Rate Allocation and Content Placement in Cache Networks

Khashayar Kamran, Armin Moharrer, Stratis Ioannidis, and Edmund Yeh
Electrical and Computer Engineering, Northeastern University, Boston, MA, USA
{kamrank, amoharrer, ioannidis, eyeh}@ece.neu.edu

***Abstract*—We introduce the problem of optimal congestion control in cache networks, whereby *both* rate allocations and content placements are optimized *jointly*. We formulate this as a maximization problem with non-convex constraints, and propose solving this problem via (a) a Lagrangian barrier algorithm and (b) a convex relaxation. We prove different optimality guarantees for each of these two algorithms; our proofs exploit the fact that the non-convex constraints of our problem involve DR-submodular functions.**

***Index Terms*—Congestion control, caching, rate control, utility maximization, DR-submadular maximization, non-convex optimization**

I. Introduction

Traffic engineering and congestion control have played a crucial role in the stability and scalability of communication networks since the early days of the Internet. They have been extremely active research areas since the seminal work by Kelly et al. [1], who studied optimal rate control subject to link capacity constraints. Formally, given a network $G(\mathcal{V}, \mathcal{E})$ with nodes $v \in \mathcal{V}$, links $e \in \mathcal{E}$, and flows $n \in \mathcal{N}$, Kelly et al. [1] studied the following convex optimization problem:

$$\max_{\boldsymbol{\lambda}} \quad \sum_{n\in\mathcal{N}} U_n(\lambda_n) \tag{1a}$$

$$\text{s.t.} \quad \rho_e(\boldsymbol{\lambda}) \leq C_e, \quad \forall e \in \mathcal{E}, \tag{1b}$$

where $\boldsymbol{\lambda} = [\lambda_n]_{n\in\mathcal{N}} \in \mathbb{R}_+^{|n|}$ is the vector of rate allocations λ_n, $n \in \mathcal{N}$ across flows, $\rho_e : \mathbb{R}^{|n|} \to \mathbb{R}_+$, $C_e \in \mathbb{R}_+$ are the loads and capacities of links $e \in \mathcal{E}$, respectively, and $U_n : \mathbb{R}_+ \to \mathbb{R}$, $n \in \mathcal{N}$, are concave utility functions of rates. Motivated by Kelly et al. [1], distributed congestion control algorithms solving Prob. (1) are now both numerous and classic [2]–[4].

In this work, we revisit this problem in the context of *cache networks* [5]–[7]. Motivated by technologies such as software defined networks [8] and network function virtualization [9], nodes in cache networks are no longer merely static routers. Instead, they are entities capable of storing data, performing computations, and making decisions. Nodes can thus fetch user-requested content [10], or perform user-specified computation tasks [11], [12], instead of simply maintaining point to point communication sessions. In turn, such functionalities can address the ever increasing interest in running data-intensive applications in large-scale networks, such as machine learning at the edge [13], IoT-enabled health care [14], and scientific data-intensive computation [15].

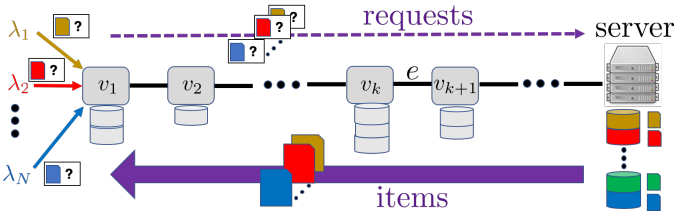


Fig. 1: Example of a cache network. N distinct flows of requests enter the network at node v_1. Each flow contains requests for an item i in a catalog $\mathcal{I}$. Requests are forwarded towards the designated server that stores all items in $\mathcal{I}$. Upon reaching it, responses carrying the requested items follow the reverse path towards node v_1. However, all intermediate nodes have caches that can be used to store items in $\mathcal{I}$. Thus, requests need not traverse the entire path, but can be satisfied upon the first hit. Hence, the load on link $e = (v_{k+1}, v_k)$ is a function of both rates $\boldsymbol{\lambda} = [\lambda_n]_{n=1}^N$ *as well as* cache allocation decisions made by the intermediate nodes $v_1, \ldots, v_k$, to name a few.

Congestion control in such networks is fundamentally different from the classic setting. When nodes can store user-requested content or provide server functionalities, network design amounts to determining not only the rate allocations per flow but also the *location* of offered network services. Put differently, to attain optimality, *cache allocation decisions need to be optimized jointly with rate allocation decisions*. In turn, this necessitates the development of novel congestion control algorithms that take cache allocation into account. To make this point clear, we illustrate the effect of cache allocations on congestion in a cache network shown in Fig. 1. Requests for content items in a catalog $\mathcal{I}$ arrive over N flows on a node on the left of a path network. They are subsequently forwarded towards a designated server on the right, that stores all items in $\mathcal{I}$. Upon reaching the server, responses carrying the requested items are sent back over the reverse path. Assuming that request traffic is negligible, the traffic load on an edge is determined by the item (i.e., response) traffic flowing through it. However, if intermediate nodes are equipped with caches that can store some of the items in $\mathcal{I}$, as in Fig. 1, requests need not be propagated all the way to the designated server. As a result, the load ρ_e caused by items traversing edge $e = (v_{k+1}, v_k)$ depends not only on the rate vector $\boldsymbol{\lambda} = [\lambda_n]_{n\in\mathcal{N}} \in \mathbb{R}_+^{|n|}$, *but also on the cache allocation decisions made at all nodes $v_1, \ldots, v_k$ preceding e in the path.* For example, the load ρ_e is zero if all items in $\mathcal{I}$ are stored in nodes $v_1, \ldots, v_k$.

Formally, in cache networks, Problem (1) becomes:

$$\max_{\boldsymbol{\lambda},\boldsymbol{x}} \quad \sum_{n\in\mathcal{N}} U_n(\lambda_n) \tag{2a}$$

$$\text{s.t.} \quad \rho_e(\boldsymbol{\lambda},\boldsymbol{x}) \leq C_e, \quad \forall e \in \mathcal{E}, \tag{2b}$$

$$\sum_{i\in\mathcal{I}} x_{vi} \leq c_v, \quad \forall v \in \mathcal{V}, \tag{2c}$$

where $\boldsymbol{x} = [x_{vi}]_{v\in V, i\in\mathcal{I}} \in \{0,1\}^{|\mathcal{V}||\mathcal{I}|}$ is the vector of cache allocation decisions $x_{vi} \in \{0,1\}$, indicating if node $v \in V$ stores $i \in \mathcal{I}$, $c_v \in \mathbb{N}$ is the storage capacity of node $v \in V$, and $\boldsymbol{\lambda}$, ρ_e, C_e, U_n are respectively the rate allocation vector, loads, link capacities, and utilities, as in Eq. (1). Crucially, the load ρ_e on links $e \in \mathcal{E}$ is a function of *both* the allocated rates *and* cache decisions. As a result, constraints (2b) define a a non-convex set. This is not just due to the combinatorial nature of cache allocation decisions $\boldsymbol{x}$: even if x_{vi} are relaxed to real values in $[0,1]$, which corresponds to making probabilistic cache allocation decisions, the resulting constraint (2b) is *still not convex*, and Problem (2) *cannot* be solved via standard convex optimization techniques. This is a significant departure from Problem (1), in which constraints (1b) are linear.

In spite of the challenges posed by the lack of convexity, *we propose algorithms solving Problem 2 with provable approximation guarantees.* Specifically:

1) We provide a unified optimization formulation for congestion control in cache networks, through joint probabilistic content placement and rate control. To the best of our knowledge, we are the first to study this class of non-convex problems and develop algorithms with approximation guarantees.
2) We propose two algorithms, each yielding different approximation guarantees. The first is a Lagrangian barrier method; the second is a convex relaxation. In both cases, we exploit the fact that constraints (2b) can be expressed in terms of DR-submodular functions [16]. Both algorithms and their corresponding analysis are novel and of independent interest, as they may be applicable for attacking problems with DR-submodular constraints beyond the cache network setting we consider here.
3) Finally, we implement both methods and compare them experimentally to greedy algorithms over several real-world and synthetic topologies, observing an improvement in aggregate utility by as much as $5.43\times$.

The remainder of this paper is structured as follows. We review related work in Section II. Our network model and problem formulation are discussed in Section III. In Section IV, we describe our two different methods for solving the optimal congestion control problem, as well as our performance guarantees. Finally, we present our evaluations in Section V, and we conclude in Section VI.

II. Related Work

Network Cache Optimization. Studies on optimal in-network cache allocation are numerous, roughly split into the *offline* and *online* solutions. Several papers study centralized, offline cache optimization in a network modeled as a bipartite graph [17], [18]. Shanmugam et al. [19] consider a femtocell network, where content is placed in caches to reduce the cost of fetching data from a base station. They do not consider congestion, and study routing costs that are linear in the traffic per link. Mahdian et al. [20] model every link with an M/M/1 queue, and consider objectives that are (non-linear) functions of the queue sizes. Similarly, Li and Ioannidis [21] model every link with an M/M/1c queue to capture the consolidation of identical responses before being forwarded downstream. The same problem was also studied, albeit in a different model, by Dehghan et al. [22]. Online cache allocation algorithms exist, e.g., for maximizing throughput [11], [23], or minimizing delay [24]. Ioannidis and Yeh [7] study a similar problem as Shanmugam et al. [19] for networks with arbitrary topology and linear link costs, seeking cache allocations that minimize routing costs across multiple hops.

Although we too consider offline algorithms, we depart substantially from prior work. First, all mentioned papers assume the input request rates are fixed, whereas we consider joint cache allocation *and* rate control. Prior works on allocation minimizing costs [7], [19]–[22] cast the problem as a sub-modular maximization problem subject to matroid constraints, for which a $(1-1/e)$-approximate solution can be constructed in polynomial time. Instead, akin to Kelly et al. [1], we treat loads on links as *constraints* rather than part of the objective. Hence, we cannot directly leverage sub-modular maximization techniques and need to design altogether new algorithms. Moreover, works which consider congestion [20]–[22] assume that the system is stable when all caches are empty; in fact, finding a cache allocation under which the system is stable is left open. We partially resolve this, jointly finding a rate and an allocation that ensure stability.

