Convex Approximation Algorithms for Back-Pressure Power Control

Evaggelia Matskani, Nicholas D. Sidiropoulos, *Fellow, IEEE*, and Leandros Tassiulas, *Fellow, IEEE*

***Abstract*—Throughput-optimal multihop wireless network operation entails a key physical-layer optimization problem: maximizing a weighted sum of link rates, with weights given by the differential queue backlogs. This emerges in joint back-pressure routing and power control, which is central in cross-layer wireless networking. We begin by showing that the core problem is not only nonconvex, but also NP-hard. This is a negative result, which however comes with a positive flip side: drawing from related developments in the digital subscriber line (DSL) literature, we propose effective ways to approximate it. Exploiting quasi-periodicity of the power allocation in stable setups due to the push-pull nature of the solution, we derive two custom algorithms that offer excellent throughput performance at reasonable, worst-case polynomial complexity. Judicious simulations illustrate the merits of the proposed algorithms.**

***Index Terms*—Back-pressure routing, convex approximation, cross-layer design, digital subscriber line (DSL), dynamic spectrum management, network optimization, NP-hard problems, power control, utility maximization.**

I. Introduction

Since its inception in the early 90s [24], back-pressure routing has won much acclaim as a (surprisingly) simple and effective adaptive routing solution that is optimal in terms of throughput. It was subsequently generalized in many ways (see [5] for a recent tutorial overview) and currently forms the backbone of emerging approaches aiming to optimize other performance objectives [17], [22].

Back-pressure policies owe their popularity to their natural agility and robustness (e.g., with respect to link failures and traffic dynamics). Back-pressure implementations for wired unicast networks are lightweight in terms of computation and signaling overhead, but the situation can be very different in wireless networks, due to fading, shadowing, and mutual interference. Along with these challenges come new opportunities, however: for example, it is possible to employ power control to obtain a more favorable "topology" from the viewpoint of maximizing throughput.

There is a lot of recent activity on optimization based approaches for network control and management, in particular for wireless networks. Advances in wireless technology over the last several years resulted in systems with much increased computational power at each radio node, while at the same time the channel sensing and measurement capabilities increased significantly as well. As a result, fairly sophisticated approaches for real-time network control become feasible. On the other hand, several recent theoretical advances in optimization-based network control increased our understanding of the theoretical performance limits that may be pursued in these systems [4], [5], [9], [11], [16], [17], [22], [23], [26].

Here we consider the joint routing and power control problem for maximal end-to-end throughput in a wireless multihop network. From the work of Tassiulas *et al.* [5], [16], [23]–[26], it is known that the following policy is optimal in the sense of enabling maximal stable throughput: choose link powers to maximize a *differential backlog*—weighted sum of link capacities; then route at each node using back-pressure. We call this policy *back-pressure power control* (BPPC). Our purpose here is to investigate the structure and properties of the BPPC problem, and come up with a suitable algorithm to solve it.

A conference version of part of this work appears in the *Proceedings of the IEEE ICASSP 2011* [15]. The conference version [15] presents the basic ideas, batch algorithms, and illustrative simulation results. This journal version adds custom algorithms, proofs and derivations, comprehensive simulations, and a fleshed-out discussion of results and insights.

II. System Model

Consider a wireless multihop network comprising N nodes. The topology of the network is represented by the directed graph $(\mathcal{N}, \mathcal{L})$, where $\mathcal{N} := \{1, \ldots, N\}$ and $\mathcal{L} := \{1, \ldots, L\}$ denote the set of nodes and the set of links, respectively. Each link $\ell \in \mathcal{L}$ corresponds to an ordered pair (i, j), where $i, j \in \mathcal{N}$ and $i \neq j$. Let $\mathrm{Tx}(\ell)$ and $\mathrm{Rx}(\ell)$ denote the transmitter and the receiver of link ℓ, i.e., when $\ell = (i, j)$, then $\mathrm{Tx}(\ell) = i$ and $\mathrm{Rx}(\ell) = j$. Data can be transmitted from any node to any other node, and each node may split its incoming traffic into multiple outgoing links. Let p_ℓ denote the power transmitted on link ℓ, and $G_{\ell k}$ the aggregate path loss between the transmitter of link

Manuscript received October 20, 2010; revised April 28, 2011 and November 27, 2011; accepted December 18, 2011. Date of publication January 12, 2012; date of current version March 06, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Josep Vidal. This work was co-financed by the European Union (European Social Fund—ESF) and Greek National Funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF)—Research Funding Program: Heracleitus II, investing in knowledge society through the European Social Fund. This work was also supported in part by ARL/ERO, under contract W911NF-09-1-0004 and EC FP7 project N-CRAVE. Part of this work has appeared in Proc. IEEE ICASSP 2011 Conference.

E. Matskani is with the Department of Electronic and Computer Engineering, Technical University of Crete, 73100 Chania, Greece (e-mail: matskani@telecom.tuc.gr).

N. D. Sidiropoulos is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: nikos@umn.edu).

L. Tassiulas is with the Department of Computer and Communication Engineering, University of Thessaly, Volos 38221, Greece (e-mail: leandros@uth.gr).

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/TSP.2012.2184097

ℓ and the receiver of link k. Then, the signal to interference plus noise ratio (SINR) experienced by the receiver of link ℓ is

$$\gamma_\ell = \frac{G_{\ell\ell}p_\ell}{\frac{1}{G}\sum_{\substack{k=1\\k\neq\ell}}^{L} G_{k\ell}p_k + V_\ell} \tag{1}$$

where V_ℓ is the background noise power, and G models spreading / beamforming gain, if any. The transmission rate on link ℓ is upper bounded by the maximum achievable rate c_ℓ, given by[1]

$$c_\ell = \log(1+\gamma_\ell) \overset{(1)}{=}$$
$$= \log\left(\sum_{k=1}^{L} G_{k\ell}p_k + V_\ell\right) - \log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} G_{k\ell}p_k + V_\ell\right) \tag{2}$$

where G has been absorbed in the coupling factors $G_{k\ell},\ k\neq\ell$. Due to (1), c_ℓ is a function of all the transmitted powers in the network.

