Nonconcave Utility Maximization in Locally Coupled Systems, With Applications to Wireless and Wireline Networks

Sem C. Borst, Mihalis G. Markakis, and Iraj Saniee, *Member, IEEE*

***Abstract*—Motivated by challenging resource allocation issues arising in large-scale wireless and wireline communication networks, we study distributed network utility maximization problems with a mixture of concave (e.g., best-effort throughputs) and nonconcave (e.g., voice/video streaming rates) utilities. In the first part of the paper, we develop our methodological framework in the context of a locally coupled networked system, where nodes represent agents that control a discrete local state. Each node has a possibly nonconcave local objective function, which depends on the local state of the node and the local states of its neighbors. The goal is to maximize the sum of the local objective functions of all nodes. We devise an iterative randomized algorithm, whose convergence and optimality properties follow from the classical framework of Markov Random Fields and Gibbs Measures via a judiciously selected neighborhood structure. The proposed algorithm is distributed, asynchronous, requires limited computational effort per node/iteration, and yields provable convergence in the limit. In order to demonstrate the scope of the proposed methodological framework, in the second part of the paper we show how the method can be applied to two different problems for which no distributed algorithm with provable convergence and optimality properties is available. Specifically, we describe how the proposed methodology provides a distributed mechanism for solving nonconcave utility maximization problems: 1) arising in OFDMA cellular networks, through power allocation and user assignment; 2) arising in multihop wireline networks, through explicit rate allocation. Several numerical experiments are presented to illustrate the convergence speed and performance of the proposed method.**

***Index Terms*—Constrained Gibbs sampler, interacting particle systems, locally coupled systems, multihop wireline networks, nonconcave utility maximization, OFDMA cellular networks.**

I. Introduction

We study distributed network utility maximization problems with a mixture of concave and nonconcave utilities. First, we develop our methodological framework in the context of an abstract locally coupled system, a networked system where nodes represent agents that control a discrete local state. Each node has a possibly nonconcave local objective function, which depends on the local state of the node and the local states of its neighbors. The implicit assumption is that neighborhoods are "small" compared to the entire system, as is typical in many applications. However, the proposed methodology can be applied to any network structure. The goal is to maximize the sum of the local objective functions of all nodes. The hardness of what is in general a nonconvex discrete optimization problem prohibits the use of standard convex optimization algorithms or efficient combinatorial approximation techniques. Instead, we devise an iterative randomized algorithm whose convergence and optimality properties follow from the classical framework of Markov Random Fields and Gibbs Measures, exploiting the fact that the set of global optima may be associated with the stationary version of a stochastic process that is governed by simple local interactions. The resulting algorithm is distributed, asynchronous, requires limited computational effort per node/iteration, and yields provable convergence in the limit. In the second part of the paper, we show how the proposed method can be applied to resource allocation problems arising in wireless and wireline networks, for which no distributed algorithm with provable convergence and optimality properties is available.

The main *motivation* for our work arises from the rapid growth of wireless and wireline communication networks in terms of size, scope, and traffic demand. Today's networks are expected to support large traffic volumes, and an increasingly complex mixture of best-effort and delay-sensitive services, with resources that are fundamentally limited. Thus, there is need to make the most efficient use of these resources and achieve optimal performance across several network layers, different technologies, and various network nodes. On the other hand, the massive size of today's networks makes the implementation of any sort of centralized optimization procedure extremely difficult. Hence, a key challenge, and the central thrust of our work, is to devise algorithms that operate in a distributed fashion, and yet offer guaranteed performance to a diverse population of users/services.

The *existing literature* on concave network utility maximization problems is significant. The problem of distributed rate control for concave utility maximization, in the context of multihop wireline networks, was first tackled in the seminal papers of Kelly *et al.* [13] and Low and Lapsley [17]. The proposed algorithms rely crucially on convexity properties and dual-based decomposition: Link "shadow prices" act as dual variables and

Manuscript received April 23, 2012; revised November 09, 2012; accepted March 25, 2013; approved by IEEE/ACM Transactions on Networking Editor S. Weber. Part of the material in this work was presented at the 49th Allerton Conference on Communication, Control, and Computing, Monticello, IL, USA, September 28–30, 2011.

S. C. Borst and I. Saniee are with the Mathematics of Networks and Communications Department, Alcatel-Lucent Bell Labs, Murray Hill, NJ 07974 USA (e-mail: sem@research.bell-labs.com; iis@research.bell-labs.com).

M. G. Markakis is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: mihalis@mit.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2013.2257181

provide the pivotal element for distributed algorithms. Flows adjust their rates in response to advertised prices, and links update their prices based on observed aggregate rates. In contrast, our approach does not rely on any convexity properties and, additionally, accommodates integrality constraints. Specifically in the context of wireless networks, even if the throughput utility functions are assumed to be concave, the interference between transmissions typically causes the global objective function to be nonconcave, prohibiting the use of similar dual-based decomposition approaches.

Significant work has also been done on distributed optimization in wireless networks, e.g., on power allocation problems in cellular networks [10], [22], [24], on packet scheduling in ad hoc networks [1], and on load balancing problems [4], [8], [9]. From a methodological standpoint, the work that comes closer to ours is that of Kauffmann *et al.* [12], which proposed distributed channel selection and node association algorithms based on the Gibbs Sampler, in the context of multichannel 802.11 WLANs. We also refer to the follow-up papers [6] and [7], which utilize this methodological approach for joint power allocation and user assignment in cellular networks. These papers, however, are restricted to a very specific utility function and hinge on a somewhat arbitrary notion of interference minimization, which is only connected to the optimization objective in a heuristic sense. A more consistent approach for achieving proportional fairness in an 802.11 WLAN setting was devised by Hou and Gupta [11]. Further related work is reported by Zhou *et al.* [25], describing an annealed Gibbs sampling approach for power control and adaptive modulation. Our methodological apparatus is developed in a more comprehensive framework, allowing for arbitrary utility functions and generic notions of local coupling [16]. Also, we pursue applications to wireless and wireline networks with scheduled access, as opposed to 802.11 WLANs with randomized access. Another recent study by Qian *et al.* [18] proposed a Gibbs Sampler-based algorithm that solves a generic power allocation problem in wireless networks. Their algorithm, however, is essentially not distributed, in the sense that the overhead associated with each iteration is of the order of magnitude of the entire network. In contrast, the computational requirements and the communication overhead of the algorithm proposed here scale with the size of typically small neighborhoods. Finally, the work of Rangan and Madan [19] proposed algorithms for solving approximately a power allocation problem in OFDMA systems. These algorithms are based on Belief Propagation, and while similar in spirit to our approach, they are only known to converge in acyclic graphs, a property that is usually violated in wireless networks. In contrast, our approach offers provable convergence for any network topology.

The *main contribution* of this paper is a distributed algorithm for network utility maximization in locally coupled systems, applicable to problems arising in both wireless and wireline networks. The proposed method utilizes the classical framework of Markov Random Fields and Gibbs Measures. Typically, algorithms that are based on this framework are: 1) distributed, but then do not have guaranteed convergence [19], or are restricted to specialized objective functions, e.g., [6], [7], [11], and [12]; or 2) allow for general objective functions, but then require global state information [18]. We identify a quite general setting, namely locally coupled systems with separable global objectives, where the framework of Markov Random Fields and Gibbs Measures can accommodate arbitrary local objective functions, and yet retain the distributed nature of updates and provable convergence properties.

The remainder of the paper is organized as follows. In Section II, we provide a nontechnical overview of our approach to distributed utility maximization in locally coupled systems. In Section III, we define formally a locally coupled system, formulate a global optimization problem, and describe a generic methodology for finding the global optimum in a distributed fashion. Also, we establish the convergence of the proposed algorithm and discuss variations and extensions, as well as special cases where the communication and computational requirements can be further reduced. In Section IV, we show how the proposed method can be applied for distributed power allocation and user assignment in wireless OFDMA cellular networks. Section V presents another application of the proposed framework, this time in the context of multihop wireline networks. In Section VI, we make some concluding remarks and suggest avenues for future research.

II. High-Level Discussion of Proposed Methodology

In this section we introduce the proposed methodology, highlighting the central features and indicating key distinctions with existing approaches, with technical details postponed for later sections.

Consider a network consisting of four nodes A, B, C, and D, with local states x_A, x_B, x_C, and x_D, respectively. The nodes could represent actual physical objects, e.g., wireless transmitters, wired links, or storage devices, but could also correspond to logical entities, e.g., network routes, radio links, or traffic flows. The local states may be interpreted as (possibly vector-valued) decision variables or parameter settings that are individually controlled by the corresponding nodes. Depending on the specific context, the local states could represent various quantities, e.g., power levels, transmission frequencies, time allocations, data rates, or sets of content items. Each node v has a local objective function $F_v(\cdot)$, which depends on the state of the node itself, and on the states of other nodes that interact with it, in an appropriate sense. For example, the local objective function could provide a measure for the throughput utility of a wireless device, which clearly depends not only on its own power level and frequency, but also on those of nearby transmitters due to interference. For concreteness, we assume the following:

$F_A(\cdot)$ depends on x_A and x_B;
$F_B(\cdot)$ depends on x_B, as well as x_A and x_C;
$F_C(\cdot)$ depends on x_C, as well as x_B and x_D;
$F_D(\cdot)$ depends on x_D and x_C.

The mutual dependence of the local objective functions may be represented in terms of an *interaction graph*, as depicted in Fig. 1(a), where two nodes are connected if the local objective function of one is affected by the local state of the other. Crucially, the global objective function $F(x_A, x_B, x_C, x_D)$ is simply the sum of the local objective functions and may be interpreted as the aggregate network utility.

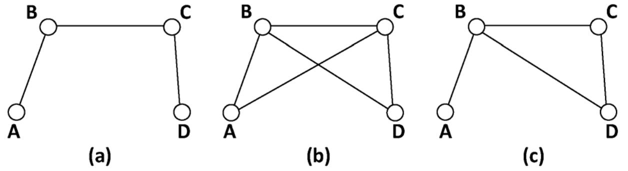


Fig. 1. Separability and functional dependencies: (a) "one-tier" interaction graph $\mathcal{G}$; (b) "two-tier" CGS graph $\mathcal{G}^*$; (c) pruned CGS graph in the case where the local objective function itself is separable into multiple components.

We tacitly assume that the various nodes are cooperative and wish to maximize the aggregate network utility, as measured by the global objective function. Since the local states are individually controlled, we seek an approach where the various nodes perform updates in an asynchronous and distributed fashion. For such an approach to have a chance of maximizing the aggregate network utility, individual nodes should somehow account for their impact on the global objective function when performing state updates. Specifically in our methodology, an individual node, say node A, will select a local state y_A with probability proportional to $q(y_A|x_B, x_C, x_D) = \exp(F(y_A, x_B, x_C, x_D)/T)$, with T some strictly positive coefficient. The latter update rule is reminiscent of a *Gibbs Sampler*, except that we allow the candidate state y_A to be limited to a random subset of the local state space, and hence we shall refer to it as a *Constrained Gibbs Sampler (CGS) algorithm*. The rule is also somewhat similar in spirit to *simulated annealing*, with T playing the role of the temperature. In that context, though, no notion of local state exists, and only a single candidate (global) state is considered and then accepted with certain probability.

