Optimization of Buffer Networks via DC Programming

Chengyan Zhao, *Member, IEEE*, Kazunori Sakurama, *Member, IEEE*, and Masaki Ogura, *Member, IEEE*

***Abstract*—This brief is concerned with the H^2 and H^∞ norm-constrained optimization problems of dynamic buffer networks. The extended network model is introduced first, wherein the weights of all edges can be tuned independently. Because of the emerging nonconvexity of the extended model, previous results of positive linear systems failed to address this situation. By resorting to the log–log convexity of a class of nonlinear functions called posynomials, the optimization problems can be reduced to differential convex programming problems. The proposed framework is illustrated for large-scale networks.**

***Index Terms*—Positive linear system, H^2 norm, H^∞ norm, buffer network, car-sharing service, posynomial, DC programming, nonconvex optimization.**

I. Introduction

THE STUDY of buffer networks has recently emerged when the nodes among them behave as buffers to exchange the inflow/outflow with their neighboring nodes. The interest for the conduct of this brief has been triggered by a variety of applications, such as environment [1], power [2], and network-on-chip design [3]. There is also an emerging application in the area of mobility systems, such as the optimization of traffic flows [4], [5], [6] and the design of efficient and effective mobility-on-demand systems [7], [8]. One of the standard models used for the study of buffer networks is the positive linear system model [9]. Based on this model, many fundamental problems related to control and optimizations have been studied. For example, stability analysis has been investigated in [10], [11]. Similarly, the positive linear system model allows the study of control and optimization problems [12]. Moreover, the robust control problem has been studied in [13], [14].

To achieve network optimization, the problem formulation in [12] assumes that the coefficient on the outlet edges of a node is constrained to have only one variable. However, in the real buffer network optimization problem, it is ideal that all edges in the network are independently adjusted to achieve maximum flexibility. Therefore, in this brief, we first extend the buffer network model in [12] to allow each edge to be designed independently. This extension expands the adjustable area of the state matrix from the diagonals to all other entries. We then formulate the H^2 and H^∞ norm-constrained optimization problems and postulate an assumption on the cost function and variable constraints. Finally, we propose a tractable but rigorous framework to solve the optimization problems via a standard nonconvex optimization programming referred to as differential convex (DC) programming [15], [16].

During the past decades, theoretical research on system control of buffer networks mainly derived from positive linear systems [17]. The authors in [18] and [19] indicated that structured stabilization problems for positive linear systems can be solved by a linear program and linear matrix inequalities, respectively. It was shown in [20] that for positive systems, the Kalman–Yakubovich–Popov lemma can be formulated in terms of a diagonal matrix. In [12], the authors presented a framework to solve the optimal control of positive linear systems via geometric programming. However, the aforementioned results reveal the limitation on the convexity of positive linear systems only when the diagonals of the state matrix are adjustable. Moreover, to efficiently reduce the buffer in the transportation and power networks, all the edges should be designed independently with complete freedom. Motivated by these results, we first extend the adjustable field from the diagonals to all entries. Although the extended problems become nonconvex optimization problems, the proposed framework using standard DC programming can also successfully solve them with high efficiency.

The brief is structured as follows. The extended buffer network model is introduced in Section II. In Section III, H^2 and H^∞ optimization problems are presented. Section IV is devoted to the proposed results in terms of DC program. Numerical simulations are presented in Section V.

The following notations are used in this brief. Let $\mathbb{R}$, $\mathbb{R}_+$, and $\mathbb{R}_{++}$ denote the set of real, nonnegative, and positive numbers, respectively. The set of corresponding vectors of size n are denoted by $\mathbb{R}^n$, $\mathbb{R}^n_+$, and $\mathbb{R}^n_{++}$, respectively. We let $\mathbb{1}$ denote a column vector with all entries set to unity. The identity and zero matrices of order n are denoted by I_n and O_n, respectively. The real matrix A is said to be non-negative (positive), and is denoted by $A \geq 0$ ($A > 0$), if all entries of A are non-negative (positive). The notion $B < A$ is defined

Manuscript received 12 September 2022; accepted 30 September 2022. Date of publication 10 October 2022; date of current version 9 February 2023. This work was supported in part by the Joint Project of Kyoto University and Toyota Motor Corporation, titled "Advanced Mathematical Science for Mobility Society." This brief was recommended by Associate Editor X. Wu. *(Corresponding author: Masaki Ogura.)*

Chengyan Zhao is with the Graduate School of Science and Engineering, Ritsumeikan University, Kusatsu 525-8577, Shiga, Japan (e-mail: c-zhao@fc.ritsumei.ac.jp).

Kazunori Sakurama is with the Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan (e-mail: sakurama@i.kyoto-u.ac.jp).