III. Maximum Throughput

We assume that the system is slotted in unit time slots, indexed by t. Let us begin by considering a single flow in the network: traffic stems from node 1 (the source) and traverses the network to reach node N (the destination). Let $W_i(t)$ denote the queue length of node i at the end of slot t. The *differential backlog* [24] of link $\ell=(i,j)$, at the end of slot t is defined as

$$D_\ell(t) := \begin{cases} \max\{0, W_i(t) - W_j(t)\}, & j \neq N \\ W_i(t), & j = N. \end{cases} \tag{3}$$

Traffic flows through the links during each slot, based on the link capacities resulting from the power allocation at the beginning of the slot. The powers for slot $t+1$ are to be determined by solving the following optimization problem [5], [6], [10], [16], [23]–[26]

$$\mathsf{BPPC}: \quad \max_{\{p_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} D_\ell(t)c_\ell \tag{4}$$
$$\text{subject to} \quad 0 \leq \sum_{\ell:\mathrm{Tx}(\ell)=i} p_\ell \leq P_i, \quad i\in\mathcal{N}. \tag{5}$$

The objective function in (4) is a weighted sum of the capacities of all network links, where the differential backlogs serve as weighting factors. The optimization problem (4) and (5) aims to maximize the throughput of the network by favoring the links whose receiver is less congested than the transmitter. Note that, from (3), links destined to more congested nodes will have low differential backlogs, and are thus down-weighted in (4). Inequalities (5) upper bound the total transmission power of each node. Per-link power constraints can be used instead of (or together with) per-node power constraints, without changing the nature of the problem and the solutions proposed in the sequel.

Extension of the above throughput-optimal policy to the case of multiple flows turns out being surprisingly simple [24]. The only difference in the case of multiple flows is that each node maintains a separate queue for each flow, and each link computes the maximum differential backlog across all flows traversing the link. The BPPC problem is solved using these maximum differential backlogs as weights on link capacities, and each link then carries a flow that achieves the link's maximum differential backlog (winner-takes-all). This policy is throughput-optimal for multiple flows [24]. A more detailed discussion of the case of multiple flows (commodities) in our particular context can be found in the Appendix (see also [6]). It follows that our results are directly applicable to the case of multiple flows; we continue with a single flow for simplicity of exposition.

Unfortunately the objective function in (4) is nonconvex in the optimization variables p_l. This can be appreciated by rewriting the problem using (2)

$$\max_{\{p_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} D_\ell(t)\left(\log\left(\sum_{k=1}^{L} G_{k\ell}p_k + V_\ell\right) - \log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} G_{k\ell}p_k + V_\ell\right)\right) \tag{6}$$
$$\text{subject to} \quad 0 \leq \sum_{\ell:\mathrm{Tx}(\ell)=i} p_\ell \leq P_i, \quad i\in\mathcal{N}. \tag{7}$$

The logarithm of an affine expression is concave, but the difference of concave functions is, in general, neither convex nor concave. This raises the question of whether or not the BPPC problem can be efficiently solved. We have the following claim, whose proof can be found in the Appendix.

Claim 1: Back-pressure power control is NP-hard.

Back-pressure power control *looks like* the multiuser sum-rate maximization problem for the interference channel, but there is an important difference between the two: the $G_{k,\ell}$ parameters are subject to certain restrictions for the former, versus completely free for the latter. This means that the NP-hardness proof in [13] cannot be directly invoked. Another difference is that back-pressure power control may be subject to per-node (instead of per-link) power constraints, which couple the transmission power across multiple links. The key to a simple and clear proof is *backlog reduction*: realizing that there is freedom to choose the backlogs in such a way that we peel off these complicating factors to reveal the multiuser sum-rate maximization problem as a special case. Details can be found in the Appendix.

While formal proof of NP-hardness of BPPC was missing, and our work closes this gap, earlier work had already recognized that BPPC is a nonconvex problem that is likely difficult to solve. It is in part for this reason that [6] proposed a greedy on-line optimization approach. In the following, we take a different, *disciplined convex approximation* approach.

IV. Convex Approximation

We develop a convex approximation algorithm by borrowing an idea originally developed in the dynamic spectrum manage-

[1]The usual SINR gap parameter Γ can be introduced in the Shannon capacity formula to account for modulation loss, coding, etc. We skip this for brevity

ment for digital subscriber lines literature [18], [19]. The main point is to lower-bound the individual link rates using a concave function. In particular, we will use the following bound [18], [19]:

$$\alpha \log(z) + \beta \leq \log(1+z) \quad \text{for} \begin{cases} \alpha = \frac{z_o}{1+z_o}, \\ \beta = \log(1+z_o) - \frac{z_o}{1+z_o}\log(z_o) \end{cases} \tag{8}$$

which is tight at z_o. Notice that as $z_o \to \infty$, the bound becomes $\log(z) \leq \log(1+z)$.

Applying (8) to

$$\max_{\{p_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} D_\ell(t)\log(1+\gamma_\ell) \tag{9}$$

$$\text{subject to} \quad 0 \leq \sum_{\ell:\mathrm{Tx}(\ell)=i} p_\ell \leq P_i, \quad i \in \mathcal{N} \tag{10}$$

results in the approximation[2]

$$\max_{\{p_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} D_\ell(t)\left(\alpha_\ell(t)\log(\gamma_\ell) + \beta_\ell(t)\right) \tag{11}$$

$$\text{subject to} \quad 0 \leq \sum_{\ell:\mathrm{Tx}(\ell)=i} p_\ell \leq P_i, \quad i \in \mathcal{N}. \tag{12}$$

The maximization problem (11)–(12) is still nonconvex, since the objective is not concave in the variables p_ℓ. However, notice that

$$\log(\gamma_\ell) \overset{(1)}{=} \log(G_{\ell\ell}p_\ell) - \log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} G_{k\ell}p_k + V_\ell\right). \tag{13}$$

Introducing a logarithmic change of variables

$$\tilde{p}_\ell := \log p_\ell \tag{14}$$

yields

$$\log(\gamma_\ell) = \tilde{G}_{\ell\ell} + \tilde{p}_\ell - \log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{G}_{k\ell}+\tilde{p}_k} + e^{\tilde{V}_\ell}\right) \tag{15}$$

where $\tilde{G}_{k_\ell} := \log(G_{k_\ell})$ and $\tilde{V}_\ell := \log(V_\ell)$ for brevity. The logarithm of a sum of exponentials is convex, the minus sign reverts the curvature, and addition with an affine expression preserves curvature; hence, (15) is a concave function of $\{\tilde{p}_\ell\}_{\ell\in\mathcal{L}}$.