A priori it is far from obvious that the CGS algorithm will produce a globally optimal state, or even converge at all. In order to establish the convergence, we interpret the local updates as transitions of a reversible Markov chain. In equilibrium, the probability of state (x_A, x_B, x_C, x_D) is proportional to $\exp(F(x_A, x_B, x_C, x_D)/T)$, which defines a *Gibbs Field* (GF) and, according to the Hammersley–Clifford theorem, a *Markov Random Field* (MRF); see [5, Ch. 7, Theorem 2.2]. As $T \downarrow 0$, the latter distribution concentrates on the set of states S^* where the global objective function attains a maximum. In that sense, the CGS algorithm ensures *provable convergence* to an approximate *global optimum* for low values of T; see Propositions 1–3 in Section III.

We note that the above arguments do not rely on specific properties of the global objective function, and convergence to an (approximate) global optimum is, in fact, guaranteed for an arbitrary function $F(\cdot)$. This allows, in particular, *nonconcave utility functions*, in sharp contrast to gradient methods for convex utility maximization, which may get trapped in a local optimum in our framework. It is important to observe, however, that for an arbitrary function $F(\cdot)$, the update of a particular local state would typically involve knowledge of all the local states across the network, which would essentially be tantamount to a centralized operation. As we will show, the notion of local coupling plays a critical role in order to avoid that, and enables a distributed implementation.

A key thrust of our methodological framework is that a *separable global objective function* only requires information exchange within the *two-tier neighborhoods* of the interaction graph. In order to illustrate that, suppose that a particular node, say node A, performs an update of its local state. By virtue of the separability of the objective function, the quantity $q(y_A|x_B, x_C, x_D)$ only depends on the candidate state y_A through the terms $F_A(y_A; x_B)$ and $F_B(x_B; y_A, x_C)$, i.e., the local objective functions of node A and its neighbor B, and *not* the terms $F_C(x_C; x_B, x_D)$ and $F_D(x_D; x_C)$. Hence, the relative selection probabilities for the various candidate states at A can be calculated based on x_B and x_C alone, without any knowledge of x_D.

As we will demonstrate in the next section, consideration of local objective functions of neighbors, and thus local state information for a two-tier neighborhood, in fact suffices in general. The graph induced by the two-tier neighborhoods, as depicted in Fig. 1(b) for the four-node network, will be called the *CGS graph*. In scenarios where these two-tier neighborhoods are relatively small compared to the size of the entire network—which we refer to as *locally coupled systems*—the CGS graph is sparse, and the amount of information exchange is commensurately small.

In other words, the *locality* and *separability* of the objective function translate into a *distributed operation* of the CGS algorithm. Fortunately, separable objective functions and local coupling arise quite naturally in network optimization problems. In wireless communications, local coupling is due to geographical proximity because interference is usually limited in range. In routing and content management, locality is not just geographical, but also due the fact that a path intersects a small fraction of all network paths.

We conclude this section with a few remarks relating to the significance of the two-tier neighborhood structure.

1) As we will show, the two-tier neighborhood structure suffices for the CGS algorithm to converge to the global optimum when the global objective function is separable across nodes. Thus, additional state information does not make any difference in the local updates. Even though the two-tier neighborhood is in general necessary, in some specific cases it is not; in particular, when the local objective functions are additionally separable across neighbors. For example, if the local objective function of node B is of the form $F_B(x_B; x_A, x_C) = F_B^{(A)}(x_B; x_A) + F_B^{(C)}(x_B; x_C)$, then the quantity $q(y_A; x_B, x_C; x_D)$ only depends on the candidate state y_A through the terms $F_A(y_A; x_B)$ and $F_B^{(A)}(x_B; y_A)$, and not on any of the other terms. Hence, node A no longer needs to know x_C, and the edge between nodes A and C can be eliminated from the CGS graph, as illustrated in Fig. 1(c).

A specific scenario of interest is $F_v^{(w)}(x_v; x_w) = -\sum_{k=1}^{K} x_{vk} x_{wk}$, with x_{uk} a 0–1 variable indicating whether node u selects option k, which in wireless networks could correspond to an access point or a transmission channel. The local objective function of a node then represents (minus) the number of competing users at the same access point, or the number of interfering nodes with the same transmission channel, as considered in the work of Kauffman *et al.* [12]. In these cases, state information from direct neighbors is sufficient, and no need exists for exchange of states with second-tier neighbors;

2) The role of the two-tier neighborhood structure may seem at odds with the fact that Gibbs Samplers for network optimization problems normally operate in terms of so-called local energy functions. These local energy functions only involve cliques, which must all be contained within single-tier neighborhoods. In order to resolve this paradox, we observe that this setting requires the global objective function to be derivable from a potential, i.e., be a sum over cliques in the interaction graph. This setting only allows for rather specialized global objective functions that can be expressed as sum over edges of the interaction graph, which is consistent with the earlier observation that single-tier neighborhoods suffice when the local objective functions are separable.

The separable global objective functions that we consider generally cannot be written as potentials with respect to the interaction graph, but can be expressed as sums over cliques in the CGS graph, as we defined it. For example, in the four-node network the global objective function is indeed the sum over the cliques $\{A, B\}$, $\{A, B, C\}$, $\{B, C, D\}$, and $\{C, D\}$ in the CGS graph. Indeed, our methodology is essentially a (Constrained) Gibbs Sampler acting on the CGS graph. Thus, the local energy function of a node in the CGS graph captures the sum of the local objective functions of that node itself and its direct neighbors in the interaction graph;

3) The significance of the two-tier neighborhood structure is also corroborated by known convergence issues of Belief Propagation methods. These only involve information exchange with direct neighbors in the interaction graph and, in general, do not guarantee convergence to a globally stable state. We expect that the two-tier neighborhood structure may help resolve such convergence issues, but we will not pursue this thread further in this paper.

III. Distributed Optimization in Locally Coupled Systems

A. Model Description and Problem Formulation

Throughout the paper we use $\mathbb{Z}$, $\mathbb{N}$, and $\mathbb{R}$ to denote the sets of integers, positive integers, and real numbers, respectively. We denote by $\mathbb{Z}_+$ and $\mathbb{R}_+$ the sets of nonnegative integers and nonnegative real numbers, and use $\mathbb{R}_+^M$ to denote the cartesian product of M copies of $\mathbb{R}_+$. We use $1_{\{A\}}$ to represent the indicator function of event A. The cardinality of set S is denoted by $|S|$. Finally, we use "$\xrightarrow{d}$" and "$\xrightarrow{as}$" to denote convergence in distribution and almost sure convergence of a sequence of random variables, respectively.

We consider a networked system, represented by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the set of edges. The edges of the graph determine a neighborhood system on the set of nodes. We denote the set of neighbors of node v by $\mathcal{N}_v^{(1)} = \{u \in \mathcal{V} | \{u, v\} \in \mathcal{E}\}$. We assume that $\mathcal{G}$ has no self loops, which implies that the neighborhood set $\mathcal{N}_v^{(1)}$ does not include node v itself. By adding node v, we obtain the extended neighborhood set of node v, denoted by $\mathcal{N}_v^+ = \mathcal{N}_v \cup \{v\}$.

Associated with each node $v \in \mathcal{V}$ are a local state x_v and a local objective function $F_v(\cdot)$.

The *local state* of node v is a D_v-dimensional vector and can be interpreted as a set of decision variables or parameter settings that are locally controlled by node v. We denote the cth component of x_v by $x_{v,c}$, where $c \in \{1, \ldots, D_v\}$. The component $x_{v,c}$ takes values in a finite set of nonnegative real numbers $S_{v,c}$. The components of the local state x_v have to satisfy a set of linear inequality constraints, i.e., there exist a nonnegative real-valued matrix B_v and vector b_v, such that $B_v \cdot x_v \leq b_v$. The local state space of node v, denoted by S_v, consists of all allowable D_v-tuples: $S_v = \{x \in \prod_{c=1}^{D_v} S_{v,c} | B_v \cdot x \leq b_v\}$. We do allow the sets $S_{v,c}$ and the linear inequality constraints to be arbitrary, but assume that the resulting local state space S_v is nonempty.

The *local objective function* of node v is a possibly nonconcave real-valued function of the local states of node v and its neighbors: $F_v : \prod_{u \in \mathcal{N}_v^+} S_u \to \mathbb{R}$.

The edges of the graph determine the functional dependencies between the various nodes: the local state of node v affects only the local objective functions of the nodes in $\mathcal{N}_v^+$, and the local objective function of node v depends only on the local states of nodes in $\mathcal{N}_v^+$. For notational convenience, we denote the set of local states of the nodes in $\mathcal{N}_v^+$ by $x_{\mathcal{N}_v^+}$.

The *global state* of the network x is defined as the set of local states of all nodes. Naturally, the global state space is the product space of all local state spaces

$$S = \prod_{v \in \mathcal{V}} S_v.$$

The *global objective function* is defined as the sum of all local objective functions

$$F(x) = \sum_{v \in \mathcal{V}} F_v(x_{\mathcal{N}_v^+}).$$

Our goal is to maximize the global objective function

$$\begin{array}{ll} \text{maximize} & F(\text{x}) \\ \text{subject to} & x \in S. \end{array} \tag{1}$$

We denote by S^* the set of solutions of this optimization problem, which we call globally optimal states, and assume that S^* is a proper subset of S.

A remark should be made on the nature of the local constraints: We assumed that they have the form of linear inequality constraints, and that they are at a "node level," as is sufficient for the applications that we investigate in the second part of the paper. However, the algorithmic framework that we introduce in Section III-B extends to any type of nonlinear constraints, possibly at a "neighborhood level," as long as a technical irreducibility condition is satisfied.

B. Constrained Gibbs Sampler

In this section, we propose a randomized algorithm that approximates an optimal solution, while operating in a distributed and asynchronous fashion. Our roadmap is as follows. We view the global state of the system as a random field on the global state space S. First, we show that there exists a neighborhood system on $\mathcal{V}$, defining a graph $\mathcal{G}^* = (\mathcal{V}, \mathcal{E}^*)$, and a Gibbs potential relative to $\mathcal{G}^*$, such that the global objective function can be written as the energy stemming from this potential. Then, we invoke the Gibbs–Markov equivalence to show that the system is a GF with steady-state distribution

$$\pi_T(x) = \frac{\exp\left[\frac{F(x)}{T}\right]}{\sum_{z \in S} \exp\left[\frac{F(z)}{T}\right]} \tag{2}$$

if and only if the random field is Markovian with respect to $\mathcal{G}^*$. This equivalence gives directly the local specification of the MRF and, thus, the distributed algorithm that we seek.