Masaki Ogura is with the Graduate School of Information Science and Technology, Osaka University, Suita 565-0871, Osaka, Japan (e-mail: m-ogura@ist.osaka-u.ac.jp).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSII.2022.3212693.

Digital Object Identifier 10.1109/TCSII.2022.3212693

as $B - A < 0$. Let the Hadamard product of matrices A and B be denoted by $A \odot B$. Let $A \otimes B$ denote the Kronecker product of matrices A and B. If A and B are square, the Kronecker sum of A and B is defined as $A \oplus B = A \otimes I_m + I_n \otimes B$, where n and m denote the orders of A and B, respectively. We define the entry-wise exponential operation of a real vector v as $\exp[v] = [\exp v_1, \dots, \exp v_n]^\top$ and the entry-wise logarithm operation as $\log[v] = [\log v_1, \dots, \log v_n]^\top$. For a vector v with scalar entries $v_1, \dots, v_n$, we use $\mathrm{diag}(v_1, \dots, v_n)$ to denote the diagonal matrix.

II. Dynamical Buffer Network

In this section, we first give the description of our model in Section II-A. We then present an example in Section II-B.

A. Model Description

Consider a weighted, directed buffer network (for example, e.g., [9], [12]) defined by the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, where $\mathcal{V} = \{v_1, \dots, v_n\}$ denotes the set of n nodes within the network and $\mathcal{E} = \{e_1, \dots, e_m\} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of directed edges. Because the graph $\mathcal{G}$ is weighted, a positive and fixed weight w_{e_ℓ} is assigned to an edge e_ℓ. The scalar w_{ij} is defined as the weight of the edge (i, j). Thus, the adjacency matrix $A_{\mathcal{G}} \in \mathbb{R}^{n\times n}$ of the graph $\mathcal{G}$ is given by

$$[A_{\mathcal{G}}]_{ij} = \begin{cases} w_{ji}, & \text{if } (j,i) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases}$$

The set of in-neighborhoods of node i is defined by $\mathcal{N}_i^{\mathrm{in}} = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$. Similarly, the set of out-neighborhoods is defined by $\mathcal{N}_i^{\mathrm{out}} = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$.

In this brief, we place the following assumption on the structure of the network. Suppose that there exist two special sets of nodes that serve as origins (i.e., the nodes having an empty in-neighborhood node) and destinations (i.e., the nodes having an empty out-neighborhood node). We let $\mathcal{V}_o = \{1, \dots, |\mathcal{V}_o|\}$ and $\mathcal{V}_d$ denote the set of origins and destinations of the buffer network, respectively. We then consider the dynamic process of the buffer network expressed by the following differential equations,

$$\frac{dx_i}{dt} = \begin{cases} f_i^{\mathrm{in}} - \sum_{j\in\mathcal{N}_i^{\mathrm{out}}} u_{ij}, & \text{if } i \in \mathcal{V}_o, \\ \sum_{j\in\mathcal{N}_i^{\mathrm{in}}} u_{ji} - \sum_{j\in\mathcal{N}_i^{\mathrm{out}}} u_{ij}, & \text{if } i \notin \mathcal{V}_o \cup \mathcal{V}_d, \\ \sum_{j\in\mathcal{N}_i^{\mathrm{in}}} u_{ji} - f_i^{\mathrm{out}}, & \text{if } i \in \mathcal{V}_d, \end{cases} \tag{1}$$

where x_i $(i = \{1, \dots, n\})$ represents buffer variable in node i, u_{ij} is the volume of flow from node i to j, and f_i^{in} and f_i^{out} denote the inlet and outlet effects, respectively.

In this brief, the flows among the buffer network are assumed to obey the following linear form:

$$f_i^{\mathrm{out}} = \beta_i x_i, \quad u_{ij} = \delta_{ij} w_{ij} x_i, \tag{2}$$

where $\beta = \{\beta_i\}_{i\in\mathcal{V}_d}$ and $\delta = \{\delta_{ij}\}_{(i,j)\in\mathcal{E}}$ are the parameters to be tuned in the next section. Herein, unlike the problem formulation in [12], where the volume of flow $u_{ij} = \phi_i w_{ij} x_i$ depends on one parameter, we allow u_{ij} to be designed independently on the parameter of each edge.