TTL caches. Time-to-Live (TTL) caches providing an elegant general framework for analyzing cache replacement policies. In TTL caches, a timer is assigned to each content, and an eviction occurs upon timer expiration. Multiple studies analyzed TTL caches as approximations to popular cache eviction policies (see [6], [25]–[27]). TTL cache optimization includes maximizing the cache hit rate [28] and the aggregate utility of cache hits [29]–[31]. In contrast to our approach, however, works on TTL caching do not provide a solution for joint cache and rate allocation, and do not guarantee network stability. Furthermore, they focus on the utility of cache hits, whereas we consider rate utility, similar to Kelly [1].

Rate admission control in cache networks. Various methods have been proposed for rate admission control in Content-Centric Networking (CCN) [32] and Named Data Networking (NDN) [10] architectures, primarily using congestion feedback from the network for rate control [33]–[37]. In contrast to our work, none of these come with optimality guarantees. Closer to us, Carofiglio et al. [38] fix a cache allocation and maximize rate utility via rate control; we depart by jointly optimizing rate and cache allocations.

DR-submodular optimization. Since their introduction by Bian et al. [16], DR-submodular functions have received much attention [39]–[41] as examples of functions which

can be maximized with performance guarantees, in spite of the fact they are not convex. Bian et al. [16] propose a constant-factor approximation algorithm for (a) maximizing monotone DR-submodular functions subject to down-closed convex constraints and (b) maximizing non-monotone DR-submodular functions subject to box constraints. In a follow-up paper, Bian et al. [39] provide a constant-factor algorithm for maximizing non-monotone continuous DR-submodular functions under general down-closed convex constraints. These works, however, do not consider DR-submodular functions in constraints, rather than in the objective. In a combinatorial setting, Crawford et al. [40] and Iyer et al. [41] provide approximate greedy algorithms for minimizing a submodular function subject to a single threshold constraint involving a submodular function. None of the above solutions, however, is applicable to our problem, which involves maximizing a concave function subject to *multiple* DR-submodular constraints. To the best of our knowledge, we are the first to study this problem and provide solutions with optimality guarantees.

III. System Model

We consider a network of caches, each capable of storing items such that the number of stored items cannot exceed a finite cache capacity. Requests are routed over fixed (and given) paths and are satisfied upon hitting the first cache that contains the requested item. Our goal is to determine the (a) items to be stored at each cache as well as (b) the request rates, so that the aggregate utility is maximized, subject to both bandwidth and storage capacity constraints in the network.

A. Network Model

Caches and Items. Following [7], [20], we represent a network by a directed graph $G(\mathcal{V}, \mathcal{E})$. We assume $G(\mathcal{V}, \mathcal{E})$ is *symmetric*, i.e., $(b, a) \in \mathcal{E}$ implies that $(a, b) \in \mathcal{E}$. There exists a catalog $\mathcal{I}$ of items (e.g., files, or file chunks) of equal size which network users can request. Each node is associated with a cache that can store a finite number of items. We describe cache contents via indicator variables: $x_{vi} \in \{0, 1\}$ for $v \in \mathcal{V}$, $i \in \mathcal{I}$, where $x_{vi} = 1$ indicates that node v stores item $i \in \mathcal{I}$. The total number of items that a node $v \in \mathcal{V}$ can store is bounded by its *node capacity* $c_v \in \mathbb{N}$ (measured in number of items). More precisely,

$$\sum_{i \in \mathcal{I}} x_{vi} \leq c_v \quad \text{for all } v \in \mathcal{V}. \tag{3}$$

We associate each item i with a fixed set of *designated servers* $S_i \subseteq \mathcal{V}$, that permanently store i; equivalently, $x_{vi} = 1$, for all $v \in S_i$. As we discuss below, these act as "caches of last resort", and ensure that all items can be eventually retrieved.

Content Requests. Item requests are routed over the network toward the designated servers. We denote by $\mathcal{N}$ the set of all requests. A request is determined by the item requested and the path that the request follows. Formally, a request n is a pair (i, p) where $i \in \mathcal{I}$ is the requested item and $p \subseteq \mathcal{V}$ is the path to be traversed to serve this request. Path p of length $|p| = K$ is a sequence of nodes $\{p_1, p_2, \ldots, p_K\} \subseteq \mathcal{V}$, where $(p_j, p_{j+1}) \in \mathcal{E}$ for all $j \in [K-1] \triangleq \{1, 2, \ldots, K-1\}$. An incoming request (i, p) is routed over the graph G and follows the path p, until it reaches a node that stores item i. At that point, a *response message* is generated, which carries the requested item. The response is propagated over p in the reverse direction, i.e., from the node that stores the item, back to the first node in p, from which the request was generated. Following [7], [20], we say that a request $n = (i, p)$ is *well-routed* if: (a) the path p is simple, i.e., it contains no loops, (b) the last node in the path is a designated server for i, i.e., $p_K \in S_i$, and (c) no other node in the path is a designated server node for i, i.e., $p_k \notin S_i$, for $k \in [K-1]$. Without loss of generality, we assume that all requests in $\mathcal{N}$ are well-routed; note that every well-routed request eventually encounters a node that contains the item requested.

Bandwidth Capacities. For each link $(a, b) \in \mathcal{E}$ there exists a positive and finite *link capacity* $C_{ab} > 0$ (measured in items/sec) indicating the bandwidth available on (a, b). We denote the vector of link capacities by $\boldsymbol{C} \in \mathbb{R}_+^{|\mathcal{E}|}$. We consider two means of controlling the rate of item transmission on a link, thereby preventing congestion in the network: (a) via the *cache allocation* strategy, i.e., by storing the requested item on a node along the path, which eliminates the flow of item on upstream links, and (b) via the *rate allocation* strategy, i.e., by controlling the rate with which requests enter the network. We describe each one in detail below.

Cache Allocation Strategy. We adopt a probabilistic cache allocation strategy. That is, we partition time into periods of equal length $T > 0$. At the beginning of the t-th time period, each node $v \in \mathcal{V}$ stores an item $i \in \mathcal{I}$ independently of other nodes and other time periods with probability $y_{vi} \in [0, 1]$, i.e., $y_{vi} = \mathrm{P}\{x_{vi}(t) = 1\} = \mathbb{E}[x_{vi}(t)]$, for all $t > 0$, where $x_{vi}(t) = 1$ indicates that node v stores item i at the t-th time period. We denote by $\boldsymbol{Y} = [y_{vi}]_{v \in \mathcal{V}, i \in \mathcal{I}} \in [0, 1]^{|\mathcal{V}||\mathcal{I}|}$ the *cache allocation strategy* vector, satisfying constraints:

$$\textstyle\sum_{i \in \mathcal{I}} y_{vi} \leq c_v \quad \text{for all } v \in \mathcal{V}. \tag{4}$$

Although condition (4) implies that cache capacity constraints are satisfied in *expectation*, it is necessary and sufficient for the existence of a probabilistic content placement (i.e., a mapping of items to caches) that satisfies capacity constraints (3) *exactly* (see, e.g., [7], [42]). We present this probabilistic placement in detail in Appendix A of our technical report [43].

Rate Allocation Strategy. Our second knob for controlling congestion is classic rate allocation, as in [1], [38]. That is, we control the input rate of requests so that the final requests injected into the network have a rate equal or smaller than original rates. We refer to the original exogenous arrival rate of a requests $n = (i, p) \in \mathcal{N}$ as the *demand rate*, and denote it by $\bar{\lambda}_n > 0$ (in requests per second). We denote the vector of demand rates by $\bar{\boldsymbol{\lambda}} = [\bar{\lambda}_n]_{n \in \mathcal{N}}$. We also denote the admitted input rate of requests into the network by λ_n, where

$$\lambda_n \leq \bar{\lambda}_n, \quad \text{for all } n \in \mathcal{N}. \tag{5}$$

We refer to the vector $\boldsymbol{\lambda} = [\lambda_n]_{n \in \mathcal{N}} \in \mathbb{R}_+^{|\mathcal{N}|}$ as the *rate allocation strategy*. We make the following assumptions on

requests admitted into the network: (a) the request process is stationary and ergodic, (b) a corresponding response message is eventually created for every admitted request, (c) the network is stable if, for all $(b,a) \in \mathcal{E}$, the following holds:

$$\rho_{(b,a)}(\boldsymbol{\lambda}, \boldsymbol{Y}) = \sum_{(i,p):(a,b)\in p} \lambda_{(i,p)} \prod_{v=p_1}^{a} (1 - y_{vi}) \leq C_{ba}. \quad (6)$$

Using the probabilistic cache allocation scheme, and the fact that the admitted request process is stationary and ergodic, $\rho_{(b,a)}(\boldsymbol{\lambda}, \boldsymbol{Y})$ is the expected rate of requests passing through link (a,b). In particular, $\prod_{v=p_1}^{a}(1-y_{vi})$ is the fraction of admitted rate $\lambda_{(i,p)}$ which is forwarded on link (a,b). Since we have assumed that for each request a response message is generated, and comes back on the reverse path, the condition in (6) ensures that the rate of items transmitted on link (b,a) is less than or equal to the link capacity C_{ba}. If the traffic rate on a link is greater than the link capacity, the network becomes unstable. In order for (6) to ensure stability, similar to [11], [23], in effect we assume that the size of requests are negligible compared to the size of requested items, and the load primarily consists of the downstream traffic of items. Note that the load on edge (b,a) depends both the rate *and* the cache allocation strategy, while constraints (6) are non-convex.