Consider the undirected graph $\mathcal{G}^* = (\mathcal{V}, \mathcal{E}^*)$, where $\{u, v\} \in \mathcal{E}^*$ if and only if $v \in \mathcal{N}_u^{(2)}$, with

$$\mathcal{N}_v^{(2)} = \left(\bigcup_{u \in \mathcal{N}_v^{(1)}} \mathcal{N}_u^+ \right) \setminus \{v\} = \left(\bigcup_{u \in \mathcal{N}_v^+} \mathcal{N}_u^{(1)} \right) \setminus \{v\}$$

representing the "two-tier" neighborhood set of node v. It is easily verified that if $v_1, v_2 \in \mathcal{N}_u^{(1)}$, for some $u \in \mathcal{V}$, then $v_1 \in \mathcal{N}_{v_2}^{(2)}$ and $v_2 \in \mathcal{N}_{v_1}^{(2)}$. Also, if $v \in \mathcal{N}_u^{(1)}$, then $v \in \mathcal{N}_u^{(2)}$ and $u \in \mathcal{N}_v^{(2)}$. It follows that the neighborhood set $\mathcal{N}_v^+$ forms a (possibly nonmaximal) clique in the graph $\mathcal{G}^*$. Let $\mathcal{C}(\mathcal{G}^*)$ represent the collection of all cliques of the graph $\mathcal{G}^*$. Then, the global objective function may be written as

$$\begin{aligned} F(x) &= \sum_{v \in \mathcal{V}} F_v(x_{\mathcal{N}_v^+}) = \sum_{C \in \mathcal{C}(\mathcal{G}^*)} \sum_{v : \mathcal{N}_v^+ = C} F_v(x_{\mathcal{N}_v^+}) \\ &= \sum_{C \in \mathcal{C}(\mathcal{G}^*)} F_C(x_C) \end{aligned}$$

where $F_C(x_C) = \sum_{v : \mathcal{N}_v^+ = C} F_v(x_{\mathcal{N}_v^+})$.

Summarizing, by defining the potential of the clique $\mathcal{N}_v^+$ to be equal to $F_v(x_{\mathcal{N}_v^+})$, for all $v \in \mathcal{V}$, and the potential of all other cliques to be zero, we have shown that the global objective function is equal to the energy that derives from this Gibbs potential relative to $\mathcal{G}^*$.

In our case, a natural positivity condition is satisfied, and the Gibbs–Markov equivalence (see [5, Section 7.2]) implies that the system is a GF following (2) if and only if the random field is Markovian with respect to $\mathcal{G}^*$. Moreover, the local specification of this MRF is given by

$$\mathbb{P}\left(x_v = y | x_{\mathcal{N}_v^{(2)}}\right) = \frac{\exp\left[\frac{\sum_{u \in \mathcal{N}_v^+} F_u\left(y, x_{\mathcal{N}_u^+ \setminus \{v\}}\right)}{T}\right]}{\sum_{z \in S_v} \exp\left[\frac{\sum_{u \in \mathcal{N}_v^+} F_u\left(z, x_{\mathcal{N}_u^+ \setminus \{v\}}\right)}{T}\right]}$$

for all $y \in S_v$.

Notice that the edge set $\mathcal{E}^*$ includes the edge set $\mathcal{E}$ of the original network graph, with the inclusion being strict, except for the special case where the original network graph is a collection of cliques. Indeed, while the global objective function $F(\cdot)$ is additive over nodes, it does not admit a clique representation with respect to the original network graph. Informally speaking, in order for the local objective functions to be potential functions, the neighborhood sets $\mathcal{N}_v^+$ should be cliques, which is not the case in general. This can be brought about by adding edges between nodes in $\mathcal{N}_v^{(1)}$ that are not directly connected in the original network graph, and this is precisely how the graph $\mathcal{G}^*$ is constructed.

Unfortunately, the following issue may arise when this procedure is viewed from an algorithmic standpoint: Before node v updates its local state, it needs to compute an exponential sum utility function for every state in its local state space S_v. In many applications of interest, and in particular in OFDMA systems, some local state spaces can be very large. In these situations, the computational and memory requirements of each iteration are prohibitive. We address this issue with the following modification of the classical framework of GF/MRF, which we call the Constrained Gibbs Sampler (CGS) algorithm.

A positive constant T is fixed, which is called the temperature of the algorithm, and is known to all nodes. Each node $v \in \mathcal{V}$ fixes a positive integer $A_v \in \{1, \ldots, D_v\}$, which represents the number of components that are modified per local state update. Also, each node v has a clock. The time periods between consecutive ticks of this clock are independent and identically distributed random variables, following the exponential distribution with rate D_v / A_v. The clocks of different nodes are mutually independent.
Whenever its clock ticks, node v:

1) requests the local states of all nodes in its two-tier neighborhood $\mathcal{N}_v^{(2)}$;
2) picks uniformly at random A_v out of the D_v components of its local state. Let a_v represent the set of these components;
3) determines the values that the components in a_v are allowed to take. Let $x_{-(v,a_v)}$ represent the components of x_v except those in a_v, together with the local states in the two-tier neighborhood of v. The local constraints of node v are satisfied only for a subset of the values in $\Pi_{c \in a_v} S_{v,c}$, which we denote by $S(x_{-(v,a_v)})$;
4) picks values for the components in a_v from the set $S(x_{-(v,a_v)})$, according to the conditional probability mass function

$$\begin{aligned} &\mathbb{P}\left(x_{v,a_v} = y | x_{-(v,a_v)}\right) \\ &\quad = \frac{\exp\left[\sum_{u \in \mathcal{N}_v^+} F_u\left(y, x_{-(v,a_v)}\right) / T\right]}{\sum_{z \in S\left(x_{-(v,a_v)}\right)} \exp\left[\sum_{u \in \mathcal{N}_v^+} F_u\left(z, x_{-(v,a_v)}\right) / T\right]}. \end{aligned} \tag{3}$$

We track the evolution of the system in discrete time, namely right after state transitions. We denote by $x(t)$ the global state of the system right after the tth state transition, $t \in \mathbb{Z}_+$. The following proposition characterizes the evolution and convergence of a locally coupled system under the CGS algorithm.

Proposition 1: (Convergence of CGS): Under the CGS algorithm the sequence $\{x(t); t \in \mathbb{Z}_+\}$ is a time-homogeneous, irreducible, aperiodic, and reversible Markov chain on S. Hence, there exists a generic random variable on S, following the Gibbs distribution of (2), such that for all $x(0) \in S$

$$x(t) \xrightarrow{d} x$$

and

$$\frac{1}{t} \sum_{\tau=0}^{t-1} F\left(x(\tau)\right) \xrightarrow{as} \mathbb{E}\left[F(x)\right].$$

Proof: First of all, it is straightforward to verify that the evolution of the global state in discrete time is a time-homogeneous Markov chain on the finite state space S.

Under the dynamics induced by the CGS algorithm, every global state has a self-loop of positive probability. This implies that this Markov chain is aperiodic.

Next, we show that the Markov chain $\{x(t); t \in \mathbb{Z}_+\}$ is irreducible. Since a global state is a collection of local states, and all constraints are purely local in nature, it is sufficient to establish irreducibility at a local level. Consider any node $v \in \mathcal{V}$ and any two local states $x_v, x'_v \in S_v$. We will establish that if the local state of node v starts from x_v, it ends in x'_v with positive probability. Denote by $x_v^{\min}$ the D_v-dimensional vector containing the minimum values of the sets $S_{v,c}$. Since all the elements of matrix A_v and vector b_v are nonegative, $x_v^{\min}$ is a feasible local state of node v, i.e., $x_v^{\min} \in S_v$. Clearly, any local state $x_v \in S_v$ dominates $x_v^{\min}$ componentwise. Hence, starting from any $x_v \in S_v$, the local state of node v can become $x_v^{\min}$ with positive probability, after $\lceil (D_v/A_v) \rceil$ local state updates. This is intuitively clear since the local constraints prevent the various components from taking higher values, not smaller ones. In state $x_v^{\min}$, all inequality constraints contain as much slack as possible, and the CGS algorithm can move to any local state $x'_v \in S$ as if unconstrained. Again, this can happen with positive probability, after $\lceil (D_v/A_v) \rceil$ local state updates.

We now show that the Markov chain is also reversible, with steady-state distribution following the Gibbs form of (2). In order to do that, we must verify that, for all $x, x' \in S$

$$\pi_T(x) \cdot \mathbb{P}(x'|x) = \pi_T(x') \cdot \mathbb{P}(x|x')$$

where $\mathbb{P}(x|x')$ is the one-step transition probability to global state x, conditional on being in global state x', and similar for $\mathbb{P}(x'|x)$.

Let us begin with some easy cases. If $x = x'$, then the reversibility condition above is trivially satisfied. If states x and x' differ in more than one local states, then both $\mathbb{P}(x|x')$ and $\mathbb{P}(x'|x)$ are equal to zero under the CGS dynamics, and the reversibility condition above is satisfied. If states x and x' differ in exactly one local state, let it be the local state of node $v \in \mathcal{V}$, but in more than A_v components, then both $\mathbb{P}(x|x')$ and $\mathbb{P}(x'|x)$ are equal to zero under the CGS dynamics, and the reversibility condition is, again, satisfied.

Thus, the only interesting case is when global states x and x' differ in exactly one local state; let it be the local state of node $v \in \mathcal{V}$, and $|\lambda(x,x')| \le A_v$, where $\lambda(x,x')$ is the set of components in which states x and x' differ. In this case, the transition probabilities $\mathbb{P}(x'|x)$ and $\mathbb{P}(x|x')$ are strictly positive, and the reversibility condition takes the form

$$\frac{\mathbb{P}(x'|x)}{\mathbb{P}(x|x')} = \frac{\pi_T(x')}{\pi_T(x)} = \exp\left(\frac{F(x') - F(x)}{T}\right).$$

At this point, let us introduce the following notation: Conditional on a local state update, denote by $B(v)$ the event that it is node v that updates its local state. Conditional on $B(v)$, denote by $\Gamma(a_v)$ the event that node v updates the components in the set a_v, where $|a_v| = A_v$. Finally, we use $C(A_v)$ for the set of combinations of A_v components of the local state of node v. We can write

$$\mathbb{P}(x'|x) = \mathbb{P}(B(v)|x) \cdot \left(\sum_{a_v \in C(A_v)} \mathbb{P}(\Gamma(a_v)|B(v), x) \cdot \mathbb{P}(x'|\Gamma(a_v), B(v), x) \right).$$

Under the CGS dynamics, the events $B(v)$ and $\Gamma(a_v)$ do not depend on the state before the transition x. Moreover, the conditional probability $\mathbb{P}(x'|\Gamma(a_v), B(v), x)$ is nonzero, only when the set of components to be updated, a_v, includes all the components in $\lambda(x,x')$. Then, this conditional probability is given by the Gibbs distribution. This implies that

$$\mathbb{P}(x'|x) = \exp\left(\frac{F(x')}{T}\right) \cdot \left(\mathbb{P}(B(v)) \cdot \sum_{a_v \in C(A_v)} \frac{\mathbb{P}(\Gamma(a_v)|B(v)) \cdot 1_{\{\lambda(x,x') \subset a_v\}}}{\sum_{z \in S\left(x_{-(v,a_v)}\right)} \exp\left(\frac{F(z)}{T}\right)} \right).$$

Similarly

$$\mathbb{P}(x|x') = \exp\left(\frac{F(x)}{T}\right) \cdot \left(\mathbb{P}(B(v)) \cdot \sum_{a_v \in C(A_v)} \frac{\mathbb{P}(\Gamma(a_v)|B(v)) \cdot 1_{\{\lambda(x,x') \subset a_v\}}}{\sum_{z \in S\left(x_{-(v,a_v)}\right)} \exp\left(\frac{F(z)}{T}\right)} \right).$$

Consequently

$$\frac{\mathbb{P}(x'|x)}{\mathbb{P}(x|x')} = \exp\left(\frac{F(x') - F(x)}{T}\right).$$

Summarizing, the sequence $\{x(t); t \in \mathbb{Z}_+\}$ is a time-homogeneous, irreducible, aperiodic, and reversible Markov chain on the finite state space S, and its (unique) steady-state distribution follows the Gibbs distribution of (2). Then, standard Markov chain theory implies the convergence properties of Proposition 1. ■

Let us summarize some qualitative properties of the CGS algorithm that relate to its applicability in real-world systems.

