overset{(A.30)}{\le} \sqrt{2}\sqrt{Ar_0^2 + Br_0\sqrt{\sum_{t=0}^{l}r_t^2}} \overset{(A.31)}{\le} r_0\sqrt{2}\sqrt{A + BC\sqrt{N}} = r_0\underbrace{\sqrt{2A + 2BC\sqrt{N}}}_{\le C} \le Cr_0.$$

Lemma 11. *Suppose that sequences of nonnegative coefficients $\{R_t\}_{t\ge0}$ and random vectors $\{\boldsymbol{\eta}^t\}_{t\ge0}$ and $\{\mathbf{a}^t\}_{t\ge0}$ are such that, for all $l = 1,\ldots,N$,*

$$\frac{1}{2}R_l^2 \le A + u\sum_{t=0}^{l-1}\left\langle\boldsymbol{\eta}^t, \mathbf{a}^t\right\rangle, \tag{A.32}$$

where A is a nonnegative constant, $d \ge 1$ is a positive constant, $\left\|\mathbf{a}^t\right\|_2 \le \widetilde{R}_t d$ and $\widetilde{R}_t = \max\{\widetilde{R}_{t-1}, R_t\}$ for all $t \ge 1$, $\widetilde{R}_0 = R_0$, and $\widetilde{R}_t$ depends only on $\boldsymbol{\eta}^0,\ldots,\boldsymbol{\eta}^t$. Additionally, suppose that $\mathbf{a}^t$ is a function of $\boldsymbol{\eta}^0,\ldots,\boldsymbol{\eta}^{t-1}$ $\forall t \ge 1$, a^0 is a constant vector, and, for any $t \ge 0$,

$$\mathbb{E}[\eta^t\,|\,\{\boldsymbol{\eta}^k\}_{k=0}^{t-1}] = 0, \quad \mathbb{E}\left[\exp\left(\left\|\boldsymbol{\eta}^t\right\|_2^2\sigma^{-2}\right)|\,\{\boldsymbol{\eta}^k\}_{k=0}^{t-1}\right] \le \exp(1).$$

Then, with probability $1 - 2\delta$, *the inequalities*

$$\widetilde{R}_l \le JR_0 \quad \text{and} \quad A + u\sum_{t=0}^{l-1}\left\langle \boldsymbol{\eta}^t, \mathbf{a}^t \right\rangle \le A + udD\sqrt{\sigma^2 g(N)NJ}\,\widetilde{R}_0^2$$

hold for all $l = 1, \ldots, N$ *simultaneously, where* D *is a positive constant*,

$$F = 2\sigma^2 d^2 N(2ud)^N \left(2A + ud\widetilde{R}_0^2 + 12ud\ln\frac{N}{\delta}\sigma^2 N\right),$$

$f = d^2\sigma^2\widetilde{R}_0^2$, $g(N) = \ln\left(\frac{N}{\delta}\right) + \ln\ln\left(\frac{F}{f}\right)$, *and*

$$J = \max\left\{1, \frac{1}{\widetilde{R}_0}udD\sqrt{\sigma^2 g(N)} + \sqrt{\frac{1}{\widetilde{R}_0^2}u^2d^2C_1^2\sigma^2 g(N) + \frac{2A}{R_0^2}}\right\}.$$

Proof. The Cauchy–Schwarz inequality is applied to the second term on the right-hand side of (A.32):

$$\frac{1}{2}R_l^2 \le A + ud\sum_{t=0}^{l-1}\left\|\boldsymbol{\eta}^t\right\|_2 \widetilde{R}_t \le A + \frac{ud}{2}\sum_{t=0}^{l-1}\widetilde{R}_t^2 + \frac{ud}{2}\sum_{t=0}^{l-1}\left\|\boldsymbol{\eta}^t\right\|_2^2. \tag{A.33}$$

By Theorem 2.1 from [35], we have

$$(\forall N \ge 1,\ \forall\gamma \ge 0): \quad \mathbb{P}\left\{\left\|\sum_{t=0}^{N-1}\boldsymbol{\eta}^t\right\|_2 \ge (\sqrt{2} + \sqrt{2}\gamma)\sqrt{\sum_{t=0}^{N-1}\sigma_t^2}\right\} \le \exp\left(-\frac{\gamma^2}{3}\right). \tag{A.34}$$

Then, with probability at least

$$1 - \frac{\delta}{N} = 1 - \exp\left(-\frac{\gamma^2}{3}\right), \tag{A.35}$$

it holds that

$$\left\|\boldsymbol{\eta}^t\right\|_2 \le \sqrt{2}\left(1 + \sqrt{3\ln\frac{N}{\delta}}\right)\sigma \le 2\sqrt{6\ln\frac{N}{\delta}}\sigma. \tag{A.36}$$

Indeed, expressing γ from (A.35) yields $\gamma = \sqrt{3\ln\frac{N}{\delta}}$. Plugging this expression into (A.34) and substituting a unified $\sigma \in \mathbf{R}_+$ for the sequence σ_t, $t = 0, \ldots, N-1$, we obtain estimate (A.36).

Combining the resulting inequalities, we see that, with probability greater than or equal to $1 - \delta$, the inequality

$$\frac{1}{2}R_l^2 \le A + \frac{ud}{2}\sum_{t=0}^{l-1}\widetilde{R}_t^2 + 12ud\ln\frac{N}{\delta}\sigma^2 l$$

holds for all $l = 1, \ldots, N$ simultaneously. Note that the last term in this estimate is a nondecreasing function of l. Define $\hat{l}$ as the largest integer for which $\hat{l} \le l$ and $\widetilde{R}_{tl} = R_{\hat{l}}$. Then $R_{\hat{l}} = \widetilde{R}_{\hat{l}} = \widetilde{R}_{\hat{l}+1} = \ldots = \widetilde{R}_l$ and, hence, with probability $\ge 1 - \delta$,

$$\frac{1}{2}\widetilde{R}_l^2 \le A + \frac{ud}{2}\sum_{t=0}^{\hat{l}-1}\widetilde{R}_t^2 + 12ud\ln\frac{N}{\delta}\sigma^2\hat{l} \le A + \frac{ud}{2}\sum_{t=0}^{l-1}\widetilde{R}_t^2 + 12ud\ln\frac{N}{\delta}\sigma^2 l \quad \forall l = 1, \ldots, N.$$