System Utility. Consistent with Kelly et al. [1], each request class $n \in \mathcal{N}$ is associated with a *utility function* $U_n : \mathbb{R}_+ \to \mathbb{R}$ of the admitted rate λ_n. The network utility is then the social welfare, i.e., the sum of all request utilities in the network:

$$U(\boldsymbol{\lambda}) = \sum_{n\in\mathcal{N}} U_n(\lambda_n). \quad (7)$$

We assume that each function U_n is twice continuously differentiable, non-decreasing, and concave for all $n \in \mathcal{N}$. Our goal is to determine a rate allocation strategy $\boldsymbol{\lambda} = [\lambda_n]_{n\in\mathcal{N}} \in \mathbb{R}_+^{|\mathcal{N}|}$ and a cache allocation strategy $\boldsymbol{Y} = [y_{vi}]_{v\in\mathcal{V},i\in\mathcal{I}} \in [0,1]^{|\mathcal{V}||\mathcal{I}|}$ that jointly maximize (7), subject to the constraints (4), (5), and (6). For technical reasons, we first transform this problem into an equivalent problem via a change of variables.

B. Problem Formulation

Change of variables. Let the *residual rate* per request be $r_n \triangleq \bar{\lambda}_n - \lambda_n$, for $n \in \mathcal{N}$. Given the *rate residual strategy* $\boldsymbol{R} \triangleq [r_n]_{n\in\mathcal{N}} \in \mathbb{R}_+^{|\mathcal{N}|}$, we rewrite the utility as

$$F(\boldsymbol{R}) \triangleq U(\bar{\boldsymbol{\lambda}} - \boldsymbol{R}) = \sum_{n\in\mathcal{N}} U_n(\bar{\lambda}_n - r_n). \quad (8)$$

Under this change of variables, we state our problem as:

UTILITYMAX

$$\text{maximize} \quad F(\boldsymbol{R}) \quad (9a)$$

$$\text{subject to} \quad (\boldsymbol{Y}, \boldsymbol{R}) \in \mathcal{D}_1, \quad (9b)$$

where $\mathcal{D}_1$ is the set of points $(\boldsymbol{Y}, \boldsymbol{R}) \in \mathbb{R}^{|\mathcal{V}||\mathcal{I}|} \times \mathbb{R}^{|\mathcal{N}|}$ satisfying the following constraints[1]:

$$g_{ba}(\boldsymbol{Y}, \boldsymbol{R}) \geq \sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}, \quad \forall (b,a) \in \mathcal{E} \quad (10a)$$

$$g_v(\boldsymbol{Y}) \leq c_v, \quad \forall v \in \mathcal{V} \quad (10b)$$

$$0 \leq y_{vi} \leq 1, \quad \forall v \in \mathcal{V}, i \in \mathcal{I} \quad (10c)$$

$$0 \leq r_n \leq \bar{\lambda}_n, \quad \forall n \in \mathcal{N}, \quad (10d)$$

where, for $(b,a) \in \mathcal{E}$ and $v \in \mathcal{V}$, $g_v(\boldsymbol{Y}) \triangleq \sum_{i\in\mathcal{I}} y_{vi}$, and

$$g_{ba}(\boldsymbol{Y}, \boldsymbol{R}) \triangleq \sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - (\bar{\lambda}_{(i,p)} - r_{(i,p)}) \prod_{v=p_1}^{a} (1 - y_{vi}). \quad (11)$$

An important consequence of this change of variables is the following lemma.

Lemma 1. *For all $(b,a) \in \mathcal{E}$, functions $g_{ba} : \mathbb{R}^{|\mathcal{V}||\mathcal{I}|} \times \mathbb{R}^{|\mathcal{N}|} \to \mathbb{R}$, are monotone DR-submodular.*

The proof is provided in Appendix A. See Section III of our technical report [43] for more information on DR-submodular functions.

IV. CACHE AND RATE ALLOCATION

The constraint set $\mathcal{D}_1$ in Problem (9) is not convex. Therefore, there is in general no efficient way to find the global optimum. Here, we propose two algorithms that come with (different) optimality guarantees. Both algorithms exploit the fact that the functions $g_{ba}(\cdot)$ are DR-submodular functions.

In our first approach, described in Section IV-A, we solve Problem (9) using a *Lagrangian barrier algorithm* [44]. We show that this converges to a Karush-Kuhn-Tucker (KKT) point (i.e., a point at which KKT necessary conditions for optimality hold) under mild assumptions. Crucially, and in contrast to general non-convex problems [44], we provide guarantees on the objective value at such KKT points. In particular, we show that the ratio of the objective value at a KKT point to the global optimum value approaches 1, asymptotically, under an appropriate proportional scaling of capacities and demand. In Section IV-B, we provide an alternative solution via *convex relaxation* of the constraint set $\mathcal{D}_1$. This turns our problem into a convex optimization problem for which efficient algorithms exist. We show that the solution obtained by solving the convex problem is feasible, and its objective value is bounded from below by the optimal value of another instance of Problem (9) with tighter constraints.

A. Lagrangian Barrier Algorithm for UTILITYMAX

Problem (9) is a maximization problem subject to the inequality constraints (10a), (10b), and the simple box constraints on the variables (10c) and (10d). Due to this structure, we propose to use the *Lagrangian Barrier with Simple Bounds* (LBSB) Algorithm, introduced by Conn et al. [44].

Algorithm description. LBSB defines the Lagrangian barrier function $\Psi(\boldsymbol{Y}, \boldsymbol{R}, \boldsymbol{\mu}, \boldsymbol{\gamma}, \boldsymbol{s})$, given by:

$$F(\boldsymbol{R}) + \sum_{(b,a)\in\mathcal{E}} \mu_{ba} s_{ba} \log(g_{ba}(\boldsymbol{Y}, \boldsymbol{R}) - \sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} + C_{ba} + s_{ba}) + \sum_{v\in\mathcal{V}} \gamma_v s_v \log(c_v - g_v(\boldsymbol{Y}) + s_v), \quad (12)$$

where the elements of vectors $\boldsymbol{\mu} \triangleq [\mu_{ba}]_{(b,a)\in\mathcal{E}} \in \mathbb{R}_+^{|\mathcal{E}|}$ and $\boldsymbol{\gamma} \triangleq [\gamma_v]_{v\in\mathcal{V}} \in \mathbb{R}_+^{|\mathcal{V}|}$ are the positive *Lagrange multiplier estimates* corresponding to (10a) and (10b) respectively, and the

[1] W.l.o.g., we implicitly set $y_{vi} = 1$, for all $v \in \mathcal{S}_i, i \in \mathcal{I}$ and do include these constraints in (10).

vector $\boldsymbol{s} \in \mathbb{R}_+^{|\mathcal{E}|+|\mathcal{V}|}$ consists of the positive values $[s_{ba}]_{(b,a)\in\mathcal{E}}$ and $[s_v]_{v\in\mathcal{V}}$ called *shifts* [44]. Intuitively, the Lagrangian barrier function in (12) penalizes the infeasibility of the link and cache constraints, and the shifts allow the constraints to be violated to some extent. Consider the following problem:

$$\begin{aligned} \max_{(\boldsymbol{Y},\boldsymbol{R})} \quad & \Psi(\boldsymbol{Y},\boldsymbol{R},\boldsymbol{\mu}_k,\boldsymbol{\gamma}_k,\boldsymbol{s}_k) \\ \text{s.t.} \quad & (\boldsymbol{Y},\boldsymbol{R}) \in \mathcal{B}, \end{aligned} \tag{13}$$

where the values $\boldsymbol{\mu}_k, \boldsymbol{\gamma}_k, \boldsymbol{s}_k$ are given, and $\mathcal{B}$ is the box constraints set defined by (10c) and (10d). Then the necessary optimality condition for Problem (13) is

$$\| P\left((\boldsymbol{Y},\boldsymbol{R}) \ , \ \nabla_{\boldsymbol{Y},\boldsymbol{R}}\Psi(\boldsymbol{Y},\boldsymbol{R},\boldsymbol{\mu}_k,\boldsymbol{\gamma}_k,\boldsymbol{s}_k)\right) \| = 0, \tag{14}$$

where $P(\boldsymbol{a} \ , \ \boldsymbol{b}) \triangleq \boldsymbol{a} - \Pi_{\mathcal{B}}(\boldsymbol{a}+\boldsymbol{b})$, and $\Pi_{\mathcal{B}}(\boldsymbol{a})$ is the projection of the vector $\boldsymbol{a}$ on the set $\mathcal{B}$. At the k-th iteration, LBSB updates $\boldsymbol{Y}_k, \boldsymbol{R}_k$ by finding a point in $\mathcal{B}$, such that

$$\| P\left((\boldsymbol{Y},\boldsymbol{R}) \ , \ \nabla_{\boldsymbol{Y},\boldsymbol{R}}\Psi(\boldsymbol{Y},\boldsymbol{R},\boldsymbol{\mu}_k,\boldsymbol{\gamma}_k,\boldsymbol{s}_k)\right) \| \leq \omega_k, \tag{15}$$

where parameter $\omega_k \geq 0$ indicates the accuracy of the solution; when $\omega_k = 0$, the point $(\boldsymbol{Y}_k, \boldsymbol{R}_k)$ satisfies the necessary optimality condition (14). In general, this point can be found by iterative algorithms such as interior-point methods or projected gradient ascent. Here, we use the trust-region algorithm [45], which we describe in Appendix C of our technical report [43].

After updating $(\boldsymbol{Y}_k, \boldsymbol{R}_k)$, LBSB checks whether the solution is in a "locally convergent regime" (with tolerance δ_k). If so, it updates the Lagrange multiplier estimates. It also updates the accuracy parameter ω_{k+1}, the tolerance parameter δ_{k+1} and the shifts $\boldsymbol{s}_{k+1}$; these updates differ depending on whether the algorithm is in a locally convergent regime or not. These iterations continue until the algorithm converges; a high-level summary of LBSB is described in Alg. 1. We refer the interested reader to Conn et al. [44] or Appendix D of our technical report [43] for a detailed description of the algorithm. Under relatively mild assumptions (see Lemma 2), the solution generated by LBSB converges to a KKT point and the Lagrange multiplier estimates converge to the Lagrange multipliers corresponding to that KKT point.