1) It is distributed, as demonstrated in (3).
2) It is asynchronous, in the sense that each node updates its local state based on its own clock, which is independent of the clocks of other nodes.
3) Each iteration requires limited computational effort, provided the A_v parameters are fixed at relatively small values.
4) The system evolves in a "smooth" way since few components of just one local state are updated per iteration.
5) It is suitable for systems with heterogeneous nodes since the rate at which each node updates its local state is proportional to the dimensionality of this space.

A remark should be made on the *communication overhead* of the proposed algorithm. Each iteration requires the updating node to acquire the local states of all nodes in its two-tier neighborhood. This may result in significant communication overhead if local state spaces are large. Alternative implementation schemes can be considered and may be more appropriate depending on the application, e.g., each neighbor computes its objective function value and communicates it to the updating node, in a "push" rather than "pull" manner. Moreover, the CGS algorithm can be modified to include asynchronous updates, i.e., a node updates its local state based on possibly outdated information. We conjecture that convergence results similar to Proposition 1 can be established in that case as well, under some additional restrictions; see [2, Ch. 6 and 7]. It should be noted, though, that the substantially reduced communication overhead will typically come at the expense of a much slower convergence rate.

Another remark should be made on the *computational effort per node/iteration*: The implicit assumption is that local state updates are done instantaneously. Thus, in the event that the clocks of neighboring nodes tick within a very short period of time, the node that follows will observe the updated local state of the node the precedes. Therefore, even though in theory the CGS algorithm would work with any values of the A_v parameters, in practice these values have to be "small." The exact values will depend on the sizes of local state spaces and the computing capabilities of each node.

As a final related remark, we have assumed the global objective function $F(x) = \sum_{v\in\mathcal{V}} F_v(x_{\mathcal{N}_v^+})$ to be separable across nodes, but have allowed the local objective functions $F_v(x_{\mathcal{N}_v^+})$ to be arbitrary. In many applications of interest, however, we may have (partly) *separable local objective functions*, e.g.,

$$F_v(x_{\mathcal{N}_v^+}) = \sum_{W\in\mathcal{W}_v} F_{v,W}(x_v, x_W)$$

where $\mathcal{W}_v$ is a collection of subsets of $\mathcal{N}_v^+$. The two-tier neighborhood sets $\mathcal{N}_v^{(2)}$ can then be restricted to

$$\widehat{\mathcal{N}}_v^{(2)} = \left(\bigcup_{u\in\mathcal{N}_v^+} \cup_{W\in W_u : W\ni v} W\right) \setminus \{v\}$$

and the functions $F_u(y, x_{\mathcal{N}_v^+\setminus\{v\}})$ in the local update rule for node v can be replaced by

$$\sum_{W\in\mathcal{W}_u} F_{u,W}\left(y, x_{W\setminus\{v\}}\right).$$

This can prune a significant amount of edges in the CGS graph.

In particular, if $\mathcal{W}_v = \cup_{u\in\mathcal{N}_v}\{u\}$ so that

$$F_v(x_{\mathcal{N}_v^+}) = \sum_{u\in\mathcal{N}_v^{(1)}} F_{v,u}(x_v, x_u)$$

i.e., the local objective functions are separable across neighbors, then $\widehat{\mathcal{N}}_v^{(2)}$ reduces to the set of direct neighbors $\mathcal{N}_v^{(1)}$, and the CGS graph coincides with the interaction graph.

C. Optimality Considerations

In this section, we illustrate how the Gibbs distribution relates to the solution of the optimization problem (1). More specifically, we focus on the probability that the system is in a globally optimal state, in steady state

$$\pi_T(S^*) = \sum_{x\in S^*} \pi_T(x) = \frac{\sum_{x\in S^*} \exp\left[\frac{F(x)}{T}\right]}{\sum_{z\in S} \exp\left[\frac{F(z)}{T}\right]}$$

and explore the optimality properties of the CGS algorithm at different temperatures. Moreover, we present a modified version of the CGS algorithm that solves the optimization problem (1) exactly.

We start with two results that stem from well-known monotonicity properties of the Gibbs distribution.

Proposition 2: The probability $\pi_T(S^*)$ is a monotonically decreasing function of the temperature T.

Proposition 3: The probability $\pi_T(S^*)$ can be made arbitrarily close to one by choosing a sufficiently small temperature, i.e.,

$$\lim_{T\downarrow 0} \pi_T(S^*) = 1.$$

Unfortunately, these qualitative insights are difficult to quantify without making specific assumptions about the state spaces and the objective functions. Moreover, there is a fundamental tradeoff: Cheeger's inequality (see [21, Theorem 2]) suggests that running the CGS algorithm at a lower temperature results, typically, in a longer time period to reach equilibrium.

Let us elaborate on the *mixing time* of the system, i.e., the time that the Markov chain needs to get close to steady state, in total variation distance. Clearly, reaching steady-state behavior rather quickly is very important in many applications. In the context of large systems, "rapid mixing" is usually defined as polynomial scaling of the mixing time in the number of nodes. It should be noted that concrete results on mixing times are available only for Markov chains with much simpler structure than ours, e.g., see [15] and the references therein. On the other hand, it is well known that the optimization problem (1) is not only NP-hard, but also inapproximable [14], i.e., a polynomial-time algorithm that approximates an optimal solution most likely does not exist. This suggests that, under the CGS algorithm, either: 1) the Markov chain mixes rapidly, but it takes exponential time to find an optimal solution in steady state (this is expected to happen at high temperatures); or 2) the Markov chain mixes slowly, but it takes polynomial time to find an optimal solution in steady state (this is expected to happen at low temperatures). Thus, in the general (worst) case, one should not expect both good transient and good steady-state behavior for any fixed temperature. However, the simulations presented in Section IV-C reveal that the CGS algorithm achieves relatively good transient and steady-state performance in the wireless OFDMA application, possibly by (implicitly) taking advantage of its special structure.

Up until now, the discussion has been limited to the case of fixed temperature. Propositions 2 and 3 motivate the following modification of CGS, which we call the *annealed CGS algorithm*: The temperature starts from a relatively high value and decreases with time at a very slow rate. We denote by $T(t)$ the temperature at the tth global state transition. The monotonically decreasing sequence $\{T(t); t \in \mathbb{Z}_+\}$ is usually called the *cooling schedule*. The following result implies that the annealed CGS algorithm solves the optimization problem (1) exactly, provided the temperature is reduced at a sufficiently slow rate.

Proposition 4 (Annealed CGS): Consider the locally coupled system described above under the annealed CGS algorithm, with cooling schedule $\{c/\log(t); t \geq 2\}$. If the constant c is sufficiently large, then

$$\lim_{t\to\infty} \mathbb{P}\left(x(t) \in S^*\right) = 1.$$

Proof: The result follows directly from [23, Proposition 4.1 and Corollary 4.1]. ■

Similar results have appeared in the simulated annealing literature; see, for instance, [3]. Unfortunately, for most systems of interest, the convergence of the corresponding inhomogeneous Markov chain occurs at an extremely slow rate, rendering it ill-suited for practical purposes.

IV. Distributed Resource Allocation in OFDMA Cellular Networks

In Section III, we presented a generic methodology for distributed optimization in locally coupled systems. We now demonstrate how this methodology can be applied for distributed resource allocation in OFDMA cellular networks.

A. Model Description and Problem Formulation

We consider the downlink of a wireless OFDMA cellular network and denote by $\mathcal{K} = \{1,\ldots,K\}$ the set of cells (base stations), by $\mathcal{M} = \{1,\ldots,M\}$ the set of users, and by $\mathcal{N} = \{1,\ldots,N\}$ the set of frequencies (subbands).

Each user $i \in \mathcal{M}$ has two attributes: 1) a throughput utility function $U_i(\cdot)$; and 2) a minimum-throughput requirement $r_i^{\min}$. The minimum-throughput requirement could reflect an intrinsic rate requirement for a real-time (delay-sensitive) application, such as voice or video streaming, but it could also represent a minimum-rate guarantee for a best-effort session. The special case $r_i^{\min} = 0$ may be interpreted as a purely best-effort user. The case where $U_i(\cdot)$ is a step or "S-shaped" function may be thought of as a purely real-time user, who has an intrinsic rate requirement, but enjoys little or no benefit from receiving a higher rate. The goal is to maximize the aggregate throughput utility of the users while satisfying their minimum-throughput requirements.

Let the 0–1 variable q_{ik} indicate whether user $i \in \mathcal{M}$ is assigned to cell $k \in \mathcal{K}$ or not. Each user can be served by only one cell, i.e., $\sum_{k\in\mathcal{K}} q_{ik} = 1$, for all $i \in \mathcal{M}$.

Each cell can allocate power to any frequency, in discrete quanta $\Delta > 0$. We denote by P_{jk} the amount of power allocated by cell $k \in \mathcal{K}$ to frequency $j \in \mathcal{N}$. Cell k operates under a maximum total transmit power constraint $P_k^{\max}$, i.e., $\sum_{j\in\mathcal{N}} P_{jk} \le P_k^{\max}$, for all $k \in \mathcal{K}$.

Let G_{ik} be the channel gain from cell k to user i, which is assumed to be fixed. We further assume that the channel gain from a cell to a user can be neglected and treated as zero outside a certain finite range. Strictly speaking, of course, channel gains are never zero, but simply neglecting all channel gains below a small threshold value provides a suitable approximation for all practical purposes.

The signal-to-noise-and-interference ratio (SINR) of user i when served by cell k on frequency j is equal to

$$S_{ijk} = \frac{G_{ik}P_{jk}}{\eta + \sum_{l\neq k} G_{il}P_{jl}}$$

with η representing the thermal background noise.