With respect to the optimization variables $\{\tilde{p}_\ell\}_{\ell\in\mathcal{L}}$, the problem of interest finally becomes

$$\max_{\{\tilde{p}_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} D_\ell(t)\left(\alpha_\ell(t)\left(\tilde{G}_{\ell\ell}+\tilde{p}_\ell-\log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{G}_{k\ell}+\tilde{p}_k}+e^{\tilde{V}_\ell}\right)\right) + \beta_\ell(t)\right) \tag{16}$$

$$\text{subject to} \quad \log\left(\sum_{\ell:\mathrm{Tx}(\ell)=i} e^{\tilde{p}_\ell}\right) \leq \tilde{P}_i := \log(P_i), \quad i \in \mathcal{N}. \tag{17}$$

[2]Since we maximize a lower bound, the solution obtained this way will yield a value for the original objective that is at least as high as the maximal lower bound.

The objective function (16) is concave, since it comprises a sum of linear and concave functions of $\{\tilde{p}_\ell\}_{\ell\in\mathcal{L}}$, cf. (15). The weights $D_\ell(t)$ are nonnegative, cf. (3), and so are the constants $\alpha_\ell(t)$ and $\beta_\ell(t)$, cf. (8). Moreover, inequalities (17) are convex, since, as noted before, the logarithm of a sum of exponentials is a convex function. Hence, optimization problem (16)–(17) is *convex*, since the maximum of a concave function is sought over a convex feasible set. The logarithmic change of variables (14) is reminiscent of geometric programming (GP); indeed, (16)–(17) is an *Extended GP* problem.

In the high-SINR regime ($\gamma_\ell \gg 1$) the bound $\log(\gamma_\ell) \leq \log(1+\gamma_\ell)$ is tight: $\log(\gamma_\ell) \approx \log(1+\gamma_\ell)$. This is an often-made (and debated) approximation; in our present context, it has been adopted in [6] to simplify the problem and enable derivation of best-response type algorithms. The issue with the high-SINR approximation is that we do not know beforehand whether it holds well at the optimum—unless worst-case interference is so low that the problem is easy to begin with.

The high-SINR approximation corresponds to using $\alpha_\ell(t) = 1$ and $\beta_\ell(t) = 0$, $\forall \ell \in \mathcal{L}$ in (16)–(17), yielding the following approximation algorithm.

Algorithm 1: `Batch High-SINR`

For each time slot t, calculate the differential backlogs and solve (16), (17) with $\alpha_\ell(t) = 1$, $\beta_\ell(t) = 0$, $\forall \ell \in \mathcal{L}$, to obtain $\mathbf{p}(t) = [p_1(t)\dots p_L(t)]^T$.

Solving the optimization problem in (16)–(17) maximizes a lower bound on the achievable differential backlog—weighted sum rate of all links. After obtaining $[p_1(t)\dots p_L(t)]^T$, the individual link rate bounds can be tightened by tuning the parameters $\alpha_\ell(t)$ and $\beta_\ell(t)$, $\forall \ell \in \mathcal{L}$ so that the bounds coincide with the link rates at $[p_1(t)\dots p_L(t)]^T$. This suggests the following successive approximation algorithm.

Algorithm 2: `Batch Successive Approximation (Batch S.A.)`

---

1) *Initialization:* For each time slot t, calculate the differential backlogs, reset iteration counter $s = 0$, and set $\alpha_\ell(t,s) = 1$ and $\beta_\ell(t,s) = 0$, $\forall \ell \in \mathcal{L}$.
2) **repeat:**
3) *Maximization step:* Solve (16), (17) to obtain $\mathbf{p}(t,s) = [p_1(t,s)\dots p_L(t,s)]^T$
4) *Tightening step:* Pick $\alpha_\ell(t,s)$, $\beta_\ell(t,s)$ according to (8) for $z_o = \gamma_\ell(\mathbf{p}(t,s))$, see (1), $\forall \ell \in \mathcal{L}$
5) $s = s+1$
6) **until** convergence of the objective value (within ϵ-accuracy).

The sequence of iterates produces a monotonically increasing objective. This is a corollary of the *majorization principle*, e.g., cf. [2]. The idea of using the bound in (8) to obtain a convex lower approximation, then tightening the bound to refine the approximation is the essence of the SCALE algorithm in [18], [19], originally proposed for spectrum balancing in cross-talk limited digital subscriber line (DSL) systems. Interestingly, the same problem structure emerges in our present multihop cross-layer networking context, where the objective is to choose link powers and flows for maximal stable end-to-end throughput.

In [19], it is shown that the SCALE algorithm converges to a KKT point of the original nonconvex power control problem. This also holds in our context, i.e., for each t, $\mathbf{p}(t,s)$ indexed by s converges to a KKT point of the original NP-hard BPPC problem. The argument in [19] carries over verbatim: the main point is that the lower bound approximation is exact at the converged solution.

V. Custom Algorithms

Unlike [18], [19], where the spectrum balancing problem is solved once (or "infrequently"), and the weights used to compute the weighted sum rate objective are fixed design parameters, we have to solve the problem on a per-slot basis with a different set of weights—the differential backlogs, which change dynamically from one scheduling slot to the next, as packets are routed in the network. The objective function changes, and, as a result, the power vector computed in the previous slot may be far from a good solution for the present slot. This calls for custom algorithms that avoid solving the problem from scratch at each slot. Towards this end, we first convert (16) and (17) into an unconstrained optimization problem using a logarithmic barrier interior point method.