Guarantees. For general non-convex problems, the KKT point to which LBSB converges *comes with no optimality guarantees*. Our main contribution is showing that due to DR-submodularity, applying LBSB to Problem (9) yields a stronger result. We first need a few additional assumptions.

Definition 1. A function $U_n : \mathbb{R}_+ \to \mathbb{R}$ has *logarithmic diminishing return* if there exists a finite number $\theta_n \in \mathbb{R}_+$ such that $\lambda \frac{dU_n(\lambda)}{d\lambda} \leq \theta_n$ for all $\lambda \in [0, \infty)$.

Assumption 1. All utility functions U_n, $n \in \mathcal{N}$, have logarithmic diminishing return.

Assumption 2. At least one of the utility functions is unbounded from above.

We want to stress that Assumptions 1 and 2 are relatively mild. For example, consider the well-known $\alpha-$fair utility functions [3]. All $\alpha-$fair utility functions with $\alpha \geq 1$ have logarithmic diminishing return. For $\alpha = 1$, the utility function is unbounded from above. Therefore, for example, a problem instance with $\alpha-$fair utility functions, where $\alpha \geq 1$ and at least one function has $\alpha = 1$, satisfies Asssumptions 1 and 2.

Algorithm 1 Summary of Lagrangian Barrier with Simple Bounds (LBSB)

Set accuracy parameter ω_0
Set tolerance parameter for locally convergent regime δ_0
Set Lagrange multiplier estimates $\boldsymbol{\mu}_0$, $\boldsymbol{\gamma}_0$, and other initial parameters
$k \leftarrow -1$
repeat
  $k \leftarrow k + 1$
  Compute shifts $\boldsymbol{s}_k$
  Find $(\boldsymbol{Y}_k, \boldsymbol{R}_k) \in \mathcal{B}$ such that $||P((\boldsymbol{Y}_k, \boldsymbol{R}_k) \ , \ \nabla_{\boldsymbol{Y},\boldsymbol{R}}\Psi(\boldsymbol{Y}_k, \boldsymbol{R}_k, \boldsymbol{\mu}_k, \boldsymbol{\gamma}_k, \boldsymbol{s}_k))|| \leq \omega_k$.
  if in locally convergent regime (with threshold δ_k) **then**
    Update Lagrange multiplier estimates $\boldsymbol{\mu}_{k+1}, \boldsymbol{\gamma}_{k+1}$
    Update ω_{k+1} using ω_k
    Update δ_{k+1} using δ_k
  else
    Update ω_{k+1} using initial parameters
    Update δ_{k+1} using initial parameters
  end if
until convergence

Definition 2. *Regular point*: If the gradients of the active inequality constraints at $(\boldsymbol{Y}, \boldsymbol{R})$ are linearly independent, then $(\boldsymbol{Y}, \boldsymbol{R})$ is called a regular point.

Our main result is the following theorem, characterizing the quality of *regular* limit points of the sequence $\{(\boldsymbol{Y}_k, \boldsymbol{R}_k)\}$ generated by Alg. 1. We note that the regularity of limit points is typically considered in the analysis of other methods in constrained optimization literature as well [46]–[48].

Theorem 1. *Consider a problem instance with link capacity vector $\boldsymbol{C} \in \mathbb{R}_+^{|\mathcal{E}|}$ and demand rate vector $\bar{\boldsymbol{\lambda}} \in \mathbb{R}_+^{|\mathcal{N}|}$. Suppose Assumptions 1 and 2 hold, and $\{(\boldsymbol{Y}_k, \boldsymbol{R}_k)\}$, $k \in \mathcal{K}$ is a sub-sequence generated by Alg. 1 which converges to a regular point $\left[\hat{\boldsymbol{Y}}(\boldsymbol{C}, \bar{\boldsymbol{\lambda}}), \hat{\boldsymbol{R}}(\boldsymbol{C}, \bar{\boldsymbol{\lambda}})\right]$. Denote the optimal solution by $\left[\boldsymbol{Y}^*(\boldsymbol{C}, \bar{\boldsymbol{\lambda}}), \boldsymbol{R}^*(\boldsymbol{C}, \bar{\boldsymbol{\lambda}})\right]$. Then, we have* $\lim_{m\to\infty} F(\hat{\boldsymbol{R}}(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}}))/F(\boldsymbol{R}^*(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}})) = 1$.

Hence, the value of the objective at a regular limit point of Alg. 1 approaches the optimal objective value, when link capacities and demand rates grow to infinity by the same factor m. Note that increasing the link capacities *does not* make the problem easier, since demand rates increase proportionally.

The proof of Theorem 1 follows from a sequence of lemmas, which we now outline.

Lemma 2. *Let $\{(\boldsymbol{Y}_k, \boldsymbol{R}_k)\}$, $k \in \mathcal{K}$, be any subsequence generated by Alg. 1 which converges to a regular point $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$. Then $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$ is a KKT point for Problem* (9).

Proof. Please see Appendix B. The lemma is proved by showing that the regularity assumption is equivalent to the assumption stated in Theorem 4.4 of Conn et. al. [44]. □

The next key technical lemma characterizes the difference between the value of the objective at a KKT point and the global optimal value:

Lemma 3. *Let $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$ be a KKT point and $(\boldsymbol{Y}^*, \boldsymbol{R}^*)$ be the optimal point for Problem* (9)*. Then $F(\hat{\boldsymbol{R}}) \geq F(\boldsymbol{R}^*) - \sum_{(b,a)\in\mathcal{E}} \hat{\mu}_{ba} \left(\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}\right)$, where $\hat{\mu}_{ba}$ is the Lagrange multiplier corresponding to link (b,a) in constraint* (10a)*, for all $(b,a) \in \mathcal{E}$.*

Proof. Please see Appendix C. Key elements in proving Lemma 3 are the concavity of $F(\cdot)$ and the fact that $g_{ba}(\cdot)$ are monotone DR-submodular functions for all $(b,a) \in \mathcal{E}$. □

Lemma 4. *Under Assumption 1,*

$$\sum_{(b,a)\in\mathcal{E}} \hat{\mu}_{ba}\Big(\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}\Big) \leq \theta \sum_{(b,a)\in\mathcal{E}} \frac{n_{ab}}{C_{ba}} \Big(\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}\Big),$$

where n_{ab} is the number of paths passing through (a,b), and $\theta \triangleq \max_{n\in\mathcal{N}} \theta_n$ is the maximum logarithmic diminishing return parameter among utilities.

Proof. Please see Appendix D. □

Proof of Theorem 1. By Lemma 2, we know that $\big[\hat{\boldsymbol{Y}}(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}}), \hat{\boldsymbol{R}}(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}})\big]$ is a KKT point for all $m \in \mathbb{R}_+$. Thus, Lemma 3 and Lemma 4 imply that, for all $m \in \mathbb{R}_+$, $F(\hat{\boldsymbol{R}}(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}}))$ is bounded from below, i.e.,

$$F(\hat{\boldsymbol{R}}(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}})) \geq F(\boldsymbol{R}^*(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}})) - \theta \sum_{(b,a)\in\mathcal{E}} \frac{n_{ab}}{C_{ba}} \Big(\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}\Big). \quad (16)$$

According to (8), $F(\boldsymbol{R}^*(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}})) = U(\boldsymbol{\lambda}^*(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}}))$. The rate vector $m\boldsymbol{\lambda}^*(\boldsymbol{C}, \bar{\boldsymbol{\lambda}})$ is feasible in Problem (9) with link capacity vector $m\boldsymbol{C}$ and demand rate vector $m\bar{\boldsymbol{\lambda}}$, and we have $U(\boldsymbol{\lambda}^*(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}})) \geq U(m\boldsymbol{\lambda}^*(\boldsymbol{C}, \bar{\boldsymbol{\lambda}}))$. Combining this and the fact that there exists an unbounded utility function $U_n(.)$ which grows without bound as the input rate goes to infinity (Assumption 2), we have $\lim_{m\to\infty} U(\boldsymbol{\lambda}^*(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}})) = \infty$, or equivalently $\lim_{m\to\infty} F(\boldsymbol{R}^*(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}})) = \infty$. This implies that there exists a $m_0 > 0$ such that $F(\boldsymbol{R}^*(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}})) > 0$, for all $m \geq m_0$. We conclude the proof by dividing both sides of (16) by $F(\boldsymbol{R}^*(m\boldsymbol{C}, m\bar{\boldsymbol{\lambda}}))$ for $m \geq m_0$, and letting $m \to \infty$. □

Convergence Rate. We briefly discuss the convergence rate studied by Conn et. al. [44] as applied to UTILITYMAX; to do so, we need the following additional assumption:

Assumption 3. The function $F(\boldsymbol{R})$, its gradient, and elements of its Hessian are Lipschitz continuous.

Proposition 1. *Suppose Assumption 3 holds, and iterates $\{(\boldsymbol{Y}_k, \boldsymbol{R}_k)\}$ generated by Alg. 1 have a single limit point $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$ which is regular, and satisfies the second-order sufficiency condition (see Section 4.3 of Bertsekas [48] or our technical report [43]). Then with proper choice of parameters, $\{(\boldsymbol{Y}_k, \boldsymbol{R}_k)\}$ converges to $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$ at least* R-linearly *for sufficiently large k, i.e., there exists $r \in (0,1)$, $P > 0$, and k_0 such that $\|(\boldsymbol{Y}_k, \boldsymbol{R}_k) - (\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})\| \leq Pr^k$, for all $k \geq k_0$.*

The proof is provided in our technical report [43]; it follows by verifying that the assumptions in Proposition 1 imply all the assumptions for part (ii) of Theorem 5.3 and Corollary 5.7 of Conn et. al. [44].