We assume the existence of a function $C(\cdot)$ that describes how the feasible transmission rates depend on the SINR. A popular choice for $C(\cdot)$ is the logarithmic function, as dictated by the Shannon–Hartley theorem. For our purposes, this function can be arbitrary, as long as it satisfies $C(0) \equiv 0$. Thus, $R_{ijk} = C(S_{ijk})$ is the rate received by user i, when served by cell k on frequency j.

Let τ_{ijk} be the fraction of time granted by cell k to user i on frequency j, with the natural constraint $\sum_{i\in\mathcal{M}} \tau_{ijk} \le 1$. The total throughput received by user i can be expressed as

$$R_i = \sum_{j\in\mathcal{N}} \sum_{k\in\mathcal{K}} q_{ik}\tau_{ijk}C(S_{ijk}), \qquad i \in \mathcal{M}.$$

Note that it only makes sense for a cell to grant time to a user that it serves, and on a frequency to which it allocates power. Hence, we should have $\tau_{ijk} = 0$ when $q_{ik} = 0$ or $P_{jk} = 0$, but the expressions for R_i and S_{ijk} show that there is no need to explicitly include such constraints. It is crucial to observe that the interference term in the SINR causes the user throughputs to be nonconcave functions of the power allocation variables P_{jk}, for all reasonable choices of $C(\cdot)$. This, in turn, causes the utility maximization problem to be nonconvex, even when the utility functions are concave and the integrality constraints are ignored.

Summarizing, our goal is to maximize the aggregate throughput utility of the users, while satisfying their minimum-throughput requirements, with the decision variables at hand being: 1) the assignment of users to cells, as represented in the variables q_{ik}, which will be called the assignment problem; 2) the allocation of power to frequencies in each cell, as described by the variables P_{jk}, which will be referred to as the power allocation problem; 3) the allocation of time fractions to users within each cell on each frequency, as specified by the variables τ_{ijk}, which will be termed the scheduling problem. In mathematical terms, we wish to solve the following nonconvex, typically large-scale, mixed integer optimization problem

$$\begin{aligned}
\text{maximize} \quad & \sum_{i\in\mathcal{M}} U_i(R_i) \\
\text{subject to} \quad & R_i = \sum_{j\in\mathcal{N}} \sum_{k\in\mathcal{K}} q_{ik}\tau_{ijk}C(S_{ijk}) \ge r_i^{\min}, \qquad i \in \mathcal{M} \\
& S_{ijk} = \frac{G_{ik}P_{jk}}{\eta + \sum_{l\neq k} G_{il}P_{jl}}, \qquad i\in\mathcal{M}, j\in\mathcal{N}, k\in\mathcal{K} \\
& \sum_{k\in\mathcal{K}} q_{ik} = 1, \qquad i \in \mathcal{M} \\
& q_{ik} \in \{0,1\}, \qquad i \in \mathcal{M}, k \in \mathcal{K} \\
& \sum_{j\in\mathcal{N}} P_{jk} \le P_k^{\max}, \qquad k \in \mathcal{K} \\
& P_{jk}/\Delta \in \mathbb{N}, \qquad j \in \mathcal{N}, k \in \mathcal{K} \\
& \sum_{i\in\mathcal{M}} \tau_{ijk} \le 1, \qquad j \in \mathcal{N}, k \in \mathcal{K} \\
& \tau_{ijk} \ge 0, \qquad i \in \mathcal{M}, j \in \mathcal{N}, k \in \mathcal{K}.
\end{aligned}$$

Throughout the remainder of the section, we tacitly assume that the above problem has at least one feasible solution.

Note that the scheduling problem is entirely local in nature, since the time fractions are locally controlled, and only have an impact on the local objective function. Thus, it can be handled by the scheduling discipline without any impact on, or from, other cells. In contrast, the other two problems are global in nature. In the power allocation problem, the variables are locally controlled but have an impact on neighboring cells as well. In the user assignment problem, the decision variables are subject to global constraints and affect surrounding cells. This observation motivates the following reformulation of the problem: Let $q = [q_{ik}]$ and $P = [P_{jk}]$ be the global user assignment and power allocation matrices. Then, the resource allocation problem at hand can be written as

$$\begin{aligned} \text{maximize} \quad & \sum_{k\in\mathcal{K}} V_k(P;q) \\ \text{subject to} \quad & \sum_{k\in\mathcal{K}} q_{ik} = 1, \qquad i \in \mathcal{M} \\ & q_{ik} \in \{0,1\}, \qquad i \in \mathcal{M}, k \in \mathcal{K} \\ & \sum_{j\in\mathcal{M}} P_{jk} \le P_k^{\max}, \qquad k \in \mathcal{K} \\ & P_{jk}/\Delta \in \mathbb{N}, \qquad j \in \mathcal{N}, k \in \mathcal{K} \end{aligned} \tag{4}$$

where $V_k(P;q)$ is the optimal value of the following scheduling problem:

$$\begin{aligned} \text{maximize} \quad & \sum_{i\in\mathcal{M}} q_{ik} U_i(R_i) \\ \text{subject to} \quad & R_i = \sum_{i\in\mathcal{N}} \tau_{ijk} C\left(\frac{G_{ik}P_{jk}}{\eta + \sum_{l\neq k} G_{il}P_{jl}}\right), \qquad i \in \mathcal{M} \\ & R_i \ge r_i^{\min}, \qquad i \in \mathcal{M} \\ & \sum_{i\in\mathcal{M}} \tau_{ijk} \le 1, \qquad j \in \mathcal{N}, k \in \mathcal{K} \\ & \tau_{ijk} \ge 0, \qquad i \in \mathcal{M}, j \in \mathcal{N}, k \in \mathcal{K}. \end{aligned}$$

The latter problem amounts to the maximization of a sum of utility functions subject to linear constraints. When the utility functions are concave, as is typically the case, this problem can be efficiently solved using standard dual-based optimization techniques. In this regard, it is worth observing that there are well-established scheduling algorithms in place for allotting time-frequency slots to the various users in LTE systems. These algorithms commonly work toward a similar aggregate throughput utility optimization objective as assumed above, be it in a possibly implicit or approximate sense. Although these algorithms typically do not solve for the time fractions in an explicit form, they usually do keep track of the smoothed throughput values of the various users, thus enabling the (approximate) calculation of the functions $V_k(P;q)$.

B. Distributed Method for Resource Allocation

We now describe how the resource allocation problem formulated in Section IV-A can be couched into the proposed methodological framework for distributed optimization.

First of all, note that $G_{ik} = 0$ implies that $S_{ijk} = 0$, for all $j \in \mathcal{N}$. We define

$$\mathcal{K}_i = \{k \in \mathcal{K} : G_{ik} > 0\}$$

to be the set of cells that have a nonzero channel gain to user i and could potentially serve this user. Similarly, we define

$$\mathcal{I}_k = \{i \in \mathcal{M} : G_{ik} > 0\}$$

to be the set of users to which cell k has a nonzero channel gain and could potentially be served by this cell. Also, let

$$\mathcal{L}_k^+ = \{l \in \mathcal{K} : \mathcal{I}_k \cap \mathcal{I}_l \neq \emptyset\} = \bigcup_{i\in\mathcal{I}_k} \mathcal{K}_i$$

be the set of cells that have at least one user "in common" with cell k, including cell k, and $\mathcal{L}_k = \mathcal{L}_k^+ \setminus \{k\}$ be the same set without cell k. Using this notation, we can write

$$V_k(P;q) \equiv V_k(P_{\mathcal{L}_k^+}, q_{\mathcal{I}_k}).$$

The joint global problem (4) may be mapped to the generic optimization problem (1) as follows. We consider two categories of nodes, one category corresponding to the cells, and the other to users. Specifically, we define $\mathcal{V} = \mathcal{V}_1 \cup \mathcal{V}_2$, with $\mathcal{V}_1 = \mathcal{K}$ and $\mathcal{V}_2 = \mathcal{M}$.

The local state variable of node $k \in \mathcal{V}_1$ corresponds to the power allocation vector $P_k = (P_{k1}, \ldots, P_{kN})$, and the local state space is $S_k = \{P_k/\Delta \in \mathbb{N}^N : \sum_{j\in\mathcal{N}} P_{jk} \le P_k^{\max}\}$ as before. The local utility function of node $k \in \mathcal{V}_1$ is $V_k(\cdot)$.

The local state variable of node $i \in \mathcal{V}_2$ consists of the assignment vector $q_i = (q_{i1}, \ldots, q_{iK})$, and the local state space is $S_i = \{q_i \in \{0,1\}^K : \sum_{k\in\mathcal{K}_i} q_{ik} \le 1\}$. Notice that in order to fit the problem formulation to the theoretical framework of Section II, we have relaxed the equality constraint to an inequality one, i.e., a user can be assigned to at most one cell. Thus, there is a possibility that a user is assigned to no cell at all. This minor complication can be dealt with by defining the objective function of node $i \in \mathcal{V}_2$ to be $V_i(q_i) = -B \cdot 1_{\{q_i=0\}}$, where B is a large positive constant, and the equality $q_i = 0$ is meant componentwise.

The neighborhood set of node $k \in \mathcal{V}_1$ is $\mathcal{N}_k^{(1)} = \mathcal{L}_k \cup \mathcal{I}_k$, and the neighborhood set of node $i \in \mathcal{V}_2$ is $\mathcal{N}_i^{(1)} = \mathcal{K}_i$.

With the above definitions, the CGS algorithm can be applied, as specified in Section III. We emphasize the distributed nature of the operation, in the sense that a local update only relies on information from the cell itself and those in a two-tier neighborhood. Specifically, conducting an update at cell k only involves the values of $V_l(P_k, P_{\mathcal{L}_l^+\setminus\{k\}}; q_{\mathcal{I}_l})$, $l \in \mathcal{L}_k^+$. In order to obtain these values, each of the cells $m \in \mathcal{N}_k^{(2)} \cup \{k\}$ can first pass its power allocation vector P_m (e.g., by broadcasting) to the cells in $\mathcal{L}_m \cap \mathcal{L}_k^+$, i.e., the neighbors it has in common with cell k, so that each of the cells $l \in \mathcal{L}_k^+$ knows P_k and $P_{\mathcal{L}_l^+\setminus\{k\}}$. (In fact, the latter values can also be simply obtained from measurements if pilot signal strengths are known.) With that knowledge, each of the cells $l \in \mathcal{L}_k^+$ can then compute $V_l(P_k, P_{\mathcal{L}_l^+\setminus\{k\}}; q_{\mathcal{I}_l})$. The latter calculation essentially amounts to solving the scheduling problem at node l, which can be done locally and efficiently in case the utility functions are concave, as mentioned earlier. Once the values of $V_l(P_k, P_{\mathcal{L}_l^+\setminus\{k\}}; q_{\mathcal{I}_l})$ have been calculated, the cells $l \in \mathcal{L}_k$ can pass them to cell k. Likewise, performing an update by user i only involves the values of $V_i(q_i) = -B \cdot 1_{\{q_i=0\}}$ and $V_k(P_{\mathcal{L}_k^+}; q_i, q_{\mathcal{I}_k\setminus\{i\}})$, $k \in \mathcal{K}_i$. The former value can be trivially determined locally, while the latter values can be obtained in a similar fashion as described above. As before, an alternative option is to take advantage of the fact that in practice there are well-established scheduling algorithms in place that work toward a similar aggregate throughput utility objective. Thus, the cells $l \in \mathcal{L}_k$ can simply use the measured user throughput values produced by these algorithms to estimate the local objective functions rather than calculate them.