To measure the performance of the buffer network, we adopt the output $y = [x^\top \ \alpha u^\top]^\top$, where $\alpha > 0$ is a weight constant and $u \in \mathbb{R}_+^{n\times n}$ includes the information of the edges. If we define the matrix B and D by

$$B_{ij} = \begin{cases} \beta_i, & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases} \qquad D_{ij} = \begin{cases} \delta_{ji}, & \text{if } (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise,} \end{cases}$$

then the dynamic model can then be expressed as

$$\Sigma : \begin{cases} \dfrac{dx}{dt} = \big(D \odot A_{\mathcal{G}} - \mathrm{diag}\big(\mathbb{1}^\top (D \odot A_{\mathcal{G}})\big) - B\big)x + G^{\mathrm{in}} f^{\mathrm{in}}, \\ y = G^{\mathrm{out}}(\delta) x, \end{cases}$$

where the vector f^{in} and the matrices G^{in}, $G^{\mathrm{out}}(\delta)$ are defined by $f^{\mathrm{in}} = [f_1^{\mathrm{in}} \cdots f_{|\mathcal{V}_o|}^{\mathrm{in}}]^\top$ and

$$G^{\mathrm{in}} = \begin{bmatrix} I_{|\mathcal{V}_o|} \\ O_{n-|\mathcal{V}_o|,|\mathcal{V}_o|} \end{bmatrix}, \quad G^{\mathrm{out}}(\delta) = \begin{bmatrix} I_n \\ \alpha H(\delta) \end{bmatrix}. \tag{3}$$

The matrix $H(\delta)$ is defined by $H(\delta)_{\ell i} = w_{e_\ell}$ if $i = e_\ell(1)$ and $H(\delta)_{\ell i} = 0$ otherwise. For each edge e_ℓ, we use the notation $e_\ell = (e_\ell(1), e_\ell(2))$, wherein the nodes $e_\ell(1)$ and $e_\ell(2)$ denote the origin and destination of the edge, respectively. Since G^{in} and $G^{\mathrm{out}}(\delta)$ are nonnegative matrices and $D \odot A_{\mathcal{G}} - \mathrm{diag}(\mathbb{1}^\top (D \odot A_{\mathcal{G}})) - B$ is the Metzler matrix, following the definition in [17], the dynamic model (3) is the positive linear system.

B. Example: Car-Sharing Service

Let us consider a one-way car-sharing service wherein the stations providing parking slots for customers renting vehicles at any stations. In spite of its convenience, this service has the drawback of the uneven distribution of vehicles, which causes parking slots or vehicles to be unavailable at particular stations. To reduce the uneven distribution, dynamics pricing is promising, which controls the demand of customers by adjusting usage prices in real-time. Here, we discuss how we can determine the prices for efficient control.

First, we construct a mathematical model of the system of the one-way car-sharing service according to the authors' previous paper [21]. Let n be the number of stations in an area. Each station corresponds to a node in $\mathcal{V}$. Let $x_i \in \mathbb{R}_+$ $(i \in \mathcal{V})$ be the expectation of the number of vehicles parking at station i. The possible usage between stations is described by an edge set $\mathcal{E}$. Let $u_{ij} \in \mathbb{R}_+$ $((i, j) \in \mathcal{E})$ be the expectation of the number of customers who travel from station i to j within a time interval. Let $f_i^{\mathrm{in}} \in \mathbb{R}_+$ $(f_i^{\mathrm{out}} \in \mathbb{R}_+)$ be the expectation of the number of vehicles moving to (from) this area from (to) other areas. Then, this system is modeled as equation (1).

Next, we construct a model of the demand of customers which can change with prices. Assume that the expectation of the demand is $\bar{u}_{ij}(t) \in \mathbb{R}_+$ when the price is $\bar{p}_{ij}(t) \in \mathbb{R}_+$ and that the change of the demand is governed with an affine model around this point. Let $p_{ij} \in \mathbb{R}_+$ be the price for traveling from stations i to j, and let $\delta_{ij} \in \mathbb{R}_+$ be the price elasticity. Then, the affine model is given as

$$u_{ij} = \bar{u}_{ij} - \delta_{ij}(p_{ij} - \bar{p}_{ij}). \tag{4}$$

As a pricing strategy, the price p_{ij} is adjusted according to the number x_i of vehicles at station i as follows:

$$p_{ij} = \hat{p}_{ij} - w_{ij} x_i, \tag{5}$$

where $\hat{p}_{ij} \in \mathbb{R}_+$ and $w_{ij} \in \mathbb{R}_+$ are design parameters. We set $\hat{p}_{ij} = \bar{p}_{ij} + \bar{u}_{ij}/\delta_{ij}$, and from (4) and (5), the demand model is reduced to $u_{ij} = \bar{u}_{ij} - \delta_{ij}(p_{ij} - \bar{p}_{ij}) = \bar{u}_{ij} - \delta_{ij}(\hat{p}_{ij} - w_{ij}x_i - \bar{p}_{ij}) = \delta_{ij}w_{ij}x_i$. This corresponds to the equation (2).