B. Convex Relaxation of UTILITYMAX

An alternative approach for solving Problem (9) is to come up with a convex relaxation of constraint set $\mathcal{D}_1$. This turns our problem into a convex optimization problem which can be solved efficiently. Similar to prior literature [7], [49], we construct concave upper and lower bounds for the non-convex and non-concave functions in constraints (10a), using the so-called Goemans and Williamson inequality:

Lemma 5 (Goemans and Williamson [50])**.** *For $\boldsymbol{Z} \in [0,1]^n$ define $A(\boldsymbol{Z}) \triangleq 1 - \prod_{i=1}^n (1 - z_i)$ and $B(\boldsymbol{Z}) \triangleq \min\{1, \sum_{i=1}^n z_i\}$. Then, $(1 - 1/e)B(\boldsymbol{Z}) \leq A(\boldsymbol{Z}) \leq B(\boldsymbol{Z})$.*

Using Lemma 5, we obtain

$$(1 - 1/e)\tilde{g}_{ba}(\boldsymbol{Y}, \boldsymbol{R}) \leq g_{ba}(\boldsymbol{Y}, \boldsymbol{R}) \leq \tilde{g}_{ba}(\boldsymbol{Y}, \boldsymbol{R}), \quad (17)$$

where

$$\tilde{g}_{ba}(\boldsymbol{Y}, \boldsymbol{R}) \triangleq \sum_{(i,p)\in\mathcal{N}:(a,b)\in p} \bar{\lambda}_{(i,p)} \min\left\{1, \frac{r_{(i,p)}}{\bar{\lambda}_{(i,p)}} + \sum_{k=1}^{a} y_{p_k i}\right\}$$

are concave functions for all $(b,a) \in \mathcal{E}$. We use this to formulate the following convex problem:

CONVEXUTILITYMAX

$$\begin{aligned} \text{maximize} \quad & F(\boldsymbol{R}) && (18a)\\ \text{subject to} \quad & (\boldsymbol{Y}, \boldsymbol{R}) \in \mathcal{D}_2, && (18b) \end{aligned}$$

where $\mathcal{D}_2 \subseteq \mathbb{R}^{|\mathcal{V}|\times|\mathcal{I}|} \times \mathbb{R}^{|\mathcal{N}|}$ is the set of $(\boldsymbol{Y}, \boldsymbol{R})$ satisfying:

$$\begin{aligned} & \tilde{g}_{ba}(\boldsymbol{Y}, \boldsymbol{R})) \geq \frac{\sum_{(i,p):(b,a)\in p} \bar{\lambda}_{(i,p)} - C_{ba}}{1 - 1/e}, && \forall (b,a) \in \mathcal{E}\\ & g_v(\boldsymbol{Y}) \leq c_v, && \forall v \in \mathcal{V}\\ & 0 \leq y_{vi} \leq 1, && \forall v \in \mathcal{V}, \forall i \in \mathcal{I}\\ & 0 \leq r_n \leq \bar{\lambda}_n, && \forall n \in \mathcal{N}. \end{aligned}$$

Although $\tilde{g}_{ba}(\cdot)$, for all $(b,a) \in \mathcal{E}$, are non-differentiable, the optimal solution can be found using sub-gradient methods as $\mathcal{D}_2$ is convex. The following theorem provides a bound on the optimal value of Problem (18) with respect to Problem (9).

Theorem 2. *Let $(\boldsymbol{Y}_{\boldsymbol{C}}^{**}, \boldsymbol{R}_{\boldsymbol{C}}^{**})$ be the optimal solution of Problem* (18) *with link capacity vector $\boldsymbol{C}$. Also let $(\boldsymbol{Y}_{\boldsymbol{C}}^{*}, \boldsymbol{R}_{\boldsymbol{C}}^{*})$ and $(\boldsymbol{Y}_{\boldsymbol{C}'}^{*}, \boldsymbol{R}_{\boldsymbol{C}'}^{*})$ be the optimal solutions of two instances of Problem* (9) *with link capacity vectors $\boldsymbol{C}$ and $\boldsymbol{C}'$, respectively, where for all $(b,a) \in \mathcal{E}$*

$$C'_{ba} = C_{ba} - \frac{1}{e-1}\Big[\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}\Big]. \quad (20)$$

TABLE I: Graph Topologies and Experiment Parameters.

Graph	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{I}\rvert$	$\lvert\mathcal{N}\rvert$	$\lvert Q\rvert$	c'_v	$\hat{F}_{\text{(loose)}}$	$\hat{F}_{\text{(tight)}}$	#Vars
cycle	30	60	10	100	10	2	9.53	9.53	344
lollipop	30	240	10	100	10	2	9.53	9.53	274
geant	22	66	10	100	10	2	9.53	9.53	228
abilene	9	26	10	40	4	2	3.81	3.81	85
dtelekom	68	546	15	125	15	3	11.91	11.91	301
balanced-tree	63	124	30	450	15	3	42.88	28.45	1434
grid-2d	64	224	30	450	15	3	42.88	37.08	1665
hypercube	64	384	15	450	15	3	42.88	35.59	1189
small-world	64	308	30	450	15	3	42.88	37.90	1349
erdos-renyi	64	378	30	450	15	3	42.88	35.06	1191

Then, $(\boldsymbol{Y}^{**}_{\boldsymbol{C}}, \boldsymbol{R}^{**}_{\boldsymbol{C}})$ *is a feasible solution to Problem* (9) *with link capacity vector* $\boldsymbol{C}$*, and* $F(\boldsymbol{R}^{*}_{\boldsymbol{C}'}) \leq F(\boldsymbol{R}^{**}_{\boldsymbol{C}}) \leq F(\boldsymbol{R}^{*}_{\boldsymbol{C}})$.

Proof. Let $\mathcal{D}_3 \subseteq \mathbb{R}^{|\mathcal{V}||\mathcal{I}|} \times \mathbb{R}^{|\mathcal{N}|}$ be the set of $(\boldsymbol{Y}, \boldsymbol{R})$ satisfying:

$$g_{ba}(\boldsymbol{Y}, \boldsymbol{R}) \geq \frac{\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}}{1 - 1/e} \qquad \forall (b,a) \in \mathcal{E}$$

$$g_v(\boldsymbol{Y}) \leq c_v \qquad \forall v \in \mathcal{V}$$

$$0 \leq y_{vi} \leq 1 \qquad \forall v \in \mathcal{V}, \forall i \in \mathcal{I}$$

$$0 \leq r_n \leq \bar{\lambda}_n \qquad \forall n \in \mathcal{N}$$

Observe that $\mathcal{D}_3$ is the constraint set for Problem (9) with the link capacity vector $\boldsymbol{C}'$. By (17), we have $\mathcal{D}_3 \subseteq \mathcal{D}_2 \subseteq \mathcal{D}_1$. By definition, $F(\boldsymbol{R}^{*}_{\boldsymbol{C}}), F(\boldsymbol{R}^{**}_{\boldsymbol{C}})$, and $F(\boldsymbol{R}^{*}_{\boldsymbol{C}'})$ are maximum values of $F(\boldsymbol{R})$ subject to $\mathcal{D}_1, \mathcal{D}_2$, and $\mathcal{D}_3$, respectively. As a result, we have $F(\boldsymbol{R}^{*}_{\boldsymbol{C}'}) \leq F(\boldsymbol{R}^{**}_{\boldsymbol{C}}) \leq F(\boldsymbol{R}^{*}_{\boldsymbol{C}})$. □

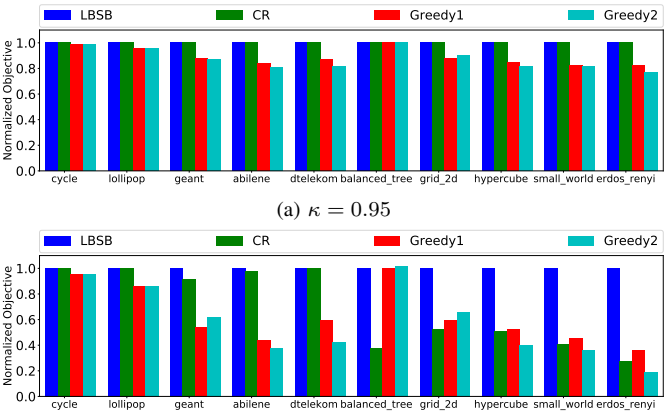
Thm. 2 states that the solution to the convex problem is no worse than the optimum of an instance of the original problem, with link capacities $C'_{ba} = C_{ba} - \frac{1}{e-1}[\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}]$. Note that C'_{ba} can be negative. In that case, Problem (9) with negative link capacities has no feasible solutions, and the solution of Problem (18) has no lower bound. For further clarification, see Corollary 3 of our technical report [43].

Comparing this to LBSB, the convex relaxation is a simpler problem, as it requires solving a convex program. On the other hand, LBSB provides better optimality guarantees, especially when the demand rate of requests exceeds the capacity of the links. In practice, as we see in the numerical evaluations (Section V), the Lagrangian barrier outperforms the convex relaxation method for a wide range of network topologies and parameter settings.

V. Numerical Evaluation

In this section, we evaluate the performance of the proposed methods, i.e., LBSB introduced in Section IV-A and the convex relaxation introduced in Section IV-B. We also implement two greedy algorithms and compare the performance of the proposed methods against these greedy algorithms.