Consider (16) under per-link power constraints (per-node power constraints as in (17) can be similarly treated), and rewrite as

$$\min_{\{\tilde{p}_\ell\}_{\ell\in\mathcal{L}}} -f(\tilde{p}_1,\ldots,\tilde{p}_L) := -\sum_{\ell=1}^{L} D_\ell(t)\left(\alpha_\ell(t)\left(\tilde{G}_{\ell\ell}+\tilde{p}_\ell - \log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{G}_{k\ell}+\tilde{p}_k}+e^{\tilde{V}_\ell}\right)\right)+\beta_\ell(t)\right) \tag{18}$$

$$\text{subject to} \quad \tilde{p}_\ell \le \tilde{P}_\ell, \quad \ell\in\mathcal{L} \tag{19}$$

where now the cost function $-f$ is convex. Converting the explicit link-power constraints into implicit ones, we seek to

$$\min_{\{\tilde{p}_\ell\}_{\ell\in\mathcal{L}}} -f(\tilde{\mathbf{p}})+\sum_{\ell=1}^{L} I_-(\tilde{p}_\ell-\tilde{P}_\ell) \tag{20}$$

where

$$I_-(u) := \begin{cases} 0, & u\le 0\\ +\infty, & u>0 \end{cases}. \tag{21}$$

Interior point methods approximate this using the convex differentiable function

$$\widehat{I}_-(u) = -\frac{1}{\tau}\log(-u), \quad \text{for } \tau>0 \tag{22}$$

which converges to $I_-(\cdot)$ as $\tau\to\infty$. In practice, it suffices to pick τ large enough.[3] This yields the unconstrained convex minimization problem

$$\min_{\{\tilde{p}_\ell\}_{\ell\in\mathcal{L}}} -\tau f(\tilde{\mathbf{p}})-\sum_{\ell=1}^{L}\log(\tilde{P}_\ell-\tilde{p}_\ell), \quad \tau\gg 0 \tag{23}$$

i.e.,

$$\min_{\{\tilde{p}_\ell\}_{\ell\in\mathcal{L}}} \tilde{f}(\tilde{\mathbf{p}}) := -\tau\left\{\sum_{\ell=1}^{L} D_\ell(t)\left(\alpha_\ell(t)\left(\tilde{G}_{\ell\ell}+\tilde{p}_\ell - \log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{G}_{k\ell}+\tilde{p}_k}+e^{\tilde{V}_\ell}\right)\right)+\beta_\ell(t)\right)\right\} - \sum_{\ell=1}^{L}\log(\tilde{P}_\ell-\tilde{p}_\ell) \tag{24}$$

which can be solved via Newton's method to obtain a solution $\tilde{\mathbf{p}}^*(\tau)$ that approaches the optimal solution $\tilde{\mathbf{p}}^*$ of (18)–(19) as $\tau\to\infty$.

Recall that the motivation for considering custom solutions to the BPPC problem is that we have to solve it for each scheduling slot, with a different set of weights (the differential backlogs) and possibly time-varying $G_{k\ell}$'s. Even when the physical layer propagation conditions vary slowly with time, the differential backlogs can change swiftly from slot to slot.

Assuming deterministic fixed-rate (or random but bounded) arrivals and fixed physical layer propagation conditions, if all queues remain bounded then the system must exhibit periodic behavior—perhaps with a very long period. This is because there is a (large but still) finite number of system states, hence the system must return to a previously visited state in due time. The same holds for finite-state time-varying physical layer propagation conditions (e.g., of the on-off type)—we only need to augment the state vector to account for those. In practice we typically observe far shorter periods, due to the need to protect links from excessive interference (including no-listen-while-you-talk considerations), which often implies that the best strategy is to schedule "independent" (quasi) noninterfering subsets of links in subsequent slots. This way the system operates in a multistage push-pull fashion, giving rise to periodic or quasi-periodic behavior for stable setups.

Given the above, it is clear that the solution at slot t can be very different from the one at slot $t-1$; thus departing from the classical setting of adaptive algorithms. Not all is lost however: the key is to exploit the aforementioned (quasi) periodicity. Even though the previously computed solution can be far from the one needed in the present slot, chances are that one of the already encountered solutions for past slots is close to the one for the present slot. This idea is exploited in the following two algorithms.

[3]Or iterate with a gradually increasing τ.

Algorithm 3: `Adaptive high-SINR`

Fix $\alpha_\ell(t) = 1, \beta_\ell(t) = 0, \forall \ell \in \mathcal{L}$

For each time slot $t \geq 1$:

1) Calculate the differential backlogs
2) Power initialization:
   For $t = 1$ draw random $\tilde{\mathbf{p}}_o(t)$ satisfying log-power constraints; else for $t \in [2, W]$ set:

$$\tilde{\mathbf{p}}_o(t) = \arg \max_{\tilde{\mathbf{p}} \in \{\tilde{\mathbf{p}}(1),\ldots,\tilde{\mathbf{p}}(t-1)\}} f(\tilde{\mathbf{p}}) \tag{25}$$

   where

$$f(\tilde{\mathbf{p}}) := \sum_{\ell=1}^{L} D_\ell(t) \left(\alpha_\ell(t) \left(\tilde{G}_{\ell\ell} + \tilde{p}_\ell - \log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{G}_{k\ell}+\tilde{p}_k} + e^{\tilde{V}_\ell}\right)\right) + \beta_\ell(t)\right) \tag{26}$$

   else ($t \geq W + 1$) set:

$$\tilde{\mathbf{p}}_o(t) = \arg \max_{\tilde{\mathbf{p}} \in \{\tilde{\mathbf{p}}(t-W),\ldots,\tilde{\mathbf{p}}(t-1)\}} f(\tilde{\mathbf{p}}). \tag{27}$$

3) Starting from $\tilde{\mathbf{p}}_o(t)$, solve (24) via Newton's method:
   a) Reset iteration counter $s = 1$ and set auxiliary variable $\widehat{\mathbf{p}}(s) = \tilde{\mathbf{p}}_o(t)$.
   b) **repeat:**
   c) Compute Gradient vector $\nabla \tilde{f}(\widehat{\mathbf{p}}(s))$ and Hessian matrix $\nabla^2 \tilde{f}(\widehat{\mathbf{p}}(s))$, see Appendix.
   d) Compute Newton direction: $\mathbf{d} = -(\nabla^2 \tilde{f}(\widehat{\mathbf{p}}(s)))^{-1} \nabla \tilde{f}(\widehat{\mathbf{p}}(s))$.
   e) Line search: Choose a step size $\nu = \arg\min_{\mu\in[01]} \tilde{f}(\widehat{\mathbf{p}}(s) + \mu\mathbf{d})$.
   f) $s = s + 1$
   g) Update powers: $\widehat{\mathbf{p}}(s) = \widehat{\mathbf{p}}(s-1) + \nu\mathbf{d}$.
   h) **until:** convergence of the cost $\tilde{f}(\widehat{\mathbf{p}}(s))$ (within ϵ-accuracy).
4) Set $\tilde{\mathbf{p}}(t) = \widehat{\mathbf{p}}(s)$.