III. Problem Formulation

Following the formulation in the previous section, we assume that the decay rates of node i and directed edge ij can be tuned by the parameters β_i and δ_{ij} to improve the performance of the buffer network. Calculation of the sum of all variables yields the cost function $L(\beta, \delta) = \sum_{i\in\mathcal{V}_d} g_i(\beta_i) + \sum_{(i,j)\in\mathcal{E}} h_{ij}(\delta_{ij})$, where the variables are tuned within the following intervals

$$0 < \beta_i \leq \bar{\beta}_i, \; 0 < \delta_{ij} \leq \bar{\delta}_{ij}. \tag{6}$$

In this brief, we adopt H^2 and H^∞ norms to measure the performance of the buffer network. If system Σ is stable, then the H^2 norm is defined by $\|\Sigma\|_2 = \sqrt{\int_0^\infty \mathrm{tr}(\Omega(t)\Omega(t)^\top)dt}$, where $\Omega(\cdot)$ is the impulse response of the system Σ, and $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. Likewise, if system Σ is stable, then the H^∞ norm of the system is defined as $\|\Sigma\|_\infty = \sup_{w\in\mathcal{L}^2} \|\Omega * w\|_2/\|w\|_2$, where $\Omega \neq 0$ and $*$ denotes a convolution product and $\mathcal{L}^2 = \{f : [0,\infty) \to \mathbb{R}^n \mid \int_0^\infty \|f(t)\|^2 dt < \infty\}$ denotes the space of Lebesgue measurement functions.

We are now ready to state the optimization problems studied in this brief.

Problem 1 (H^2 Norm-Constrained Optimization): Given the desired H^2 performance $\gamma_2 > 0$, find the parameters β and δ minimizing the parameter tuning cost $L(\beta, \delta)$, under the constraint that the parameter constraints (6) are satisfied, Σ is stable, and the system requirement $\|\Sigma\|_2 < \gamma_2$ is satisfied.

Problem 2 (H^∞ Norm-Constrained Optimization): Given the desired H^∞ performance $\gamma_\infty > 0$. Find the parameters β and δ minimizing the parameter tuning cost $L(\beta, \delta)$, under the constraint that the parameter constraints (6) are satisfied, Σ is stable, and the system requirement $\|\Sigma\|_\infty < \gamma_\infty$ is satisfied.

The difficulty of solving Problem 1 and Problem 2 mainly stems from the extended model (3), with the introduction of δ_{ij}, the problems are no longer convex and can not be solved by GP programming.

IV. Main Results

In this section, we present the solutions to the H^2 and H^∞ norm-constrained optimization problems in terms of DC programming [15], [16]. We begin this section by introducing the preliminary knowledge of posynomials, DC functions, and DC program to derive the main results.

Definition 1: Let $v_1, \ldots,$ and v_n denote n real positive variables. We state that a real function $g(v)$ is a *monomial* if $c > 0$ and $a_1, \ldots, a_n \in \mathbb{R}$ such that $g(v) = cv_1^{a_1} \cdots v_n^{a_n}$. We state that a real function $f(v)$ is a *posynomial* [22] if f is the sum of the monomials of v.

The following lemma shows the log-convexity of posynomials [22].

Lemma 1: If $f : \mathbb{R}^n_{++} \to \mathbb{R}_{++}$ is a posynomial function, then the log-transformed function $F : \mathbb{R}^n \to \mathbb{R} : w \mapsto \log[f(\exp[w])]$ is convex.

The property shown in Lemma 1 enables us to build a relationship with a general class of mathematical programming that deals with the difference between two convex functions, called DC programming.

Definition 2 (DC Functions [15]): Let C be a convex subset of $\mathbb{R}^n$. A real-valued function $f : C \to \mathbb{R}$ is called a *DC function* on C if there exist two convex functions $g, h : C \to \mathbb{R}$ such that f can be expressed in the form $f(x) = g(x) - h(x)$.

In principle, every continuous function can be approximated by a DC function with the desired precision. Based on decomposition methods [15], it is possible to convert a nonconvex optimization problem to a DC programming problem.

Definition 3 (DC Programming Problem [15]): Programming problems dealing with DC functions are called *DC programming problems*. Let C be a closed convex subset of $\mathbb{R}^n$, and the general form of DC programming problem considered in this brief is

$$\begin{aligned} &\underset{x\in C}{\text{minimize}} \;\; f_0(x) \\ &\text{subject to} \;\; f_i(x) \leq 0, \; i = 1, \ldots, m, \end{aligned}$$

where $f_0(x) = g_0(x) - h_0(x)$ and $f_i(x) = g_i(x) - h_i(x)$ are the differences of the two convex functions.