We note that the above formulation can be modified slightly in order to solve the "Channel Selection" problem considered in Kauffmann *et al.* [12], even though the latter is cast in the context of IEEE 802.11 WLANs rather than OFDMA cellular networks. Cell $k \in \mathcal{K}$ can use any frequency, but is only allowed to select a single one to operate on and is assumed to transmit at a fixed power level P_k^{fixed}. In our formulation, this can be represented with power quanta $\Delta_k = P_k^{\max} = P_k^{\text{fixed}}$. By choosing the local objective function of cell k to be

$$V_k(P) = -\eta - \sum_{j \in \mathcal{N}} \sum_{l \neq k} P_{jl} \cdot 1_{\{P_{jk} \cdot P_{jl} > 0\}}$$

the CGS algorithm minimizes the total interference.

C. Numerical Experiments

We now present the numerical experiments that we have conducted to examine the performance of the proposed distributed optimization approach in the context of OFDMA cellular networks.

In the numerical experiments presented below, we have assumed that the assignment of users to cells is fixed, and that each user is assigned to its nearest cell, as it is usually done in practice. Moreover, we implement the following approximation to the CGS algorithm: Every time a cell updates its power allocation, it selects one of three possible local states with a certain probability: 1) lower the power level by Δ on a randomly selected frequency; 2) raise the power level by Δ on a randomly selected frequency; 3) swap an amount of power Δ between two randomly selected frequencies. If the selected candidate state is infeasible (because the power level allocated to one of the frequencies would become negative or because the maximum available power would be exceeded) or identical to the current one, then we simply keep the current state. Otherwise, the selected candidate state becomes the new local state with probability

$$\min\left\{1, \exp\left(-\frac{F(\text{current}) - F(\text{candidate})}{T}\right)\right\}.$$

This mechanism falls in the general framework of Metropolis–Hastings sampling and induces dynamics very similar to CGS.

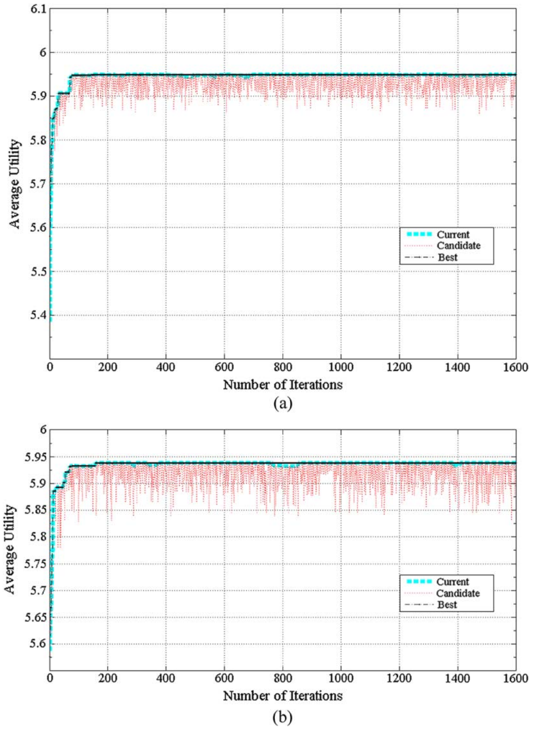
In order to benchmark the performance of the proposed distributed optimization approach, the litmus test would obviously be to compare the solution to the global optimum. Unfortunately, no computationally viable method is available to find the global optimum in all but the smallest networks. Note that the size of the local state spaces at each of the cell sites equals the number of ways to allocate up to $P^{\max}/\Delta$ power quanta among N frequencies, i.e., $S_0 = \binom{P^{\max}/\Delta + N}{N}$, and the size of the collective state space is S_0^K. Also, recall that the evaluation of the global objective function for a given state involves the calculation of K local objective functions, which in turn entails the solution to the local scheduling problem. This solution can be found efficiently for concave utility functions, but, typically, still requires a nontrivial computation effort. Thus, in order for the global optimum to be tractable, it is critical to limit the size of the global state space and confine to relatively small values of K, N, and $P^{\max}/\Delta$. Moreover, if the assignment of users to cells is included as part of the problem, then this will multiply the size of the global state space by an additional factor K^M.

In light of the above considerations, in the first experiment we focus on a system that is sufficiently small so that the global optimum can be found by sheer enumeration. Specifically, we consider a system with $K = 4$ cell sites located in a rectangular coverage area of $X_0 = 4.000$ by $Y_0 = 3.464$ km. In order to avoid boundary effects, we adopt the usual assumption that the edges are connected in a "wraparound" manner. We distinguish two scenarios for the locations of the cell sites: 1) a regular hexagonal pattern; and 2) a pseudo-random placement. In scenario 1, the four cell sites have coordinates $(x_1, y_1) = ((3/8)X_0, (1/4)Y_0)$, $(x_2, y_2) = ((7/8)X_0, (1/4)Y_0)$, $(x_3, y_3) = ((1/8)X_0, (3/4)Y_0)$, $(x_4, y_4) = ((5/8)X_0, (3/4)Y_0)$. Due to the wraparound boundary, the locations form a hexagonal pattern, with a distance between each pair of cell sites of $d = 2$ km. In scenario 2, the coordinates of the four cell sites are randomly perturbed, and the kth cell site is located in a rectangle of size $X_0/2$ by $Y_0/2$ km centered around (x_k, y_k).

The system supports a population of $M = 64$ users, which are distributed uniformly at random across the coverage area. In order to limit the size of the state space for the above-mentioned reasons, we fix the assignment of users to cells and assume that each user is served by the nearest cell site. In addition, we assume that there are (only) $N = 4$ frequencies.

The assumptions concerning channel gains are broadly consistent with the standard 3GPP propagation models. In particular, the channel gain value from cell site k to user i is $G_{ik} = H(D_{ik})$, with D_{ik} the distance between cell site k and user i (in kilometers), and $H(d) = 10^{h_0} d^{-\kappa}$, i.e., $H(d) = 10h_0 - 10\kappa \log_{10}(d)$ (dB), with a path-loss exponent $\kappa = 3.5$ and $h_0 = -14.4$. The thermal background noise is $\eta = -174$ dBm (Hz^{-1}), and the bandwidth per frequency is 1 MHz. Each cell site has a maximum total transmit power budget of $P_k^{\max} = 16$ W. In order to limit the size of the state space, we assume that power can only be allocated to the various frequencies in quanta of $\Delta = 4$ W. The feasible transmission rate as a function of SINR is given by $C(s) = c_0 \log(1 + s)$ (kb/s), with $c_0 = 1000$.

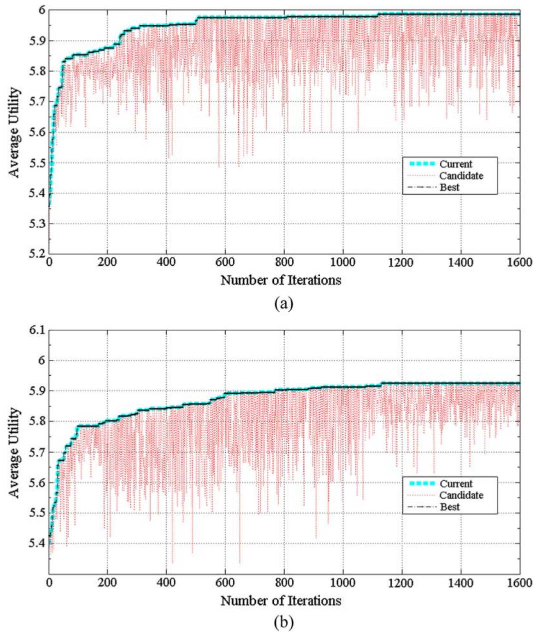
We consider Proportional Fairness as the optimization objective, i.e., each user has a logarithmic throughput utility function $U(r) = \log(r)$, and assume that there are no explicit minimum-throughput requirements specified. In order to facilitate a comparison to later experiments with larger numbers of users, we normalize the aggregate throughput utility by the number of users so as to obtain the average throughput utility per user: $(1/M) \sum_{i \in \mathcal{M}} \log(R_i) = (1/M) \log(\prod_{i=1}^{M} R_i)$. A brute-force search (the global state space consists of $70^4 = 2.401 \cdot 10^7$ states) yields that, in the globally optimal solution, the average throughput utility per user equals 5.947 and 5.938 in the case of a strictly hexagonal cell site arrangement and irregular placement, respectively. Fig. 2(a) and (b) plots the objective value produced by the proposed distributed



(a)

(b)

Fig. 2. Each chart consists of three curves: 1) the current solution resulting from the CGS algorithm (light dashed line); 2) the candidate solution, which is considered but not necessarily accepted by CGS (dotted line); and 3) the best solution found up to given iteration (black dashed line). The charts correspond to $N = 4$ frequencies, $K = 4$ cell sites, and $M = 64$ users, for a (a) hexagonal and (b) irregular cell site placement.



(a)

(b)

Fig. 3. Each chart consists of three curves: 1) the current solution resulting from the CGS algorithm (light dashed line); 2) the candidate solution, which is considered but not necessarily accepted by CGS (dotted line); and 3) the best solution found up to given iteration (black dashed line). The charts correspond to $N = 16$ frequencies, $K = 4$ cell sites, and $M = 64$ users, for a (a) hexagonal and (b) irregular cell site placement.

optimization approach with a temperature value $T = 0.1$ as function of the number of iterations for a random initial state. The various curves correspond to the current solution, the candidate solution (the spikes), and the best solution so far (the upper envelope). We observe convergence to the global optimum in about 80 iterations for the regular cell site arrangement and about 150 iterations for the perturbed placement. We obtained similar results for two other initial states: 1) a full reuse pattern with the maximum total power uniformly distributed across all frequencies; and 2) a factor-four reuse pattern with the maximum total power equally allocated to a quarter of the frequencies (so just a single frequency when $N = 4$), with the latter initial state being somewhat worse and causing slower convergence.

We next consider the same system with $N = 16$ rather than $N = 4$ frequencies, and reduce the size of the power quanta from $\Delta = 4$ to $\Delta = 1$ W. This increases the size of each of the local state spaces from 70 to 4845 (almost by a factor 70), and thus increases the size of the collective state space by almost a factor $70^4 = 2.401 \cdot 10^7$ to around $6 \cdot 10^{14}$, making it impossible to obtain the global optimum via enumeration. Since the same resource allocation can be reproduced (by simply bundling the frequencies into groups of four), the globally optimal solution should, however, improve. As before, Fig. 3(a) and (b) plots the objective value as function of the number of iterations for a random initial state and indeed shows comparable normalized throughput utility values as in the previous scenario. We continue to observe convergence, be it at a somewhat slower rate now due to the larger number of frequencies. For the initial state with a factor-four reuse pattern (results are not shown), we saw sluggish convergence, suggesting a persistent local optimum.