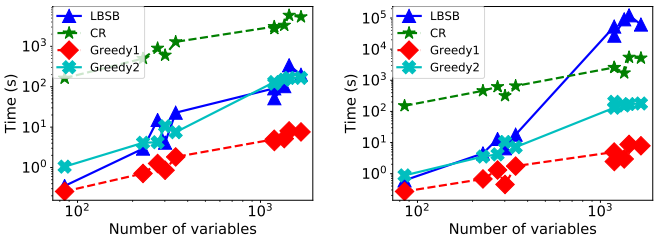
Experiment Setup. We evaluate our algorithms on 10 graphs summarized in Table I. Given a graph $G(\mathcal{V}, \mathcal{E})$, we generate a catalog $\mathcal{I}$, and assign a cache to each node in the graph. For every item $i \in \mathcal{I}$, we designate a source node selected uniformly at random (u.a.r.) from $\mathcal{V}$. We set the capacity c_v of every node v so that $c'_v = c_v - |\{i : v \in S_i\}|$ is constant among all nodes in $\mathcal{V}$. We then generate a set of requests



(a) $\kappa = 0.95$

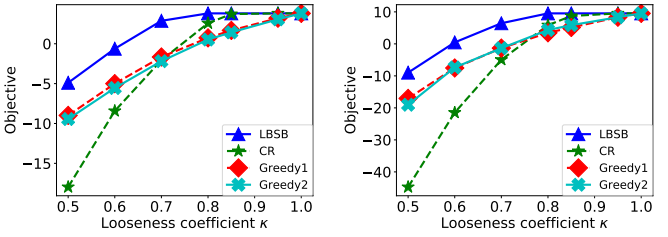
(b) $\kappa = 0.85$

Fig. 2: Objectives performance. The figure shows the normalized objective obtained by different algorithms across different topologies and in two settings, i.e., the loose setting, with $\kappa = 0.95$ (Fig 2a), and the tight setting with $\kappa = 0.85$ (Fig 2b). Note that solutions for all 4 algorithms are feasible in all cases.



(a) $\kappa = 0.95$ (b) $\kappa = 0.85$

Fig. 3: Execution Times. Figures 3a and 3b show the execution times for algorithms w.r.t. the number of variables, for the loose and tight settings, respectively.



(a) `abilene` (b) `geant`

Fig. 4: Effects of tightening the constraints. The figure shows the objective values w.r.t. the looseness coefficient κ, for two topologies `abilene` and `geant`. Note that solutions for all 4 algorithms are feasible in all cases.

$\mathcal{N}$ as follows. First, we select a set Q nodes in $\mathcal{V}$ selected u.a.r., that we refer to as *query nodes*: these are the only nodes that generate requests. More specifically, for each query node $v \in Q$, we generate $\approx |\mathcal{N}|/|Q|$ requests according to a Zipf distribution with parameter 1.2 and without replacement from the catalog $\mathcal{I}$. Each request is then routed over the shortest path between the query node and the designated source for the requested item. We assign a demand rate $\bar{\lambda}_{(i,p)} = 1$ to every request $n \in \mathcal{N}$. The values of $|\mathcal{I}|$, $|\mathcal{N}|$, $|Q|$, and c_v for each topology are given in Table I. Our process also makes sure that each item $i \in \mathcal{I}$ is requested at least once.

We determine the link capacities C_{ba}, $(b,a) \in \mathcal{E}$, as follows.

First, note that the maximum possible load on each link $(b,a) \in \mathcal{E}$ is $\lambda_{ba}^{(\max)} \triangleq \sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)}$. We set the link capacities as $C_{ba} = \kappa\lambda_{ba}^{(\max)}$, where $\kappa \in (0,1]$ is a *looseness coefficient*: the higher κ is, the easier it becomes to satisfy the demand. Note that for every link (b,a), if $C_{ba} \geq \lambda_{ba}^{(\max)}$ (or equivalently $\kappa \geq 1$), then the link constraint corresponding to (b,a) in (10a) is trivially satisfied. As κ decreases below 1, the link constraints are tightened, and finding optimal rate and cache allocation strategies becomes non-trivial. For each topology in Table I, we study two settings: (1) a *loose* setting, where $\kappa = 0.95$ and (2) a *tight* setting, where $\kappa = 0.85$.

Algorithms. We implement[2] Alg. 1 and refer to it as `LBSB`. We run this algorithm until the convergence criterion in [44] is met (with $\delta_* = 10^{-4}, \omega_* = 10^{-4}$). We also solve the convex relaxation in (18) via a sub-gradient method, as described in Section 7.5 of Bertsekas [48]; we refer to this algorithm by `CR`, for convex relaxation. We run `CR` for a fixed number of iterations (500 iterations). In addition, we implement two greedy algorithms, i.e., `Greedy1` and `Greedy2`:

- `Greedy1` consists of three steps; in Step 1, we initialize $\boldsymbol{Y} = 0$ and solve (9) only w.r.t. $\boldsymbol{R}$. This is a convex optimization problem. In Step 2, `Greedy1` keeps $\boldsymbol{R}$ fixed, as computed by Step 1, and updates $\boldsymbol{Y}$ by maximizing the sum $\sum_{(b,a)\in\mathcal{E}} g_{ba}(\boldsymbol{Y},\boldsymbol{R})$, subject to the constraints (10b) and (10c) and only w.r.t. $\boldsymbol{Y}$. This is equivalent to minimizing the total long-term time-average item load over the links of the graph (c.f. Section III-A). Note that Step 2 is a monotone DR-submodular maximization subject to a polytope, which we solve via the Frank-Wolfe algorithm proposed by Bian et. al. [16]. Finally in Step 3, `Greedy1` updates $\boldsymbol{R}$ by solving (9) w.r.t. $\boldsymbol{R}$, one more time, while $\boldsymbol{Y}$ is fixed to the value computed by Step 2.
- `Greedy2` initializes $\boldsymbol{Y} = 0$, and then alternatively updates $\boldsymbol{Y}$ and $\boldsymbol{R}$; we refer to the former step as the cache allocation step and to the latter as the rate allocation step. In the cache allocation step, `Greedy2` "greedily" places one item to a cache: it changes one zero variable ($y_{vi} = 0$) to 1, where (v,i) is the feasible pair with the largest marginal gain in load reduction. Formally, (a) node v has not fully used its cache capacity ($g_v(\boldsymbol{Y}) < c'_v$) and (b) changing y_{vi} from 0 to 1 has the highest increase in the total sum $\sum_{(b,a)\in\mathcal{E}} g_{ba}(\boldsymbol{Y},\boldsymbol{R})$ (or equivalently the highest decrease in the aggregate item load). In the rate allocation step, `Greedy2` keeps $\boldsymbol{Y}$ constant from the previous step, and solves (9) w.r.t. $\boldsymbol{R}$. Finally, `Greedy2` terminates once node storage capacities are depleted, i.e., there is no pair (v,i) left, s.t., $y_{vi} = 0$ and $y_v < c'_v$.

Metrics. Throughout the experiments we report the objective function $F(\boldsymbol{R})$ obtained by different algorithms for the choice of the logarithmic utility functions $U_n(\lambda) = \log(\lambda + 0.1)$, for all $n \in \mathcal{N}$. Note that these utility functions satisfy Assumption 1 and Assumption 2. We observe that all algorithms generate feasible solutions for all cases.

[2]Our code is publicly available at https://github.com/neu-spiral/UtilityMaximizationProbCaching.

Objective Performance. Fig. 2 shows objectives attained by different algorithms, normalized by the objective under `LBSB`; the latter is given in the $\hat{F}_{\text{(loose)}}$ and $\hat{F}_{\text{(tight)}}$ columns of Table I, for the loose ($\kappa = 0.95$) and tight ($\kappa = 0.85$) settings, respectively. We observe that `LBSB` outperforms all its competitors across all topologies in both settings, except for one case (`balanced-tree` with $\kappa = 0.85$.) In Fig. 2a, we see that in the loose setting, `CR` performance almost matches `LBSB` and achieves better objective values in comparison with `Greedy1` and `Greedy2`. However, in Fig. 2b we see that as the constraint set is tightened, the performance of `CR` deteriorates. Moreover, by comparing Fig. 2a and Fig. 2b we see that for some topologies (e.g., `grid-2d`, `hypercube`, `small-world`, and `erdos-renyi`), the gap between the objective values obtained by `LBSB` and other algorithms is significantly higher in the tight constraints regime.

Execution Time. In Fig. 3, we plot the execution times of all algorithms for each scenario as a function of the number of variables in the corresponding instance of problem UTILITYMAX (reported in the last column of Table I). Figures 3a and 3b correspond to Figures 2a and 2b (the loose and tight settings), respectively. In particular, in the loose setting (Fig. 3a) we see that the execution times for `LBSB` almost match the execution times for `Greedy2`, and `LBSB` is much faster than `CR`. In the tight setting (Fig. 3b), however, we see that the execution time of `LBSB` is higher, particularly when the number of variables is large (corresponding to larger topologies, i.e., `grid-2d`, `balanced-tree`, `hypercube`, `small-world`, and `erdos-renyi`). The main reason is that, in the tight setting, the trust-region algorithm used as a subroutine at each iteration requires a higher number of iterations to satisfy (15); as a result, the execution time for `LBSB` increases. Nonetheless, as we observed in Fig. 2b, `LBSB` achieves significantly improved objective performance compared to other algorithms.

Effect of Tightening Constraints. In Fig. 4, we plot the objective values achieved by different algorithms for looseness coefficients $\kappa = 0.5, 0.6, 0.7, 0.8, 0.85, 0.95$, and 1. We observe that for both topologies, when $\kappa = 1$, all algorithms achieve the optimal objective value; this is expected, because as explained, when $C_{ba} \geq \lambda_{ba}^{(\max)}$, the non-convex constraints (10a) are trivially satisfied. As we tighten the constraints by decreasing κ, we observe that all algorithms obtain smaller objectives, which is also expected. Crucially, `LBSB` significantly outperforms other algorithms and remains quite resilient to tightening of the constraints. Moreover, we see in Fig. 4 that for moderate tightness of constraints (e.g., $\kappa \geq 0.8$, `CR`) shows decent performance, and outperforms `Greedy1` and `Greedy2`; however, tightening the constraints further, e.g., for $\kappa \leq 0.7$, grossly affects the performance of `CR`: the objective values falls below that of the greedy algorithms. In fact, this is expected from Thm. 2. To see this, note that when $\kappa < 1/e \approx 0.36$, for the capacities in (20) we have $C'_{ba} < 0$, for all $(a,b) \in \mathcal{E}$. As a result, based on Thm. 2, for $\kappa < 1/e$, the lower bound on the optimal objective of `CR` is non-existent, as the constraint set $\mathcal{D}_3$ (see (21)) is an empty set. In other

words, in this regime, CR has no guaranteed lower bound.