To optimally solve the DC programming problem, we can choose the branch-and-bound type and outer-approximation algorithms [23], which lead to more efficient procedures. Subject to the proper assumption for the cost function, we show that Problems 1 and 2 can be transformed to DC programming problems.

Assumption 1: The functions $g_i(\beta_i)$ and $h_{ij}(\delta_{ij})$ are posynomials for all i and j.

To present the results concisely, we let G_j^{in} and $G_i^{\mathrm{out}}(\delta)$ denote the jth column and ith row of the matrices G^{in} and $G^{\mathrm{out}}(\delta)$, respectively, and define the n^2-dimensional column and row vectors $\widetilde{G}^{\mathrm{in}} = \sum_{j=1}^{|\mathcal{V}_o|} G_j^{\mathrm{in}} \otimes G_j^{\mathrm{in}}$ and $\widetilde{G}^{\mathrm{out}}(\delta) = \sum_{i=1}^{n+m} G_i^{\mathrm{out}}(\delta) \otimes G_i^{\mathrm{out}}(\delta)$.

Theorem 1: Under Assumption 1, the solution of Problem 1 is given by the solution of the following DC programming problem,

$$\begin{aligned} &\underset{\substack{\gamma,\eta\in\mathbb{R} \\ \mu\in\mathbb{R}^{n^2}_{++}}}{\text{minimize}} && \log L(\exp[\gamma], \exp[\eta]) \\ &\text{subject to} && \log[\widetilde{G}^{\mathrm{out}}(\exp[\eta])\mu] - \log[\gamma_2^2] < 0, \\ &&& \log[(\exp[\eta] \odot A_{\mathcal{G}}) \oplus (\exp[\eta] \odot A_{\mathcal{G}})\mu + \widetilde{G}^{\mathrm{in}}] \\ &&& - \log[(\mathrm{diag}(\mathbb{1}^\top \exp[\eta] \odot A_{\mathcal{G}}) + \exp[\gamma]) \\ &&& \oplus(\mathrm{diag}(\mathbb{1}^\top \exp[\eta] \odot A_{\mathcal{G}}) + \exp[\gamma])\mu] < 0, \\ &&& \log[\exp[\gamma]] - \log[\exp[\bar{\gamma}]] < 0, \\ &&& \log[\exp[\eta]] - \log[\exp[\bar{\eta}]] < 0. \end{aligned}$$

The solution of Problem 1 is then given by

$$B = \exp[\gamma], \;\; D = \exp[\eta]. \tag{7}$$

Proof: Based on Section II, model (3) is a standard positive linear system. Resorting to the stability result of positive linear system in Proposition 3.4 [12], we can show that if the buffer network Σ is internally stable and $\|\Sigma\|_2 < \gamma_2$, then the following problem is equivalent to Problem 1:

$$\underset{\beta,\delta\in\Theta,\mu\in\mathbb{R}_{++}^{n^2}}{\text{minimize}} \quad L(\beta,\delta) \tag{8a}$$
$$\text{subject to} \quad \widetilde{G}^{\text{out}}(\delta)\mu < \gamma_2^2, \tag{8b}$$
$$(A_{\mathcal{G}}(\beta,\delta)\oplus A_{\mathcal{G}}(\beta,\delta))\mu+\widetilde{G}^{\text{in}} < 0, \tag{8c}$$
$$(6). \tag{8d}$$

From the observation in (3), $A_{\mathcal{G}}(\beta,\delta)$ includes the difference of posynomials, therefore, (8) is no longer a geometric programming (GP) programming problem [22]. For this situation, we resort to the log–log convexity of the posynomials in Lemma 1 for reducing (8) into DC programming problem. According to Definition 1 and Assumption 1, the sum of g_i and h_{ij} in (8a) is the sum of monomials in essence. Thus, (8a) is also a posynomial function. The log–log transformation of (8a) subtracts 0 that satisfies the DC functions in Definition 3. For the constraint (8b), the entries in which the vector obtained by the sum of the Kronecker product of the non-negative vectors $\widetilde{G}^{\text{in}}$ and $\widetilde{G}^{\text{out}}$ are also posynomials. Because μ is a positive vector and γ_2^2 is a positive number, (8b) is the difference between two posynomials, which can be successfully transformed to DC functions by the log–log transformation. For (8c), we let $A_{\mathcal{G}}(\beta,\delta)=\tilde{A}_{\mathcal{G}}(\delta)-R(\beta,\delta)$, where all the elements in $\tilde{A}_{\mathcal{G}}(\delta)=D\odot A_{\mathcal{G}}$ and $R(\beta,\delta)=\text{diag}(\mathbb{1}^{\top}(D\odot A_{\mathcal{G}}))+B$ are either posynomials or zero. Thus, (8c) is equivalent to $(\tilde{A}_{\mathcal{G}}(\delta)\oplus\tilde{A}_{\mathcal{G}}(\delta))\mu+\widetilde{G}^{\text{in}}-(R(\beta,\delta)\oplus R(\beta,\delta))\ \mu<0$, which shows the difference between the two materials. Similarly, (8c) can also be transformed to DC functions. For (8d), we can directly obtain the variable constraints in the form of DC functions from (6). Hence, Theorem 1 is a DC programming problem. This completes the proof of theorem. ■