Notice that (25) and (27) simply evaluate the current objective function at previously computed solutions (for past slots). The parameter W should be large enough to capture emerging periodic behavior, but using larger than necessary W's is not a problem, as function evaluations are cheap relative to the Newton steps that follow.

Similar to the batch case, it is also possible to begin with a high-SINR approximation and successively refine it by tuning the parameters $\alpha_\ell(t)$ and $\beta_\ell(t)$ to tighten the individual link rate bounds. This yields the following algorithm.

Algorithm 4: `Adaptive Successive Approximation (Adaptive S.A.)`

Initialize link rate bound parameters: set $\alpha_\ell(t) = 1, \beta_\ell(t) = 0, \forall \ell \in \mathcal{L}$

For each time slot $t \geq 1$:

1) Calculate the differential backlogs
2) Power initialization:
   For $t = 1$ draw random $\tilde{\mathbf{p}}_o(t)$ satisfying log-power constraints; else for $t \in [2, W]$ set:

$$\tilde{\mathbf{p}}_o(t) = \arg \max_{\tilde{\mathbf{p}} \in \{\tilde{\mathbf{p}}(1),\ldots,\tilde{\mathbf{p}}(t-1)\}} f(\tilde{\mathbf{p}}) \tag{28}$$

   where

$$f(\tilde{\mathbf{p}}) := \sum_{\ell=1}^{L} D_\ell(t) \left(\alpha_\ell(t) \left(\tilde{G}_{\ell\ell} + \tilde{p}_\ell - \log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{G}_{k\ell}+\tilde{p}_k} + e^{\tilde{V}_\ell}\right)\right) + \beta_\ell(t)\right) \tag{29}$$

   else ($t \geq W + 1$) set:

$$\tilde{\mathbf{p}}_o(t) = \arg \max_{\tilde{\mathbf{p}} \in \{\tilde{\mathbf{p}}(t-W),\ldots,\tilde{\mathbf{p}}(t-1)\}} f(\tilde{\mathbf{p}}). \tag{30}$$

3) Outer initialization: Reset outer iteration counter $s_1 = 1$, set $\tilde{\mathbf{p}}_1(t, s_1) = \tilde{\mathbf{p}}_o(t)$, and pick $\alpha_\ell(t, s_1)$, $\beta_\ell(t, s_1)$ according to (8) for $z_o = \gamma_\ell(\tilde{\mathbf{p}}_1(t, s_1))$, see (1), $\forall \ell \in \mathcal{L}$.
4) **repeat:**
5) Starting from $\tilde{\mathbf{p}}_o(t)$, solve (24) via Newton's method:
   a) Inner initialization: Reset inner iteration counter $s_2 = 1$ and set auxiliary variable $\tilde{\mathbf{p}}_2(s_2) = \tilde{\mathbf{p}}_1(t, s_1)$.
   b) **repeat:**
   c) Compute Gradient vector $\nabla \tilde{f}(\tilde{\mathbf{p}}_2(s_2))$ and Hessian matrix $\nabla^2 \tilde{f}(\tilde{\mathbf{p}}_2(s_2))$, see Appendix.
   d) Compute Newton direction: $\mathbf{d} = -(\nabla^2 \tilde{f}(\tilde{\mathbf{p}}_2(s_2)))^{-1} \nabla \tilde{f}(\tilde{\mathbf{p}}_2(s_2))$.
   e) Line search: Choose a step size $\nu = \arg\min_{\mu\in[01]} \tilde{f}(\tilde{\mathbf{p}}_2(s_2) + \mu\mathbf{d})$.
   f) $s_2 = s_2 + 1$
   g) Update powers: $\tilde{\mathbf{p}}_2(s_2) = \tilde{\mathbf{p}}_2(s_2 - 1) + \nu\mathbf{d}$.
   h) **until:** convergence of the cost $\tilde{f}(\tilde{\mathbf{p}}_2(s_2))$ (within ϵ-accuracy).
6) $s_1 = s_1 + 1$
7) Set $\tilde{\mathbf{p}}_1(t, s_1) = \tilde{\mathbf{p}}_2(s_2)$
8) Tightening step: pick $\alpha_\ell(t, s_1)$, $\beta_\ell(t, s_1)$ according to (8) for $z_o = \gamma_\ell(\tilde{\mathbf{p}}_1(t, s_1))$, see (1), $\forall \ell \in \mathcal{L}$.
9) **until** convergence of the cost $\tilde{f}(\tilde{\mathbf{p}}_1(t, s_1))$ (within ϵ-accuracy).
10) Set $\tilde{\mathbf{p}}(t) = \tilde{\mathbf{p}}_1(t, s_1)$.

VI. Complexity of Convex Approximation

The worst-case complexity order of the batch algorithms is $O(L^{3.5})$, where L is the number of links (optimization variables) [1], [12]. In dense networks where every node is within range of every other node, $L = O(N^2)$, where N is the number of nodes; thus worst-case complexity is then $O(N^7)$. This is relatively high, but it is important to note that even the solution of a system of linear equations in the link powers would entail complexity $O(L^3) = O(N^6)$. The successive approximation algorithm typically converges in just a few (3–4 in our experiments) tightening steps, so complexity order is the same as the high-SINR one.

The worst-case complexity of the adaptive approximation algorithms is the same as that of the batch algorithms—mainly due to the matrix inversion in the Newton step which is cubic in

L, the remainder being the number of Newton steps needed to converge in the worst case. The constants that are hidden in the big O notation are of course far smaller for the adaptive algorithms, and so is their average complexity—due to their "reuse" of past solutions for warm restart. This will be illustrated in the simulations section. It is also possible to use quasi-Newton methods such as BFGS to further reduce the average complexity of the adaptive algorithms.