As a result, with probability $\ge 1 - \delta$, we have the estimate

$$\widetilde{R}_l^2 \le 2A + ud\sum_{t=0}^{l-1}\widetilde{R}_k^2 + 24ud\ln\frac{N}{\delta}\sigma^2 l \le 2A\underbrace{(1 + ud)}_{\le 2ud} + \underbrace{(ud + u^2d^2)}_{\le 2u^2d^2}\sum_{t=0}^{l-2}\widetilde{R}_t^2$$

$$+ 24ud\ln\frac{N}{\delta}\sigma^2\underbrace{(l + ud(l-1))}_{\le 2udl} \le 2ud\left(2A + ud\sum_{t=0}^{l-2}\widetilde{R}_t^2 + 24ud\ln\frac{N}{\delta}\sigma^2 l\right) \quad \forall l = 1, \ldots, N.$$

Applying this estimate recursively, we conclude that, with probability $\geq 1-\delta$,

$$\widetilde{R}_l^2 \leq (2ud)^l \left(2A + ud\widetilde{R}_0^2 + 24ud\ln\frac{N}{\delta}\sigma^2 l\right).$$

Next, consider the sequence of random variables $\xi^t = \left\langle \boldsymbol{\eta}^t, \mathbf{a}^t \right\rangle$. Note that $\mathbb{E}[\xi^t | \xi^0, \ldots, \xi^{t-1}] = \left\langle \mathbb{E}[\boldsymbol{\eta}^t | \boldsymbol{\eta}^0, \ldots, \boldsymbol{\eta}^{k-1}], \mathbf{a}^t \right\rangle = 0$. Then, using the Cauchy–Schwarz inequality yields

$$\mathbb{E}\left[\exp\left(\frac{(\xi^t)^2}{\sigma^2 d^2 \widetilde{R}_t^2}\right) | \xi^0, \ldots, \xi^{t-1}\right] \leq \mathbb{E}\left[\exp\left(\frac{\left\|\boldsymbol{\eta}^t\right\|_2^2 d^2 \widetilde{R}_t^2}{\sigma^2 d^2 \widetilde{R}_t^2}\right) | \boldsymbol{\eta}^0, \ldots, \boldsymbol{\eta}^{t-1}\right]$$
$$= \mathbb{E}\left[\exp\left(\frac{\left\|\boldsymbol{\eta}^t\right\|_2^2}{\sigma^2}\right) | \boldsymbol{\eta}^0, \ldots, \boldsymbol{\eta}^{t-1}\right] \leq \exp(1).$$

Define $\hat{\sigma}_t^2 = \sigma^2 d^2 \widetilde{R}_t^2$. Then, with probability $\geq 1-\delta$, it is true that

$$\sum_{t=0}^{l-1} \hat{\sigma}_t^2 \leq \sigma^2 d^2 l (2ud)^l \left(2A + ud\widetilde{R}_0^2 + 24ud\ln\frac{N}{\delta}\sigma^2 l\right)$$
$$\leq \sigma^2 d^2 N (2ud)^N \left(2A + ud\widetilde{R}_0^2 + 24ud\ln\frac{N}{\delta}\sigma^2 N\right) := \frac{F}{2}$$

for all $l = 1, \ldots, N$ simultaneously, where

$$F = 2\sigma^2 d^2 N (2ud)^N \left(2A + ud\widetilde{R}_0^2 + 24ud\ln\frac{N}{\delta}\sigma^2 N\right).$$

Using Corollary 8 from [34] for $b = \hat{\sigma}_0^2$, we see that, for any $l = 1, \ldots, N$ with probability $\geq 1 - \frac{\delta}{N}$,

$$\text{either} \quad \sum_{t=0}^{l-1} \hat{\sigma}_t^2 \geq F \quad \text{or} \quad \left|\sum_{t=0}^{l-1} \xi^t\right| \leq C_1 \sqrt{\sum_{t=0}^{l-1} \hat{\sigma}_t^2 \left(\ln\left(\frac{N}{\delta}\right) + \ln\ln\left(\frac{F}{f}\right)\right)}, \tag{A.37}$$

where $C_1 > 0$ is a constant independent of F or f.

Combining the resulting estimates, we conclude that, with probability $\geq 1-\delta$, estimate (A.37) holds for all $l = 1, \ldots, N$ simultaneously.

Taking into account the choice of F, with probability $\geq 1-2\delta$, the estimate

$$\left|\sum_{t=0}^{l-1} \xi^t\right| \leq C_1 \sqrt{\sum_{t=0}^{l-1} \hat{\sigma}_t^2 \left(\ln\left(\frac{N}{\delta}\right) + \ln\ln\left(\frac{F}{f}\right)\right)}$$

holds for all $l = 1, \ldots, N$ simultaneously.

For convenience in what follows, we introduce $g(N) := \ln\left(\frac{N}{\delta}\right) + \ln\ln\left(\frac{F}{f}\right) \approx \ln\left(\frac{N}{\delta}\right)$, neglecting the constant. Using $\hat{\sigma}_t^2 = \sigma^2 d^2 \widetilde{R}_t^2$, we find that, with probability $\geq 1-2\delta$, the estimate

$$\frac{1}{2}\widetilde{R}_l^2 \leq A + u\sum_{t=0}^{l-1} \underbrace{\left\langle \boldsymbol{\eta}^t, \mathbf{a}^t \right\rangle}_{\xi^t} \leq A + udD\sqrt{\sigma^2 g(N)}\sqrt{\sum_{t=0}^{l-1} \widetilde{R}_t^2} \tag{A.38}$$

holds for all $l = 1, \ldots, N$ simultaneously. After choosing $A = \frac{A}{\widetilde{R}_0^2}$, $B = \frac{1}{\widetilde{R}_0} udC_1\sqrt{\sigma^2 g(N)}$, and $r_t = \widetilde{R}_t$, Lemma 10 implies that, with probability $1-2\delta$,

$$\widetilde{R}_l \leq JR_0$$

for all $l = 1,\ldots,N$ simultaneously, where

$$J = \max\left\{1, \frac{1}{\widetilde{R}_0} udC_1\sqrt{\sigma^2 g(N)} + \sqrt{\frac{1}{\widetilde{R}_0^2} u^2 d^2 C_1^2 \sigma^2 g(N) + \frac{2A}{R_0^2}}\right\}.$$

It follows that, with probability $1 - 2\delta$, the estimate

$$A + u\sum_{t=0}^{l-1}\left\langle \boldsymbol{\eta}^t, \mathbf{a}^t\right\rangle \le A + udC_1\sqrt{\sigma^2 g(N) lJ}\,\widetilde{R}_0^2 \le A + udC_1\sqrt{\sigma^2 g(N) NJ}\,\widetilde{R}_0^2$$

holds for all $l = 1,\ldots,N$ simultaneously.