Theorem 1 shows that Problem 1 can be turned into an equivalent DC program. A similar result is true also for Problem 2.

Theorem 2: Under the aforementioned assumptions and lemma, the solution of Problem 2 is given by the solution of the following DC programming problem,

$$\begin{aligned}
\underset{\substack{\gamma,\eta\in\mathbb{R},\xi,\zeta\in\mathbb{R}_{++}^{n}\\ \mu\in\mathbb{R}_{++}^{n_{\text{in}}},\nu\in\mathbb{R}_{++}^{n_{\text{out}}}}}{\text{minimize}} \quad & \log L(\exp[\gamma],\exp[\eta])\\
\text{subject to} \quad & \log[G^{\text{out}}(\exp[\eta])\xi]-\log[\gamma_\infty\nu]<0,\\
& \log[(\exp[\eta]\odot A_{\mathcal{G}})\xi+G^{\text{in}}\mu]-\\
& \log[(\text{diag}(\mathbb{1}^{\top}\exp[\eta]\odot A_{\mathcal{G}})+\exp[\gamma])\xi]<0,\\
& \log[G^{\text{in}\top}\zeta]-\log[\gamma_\infty\mu]<0,\\
& \log[(\exp[\eta]\odot A_{\mathcal{G}})^{\top}\zeta+G^{\text{out}}(\exp[\eta])^{\top}\nu]-\\
& \log[(\text{diag}(\mathbb{1}^{\top}\exp[\eta]\odot A_{\mathcal{G}})+\exp[\gamma])^{\top}\zeta]<0,\\
& \log[\exp[\gamma]]-\log[\exp[\bar{\gamma}]]<0,\\
& \log[\exp[\eta]]-\log[\exp[\bar{\eta}]]<0.
\end{aligned}$$

The solution of Problem 2 is then given by (7).

Proof: Relying on the H^{∞} stability results in Proposition 4.4 [12] which shows that if the positive linear system Σ is internally stable and $\|\Sigma\|_\infty<\gamma_\infty$, we can show that the following optimization problem is equivalent to

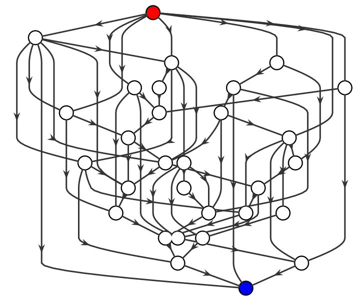

Fig. 1. Buffer network comprising 30 nodes (red: origin, blue: destination).

Problem 2:

$$\underset{\substack{\beta,\delta\in\Theta,\xi,\zeta\in\mathbb{R}_{++}^{n}\\ \mu\in\mathbb{R}_{++}^{n_{\text{in}}},\nu\in\mathbb{R}_{++}^{n_{\text{out}}}}}{\text{minimize}} \quad L(\beta,\delta) \tag{9a}$$
$$\text{subject to} \quad G^{\text{out}}(\delta)\xi<\gamma_\infty\nu, \tag{9b}$$
$$A_{\mathcal{G}}(\beta,\delta)\xi+G^{\text{in}}\mu<0, \tag{9c}$$
$$G^{\text{in}\top}\zeta<\gamma_\infty\mu, \tag{9d}$$
$$A_{\mathcal{G}}(\beta,\delta)^{\top}\zeta+G^{\text{out}\top}(\delta)\nu<0, \tag{9e}$$
$$(6). \tag{9f}$$

Based on the results in Theorem 1, we start the proof from (9b). Because $G^{\text{out}}(\delta)$, ξ, and ν are non-negative matrices, the entries in the product vectors $G^{\text{out}}(\delta)\xi$ and $\gamma_\infty\nu$ are either posynomials or zero. In a manner similar to the proof of (8c), constraint (9c) is equivalent to $\tilde{A}_{\mathcal{G}}(\delta)\xi+G^{\text{in}}\mu-R(\beta,\delta)\xi<0$, which shows the difference between the two non-negative matrices. Similarly, the existence of DC functions from (9d) and (9e) can also be obtained using the ways of (9b) and (9c). By taking Lemma 1, problem (9) is reduced to the DC programming problem. ■