VII. Baselines

A. Assessing the Quality of Approximation

Given that the proposed convex approximation algorithms only find approximate solutions to the original NP-hard problem, we would like to develop means of assessing how far an approximate solution is from an optimal one. Returning to the original objective, we could use the inequality $\log(1+z) \leq \log(1+z_o) + \frac{z-z_o}{1+z_o}$ (follows from $\log(x) \leq x-1$), met with equality at z_o, to upper bound the individual terms of the objective function. This yields an upper bound, but its maximization is difficult, as it involves products of variables.

The classical way to obtain an upper bound is via duality—i.e., by considering the Lagrange dual problem. The dual problem is convex by definition; yet computing the dual function (objective of the dual problem) is also NP-hard. This can be established by showing that it contains (see Appendix) the corresponding computation for single-carrier sum-rate maximization in DSL systems, which is known to be NP-hard [14].

In [20], an algorithm called MAPEL is proposed, based on increasingly accurate approximation of the feasible SINR region to find an optimal solution to the weighted sum-rate maximization problem. When optimal solution is sought, NP-hardness implies that MAPEL's worst-case complexity is exponential. MAPEL can also be used for approximation, however our simulations indicated that it is too complex to be used even as a benchmark in our setting. For this reason, we will resort to an algorithm of Yu and Lui [27], originally developed for spectrum balancing in DSL systems, which yields an approximate solution of the dual problem—hence an approximate upper bound of our objective. This algorithm is briefly reviewed next.

1) Iterative Spectrum Balancing Algorithm: Considering per-link power constraints, the dual objective function $g(\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ denotes the vector of Lagrange dual variables, is the result of the unconstrained maximization

$$\begin{aligned} g(\boldsymbol{\lambda}) =& \max_{\{p_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} D_\ell(t) \log\left(1 + \frac{G_{\ell\ell}p_\ell}{\frac{1}{G}\sum_{\substack{k=1\\k\neq\ell}}^{L} G_{k\ell}p_k + V_\ell}\right) \\ &+ \boldsymbol{\lambda}^T(\mathbf{P} - \mathbf{p}) \\ =& \max_{\{p_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} D_\ell(t) \log\left(1 + \frac{G_{\ell\ell}p_\ell}{\frac{1}{G}\sum_{\substack{k=1\\k\neq\ell}}^{L} G_{k\ell}p_k + V_\ell}\right) \\ &+ \sum_{\ell=1}^{L} \lambda_\ell(P_\ell - p_\ell) \\ =& \max_{\{p_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} \left(D_\ell(t) \log\left(1 + \frac{G_{\ell\ell}p_\ell}{\frac{1}{G}\sum_{\substack{k=1\\k\neq\ell}}^{L} G_{k\ell}p_k + V_\ell}\right)\right. \\ &\left. - \lambda_\ell p_\ell\right) + \sum_{\ell=1}^{L} \lambda_\ell P_\ell. \end{aligned}$$

In [27], an iterative algorithm that alternates between primal (link powers) and dual variables is proposed to obtain an approximate solution to the dual problem. Following straightforward adaptation in our present (single-carrier, weighted sum rate) context, this algorithm can be summarized as follows:

Algorithm 5: `Iterative Spectrum Balancing (ISB)`

For each time slot t:
1) Initialize $\boldsymbol{\lambda} = [\lambda_1 \dots \lambda_L]^T$
2) **repeat:**
3) initialize $\mathbf{p} = [p_1 \dots p_L]^T$
4) **repeat:**
5) for $\ell = 1$ to L, set

$$p_\ell = \arg\max_{\mathrm{p}_\ell} \sum_{\mathrm{m}=1}^{\mathrm{L}} (\mathrm{D_m(t)} \\ \log\left(1 + \frac{G_{mm}p_m}{\frac{1}{G}\sum_{\substack{k=1\\k\neq m}}^{L} G_{km}p_k + V_m}\right) - \lambda_m p_m\Bigg)$$

end
6) **until** $\mathbf{p} = [p_1 \dots p_L]^T$ converges (within ϵ- accuracy)
7) Update $\boldsymbol{\lambda} = [\lambda_1 \dots \lambda_L]^T$ using subgradient or ellipsoid method
8) **until** $\boldsymbol{\lambda} = [\lambda_1 \dots \lambda_L]^T$ converges (within ϵ- accuracy).

In the above, the element-wise power optimization is nonconvex in p_ℓ, but can be accomplished via either line-search, or polynomial rooting (finding the roots of the first derivative and examining the second derivative as well). A subgradient iteration is employed for the $\boldsymbol{\lambda}$-update step (7)

$$\boldsymbol{\lambda}^{(m+1)} = \max\left(\boldsymbol{\lambda}^{(m)} - s^{(m)}(\mathbf{P} - \mathbf{p}), \mathbf{0}\right)$$

where m is the iteration number, and the step-size sequence is usually taken as $s^{(m)} = \frac{\beta}{m}$, for some $0 < \beta \leq 1$.

B. Prior Art: Back-Pressure Best Response Algorithm

In addition to a (possibly unattainable, by often tight) upper bound, we will also compare the proposed algorithms to the earlier state-of-art for the BPPC problem—namely [6], where two low-complexity algorithms were derived under the high-SINR assumption. These are the *Best Response* algorithm, and the *Gradient Projection* algorithm. Best Response outperforms Gradient Projection; we therefore only consider the former in the sequel. For every time slot $t = 1, 2, \dots$, and for each link $\ell = 1, \dots, L$, the Best Response algorithm computes the differential backlog $D_\ell(t)$ and the *interference price* $\pi_\ell(t) = \frac{D_\ell(t)}{I_\ell(\mathbf{p}(t)) + V_\ell}$, where $I_\ell(\mathbf{p}(t) := \frac{1}{G}\sum_{\substack{k=1\\k\neq\ell}}^{L} G_{k\ell}p_k(t)$ is the total interference to link ℓ from other links. The price is subsequently communicated to all links, and link ℓ transmits with power

$$p_\ell(t) = \min\left(\frac{D_\ell(t-1)}{\sum_{\substack{k=1\\k\neq\ell}}^{L} \pi_k(t-1)\frac{1}{G}G_{\ell k}}, P_\ell^{\max}\right).$$

VIII. Simulation Results

Simulation experiments have been performed in three different scenarios, in order to assess the performance of the

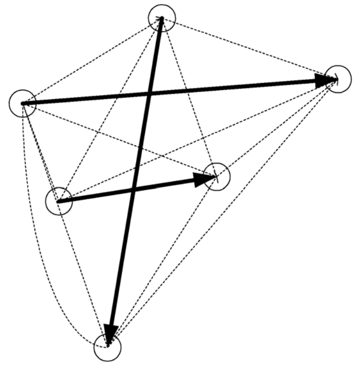

Fig. 1. Illustration of network construction in proof of Claim 1. Thick lines indicate links with nonzero differential backlog.