Proof of Lemma 2. For $\boldsymbol{\lambda} \in \mathbf{R}_+^m$

$$\left\|\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}\right\|_2^2 \left\|[\boldsymbol{\lambda}^t - \beta\nabla\varphi(\boldsymbol{\lambda}^t, \xi^t)]_+ - \boldsymbol{\lambda}\right\|_2^2 \le \left\|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}\right\|_2^2 - 2\beta\left\langle \nabla\varphi(\boldsymbol{\lambda}^t, \xi^t), \boldsymbol{\lambda}^t - \boldsymbol{\lambda}\right\rangle + \beta^2\left\|\nabla\varphi(\boldsymbol{\lambda}^t, \xi^t)\right\|_2^2,$$

i.e.,

$$0 \le \frac{1}{2\beta}\left(\left\|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}\right\|_2^2 - \left\|\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}\right\|_2^2\right) + \left\langle \nabla\varphi(\boldsymbol{\lambda}^t, \xi^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t\right\rangle + \frac{\beta}{2}\left\|\nabla\varphi(\boldsymbol{\lambda}^t, \xi^t)\right\|_2^2. \tag{A.39}$$

Adding $\varphi(\boldsymbol{\lambda}^t)$ to both sides of inequality (A.39), multiplying it by N, and summing the result from 0 to $N-1$, we obtain

$$\begin{aligned}\frac{1}{N}\sum_{t=0}^{N-1}\varphi\boldsymbol{\lambda}^t) \le \frac{1}{N}\sum_{t=0}^{N-1}\Bigg\{&\varphi(\boldsymbol{\lambda}^t) + \left\langle \nabla\varphi(\boldsymbol{\lambda}^t, \xi^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t\right\rangle + \frac{\beta}{2}\left\|\nabla\varphi(\boldsymbol{\lambda}^t, \xi^t)\right\|_2^2 \\ &+ \frac{1}{2\beta}\left(\left\|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}\right\|_2^2 - \left\|\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}\right\|_2^2\right)\Bigg\}.\end{aligned} \tag{A.40}$$

Since $\varphi(\boldsymbol{\lambda})$ is convex, for $\hat{\boldsymbol{\lambda}}^N = \frac{1}{N}\sum_{t=0}^{N-1}\boldsymbol{\lambda}^t$, we have

$$N\varphi(\hat{\boldsymbol{\lambda}}^N) \le \sum_{t=0}^{N-1}\left\{\varphi(\boldsymbol{\lambda}^t) + \left\langle \nabla\varphi(\boldsymbol{\lambda}^t, \xi^t), \boldsymbol{\lambda} - \boldsymbol{\lambda}^t\right\rangle\right\} + \frac{\beta}{2}\left\|\nabla\varphi(\boldsymbol{\lambda}^t, \xi^t)\right\|_2^2 + \frac{1}{2\beta}\left(\left\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}\right\|_2^2 - \left\|\boldsymbol{\lambda}^N - \boldsymbol{\lambda}\right\|_2^2\right). \tag{A.41}$$

Setting $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$ and adding and subtracting $\sum_{t=0}^{N-1}\left\langle \nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^t\right\rangle$ on the right-hand side, we obtain

$$\begin{aligned}N\varphi(\hat{\boldsymbol{\lambda}}^N) \le &\sum_{t=0}^{N-1}\left\{\varphi(\boldsymbol{\lambda}^t) + \left\langle \nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^t\right\rangle\right\} + \frac{\beta}{2}\left\|\nabla\varphi(\boldsymbol{\lambda}^t, \xi^t)\right\|_2^2 \\ &+ \sum_{t=0}^{N-1}\left\langle \nabla\varphi(\boldsymbol{\lambda}^t, \xi^t) - \nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^t\right\rangle + \frac{1}{2\beta}\left(\left\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\right\|_2^2 - \left\|\boldsymbol{\lambda}^N - \boldsymbol{\lambda}^*\right\|_2^2\right).\end{aligned} \tag{A.42}$$

The convexity of $\varphi(\boldsymbol{\lambda})$ implies that

$$\sum_{t=0}^{N-1}\left\{\varphi(\boldsymbol{\lambda}^t) + \left\langle \nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^t\right\rangle\right\} \le \sum_{t=0}^{N-1}\left\{\varphi(\boldsymbol{\lambda}^t) + \varphi(\boldsymbol{\lambda}^*) - \varphi(\boldsymbol{\lambda}^t)\right\} \le \sum_{t=0}^{N-1}\varphi(\boldsymbol{\lambda}^*) \le N\varphi(\boldsymbol{\lambda}^*).$$

Substituting this estimate into (A.42) yields

$$\frac{1}{2\beta}\left\|\boldsymbol{\lambda}^N - \boldsymbol{\lambda}^*\right\|_2^2 \le \frac{1}{2\beta}\left\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\right\|_2^2 + \sum_{t=0}^{N-1}\left\langle \nabla\varphi(\boldsymbol{\lambda}^t, \xi^t) - \nabla\varphi(\boldsymbol{\lambda}^t), \boldsymbol{\lambda}^* - \boldsymbol{\lambda}^t\right\rangle + \frac{\beta}{2}\left\|\nabla\varphi(\boldsymbol{\lambda}^t, \xi^t)\right\|_2^2. \tag{A.43}$$