In the third set of experiments presented in Fig. 4, we consider the same system with $N = 4$ frequencies, but now with $K = 64$ rather than $K = 4$ cell sites, and $M = 1024$ rather than $M = 64$ users, scaled up by factor 16 as well. This increases the size of the collective state space by a factor 70^{60}, again rendering it prohibitively time-consuming to generate the global optimum via an exhaustive search. The results continue to show convergence, although the achievable throughput per user is somewhat lower now, as the larger number of cells causes a higher degree of interference. For cross comparison of the convergence rates, note that a total of 6400 iterations amounts to 100 iterations per cell, which corresponds to a total of 400 in the previous two experiments.

In the fourth and final set of experiments presented in Fig. 5, we consider a system with $K = 64$ cell sites, $M = 1024$ users, and $N = 16$ frequencies, yielding a collective state space of a size of 4845^{64}. The results again demonstrate convergence, be it that the larger number of frequencies further slows down the rate.

V. Distributed Nonconcave Utility Maximization in Multihop Networks

In this section, we demonstrate how the proposed methodology can be applied for distributed nonconcave utility maximization in multihop networks.

A. Model Description and Problem Formulation

We consider a multihop network with fixed routing, where links labeled by $l \in \mathcal{L}$ and users indexed by $i \in \mathcal{M}$.

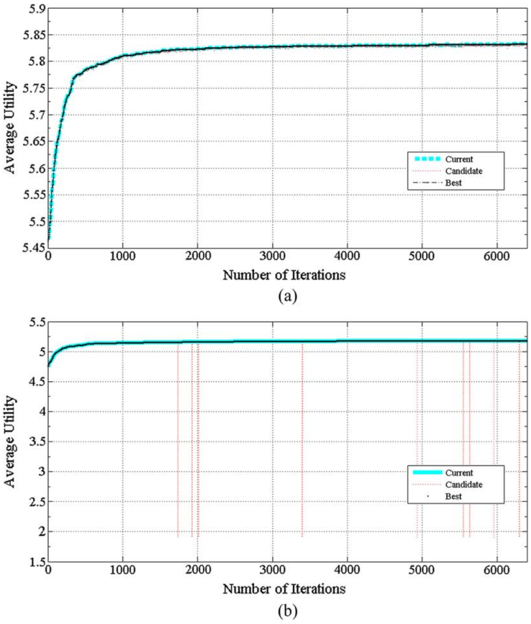


Fig. 4. Each chart consists of three curves: 1) the current solution resulting from the CGS algorithm (light dashed line); 2) the candidate solution, which is considered but not necessarily accepted by CGS (dotted line; coinciding with the light and black lines); and 3) the best solution found up to given iteration (black dashed line). The charts correspond to $N = 4$ frequencies, $K = 64$ cell sites, and $M = 1024$ users, for a (a) hexagonal and (b) irregular cell site placement.

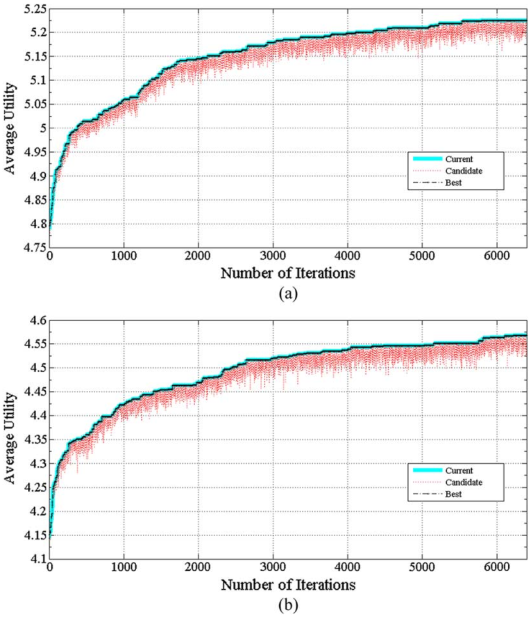


Fig. 5. Each chart consists of three curves: 1) the current solution resulting from the CGS algorithm (light dashed line); 2) the candidate solution, which is considered but not necessarily accepted by CGS (dotted line); and 3) the best solution found up to given iteration (black dashed line). The charts correspond to $N = 16$ frequencies, $K = 64$ cell sites, and $M = 1024$ users, for a (a) hexagonal and (b) irregular cell site placement.

Each user $i \in \mathcal{M}$ has three attributes: 1) a throughput R_i, which takes values in a finite set of nonnegative real numbers, X_i; 2) a throughput utility function $U_i(\cdot)$. We stress the fact that this function may be nonconcave. Similar to the application in wireless cellular networks, a concave utility function may be indicative of a best-effort user, while a nonconcave function, e.g., an "S-shaped" function, may be indicative of a delay-sensitive user; 3) a fixed route (sequence of links) along which the traffic of the user is carried.

We denote by A the 0–1 user-link incidence matrix, with A_{il} being equal to one if link l is traversed by the traffic of user i, and zero otherwise. The route set

$$\mathcal{R}_i = \{l \in \mathcal{L} : A_{il} = 1\}$$

consists of all the links traversed by the data flow of user i, and the set

$$\mathcal{S}_l = \{i \in \mathcal{M} : A_{il} = 1\}$$

contains all the users whose data flows traverse link l.

We denote by

$$C_l = \sum_{i \in \mathcal{M}} A_{il} R_i$$

the total throughput to be carried on link $l \in \mathcal{L}$. The feasible C_l values are assumed to be characterized by N capacity constraints: If $\mathcal{L}_j \subset \mathcal{L}$ is the set of links associated with the jth constraint, and $C_{\mathcal{L}_j}$ represents the total throughputs of the links in $\mathcal{L}_j$, then the capacity constraints take the generic form

$$G_j(C_{\mathcal{L}_j}) \geq 0, \qquad j \in \mathcal{N}$$

where $\mathcal{N} = \{1, \ldots, N\}$. For now, we only make the natural monotonicity assumption that each of the functions $G_j(\cdot)$ is decreasing in all arguments. We also use

$$\mathcal{J}_l = \{j \in \mathcal{N} : l \in \mathcal{L}_j\}$$

to denote the set of all constraints in which link l is involved.

This framework covers the special case of fixed link capacities $(\Gamma_1, \ldots, \Gamma_L)$, where $G_l(C_l) = \Gamma_l - C_l, l \in \mathcal{L}$, as is common in wireline networks with noninterfering transmission links. However, it also allows for modeling interfering links in wireless networks with power control, where the sustainable throughputs on the various links are mutually dependent.

It is worth observing that the functions $G_j(\cdot)$ may also be used to capture "soft" capacity constraints, e.g., average delay bounds. For example, in the special case of fixed link capacities $(\Gamma_1, \ldots, \Gamma_l)$, we could adopt

$$\Delta_l(C_l) = \frac{1}{\Gamma_l - C_l}$$

as a proxy for the average delay experienced on link l as function of the total carried throughput C_l on that link, and define

$$G_i(C_{\mathcal{R}_i}) = \tau_{\mathcal{R}_i} - \sum_{l \in \mathcal{R}_i} \Delta_l(C_l)$$

to impose an upper bound $\tau_{\mathcal{R}_i}$ on the average end-to-end delay of user $i \in \mathcal{M}$. This is similar in spirit to the problem considered in [20].

The goal is to maximize the sum of the users' throughput utility functions, subject to the given capacity constraints

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in \mathcal{M}} U_i(R_i) \\ \text{subject to} \quad & R_i \in X_i, \qquad i \in \mathcal{M} \\ & C_l = \sum_{i \in \mathcal{M}} A_{il} R_i, \qquad l \in \mathcal{L} \\ & G_j(C_{\mathcal{L}_j}) \geq 0, \qquad j \in \mathcal{N}. \end{aligned}$$

Throughout the remainder of the section, we tacitly assume that the above problem has at least one feasible solution.

The problem may be rewritten in the form

$$\begin{aligned} \text{maximize} \quad & \sum_{i \in \mathcal{M}} U_i(R_i) - \sum_{j \in \mathcal{N}} \hat{G}_j(C_{\mathcal{L}_j}) \\ \text{subject to} \quad & R_i \in X_i, \qquad i \in \mathcal{M} \\ & C_l = \sum_{i \in \mathcal{M}} A_{il} R_i, \qquad l \in \mathcal{L} \end{aligned}$$

where $\hat{G}_j(C_{\mathcal{L}_j}) = B \cdot 1_{\{G_j(C_{\mathcal{L}_j}) < 0\}}$ and B is a large positive constant.

B. Distributed Method for Throughput Utility Maximization

We now describe how the throughput utility maximization problem formulated above can be couched into the proposed methodological framework for distributed optimization.

In view of the structure of the objective function, we define the set of nodes $\mathcal{V} = \mathcal{M} \cup \mathcal{N}$ to comprise the set of users as well as the set of link capacity constraints. The local state variable of a node $i \in \mathcal{M}$ is the throughput of the associated user, R_i, and the corresponding local state space is X_i. Nodes $j \in \mathcal{N}$ do not have any local state variables.

A user $i \in \mathcal{M}$ and a link capacity constraint $j \in \mathcal{N}$ are neighbors when $\mathcal{R}_i \cap \mathcal{L}_j \neq \emptyset$, i.e., the jth capacity constraint involves a link that is traversed by user i. Thus, the neighborhood set of a node $i \in \mathcal{M}$ is

$$\mathcal{N}_i^{(1)} = \{j : \mathcal{R}_i \cap \mathcal{L}_j \neq \emptyset\} = \bigcup_{l \in \mathcal{R}_i} \mathcal{J}_l$$

and the neighborhood set of a node $j \in \mathcal{N}$ is

$$\mathcal{N}_j^{(1)} = \{i : \mathcal{L}_j \cap \mathcal{R}_i \neq \emptyset\} = \bigcup_{l \in \mathcal{L}_j} \mathcal{S}_l.$$

Consequently, the two-tier neighborhood of node $i \in \mathcal{M}$ is

$$\mathcal{N}_i^{(2)} = \left(\bigcup_{l \in \mathcal{R}_i} \left(\mathcal{J}_l \cup \bigcup_{j \in \mathcal{J}_l} \bigcup_{m \in \mathcal{L}_j} \mathcal{S}_m \right) \right) \setminus \{i\}$$

consisting of all the link capacity constraints that involve a link traversed by user i, along with all other users that traverse a link that is involved in any of these constraints. Also, the two-tier neighborhood of a node $j \in \mathcal{N}$ is

$$\mathcal{N}_j^{(2)} = \left(\bigcup_{l \in \mathcal{L}_j} \left(\mathcal{S}_l \cup \bigcup_{i \in \mathcal{S}_l} \bigcup_{m \in \mathcal{R}_i} \mathcal{J}_m \right) \right) \setminus \{j\}$$

consisting of all the users that traverse a link that is involved in the jth constraint, along with all other constraints that involve a link that is traversed by any of these users.