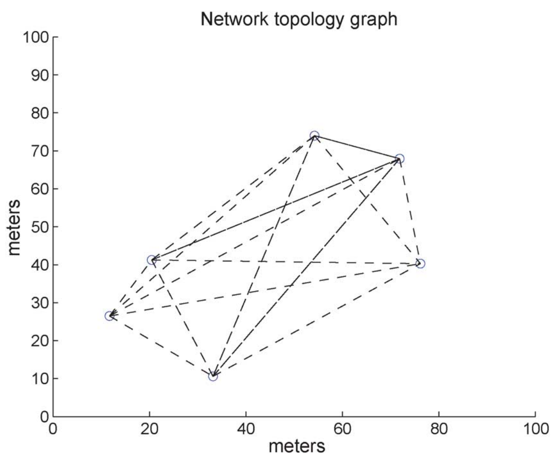

Fig. 2. Network topology.

various algorithms. In addition to the batch high-SINR / S.A. and their adaptive counterparts developed herein, we also included ISB and the Best Response algorithm as baselines for comparison.

Scenario 1—Small Network With Moderate Interference: The first scenario considered is a network with $N = 6$ nodes, randomly drawn from a uniform distribution over a 100×100 square. The resulting topology is shown in Fig. 2. The network remains fixed throughout the simulations for scenario 1. The leftmost node is the source, the rightmost is the destination, and the intermediates are relays. Since the destination acts as a sink, and no node transmits to itself, there are $L = 25$ links in the network (note that intermediate nodes are in principle allowed to send packets back to the source). Direct and crosstalk link power losses are taken inversely proportional to d^4, where d is the distance between transmitter and receiver. It is assumed that nodes cannot listen while they talk: if link ℓ terminates at node i and link k departs from i, then $G_{k,\ell} = 1/\mathrm{eps}$, where eps is machine precision. A spreading gain $G = 128$ is assumed and absorbed in the crosstalk factors $G_{k\ell},\ k \neq \ell$. The background noise power is set to $V_\ell = 10^{-12}, \forall \ell$. Per-link power constraints are adopted to be consistent with ISB and Best Response, used here as baselines; $P_\ell^{\max} = 5,\ \forall \ell$. We assume deterministic arrivals at the source. Simulating the network under control of each algorithm, we consider various arrival rates in order to narrow down the maximal arrival rate that each algorithm can support. Unless otherwise noted, the network is simulated for 100 packet / control slots. For the batch algorithms, each problem instance is solved using the CVX toolbox [7] for Matlab, on a per-slot basis. The desired accuracy in step 6 of the batch S.A. algorithm was set to $\epsilon = 0.1$.

Simulations verified the expected push-pull "wave" propagation over the network, and that periodic behavior emerges for stable setups. For arrival rates up to 9 packets per slot (pps), both batch algorithms stabilize the system, and the average throughput equals the incoming traffic at the source. Gradually increasing the arrival rate, we found that 9.7 pps was the maximal value for which the batch high-SINR algorithm managed to stabilize the system. Beyond that, all backlogs grow to infinity. Plots of the evolution of source and relay backlogs, power allocation, and end-to-end throughput for the batch high-SINR algorithm are shown in Fig. 3 (left) for arrival rate 9.7 pps, and Fig. 3 (right) for arrival rate 9.9 pps. Note that for the stable setup of Fig. 3 (left), and after a short transient, all relevant quantities (backlogs, powers, end-to-end throughput) converge to a periodic pattern. In this case, the emerging period is two slots. Simulations also verified that the batch S.A. algorithm is able to stabilize the system at higher input loads than the batch high-SINR algorithm. The maximum arrival rate that batch S.A. can support in this case is 10.4 pps. The queues become unstable at higher arrival rates. This can be easily visualized at 10.8 pps. Plots for the batch S.A. are shown in Fig. 4 (left) for arrival rate 10.4 pps, and Fig. 4 (right) for arrival rate 10.8 pps. For the stable setup in Fig. 4 (left), note again that all relevant quantities quickly settle in to a periodic pattern.

A first check point for the custom adaptive algorithms is to verify that they indeed reproduce the results of their batch counterparts, ideally at far lower complexity. We used a window of $W = 5$ slots for power initialization, and τ in (24) was set to 10^6. For the line search in step 3e of the adaptive high-SINR (5e of the adaptive S.A., respectively), a grid search of accuracy 10^{-3} was used. The desired accuracy in step 9 of the adaptive S.A. was set to $\epsilon = 0.1$. Simulation results concerning the adaptive high-SINR algorithm are shown in Fig. 5, for arrival rates 9.7 and 9.9 pps, passing sanity check (compare to Fig. 3). The adaptive S.A. algorithm likewise keeps up with its batch counterpart, as shown in Fig. 6 (compare to Fig. 4). Notice that the adaptive version of each algorithm yields qualitatively similar results (and in particular attains the same stable throughput) as its batch counterpart, however the respective power allocations do not coincide. This is because the underlying problem does not have a unique solution in general. To see this, consider an interference-limited scenario comprising two intermediate nodes transmitting to the common destination. Assuming symmetric loads and channels, activating either one of the two will yield the same reward.