Define $R_t = \left\|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^*\right\|_2$ and $\widetilde{R}_t = \max\{\widetilde{R}_{t-1}, R_t\}$, where $R_0 = \widetilde{R}_0$. Since $\boldsymbol{\lambda}^0 = \mathbf{0}$ and $\left\|\boldsymbol{\lambda}^*\right\|_2 \le R$, we have $R_0 = R$. Moreover, by construction, $\boldsymbol{\lambda}^t \in B_{\widetilde{R}_t}(\boldsymbol{\lambda}^*)$. In a similar manner, we define $\left\|\mathbf{a}^t\right\|_2 = \left\|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^*\right\|_2 \le \widetilde{R}_t$. Then (A.43) can be rewritten as

$$\frac{1}{2\beta}\widetilde{R}_N^2 \le \frac{1}{2\beta}\widetilde{R}_0^2 + \sum_{t=0}^{N-1}\left\langle \nabla\varphi(\boldsymbol{\lambda}^t, \xi^t) - \nabla\varphi(\boldsymbol{\lambda}^t), \mathbf{a}^t\right\rangle + \frac{\beta}{2}\left\|\nabla\varphi(\boldsymbol{\lambda}^t, \xi^t)\right\|_2^2.$$

Define $\eta^t = \nabla\varphi(\lambda^t, \xi^t) - \nabla\varphi(\lambda^t)$. By Theorem 2.1 from [35],

$$\mathbb{P}\left\{\left\|\sum_{t=0}^{N-1}\eta^t\right\|_2 \geq (\sqrt{2}+\sqrt{2}\gamma)\sqrt{\sum_{t=0}^{N-1}\sigma_t^2}\,|\,\{\xi^t\}_{t=0}^{N-1}\right\} \leq \exp\left(-\frac{\gamma^2}{3}\right). \tag{A.44}$$

Using Lemma 2 from [34], we obtain

$$\mathbb{E}\left[\exp\left(\frac{\|\eta^t\|_2^2}{\sigma^2}\right)|\,\{\xi^k\}_{k=0}^{t-1}\right] \leq \exp(1),$$

where η^t depends only on $\xi^{t-1},\ldots,\xi^0$. Using the new notation and (8), we have

$$\widetilde{R}_N^2 \leq \widetilde{R}_0^2 + 2\beta\sum_{t=0}^{N-1}\left\langle\eta^t, a^t\right\rangle + \beta^2 M^2.$$

Then, by Lemma 11 with constants $A = \widetilde{R}_0^2 + \beta^2 M^2$, $d = 1$, and $u = \beta$, we conclude that, with probability $1 - 2\delta$, where $\frac{\delta}{N} = \exp\left(-\frac{\gamma^2}{3}\right)$, the estimates

$$\widetilde{R}_l \leq JR_0 \quad \text{and} \quad \sum_{t=0}^{l-1}\left\langle\eta^t, a^t\right\rangle \leq D\sqrt{\sigma^2 g(N)NJ}\,\widetilde{R}_0^2 \tag{A.45}$$

hold for all $l = 1,\ldots,N$ simultaneously, where D is a positive constant,

$$F = 2\sigma^2 N(2\beta)^N\left(2A + \beta\widetilde{R}_0^2 + 24\ln\frac{N}{\delta}\beta\sigma^2 N\right),$$

$f = \sigma^2\widetilde{R}_0^2$, $g(N) = \ln\left(\frac{N}{\delta}\right) + \ln\ln\left(\frac{F}{f}\right)$, and

$$J = \max\left\{1, \frac{1}{\widetilde{R}_0}\beta C_1\sqrt{\sigma^2 g(N)} + \sqrt{\frac{1}{\widetilde{R}_0^2}\beta^2 C_1^2\sigma^2 g(N) + \frac{2A}{R_0^2}}\right\}.$$

To estimate the duality gap, we use (A.41), noting that this estimate holds for any $\lambda \in \mathbf{R}_+^m$. Therefore, taking the minimum over all λ from the set $\Lambda_{2R} = \left\{\lambda \in \mathbb{R}_+^m : \|\lambda\|_2 \leq 2R\right\}$ yields

$$N\varphi(\hat{\lambda}^N) \leq \min_{\lambda\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left(\varphi(\lambda^t) + \left\langle\nabla\varphi(\lambda^t,\xi^t), \lambda - \lambda^t\right\rangle\right) + \frac{1}{2\beta}\left\|\lambda^0 - \lambda\right\|_2^2\right\} + \frac{N\beta M^2}{2},$$

where the last term was estimated using assumption (8). Additionally, the inequality $\left\|\lambda^N - \lambda\right\|_2^2 \geq 0$ was taken into account. By virtue of (A.29), we obtain the estimate

$$\varphi(\hat{\lambda}^N) \leq \frac{1}{N}\min_{\lambda\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left(\varphi(\lambda^t) + \left\langle\nabla\varphi(\lambda^t,\xi^t), \lambda - \lambda^t\right\rangle\right)\right\} + \frac{2R^2}{\beta N} + \frac{\beta M^2}{2}.$$

Adding and subtracting $\sum_{t=0}^{N-1}\left\langle\nabla\varphi(\lambda^t), \lambda - \lambda^t\right\rangle$ from the expression under the minimum sign yields

$$\min_{\lambda\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left(\varphi(\lambda^t) + \left\langle\nabla\varphi(\lambda^t), \lambda - \lambda^t\right\rangle\right)\right\} \leq \min_{\lambda\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left(\varphi(\lambda^t) + \left\langle\nabla\varphi(\lambda^t,\xi^t), \lambda - \lambda^t\right\rangle\right)\right\}$$
$$+ \max_{\lambda\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left\langle\nabla\varphi(\lambda^t,\xi^t) - \nabla\varphi(\lambda^t), \lambda\right\rangle\right\} + \sum_{t=0}^{N-1}\left\langle\nabla\varphi(\lambda^t,\xi^t) - \nabla\varphi(\lambda^t), -\lambda^t\right\rangle.$$

Note that $-\boldsymbol{\lambda}^* \in \Lambda_{2R}$. Then we have

$$\sum_{t=0}^{N-1}\left\langle \nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t),-\boldsymbol{\lambda}^t\right\rangle = \sum_{t=0}^{N-1}\left\langle \nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}^*-\boldsymbol{\lambda}^t\right\rangle$$
$$+\sum_{t=0}^{N-1}\left\langle \nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t),-\boldsymbol{\lambda}^*\right\rangle \le \max_{\boldsymbol{\lambda}\in\Lambda_{2R}}\sum_{t=0}^{N-1}\left\langle \nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}\right\rangle$$
$$+\sum_{t=0}^{N-1}\left\langle \nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}^*-\boldsymbol{\lambda}^t\right\rangle.$$