The local objective function of a node $i \in \mathcal{M}$ is $U_i(\cdot)$, and the local objective function of a node $j \in \mathcal{N}$ is $-\hat{G}_j(\cdot)$. With the above definitions, the global objective function may indeed be written as sum of the local objective functions, and the CGS algorithm can be applied as specified in Section III. Note that the local state spaces of users are one-dimensional for the particular problem, which implies that the CGS algorithm reduces to the original Gibbs Sampler.

In the special case of fixed link capacities, there is a one-to-one correspondence between the set of links $\mathcal{L}$ and the set of capacity constraints $\mathcal{N}$. Therefore, $\mathcal{J}_l = \{l\}$ and $\mathcal{L}_j = \{j\}$, and the above neighborhood sets reduce to $\mathcal{N}_i = \mathcal{R}_i$ and $\mathcal{N}_j = \mathcal{S}_j$. Moreover, the two-tier neighborhoods reduce to

$$\mathcal{N}_i^{(2)} = \left(\mathcal{R}_i \cup \bigcup_{l \in \mathcal{R}_i} \mathcal{S}_l \right) \setminus \{i\} = \left(\bigcup_{l \in \mathcal{R}_i} (\{l\} \cup \mathcal{S}_l) \right) \setminus \{i\}$$

and

$$\mathcal{N}_j^{(2)} = \left(\mathcal{S}_j \cup \bigcup_{i \in \mathcal{S}_j} \mathcal{R}_i \right) \setminus \{j\} = \left(\bigcup_{i \in \mathcal{S}_j} (\{i\} \cup \mathcal{R}_i) \right) \setminus \{j\}.$$

Finally, the local objective function of node $j \in \mathcal{N}$ reduces to $-\hat{G}_j(R_{\mathcal{S}_j}) = -B \cdot 1_{\{\sum_{i \in \mathcal{S}_j} R_i > \Gamma_j\}}$.

We emphasize the distributed nature of the operation, in the sense that a local update of user i only relies on the throughput of the user itself, and the throughputs of the users $h \in \mathcal{N}_i^{(2)} \setminus \mathcal{N}$ in a two-tier neighborhood. The latter dependence arises through the link throughputs C_m, where $m \in \mathcal{K}_i = \cup_{l \in \mathcal{R}_i} \cup_{j \in \mathcal{J}_l} \mathcal{L}_j$. The actual size of the set of links $\mathcal{K}_i$ strongly depends on the nature of the link capacity constraints. When the individual link capacity constraints are local in the sense that they only involve a single link, then the set of links $\mathcal{K}_i$ coincides with $\mathcal{R}_i$, the route set consisting of the links traversed by the data flow of user i. On the other hand, when the link capacity constraints are global in the sense that they involve most or all of the links, then the set $\mathcal{K}_i$ could contain most, if not all, the network links.

VI. Conclusion

In this paper, we proposed a general methodology for distributed optimization in locally coupled systems. In contrast to earlier studies, our method does not rely on any on convexity/concavity assumptions and accommodates arbitrary utility functions as well as integrality constraints. We applied the proposed approach to a joint power allocation and user assignment problem arising in wireless OFDMA cellular networks and to a rate allocation problem arising in multihop wireline networks. We established analytically the convergence of this algorithm and presented numerical experiments to illustrate its overall performance.

Future work involves further numerical experiments to examine how the rate of convergence depends on the temperature value and other design parameters used in the local update algorithm. Also, in the present paper, we have assumed a static user population with fixed channel gains. A crucial direction for future work is to extend the approach to scenarios with a dynamic user population (due to arrivals and departures) and random variations in channel gains (due to fading and user mobility).

REFERENCES

[1] U. Akyol, M. Andrews, P. Gupta, J. Hobby, I. Saniee, and A. Stolyar, “Joint scheduling and congestion control in mobile ad-hoc networks,” in *Proc. 27th IEEE INFOCOM*, Phoenix, AZ, USA, 2008, pp. 619–627.

[2] D. P. Bertsekas and J. N. Tsitsiklis*, Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.

[3] D. Bertsimas and J. N. Tsitsiklis, “Simulated annealing,” *Stat. Sci.*, vol. 8, no. 1, pp. 10–15, 1993.

[4] S. C. Borst, I. Saniee, and P. A. Whiting, “Distributed dynamic load balancing in wireless networks,” in *Proc. 20th Int. Teletraffic Conf.*, Ottawa, ON, Canada, 2007, pp. 1024–1037.

[5] P. Bremaud*, Markov Chains: Gibbs fields, Monte Carlo Simulation, and Queues*. New York, NY, USA: Springer, 1999.

[6] C. S. Chen and F. Baccelli, “Self-optimization in mobile cellular networks: Power control and user association,” in *Proc. IEEE ICC*, Cape Town, South Africa, 2010, pp. 1–6.

[7] C. S. Chen, F. Baccelli, and L. Roullet, “Joint optimization of radio resources in small and macro cell networks,” in *Proc. 74th IEEE VTC*, San Francisco, CA, USA, 2011, pp. 1–5.

[8] S. Das, H. Viswanathan, and G. Rittenhouse, “Dynamic load balancing through coordinated scheduling in packet data systems,” in *Proc. 22nd IEEE INFOCOM*, San Francisco, CA, USA, 2003, vol. 1, pp. 786–796.

[9] P. Hande, S. Patil, and H. Myung, “Distributed load balancing in a multi-carrier wireless system,” in *Proc. IEEE Conf. Wireless Commun. Netw.*, 2009, pp. 1–6.

[10] S. V. Hanly, L. Andrew, and T. Thanabalasingham, “Dynamic allocation of subcarriers and transmit powers in an OFDMA cellular network,” *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5445–5462, Dec. 2009.

[11] I.-H. Hou and P. Gupta, “Distributed resource allocation for proportional fairness in multi-band wireless systems,” in *Proc. IEEE ISIT*, Saint Petersburg, Russia, 2011, pp. 1975–1979.

[12] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mharte, K. Papagiannaki, and C. Diot, “Measurement-based self organization of interfering 802.11 wireless access networks,” in *Proc. 26th IEEE INFOCOM*, Anchorage, AK, USA, 2007, pp. 1451–1459.

[13] F. P. Kelly, A. Maulloo, and D. Tan, “Rate control in communication networks: Shadow prices, proportional fairness and stability,” *J. Oper. Res. Soc.*, vol. 49, no. 3, pp. 237–252, 1998.

[14] M. Koppe, “On the complexity of mixed-integer nonlinear optimization,” in *Mixed-Integer Nonlinear Programming*, J. Lee and S. Leyffer, Eds. Berlin, Germany: Springer, 2011.

[15] D. A. Levin, Y. Peres, and E. L. Wilmer*, Markov Chains and Mixing Times*. Providence, RI, USA: Amer. Math. Soc., 2009.

[16] T. Liggett*, Interacting Particle Systems*. New York, NY, USA: Springer, 1985.

[17] S. Low and D. Lapsley, “Optimization flow control: Basic algorithm and convergence,” *IEEE/ACM Trans. Netw.*, vol. 7, no. 2, pp. 861–874, Apr. 1999.

[18] L. P. Qian, Y. J. Zhang, and M. Chiang, “Globally optimal distributed power control for nonconcave utility maximization,” in *Proc. IEEE GLOBECOM*, Miami, FL, USA, 2010.

[19] S. Rangan and R. Madan, “Belief propagation methods for intercell interference coordination,” in *Proc. 30th IEEE INFOCOM*, Shanghai, China, 2011, pp. 2543–2551.

[20] I. Saniee, “Decentralized control and optimization of networks with QoS-constrained services,” in *Proc. IEEE ICC*, Dresden, Germany, 2009, pp. 1–6.

[21] A. Sinclair, “Improved bounds on mixing rates of Markov chains and multicommodity flow,” *Combin., Probab., Comput.*, vol. 1, no. 4, pp. 351–370, 1992.

[22] A. Stolyar and H. Viswanathan, “Self-organizing dynamic fractional frequency reuse in OFDMA systems,” in *Proc. 27th IEEE INFOCOM*, Phoenix, AZ, USA, 2008, pp. 691–699.

[23] J. N. Tsitsiklis, “Markov chains with rare transitions and simulated annealing,” *Math. Oper. Res.*, vol. 14, no. 1, pp. 70–90, 1989.

[24] X. Wu, A. Das, J. Li, and R. Laroia, “Fractional power reuse in cellular networks,” in *Proc. 44th Annu. Allerton Conf.*, Monticello, IL, USA, 2006, pp. 372–381.

[25] S. Zhou, X. Wu, and L. Ying, “Distributed power control and coding-modulation adaptation in wireless networks using annealed Gibbs sampling,” in *Proc. 31st IEEE INFOCOM*, Orlando, FL, USA, 2012, pp. 3016–3020.

Sem C. Borst received the M.Sc. degree in applied mathematics from the University of Twente, Enschede, The Netherlands, in 1990, and the Ph.D. degree from the University of Tilburg, Tilburg, The Netherlands, in 1994.

He has been with the Mathematics of Networks and Communications Department, Bell Laboratories, Murray Hill, NJ, USA, since 1995. From 1998 to 2006, he was a Senior Researcher with the Center for Mathematics and Computer Science (CWI), Amsterdam, The Netherlands. Since 1998, he also has a (part-time) appointment as a Professor in stochastic operations research with Eindhoven University of Technology, Eindhoven, The Netherlands. His main research areas are performance evaluation and resource allocation for stochastic systems, in particular computer-communication networks.

Dr. Borst serves or has served on the editorial boards of multiple journals and has been a program committee member of numerous conferences. He was (co-)recipient of the Best Paper awards at SIGMETRICS/Performance 1992 and IEEE INFOCOM 2003, the 2001 Yosef Levy Prize, and the 2005 Van Dantzig Prize.

Mihalis G. Markakis received the B.S. degree from the National Technical University of Athens, Athens, Greece, in 2005, and the M.S. degree from the University of Southern California, Los Angeles, CA, USA, in 2008, both in electrical engineering, and is currently pursuing the Ph.D. degree within the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA.

He held an internship position with Alcatel-Lucent Bell Labs, Murray Hill, NJ, USA, in the summer of 2010.

Iraj Saniee (M’98) received the B.A. (Hon.) and M.A. (Hon.) degrees in mathematics and Ph.D. degree in operations research and control theory from the University of Cambridge, U.K.

He is Head of the Mathematics of Networks and Communications Department, Bell Labs—Research, Alcatel-Lucent, Murray Hill, NJ, USA. His current research interests include distributed optimization, computations in large graphs, and applications to capacity and performance management in communication networks.

Dr. Saniee is a member of INFORMS and recently completed a six-year term on the Editorial Board of *Operations Research*. He has served on technical program committees of numerous conferences and journals.