VI. Conclusion

We studied a new class of non-convex optimization problems for joint content placement and rate allocation in cache networks, and proposed solutions with optimality guarantees. Our solutions establish a foundation for several possible future investigations. First, in the spirit of Kelly et al. [1], studying distributed algorithms that converge to a KKT point, and providing similar guarantees as Thm. 1, is an important open question. Second, designing new rounding techniques for deterministic content placement is another open question.

Acknowledgment

The authors gratefully acknowledge support from National Science Foundation grants NeTS-1718355 and CCF-1750539, and a research grant from American Tower Corp.

Appendix A
Proof of Lemma 1

By Eq. (11), $\frac{\partial^2 g_{ba}(\boldsymbol{Y},\boldsymbol{R})}{\partial y_{vi}\partial y_{v'i'}} \leq 0$, $\frac{\partial^2 g_{ba}(\boldsymbol{Y},\boldsymbol{R})}{\partial y_{vi}\partial r_n} \leq 0$, and $\frac{\partial^2 g_{ba}(\boldsymbol{Y},\boldsymbol{R})}{\partial r_n \partial r_{n'}} \leq 0$, for all $v, v' \in \mathcal{V}$, $i, i' \in \mathcal{I}$, and $n, n' \in \mathcal{N}$. This proves the DR-submodularity of $g_{ba}(\cdot)$, for all $(b,a) \in \mathcal{E}$ (See [16]). In addition, since $\frac{\partial g_{ba}(\boldsymbol{Y},\boldsymbol{R})}{\partial y_{vi}} \geq 0$, and $\frac{\partial g_{ba}(\boldsymbol{Y},\boldsymbol{R})}{\partial r_n} \geq 0$, for all $v \in \mathcal{V}, i \in \mathcal{I}, n \in \mathcal{N}$, $g_{ba}(\cdot)$ are monotone, for all $(b,a) \in \mathcal{E}$. □

Appendix B
Proof of Lemma 2

Here we show that the regularity of a point $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$ is equivalent to the assumption stated in Theorem 4.4 of Conn et. al. [44]. We divide the variables $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$ into two distinct class $\mathcal{F}$ and $\mathcal{F}'$, such that the variables in $\mathcal{F}'$ are equal to their upper or lower bound and the variables in $\mathcal{F}$ are strictly between their upper or lower bound. As a result, we have exactly $|\mathcal{F}'|$ active bounding constraints. We denote by $\mathcal{A}$ the set of active link and cache constraints (10a), (10b). Thus, we can decompose the Jacobian matrix for the active constraints at point $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$ as $J = \begin{bmatrix} J_{[\mathcal{A}\ \mathcal{F}]} & J_{[\mathcal{A}\ \mathcal{F}']} \\ \mathbf{0}_{|\mathcal{F}'|\times|\mathcal{F}|} & Q_{|\mathcal{F}'|\times|\mathcal{F}'|} \end{bmatrix}$. We denote by $M_{[s_1\ s_2]}$ is a submatrix of the matrix M where rows are picked according to set s_1 and columns are picked according to set s_2. The last $|\mathcal{F}'|$ rows correspond to active box constraints (10c), (10d). It can be easily seen that $Q_{|\mathcal{F}'|\times|\mathcal{F}'|}$ is a diagonal matrix with elements of diagonal being $+1$ or -1 depending on whether the variable is at its upper bound or lower bound. If $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$ is a regular point, then J is full rank at $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$. Hence, $J_{[\mathcal{A}\ \mathcal{F}]}$ is also full rank, which is exactly the assumption in Theorem 4.4 of Conn et. al. [44]. □

Appendix C
Proof of Lemma 3

By definition, the KKT necessary conditions for optimality hold at $(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})$. Hence, there exist Lagrange multipliers $[\hat{\mu}_{ba}]_{(b,a)\in\mathcal{E}}$ associated with (10a), $[\hat{\gamma}_v]_{v\in\mathcal{V}}$ associated with (10b), $[\hat{\xi}_{vi}]_{v\in\mathcal{V},i\in\mathcal{I}}$, $[\hat{\xi}'_{vi}]_{v\in\mathcal{V},i\in\mathcal{I}}$ associated with (10c), $[\hat{\eta}_{(i,p)}]_{(i,p)\in\mathcal{N}}$, $[\hat{\eta}'_{(i,p)}]_{(i,p)\in\mathcal{N}}$ associated with (10d). Due to the concavity of F, we can write $F(\hat{\boldsymbol{R}}) \geq F(\boldsymbol{R}^*) - \nabla_{\boldsymbol{Y},\boldsymbol{R}} F(\hat{\boldsymbol{R}})^T \big[(\boldsymbol{Y}^*, \boldsymbol{R}^*) - (\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})\big]$. After applying the first order necessary condition and complementary slackness, we can write $F(\hat{\boldsymbol{R}}) \geq F(\boldsymbol{R}^*) + \sum_{(b,a)\in\mathcal{E}} \hat{\mu}_{ba} \nabla_{\boldsymbol{Y},\boldsymbol{R}} g_{ba}(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})^T \big[(\boldsymbol{Y}^*, \boldsymbol{R}^*) - (\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})\big]$. As stated in Lemma 1, functions $g_{ba}(\cdot)$, for all $(b,a) \in \mathcal{E}$ are monotone DR-submodular. Therefore, we can use the following Lemma from Bian et al. [39]:

Lemma 6 (Bian et al. [39])**.** *For any differentiable DR-submodular function $f : \mathcal{X} \to \mathbb{R}$ and any two points $\boldsymbol{a}, \boldsymbol{b}$ in $\mathcal{X}$, we have*

$$(\boldsymbol{b} - \boldsymbol{a})^T \nabla f(\boldsymbol{a}) \geq f(\boldsymbol{a} \vee \boldsymbol{b}) + f(\boldsymbol{a} \wedge \boldsymbol{b}) - 2f(\boldsymbol{a}),$$

where $\vee$ and $\wedge$ are coordinate-wise maximum and minimum operations, respectively.

By Lemma 6, we have $F(\hat{\boldsymbol{R}}) \geq F(\boldsymbol{R}^*) + \sum_{(b,a)\in\mathcal{E}} \hat{\mu}_{ba} \big(g_{ba}(\hat{\boldsymbol{Y}} \vee \boldsymbol{Y}^*, \hat{\boldsymbol{R}} \vee \boldsymbol{R}^*) + g_{ba}(\hat{\boldsymbol{Y}} \wedge \boldsymbol{Y}^*, \hat{\boldsymbol{R}} \wedge \boldsymbol{R}^*) - 2g_{ba}(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})\big)$. By Lemma 1, $g_{ba}(\cdot)$ are monotone and we know that they are positive for all $(b,a) \in \mathcal{E}$. Hence, $g_{ba}(\hat{\boldsymbol{Y}} \vee \boldsymbol{Y}^*, \hat{\boldsymbol{R}} \vee \boldsymbol{R}^*) \geq g_{ba}(\boldsymbol{Y}^*, \boldsymbol{R}^*)$ and $g_{ba}(\hat{\boldsymbol{Y}} \wedge \boldsymbol{Y}^*, \hat{\boldsymbol{R}} \wedge \boldsymbol{R}^*) \geq 0$ for all $(b,a) \in \mathcal{E}$. Therefore, $F(\hat{\boldsymbol{R}}) \geq F(\boldsymbol{R}^*) + \sum_{(b,a)\in\mathcal{E}} \hat{\mu}_{ba} \big(g_{ba}(\boldsymbol{Y}^*, \boldsymbol{R}^*) - 2g_{ba}(\hat{\boldsymbol{Y}}, \hat{\boldsymbol{R}})\big) \overset{(**)}{\geq} F(\boldsymbol{R}^*) - \sum_{(b,a)} \hat{\mu}_{ba} \big(\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}\big)$, where $(**)$ is due to complementary slackness. □

Appendix D
Proof of Lemma 4

We call a link (b,a) an *active link* if $\sum_{(i,p):(a,b)\in p} \lambda_{(i,p)} \prod_{v=p_1}^{a} (1 - y_{vi}) = C_{ba}$. Since C_{ba} is positive, for an active link (b,a) there exists a set

$$\hat{\mathcal{N}}^{\text{active}}_{(b,a)} \triangleq \Big\{(i,p) : (a,b) \in p,\ (\bar{\lambda}_{(i,p)} - \hat{r}_{(i,p)}) \prod_{v=p_1}^{a} (1 - \hat{y}_{vi}) > 0,$$

$$\sum_{(i,p)\in\hat{\mathcal{N}}^{\text{active}}_{(b,a)}} (\bar{\lambda}_{(i,p)} - \hat{r}_{(i,p)}) \prod_{v=p_1}^{a} (1 - \hat{y}_{vi}) = C_{ba}\Big\}$$