It follows that

$$\varphi(\hat{\boldsymbol{\lambda}}^N) \le \frac{1}{N}\min_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left(\varphi(\boldsymbol{\lambda}^t)+\left\langle\nabla\varphi(\boldsymbol{\lambda}^t,\xi^t),\boldsymbol{\lambda}-\boldsymbol{\lambda}^t\right\rangle\right)\right\}+\frac{2R^2}{\beta N}+\frac{\beta M^2}{2}$$
$$\le \frac{1}{N}\min_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left(\varphi(\boldsymbol{\lambda}^t)+\left\langle\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}-\boldsymbol{\lambda}^t\right\rangle\right)\right\}+\frac{1}{N}\sum_{t=0}^{N-1}\left\langle\nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}^*-\boldsymbol{\lambda}^t\right\rangle$$
$$+\frac{2}{N}\max_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left\langle\nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}\right\rangle\right\}+\frac{2R^2}{\beta N}+\frac{\beta M^2}{2}.$$

The definition of the norm implies that

$$\max_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left\langle\nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}\right\rangle\right\} \le 2R\left\|\sum_{t=0}^{N-1}\left(\nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^l)\right)\right\|_2. \tag{A.46}$$

Using (A.44), we conclude that, with probability $1-\delta$,

$$\left\|\sum_{t=0}^{N-1}\left(\nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^l)\right)\right\|_2 \le \sigma\sqrt{2N}\left(1+\sqrt{3\ln\frac{1}{\delta}}\right). \tag{A.47}$$

Substituting the values of $\varphi(\boldsymbol{\lambda}^t)$ and $\nabla\varphi(\boldsymbol{\lambda}^t)$ into the expression $\sum_{t=0}^{N-1}\left(\varphi(\boldsymbol{\lambda}^t)+\left\langle\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}-\boldsymbol{\lambda}^t\right\rangle\right)$ in (A.46) yields

$$\sum_{t=0}^{N-1}\left(\left\langle\boldsymbol{\lambda}^t,\mathbf{b}\right\rangle+\sum_{k=1}^{n}(u_k(x_k(\boldsymbol{\lambda}^t))-\left\langle\boldsymbol{\lambda}^t,\mathbf{C}_k x_k(\boldsymbol{\lambda}^t)\right\rangle+\left\langle\mathbf{b}-C\mathbf{x}^t(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}-\boldsymbol{\lambda}^t\right\rangle\right)$$
$$=\sum_{t=0}^{N-1}\left(\sum_{k=1}^{n}(u_k(x_k(\boldsymbol{\lambda}^t))+\left\langle\mathbf{b}-C\mathbf{x}^t(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}\right\rangle\right).$$

Then, since the functions $u_k(x_k)$ are concave, it holds that

$$\frac{1}{N}\min_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left(\varphi(\boldsymbol{\lambda}^t)+\left\langle\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}-\boldsymbol{\lambda}^t\right\rangle\right)\right\} \le U(\hat{\mathbf{x}}^N)-\frac{1}{N}\max_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left\langle C\mathbf{x}^t(\boldsymbol{\lambda}^t)-\mathbf{b},\boldsymbol{\lambda}\right\rangle\right\}.$$

Combining this inequality with (A.46) gives

$$\varphi(\hat{\boldsymbol{\lambda}}^N) \le U(\hat{\mathbf{x}}^N)-\frac{1}{N}\max_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\sum_{t=0}^{N-1}\left\langle C\mathbf{x}^t(\boldsymbol{\lambda}^t)-\mathbf{b},\boldsymbol{\lambda}\right\rangle\right\}+\frac{2R^2}{\beta N}+\frac{\beta M^2}{2}$$
$$+\frac{2R}{N}\left\|\sum_{t=0}^{N-1}\left(\nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t)\right)\right\|_2+\frac{1}{N}\sum_{t=0}^{N-1}\left\langle\nabla\varphi(\boldsymbol{\lambda}^t,\xi^t)-\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}^*-\boldsymbol{\lambda}^t\right\rangle.$$

From this, taking into account estimate (A.47) and result (A.44), we see that, with probability $1-3\delta$,

$$\varphi(\hat{\boldsymbol{\lambda}}^N)-U(\hat{\mathbf{x}}^N)+2R\left\|[C\hat{\mathbf{x}}^N-\mathbf{b}]_+\right\|_2 \le \frac{2R\sigma\sqrt{2}\left(1+\sqrt{3\ln\frac{1}{\delta}}\right)}{\sqrt{N}}+\frac{2R^2}{\beta N}+\frac{\beta M^2}{2}+C_1\frac{\sigma\sqrt{g(N)J}R^2}{\sqrt{N}}. \tag{A.48}$$

By Theorem 2.1 in [35], for all $\gamma > 0$, it is true that

$$P\left\{\left\|\sum_{t=0}^{N-1}\left(\mathbf{x}(\boldsymbol{\lambda}^t,\xi^t)-\mathbf{x}(\boldsymbol{\lambda}^t)\right)\right\|_2 \geq (\sqrt{2}+\sqrt{2}\gamma)\sqrt{\sum_{t=0}^{N-1}\sigma_x^2}\,|\,\{\xi^t\}_{t=0}^{N-1}\right\} \leq \exp\left(-\frac{\gamma^2}{3}\right).$$

Setting $\gamma = \sqrt{3\ln\frac{1}{\delta}}$, we conclude that, with probability $1-\delta$,

$$\left\|\tilde{\mathbf{x}}^N-\hat{\mathbf{x}}^N\right\|_2 = \frac{1}{N}\left\|\sum_{t=0}^{N-1}\left(\mathbf{x}(\boldsymbol{\lambda}^t,\xi^t)-\mathbf{x}(\boldsymbol{\lambda}^t)\right)\right\|_2 \leq \sigma_x\sqrt{\frac{2}{N}}\left(1+\sqrt{3\ln\frac{1}{\delta}}\right).$$