V. Numerical Simulation

In the numerical simulation, we build a buffer network with a relatively large size (30 nodes), as shown in Fig. 1. We set the network structure to contain one origin, one destination, and 60 edges. Thus, we fix $A_{\mathcal{G}}$ and $\beta=[0,\ldots,0,\beta_1]$. In the traffic control problem, since the buffer content on each node is the major concern, we set $\alpha=0$. In addition, we let $G^{\text{in}}=[1,0,\ldots,0]$ and $G^{\text{out}}=\mathbb{1}^{\top}$ for the network. We simply set the linear correlation $g_i(\beta_i)=\beta_i$ and $h_{ij}(\delta_{ij})=\delta_{ij}$ for the decay rate on the destination and the flow rate on each edge, respectively, thus indicating that the increment in the variable has a positive correlation with the tuning cost.

We adopt the H^{∞} norm-optimized DC programming problem under the total parameter cost constraint $\bar{L}$ varying between [100, 500]. As the solver for simulation, we resort to the proximal bundle method [23] to solve the DC program. As a comparison between the model in [12] and this brief and their corresponding computation frameworks, we reduce the DC problem to GP problem through adding the constraint $\delta_{ij}=\delta_{ik}$ for all k, which coincides with that in [12]. In Fig. 2, the triangles show the optimized γ_∞ for various $\bar{L}$, and the circles show the optimal solution of the original problem [12] solved by geometric programming. Fig. 2 shows that under the same

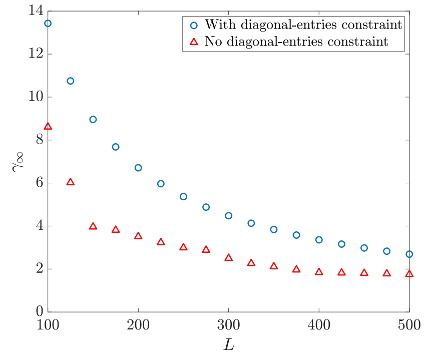

Fig. 2. Optimized H^∞ norm versus the cost constraints.

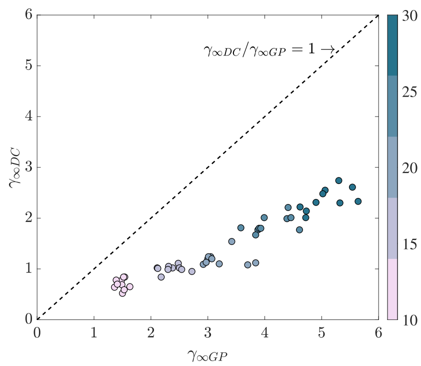

Fig. 3. Plot of the $\gamma_{\infty DC}/\gamma_{\infty GP}$ ratio versus the size of network. $\gamma_{\infty DC}$: the optimized H^∞ norm solved by Theorem 2. $\gamma_{\infty GP}$: optimal H^∞ norm solved by geometric programming (colorbar: size of network).

cost constraint $\bar{L}$, our results exhibit a higher performance in the design of the buffer network. Due to the non-convexity of the non-constrained problem, DC programming would not necessarily ensure the globally optimal solution. In other words, it is within the bounds of possibility that DC programming shows the lower performance than GP. To make a comprehensive research on various network sizes and structures, we set the ratio between the number of variables and the cost constraint to be a constant $N/L = 1/5$ as the offset against the variety of the network size. This effectiveness is reinforced by the comprehensive simulations in Fig. 3 that the $\gamma_{\infty DC}/\gamma_{\infty GP}$ ratio represented by the color dots are all located below the $\gamma_{\infty DC}/\gamma_{\infty GP} = 1$ dashed line.

VI. Conclusion

In this brief, we studied the H^2 and H^∞ norm-constrained optimization problems for dynamic buffer networks. In the problem formulation, we relaxed the adjustable region of the buffer network to all its edges, and formulated the extended optimization problems. Adhering to the mild assumption of the cost function, the extended optimization problems can be reduced to standard DC programming problems through the log–log convexity property of the posynomials. Numerical simulations are presented to illustrate the numerical scalability of the proposed optimization framework as well as its effectiveness. Several research directions should be pursued. One such direction is considering the L^1-induced norm. Another direction is the application of our findings to the socio-technical systems and higher-order multi-agent systems [24].