Suppose (b,a) is an active link. By definition, we have $\hat{r}_{(i,p)} < \bar{\lambda}_{(i,p)}$, $\forall (i,p) \in \hat{\mathcal{N}}^{\text{active}}_{(b,a)}$. By writing the KKT conditions with respect to $\hat{r}_{(i,p)}$ for $(i,p) \in \hat{\mathcal{N}}^{\text{active}}_{(b,a)}$, we have $\frac{dU_{(i,p)}(\bar{\lambda}_{(i,p)} - \hat{r}_{(i,p)})}{d\lambda} = \sum_{(c,d):(c,d)\in p} \hat{\mu}_{dc} \prod_{v=p_1}^{c} (1 - \hat{y}_{vi}) + \hat{\eta}_{(i,p)}$. This implies $\frac{dU_{(i,p)}(\bar{\lambda}_{(i,p)} - \hat{r}_{(i,p)})}{d\lambda} \geq \hat{\mu}_{ba} \prod_{v=p_1}^{a} (1 - \hat{y}_{vi})$. After multiplying both sides by $(\bar{\lambda}_{(i,p)} - \hat{r}_{(i,p)})$, and using Assumption 1 and the fact that θ is the maximum among logarithmic diminishing return parameters, we can write $\theta \geq \hat{\mu}_{ba} (\bar{\lambda}_{(i,p)} - \hat{r}_{(i,p)}) \prod_{v=p_1}^{a} (1 - \hat{y}_{vi})$. By summing over all $(i,p) \in \hat{\mathcal{N}}^{\text{active}}_{(b,a)}$, we have $\theta |\hat{\mathcal{N}}^{\text{active}}_{(b,a)}| \geq \hat{\mu}_{ba} C_{ba}$, or equivalently $\hat{\mu}_{ba} \leq \theta \frac{|\hat{\mathcal{N}}^{active}_{(b,a)}|}{C_{ba}}$. If (b,a) is not an active link, $\hat{\mu}_{ba} = 0$. As a result, we have $\sum_{(b,a)\in\mathcal{E}} \hat{\mu}_{ba} \big(\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}\big) \leq \theta \sum_{(b,a)\in\mathcal{E}} n_{ab} \big(\sum_{(i,p):(a,b)\in p} \bar{\lambda}_{(i,p)} - C_{ba}\big)/C_{ba}$, where the last inequality is due to $|\hat{\mathcal{N}}^{\text{active}}_{(b,a)}| \leq n_{ab}$. □

REFERENCES

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

[2] R. Srikant, *The mathematics of Internet congestion control*. Springer Science & Business Media, 2012.

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

[4] S. H. Low and D. E. Lapsley, "Optimization flow control. i. basic algorithm and convergence," *IEEE/ACM Transactions on networking*, vol. 7, no. 6, pp. 861–874, 1999.

[5] E. J. Rosensweig, J. Kurose, and D. Towsley, "Approximate models for general cache networks," in *INFOCOM*, 2010, pp. 1–9.

[6] N. C. Fofack, P. Nain, G. Neglia, and D. Towsley, "Analysis of ttl-based cache networks," in *6th International ICST Conference on Performance Evaluation Methodologies and Tools*, 2012, pp. 1–10.

[7] S. Ioannidis and E. Yeh, "Adaptive caching networks with optimality guarantees," *SIGMETRICS Perform. Eval. Rev.*, vol. 44, no. 1, p. 113–124, 2016.

[8] S. Shenker, M. Casado, T. Koponen, N. McKeown *et al.*, "The future of networking, and the past of protocols," *Open Networking Summit*, vol. 20, pp. 1–30, 2011.

[9] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, "Network function virtualization: Challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 90–97, 2015.

[10] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. D. Thornton, D. K. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos *et al.*, "Named data networking (ndn) project," *Relatório Técnico NDN-0001, Xerox Palo Alto Research Center-PARC*, vol. 157, p. 158, 2010.

[11] K. Kamran, E. Yeh, and Q. Ma, "Deco: Joint computation, caching and forwarding in data-centric computing networks," in *Mobihoc*, 2019, p. 111–120.

[12] H. Feng, J. Llorca, A. M. Tulino, and A. F. Molisch, "Optimal dynamic cloud network control," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2118–2131, 2018.

[13] H. Li, K. Ota, and M. Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE Network*, vol. 32, no. 1, pp. 96–101, 2018.

[14] R. Mahmud, F. L. Koch, and R. Buyya, "Cloud-fog interoperability in iot-enabled healthcare solutions," in *ICDCN*, 2018.

[15] J. Ekanayake, S. Pallickara, and G. Fox, "Mapreduce for data intensive scientific analyses," in *IEEE Fourth International Conference on eScience*, 2008, pp. 277–284.

[16] A. A. Bian, B. Mirzasoleiman, J. Buhmann, and A. Krause, "Guaranteed Non-convex Optimization: Submodular Maximization over Continuous Domains," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, 2017, pp. 111–120.

[17] I. Baev, R. Rajaraman, and C. Swamy, "Approximation algorithms for data placement problems," *SIAM J. Comput.*, vol. 38, no. 4, p. 1411–1429, 2008.

[18] L. Fleischer, M. X. Goemans, V. S. Mirrokni, and M. Sviridenko, "Tight approximation algorithms for maximum general assignment problems," in *SODA*, 2006, p. 611–620.

[19] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, 2013.

[20] M. Mahdian, A. Moharrer, S. Ioannidis, and E. Yeh, "Kelly cache networks," in *INFOCOM*, 2019, pp. 217–225.

[21] Y. Li and S. Ioannidis, "Universally stable cache networks," in *INFOCOM*, 2020.

[22] M. Dehghan, A. Seetharam, B. Jiang, T. He, T. Salonidis, J. Kurose, D. Towsley, and R. Sitaraman, "On the complexity of optimal routing and content caching in heterogeneous networks," in *INFOCOM*, 2015, pp. 936–944.

[23] E. Yeh, T. Ho, Y. Cui, M. Burd, R. Liu, and D. Leong, "Vip: A framework for joint dynamic forwarding and caching in named data networks," in *ACM-ICN*, 2014, p. 117–126.

[24] M. Mahdian and E. Yeh, "Mindelay: Low-latency joint caching and forwarding for multi-hop networks," in *ICC*, 2018, pp. 1–7.

[25] Hao Che, Ye Tung, and Zhijun Wang, "Hierarchical web caching systems: modeling, design and experimental results," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 7, pp. 1305–1314, 2002.

[26] C. Fricker, P. Robert, and J. Roberts, "A versatile and accurate approximation for lru cache performance," in *ITC*, 2012, pp. 1–8.

[27] V. Martina, M. Garetto, and E. Leonardi, "A unified approach to the performance analysis of caching systems," in *INFOCOM*, 2014, pp. 2040–2048.

[28] A. Ferragut, I. Rodriguez, and F. Paganini, "Optimizing ttl caches under heavy-tailed demands," *SIGMETRICS Perform. Eval. Rev.*, vol. 44, no. 1, p. 101–112, 2016.

[29] M. Dehghan, L. Massoulié, D. Towsley, D. S. Menasché, and Y. C. Tay, "A utility optimization approach to network cache design," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1013–1027, 2019.

[30] N. K. Panigrahy, J. Li, F. Zafari, D. Towsley, and P. Yu, "A ttl-based approach for content placement in edge networks," *arXiv*, 2017.

[31] N. K. Panigrahy, J. Li, and D. Towsley, "Hit rate vs. hit probability based cache utility maximization," *SIGMETRICS Perform. Eval. Rev.*, vol. 45, no. 2, p. 21–23, 2017.

[32] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *CoNEXT*, 2009, pp. 1–12.

[33] N. Rozhnova and S. Fdida, "An effective hop-by-hop interest shaping mechanism for ccn communications," in *INFOCOM*, 2012, pp. 322–327.

[34] Y. Ren, J. Li, S. Shi, L. Li, G. Wang, and B. Zhang, "Congestion control in named data networking – a survey," *Computer Communications*, vol. 86, pp. 1 – 11, 2016.

[35] M. Mahdian, S. Arianfar, J. Gibson, and D. Oran, "Mircc: Multipath-aware icn rate-based congestion control," in *ACM-ICN*, 2016, p. 1–10.

[36] M. Badov, A. Seetharam, J. Kurose, V. Firoiu, and S. Nanda, "Congestion-aware caching and search in information-centric networks," in *ACM-ICN*, 2014, p. 37–46.

[37] D. Tanaka and M. Kawarasaki, "Congestion control in named data networking," in *LANMAN*, 2016, pp. 1–6.

[38] G. Carofiglio, M. Gallo, L. Muscariello, M. Papalini, and Sen Wang, "Optimal multipath congestion control and request forwarding in information-centric networks," in *ICNP*, 2013, pp. 1–10.

[39] A. Bian, K. Levy, A. Krause, and J. M. Buhmann, "Continuous dr-submodular maximization: Structure and algorithms," in *Advances in Neural Information Processing Systems*, 2017, pp. 486–496.

[40] V. G. Crawford, A. Kuhnle, and M. T. Thai, "Submodular cost submodular cover with an approximate oracle," *arXiv:1908.00653*, 2019.

[41] R. K. Iyer and J. A. Bilmes, "Submodular optimization with submodular cover and submodular knapsack constraints," in *NeurIPS*, 2013, pp. 2436–2444.

[42] B. Blaszczyszyn and A. Giovanidis, "Optimal geographic caching in cellular networks," in *ICC*, 2015, pp. 3358–3363.

[43] K. Kamran, A. Moharrer, S. Ioannidis, and E. Yeh, "Rate allocation and content placement in cache networks," 2021. [Online]. Available: https://arxiv.org/abs/2101.03441

[44] A. Conn, N. Gould, and P. Toint, "A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds," *Mathematics of Computation of the American Mathematical Society*, vol. 66, no. 217, pp. 261–288, 1997.

[45] A. Conn, P. Toint, and N. Gould, "Global convergence of a class of trust region algorithms for optimization with simple bounds," *SIAM Journal on Numerical Analysis*, vol. 25, 1988.

[46] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Method*. Academic Press, 1982.

[47] R. Fletcher, *Practical Methods of Optimization, Vol. 2*. John Wiley, 1981.

[48] D. P. Bertsekas, *Nonlinear programming*. Athena scientific, 1999.

[49] M. Karimi, M. Lucic, H. Hassani, and A. Krause, "Stochastic submodular maximization: The case of coverage functions," in *NeurIPS*, 2017, pp. 6853–6863.

[50] M. X. Goemans and D. P. Williamson, "New 34-approximation algorithms for the maximum satisfiability problem," *SIAM Journal on Discrete Mathematics*, vol. 7, no. 4, pp. 656–666, 1994.