Then, with probability $1-\delta$,

$$\left\|C\tilde{\mathbf{x}}^N-C\hat{\mathbf{x}}^N\right\|_2 \leq \|C\|_2\cdot\left\|\tilde{\mathbf{x}}^N-\hat{\mathbf{x}}^N\right\|_2 \leq \sigma_x\sqrt{\frac{2\lambda_{\max}(C^{\mathrm{T}}C)}{N}}\left(1+\sqrt{3\ln\frac{1}{\delta}}\right).$$

Note that

$$\begin{aligned}
2R\left\|[C\tilde{\mathbf{x}}^N-\mathbf{b}]_+\right\|_2 &= \max_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\left\langle C\tilde{\mathbf{x}}^N-\mathbf{b},\boldsymbol{\lambda}\right\rangle+\left\langle C\tilde{\mathbf{x}}^N-C\hat{\mathbf{x}}^N-\mathbf{b}+\mathbf{b},\boldsymbol{\lambda}\right\rangle\right\}\\
\leq \max_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\left\langle C\hat{\mathbf{x}}^N-\mathbf{b},\boldsymbol{\lambda}\right\rangle\right\}+\max_{\boldsymbol{\lambda}\in\Lambda_{2R}}\left\{\left\langle C\tilde{\mathbf{x}}^N-C\hat{\mathbf{x}}^N,\boldsymbol{\lambda}\right\rangle\right\} &\leq 2R\left\|[C\tilde{\mathbf{x}}^N-\mathbf{b}]_+\right\|_2+2R\|C\tilde{\mathbf{x}}^N-C\hat{\mathbf{x}}^N\|_2 \qquad \text{(A.49)}\\
&\leq 2R\left\|[C\tilde{\mathbf{x}}^N-\mathbf{b}]_+\right\|_2+2R\sigma_x\sqrt{\frac{2\lambda_{\max}(C^{\mathrm{T}}C)}{N}}\left(1+\sqrt{3\ln\frac{1}{\delta}}\right).
\end{aligned}$$

Since the function U is Lipschitz continuous, we obtain

$$\left|U(\tilde{\mathbf{x}}^N)-U(\hat{\mathbf{x}}^N)\right| \leq M_U\left\|\tilde{\mathbf{x}}^N-\hat{\mathbf{x}}^N\right\|_2 \leq M_U\sigma_x\sqrt{\frac{2}{N}}\left(1+\sqrt{3\ln\frac{1}{\delta}}\right).$$

Then

$$U(\hat{\mathbf{x}}^N) = U(\tilde{\mathbf{x}}^N)+(U(\hat{\mathbf{x}}^N)-U(\tilde{\mathbf{x}}^N)) \geq U(\tilde{\mathbf{x}}^N)-M_U\sigma_x\sqrt{\frac{2}{N}}\left(1+\sqrt{3\ln\frac{1}{\delta}}\right). \qquad \text{(A.50)}$$

Substituting (A.49) and (A.50) into (A.48), we conclude that, with probability $1-4\delta$,

$$\begin{aligned}
\varphi(\hat{\boldsymbol{\lambda}}^N)-U(\tilde{\mathbf{x}}^N)+2R\left\|[C\tilde{\mathbf{x}}^N-\mathbf{b}]_+\right\|_2 &\leq C_1\frac{\sigma\sqrt{g(N)J}R^2}{\sqrt{N}}+\frac{2R^2}{\beta N}+\frac{\beta M^2}{2}\\
&+\frac{\sqrt{2}\left(1+\sqrt{3\ln\frac{1}{\delta}}\right)}{\sqrt{N}}\left(M_U\sigma_x+2R\left(\sigma+\sigma_x\sqrt{\lambda_{\max}(C^{\mathrm{T}}C)}\right)\right).
\end{aligned}$$

Proof of Theorem 3. Since $\left\|\nabla\varphi(\lambda)\right\|_2 \leq M$ for any $\boldsymbol{\lambda}\in\Lambda_{2R}$ (see (5)), we have the estimate

$$\sup_{\boldsymbol{\lambda}^1,\boldsymbol{\lambda}^2\in\Lambda_{2R}}\left\langle\nabla\varphi(\boldsymbol{\lambda}^1),\boldsymbol{\lambda}^2-\boldsymbol{\lambda}^1\right\rangle \leq M\cdot 4R.$$

Theorem 4.1 from [26] yields

$$\max_{\boldsymbol{\lambda}\in\Lambda_{2R}}\sum_{t=1}^{N}\xi^t\left\langle\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}^t-\boldsymbol{\lambda}\right\rangle \leq \varepsilon_N,$$

where $\varepsilon_N = 32\times 4MR\exp\left\{-\frac{N}{2m(m+1)}\right\}$. Then

$$\forall\boldsymbol{\lambda}\in\Lambda_{2R}\sum_{t\in I_N}\xi^t\left\langle\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}^t-\boldsymbol{\lambda}\right\rangle \leq \sum_{t=1}^{N}\xi^t\left\langle\nabla\varphi(\boldsymbol{\lambda}^t),\boldsymbol{\lambda}^t-\boldsymbol{\lambda}\right\rangle \leq \varepsilon_N.$$

It follows that

$$\sum_{t\in I_N}\xi^t\left\langle \mathbf{b}-C\mathbf{x}^t,\boldsymbol{\lambda}^t\right\rangle+\max_{\boldsymbol{\lambda}\in\Lambda_R}\left\langle -\sum_{t\in I_N}\xi^t(\mathbf{b}-C\mathbf{x}^t),\boldsymbol{\lambda}\right\rangle\le\varepsilon_N,$$

which can be rewritten as

$$\sum_{t\in I_N}\xi^t\left\langle \mathbf{b}-C\mathbf{x}^t,\boldsymbol{\lambda}^t\right\rangle\le\varepsilon_N-2R\left\|[C\hat{\mathbf{x}}^N-\mathbf{b}]_+\right\|_2. \tag{A.51}$$