References

[1] A. Michez, H. Piégay, P. Lejeune, and H. Claessens, “Multi-temporal monitoring of a regional riparian buffer network (>12,000 km) with LiDAR and photogrammetric point clouds,” *J. Environ. Manag.*, vol. 202, pp. 424–436, Nov. 2017.
[2] N. Vafamand, M. H. Khooban, T. Dragicevic, and F. Blaabjerg, “Networked fuzzy predictive control of power buffers for dynamic stabilization of DC microgrids,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1356–1362, Feb. 2019.
[3] C. Pu, W. Cui, J. Wu, and J. Yang, “Bufferless transmission in complex networks,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 7, pp. 893–897, Jul. 2018.
[4] G. Como, K. Savla, D. Acemoglu, M. A. Dahleh, and E. Frazzoli, “Stability analysis of transportation networks with multiscale driver decisions,” *SIAM J. Control Optim.*, vol. 51, no. 1, pp. 230–252, 2013.
[5] J. Wu and Y. Xia, “Complex-network-inspired design of traffic generation patterns in communication networks,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 64, no. 5, pp. 590–594, May 2017.
[6] P. Grandinetti, C. Canudas-de-Wit, and F. Garin, “Distributed optimal traffic lights design for large-scale urban networks,” *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 950–963, May 2019.
[7] G. C. Calafiore, C. Bongiorno, and A. Rizzo, “A robust MPC approach for the rebalancing of mobility on demand systems,” *Control Eng. Pract.*, vol. 90, pp. 169–181, Sep. 2019.
[8] S. Illgen and M. Höck, “Literature review of the vehicle relocation problem in one-way car sharing networks,” *Transp. Res. B Methodol.*, vol. 120, pp. 193–204, Feb. 2019.
[9] A. Rantzer and M. E. Valcher, “A tutorial on positive systems and large scale control,” in *Proc. 57th IEEE Conf. Decis. Control*, 2018, pp. 3686–3697.
[10] E. Lovisari, G. Como, and K. Savla, “Stability of monotone dynamical flow networks,” in *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 2384–2389.
[11] S. Coogan and M. Arcak, “A compartmental model for traffic networks and its dynamical behavior,” *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2698–2703, Oct. 2015.
[12] M. Ogura, M. Kishida, and J. Lam, “Geometric programming for optimal positive linear systems,” *IEEE Trans. Autom. Control*, vol. 65, no. 11, pp. 4648–4663, Nov. 2020.
[13] M. Rotkowitz, R. Cogill, and S. Lall, “Convexity of optimal control over networks with delays and arbitrary topology,” *Int. J. Syst. Control Commun.*, vol. 2, nos. 1–3, pp. 30–54, 2010.
[14] K. Savla, G. Como, and M. A. Dahleh, “Robust network routing under cascading failures,” *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 1, pp. 53–66, Jan.-Jun. 2014.
[15] R. Horst and N. V. Thoai, “DC programming: Overview,” *J. Optim. Theory Appl.*, vol. 103, no. 1, pp. 1–43, 1999.
[16] H. A. Le Thi and T. P. Dinh, “DC programming and DCA: thirty years of developments,” *Math. Program.*, vol. 169, no. 1, pp. 5–68, 2018.
[17] L. Farina and S. Rinaldi, *Positive Linear Systems: Theory and Applications*, New York, NY, USA: Wiley, 2000.
[18] M. A. Rami and F. Tadeo, “Controller synthesis for positive linear systems with bounded controls,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 2, pp. 151–155, Feb. 2007.
[19] H. Gao, J. Lam, C. Wang, and S. Xu, “Control for stability and positivity: Equivalent conditions and computation,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 52, no. 9, pp. 540–544, Sep. 2005.
[20] A. Rantzer, “An extended Kalman-Yakubovich-Popov lemma for positive systems,” *IFAC-PapersOnLine*, vol. 48, no. 11, pp. 242–245, 2015.
[21] T. Ikeda, K. Sakurama, and K. Kashima, “Multiple sparsity constrained control node scheduling with application to rebalancing of mobility networks,” *IEEE Trans. Autom. Control*, vol. 67, no. 8, pp. 4314–4321, Aug. 2022.
[22] S. Boyd, S.-J. Kim, L. Vandenberghe, and A. Hassibi, “A tutorial on geometric programming,” *Optim. Eng.*, vol. 8, no. 1, pp. 67–127, 2007.
[23] W. Oliveira, “Proximal bundle methods for nonsmooth DC programming,” *J. Glob. Optim.*, vol. 75, no. 2, pp. 523–563, 2019.
[24] X. Gong, Y. Cui, T. Wang, J. Shen, and T. Huang, “Distributed prescribed-time consensus observer for high-order integrator multi-agent systems on directed graphs,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 69, no. 4, pp. 2216–2220, Apr. 2022.