Next, by virtue of (3), for each $\mathbf{x}\ge 0$ and $t\in I_N$, we have

$$U(\mathbf{x}^t(\boldsymbol{\lambda}^t))-\left\langle C\mathbf{x}^t(\boldsymbol{\lambda}^t)-\mathbf{b},\boldsymbol{\lambda}^t\right\rangle\ge U(\mathbf{x})-\left\langle C\mathbf{x}-\mathbf{b},\boldsymbol{\lambda}^t\right\rangle.$$

Multiplying the tth inequality by ξ^t, summing the result over all indices from I_N, and taking into account that $\sum_{t\in I_N}\xi^tU(\mathbf{x}^t)\le U(\hat{\mathbf{x}}^N)$ and the functions $u_k(x_k)$, $k=1,\ldots,N$, are concave, we obtain

$$U(\mathbf{x})-U(\hat{\mathbf{x}}^N)+\left\langle \mathbf{b}-C\mathbf{x},\hat{\boldsymbol{\lambda}}^N\right\rangle\le\sum_{t\in I_N}\xi^t\left\langle \mathbf{b}-C\mathbf{x}^t,\boldsymbol{\lambda}^t\right\rangle,$$

where $\hat{\boldsymbol{\lambda}}^N=\sum_{t\in I_N}\xi^t\boldsymbol{\lambda}^t$. Using estimate (A.51), we derive

$$2R\left\|[C\hat{\mathbf{x}}^N-\mathbf{b}]_+\right\|_2+U(\mathbf{x}^*)-U(\hat{\mathbf{x}}^N)+\left\langle \mathbf{b}-C\mathbf{x}^*,\hat{\boldsymbol{\lambda}}^N\right\rangle\le\varepsilon_N. \tag{A.52}$$

Since $\hat{\boldsymbol{\lambda}}^N\in\Lambda_{2R}$ and, hence, $\hat{\boldsymbol{\lambda}}^N\ge 0$, whence $\left\langle \mathbf{b}-C\mathbf{x}^*\hat{\boldsymbol{\lambda}}^N\right\rangle\ge 0$, it follows from (A.51) that $U(\mathbf{x}^*)-U(\hat{\mathbf{x}}^N)\le\varepsilon_N$. Furthermore, since, by the definition of $\boldsymbol{\lambda}^*$, $U(\mathbf{x}^*)\ge U(\mathbf{x})-\left\langle \boldsymbol{\lambda}^*,C\mathbf{x}-\mathbf{b}\right\rangle$ for all $\mathbf{x}\ge 0$, we obtain

$$\begin{aligned}U(\hat{\mathbf{x}}^N)&\le U(\mathbf{x}^*)-\left\langle \boldsymbol{\lambda}^*,\mathbf{b}-C\hat{\mathbf{x}}^N\right\rangle\le U(\mathbf{x}^*)-\min_{\boldsymbol{\lambda}\in\Lambda_R}\left\{\left\langle \boldsymbol{\lambda},\mathbf{b}-C\hat{\mathbf{x}}^N\right\rangle\right\}\\&=U(\mathbf{x}^*)+\max_{\boldsymbol{\lambda}\in\Lambda_R}\left\{\left\langle \boldsymbol{\lambda},C\hat{\mathbf{x}}^N-\mathbf{b}\right\rangle\right\}\le U(\mathbf{x}^*)+R\left\|[C\hat{\mathbf{x}}^N-\mathbf{b}]_+\right\|_2.\end{aligned}$$

Combining this relation with (A.52) yields $R\left\|[C\hat{\mathbf{x}}^N-\mathbf{b}]_+\right\|_2\le\varepsilon_N$. Estimate (11) for the number of iterations of the method follows from the continued inequality

$$\begin{aligned}\varepsilon_N&=32\times 4MR\exp\left\{-\frac{N}{2m(m+1)}\right\}\le\varepsilon\Rightarrow-\frac{N}{2m(m+1)}\\&\le\ln\left(\frac{\varepsilon}{32\times 4MR}\right)\Rightarrow N\ge 2m(m+1)\ln\left(\frac{32\times 4MR}{\varepsilon}\right).\end{aligned}$$

FUNDING

Gasnikov's research was supported by the Russian Foundation for Basic Research, grant no. 18-31-20005 mol_a_ved and 19-31-51001 Scientific mentoring. Dvurechensky's research was supported by the Russian Foundation for Basic Research, grant no. 18-29-03071 mk. Vorontsova's research was supported by the Ministry of Science and Higher Education of the Russian Federation (state assignment no. 075-00337-20-03), project no. 0714-2020-0005.

REFERENCES

1. F. P. Kelly, A. K. Maulloo, and D. K. H. Tan, "Rate control for communication networks: Shadow prices, proportional fairness, and stability," J. Oper. Res. Soc. **49**, 237–252 (1998).
2. D. B. Rokhlin, "Resource allocation in communication networks with large number of users: The stochastic gradient descent method" (2019). https://arxiv.org/abs/1905.04382
3. K. J. Arrow and L. Hurwicz, *Decentralization and Computation in Resource Allocation* (Department of Economics, Stanford Univ., Stanford, CA, 1958).
4. A. Kakhbod, *Resource Allocation in Decentralized Systems with Strategic Agents: An Implementation Theory Approach* (Springer Science & Business Media, New York, 2013).

5. D. E. Campbell, *Resource Allocation Mechanisms* (Cambridge Univ. Press, Cambridge, 1987).
6. E. J. Friedman and S. S. Oren, "The complexity of resource allocation and price mechanisms under bounded rationality," Econ. Theory **6**, 225–250 (1995).
7. Yu. Nesterov and V. Shikhman, "Dual subgradient method with averaging for optimal resource allocation," Eur. J. Oper. Res. **270**, 907–916 (2018).
8. A. Ivanova, P. Dvurechensky, A. Gasnikov, and D. Kamzolov, "Composite optimization for the resource allocation problem" (2018). arXiv preprint arXiv:1810.00595
9. Yu. E. Nesterov, "Method of minimizing convex functions with convergence rate $O(1/k^2)$," Dokl. Akad. Nauk SSSR **269** (3), 543–547 (1983).
10. A. V. Gasnikov, E. V. Gasnikova, Yu. E. Nesterov, and A. V. Chernov, "Efficient numerical methods for entropy-linear programming problems," Comput. Math. Math. Phys. **56**, 514–524 (2016).
11. A. Chernov, P. Dvurechensky, and A. Gasnikov, "Fast primal-dual gradient method for strongly convex minimization problems with linear constraints," *Discrete Optimization and Operations Research: Proceedings of the 9th International Conference, DOOR 2016, Vladivostok, Russia, September 19–23, 2016* (Springer International, Berlin, 2016), pp. 391–403.
12. P. Dvurechensky, A. Gasnikov, E. Gasnikova, S. Matsievsky, A. Rodomanov, and I. Usik, "Primal-dual method for searching equilibrium in hierarchical congestion population games," *Supplementary Proceedings of the 9th International Conference on Discrete Optimization and Operations Research and Scientific School* (*DOOR 2016*) *Vladivostok, Russia, September 19–23, 2016*, pp. 584–595. arXiv:1606.08988
13. A. Anikin, A. Gasnikov, A. Turin, and A. Chernov, "Dual approaches to the minimization of strongly convex functionals with a simple structure under affine constraints," Comput. Math. Math. Phys. **57**, 1262–1276 (2017).
14. P. Dvurechensky, A. Gasnikov, and A. Kroshnin, "Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm," *Proceedings of the 35th International Conference on Machine Learning* (2018), Vol. 80, pp. 1367–1376. arXiv:1802.04367
15. Yu. Nesterov, A. Gasnikov, S. Guminov, and P. Dvurechensky, "Primal-dual accelerated gradient methods with small-dimensional relaxation oracle" (2018). arXiv:1809.05895
16. S. Guminov, P. Dvurechensky, and A. Gasnikov, "On accelerated alternating minimization" (2019). arXiv:1906.03622
17. S. V. Guminov, Yu. E. Nesterov, P. E. Dvurechensky, and A. V. Gasnikov, "Accelerated primal-dual gradient descent with linesearch for convex, nonconvex, and nonsmooth optimization problems," Dokl. Math. **99**, 125–128 (2019).
18. A. Kroshnin, N. Tupitsa, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and C. A. Uribe, "On the complexity of approximating Wasserstein barycenters," *Proceedings of the 36th International Conference on Machine Learning*, Ed. by K. Chaudhuri and R. Salakhutdinov (PMLR, California, US, 2019), Vol. 97, pp. 3530–3540. arXiv:1901.08686
19. C. A. Uribe, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and A. Nedich, "Distributed computation of Wasserstein barycenters over networks," *2018 IEEE Conference on Decision and Control* (*CDC*) (2018), pp. 6544–6549. arXiv:1803.02933
20. D. Dvinskikh, E. Gorbunov, A. Gasnikov, P. Dvurechensky, and C. A. Uribe, "On primal and dual approaches for distributed stochastic convex optimization over networks," *2019 IEEE Conference on Decision and Control* (*CDC*) (2019). arXiv:1903.09844
21. P. Dvurechensky, D. Dvinskikh, A. Gasnikov, C. A. Uribe, and A. Nedic, "Decentralize and randomize: Faster algorithm for Wasserstein barycenters," Adv. Neural Inf. Process. Syst. **31**, 10783–10793 (2018). arXiv:1806.03915
22. D. M. Danskin, *Theory of Maximin* (Sovetskoe Radio, Moscow, 1970) [in Russian].
23. V. F. Demyanov and V. N. Malozemov, *Introduction to Minimax* (Nauka, Moscow, 1972; Wiley, New York, 1974).
24. Yu. Nesterov, "Smooth minimization of nonsmooth functions," Math. Program. **103**, 127–152 (2005).
25. D. B. Yudin and A. S. Nemirovski, "Information complexity and efficient methods for solving convex optimization problems," Ekon. Mat. Metody, No. 2, 357–369 (1976).
26. A. Nemirovski, S. Onn, and U. G. Rothblum, "Accuracy certificates for computational problems with convex structure," Math. Oper. Res. **35**, 52–78 (2010).
27. G. Lan and Y. Zhou, "Random gradient extrapolation for distributed and stochastic optimization," SIAM J. Optim. **28**, 2753–2782 (2018).
28. S. Bubeck, "Convex optimization: Algorithms and complexity," Found. Trends Mach. Learn. **8** (3–4), 231–357 (2015).

29. Yu. Nesterov, "Implementable tensor methods in unconstrained convex optimization," Tech. Rep. (Universite catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2018).
30. A. Gasnikov, P. Dvurechensky, E. Gorbunov, E. Vorontsova, D. Selikhanovych, C. A. Uribe, B. Jiang, H. Wang, S. Zhang, S. Bubeck, Q. Jiang, Y. T. Lee, Y. Li, and A. Sidford, "Near optimal methods for minimizing convex functions with Lipschitz *p*th derivatives," *Proceedings of the Thirty-Second Conference on Learning Theory* (2019), Vol. 99, pp. 1392–1393. http://proceedings.mlr.press/v99/gasnikov19b.html
31. K. Zhou, F. Shang, and J. Cheng, "A simple stochastic variance reduced algorithm with fast convergence rates" (2018). arXiv preprint arXiv:1806.11027
32. K. Zhou, "Direct acceleration of SAGA using sampled negative momentum" (2018). arXiv preprint arXiv:1806.11048
33. F. Niu, B. Recht, C. Re, and S. J. Wright, "Hogwild: A lock-free approach to parallelizing stochastic gradient descent," in *Advances in Neural Information Processing Systems* (2011), pp. 693–701.
34. C. Jin, P. Netrapalli, R. Ge, S. M. Kakade, and M. I. Jordan, "A short note on concentration inequalities for random vectors with subgaussian norm" (2019). arXiv preprint arXiv:1902.03736
35. A. Juditsky and A. Nemirovski, "Large deviations of vector-valued martingales in 2-smooth normed spaces," Tech. Rep. (2008). http://hal.archives-ouvertes.fr/hal-00318071
36. Yu. Nesterov, *Lectures on Convex Optimization*, 2nd ed. (Springer, 2018).
37. Yu. E. Nesterov, Doctoral Dissertation in Mathematics and Physics (Moscow Inst. of Physics and Technology, Dolgoprudnyi, 2013).
38. A. Beck, *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB* (SIAM, Philadelphia, 2014).

Translated by I. Ruzanova