As an indication of the complexity of the various algorithms, we note that the batch high-SINR one required on average about 8 seconds per slot (problem instance); batch S.A. averaged 3–4 outer iterations for a total of 20 seconds per slot. The adaptive high-SINR algorithm averaged about four inner iterations (Newton steps) for a total of 1.5 seconds per slot. Adaptive S.A. averaged 1–2 outer iterations of up to four inner Newton steps,

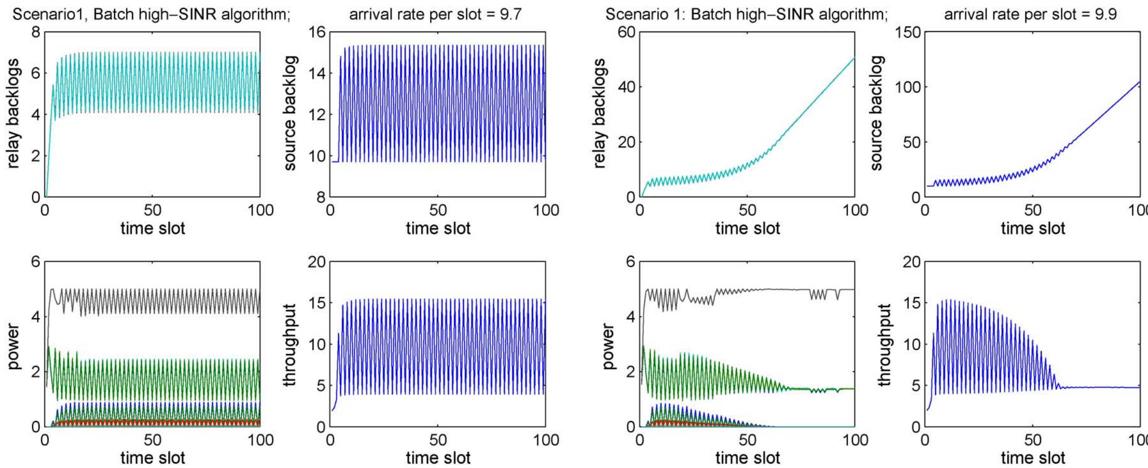

Fig. 3. Scenario 1: Batch high-SINR, arrival rate $= 9.7$ (left) and 9.9 (right).

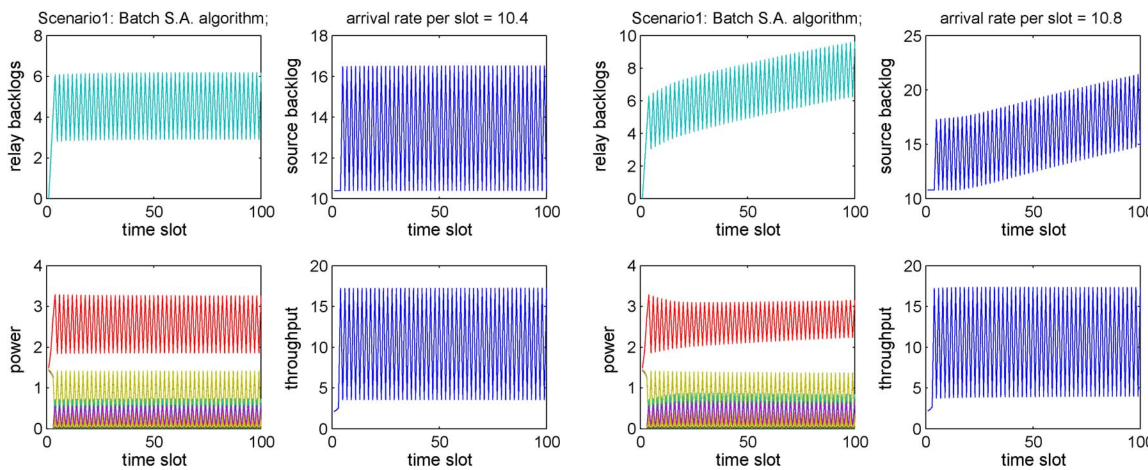

Fig. 4. Scenario 1: Batch Successive Approximation, arrival rate $= 10.4$ (left) and 10.8 (right).

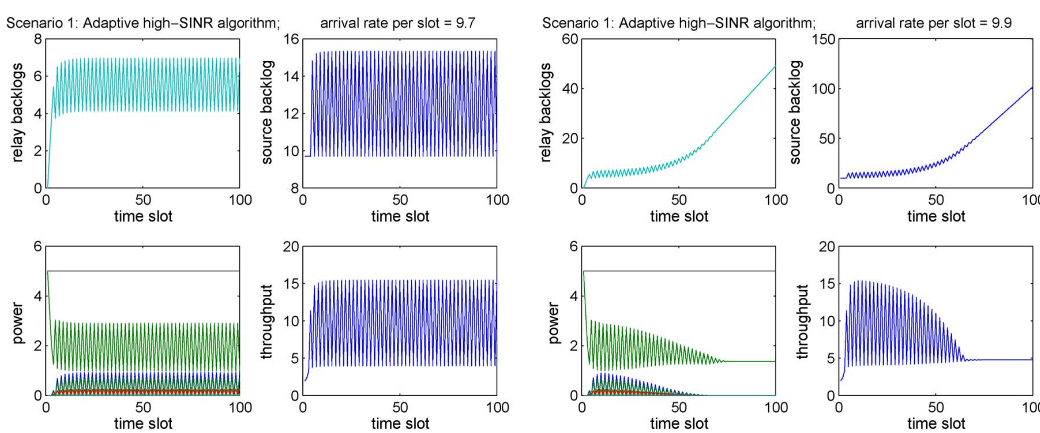

Fig. 5. Scenario 1: Adaptive high-SINR, arrival rate $= 9.7$ (left) and 9.9 (right).

for a total of about 1.5 seconds per slot, when initialized at the best (for the present slot) of the W past solutions for the previous slots. The difference is more significant (order of magnitude) in the case of the successive approximation algorithms, and the gap widens quickly with the size of the network. We also note that for time-invariant (or slowly-varying) channels, the rate of power updates (i.e., how often one solves the BPPC problem) only affects delay, not the attainable stable throughput of the algorithm, as shown in [23], [24]. Thus one can take more time between power updates at the expense of increased packet delay, without sacrificing throughput.

Next, we present results for the Best Response algorithm proposed in [6]. We also tried the gradient projection algorithm in [6], but it proved consistently inferior[4] to Best Response, so we skip the associated plots. Best Response was initialized with the

[4]Failed to stabilize the network in cases where Best Response did.

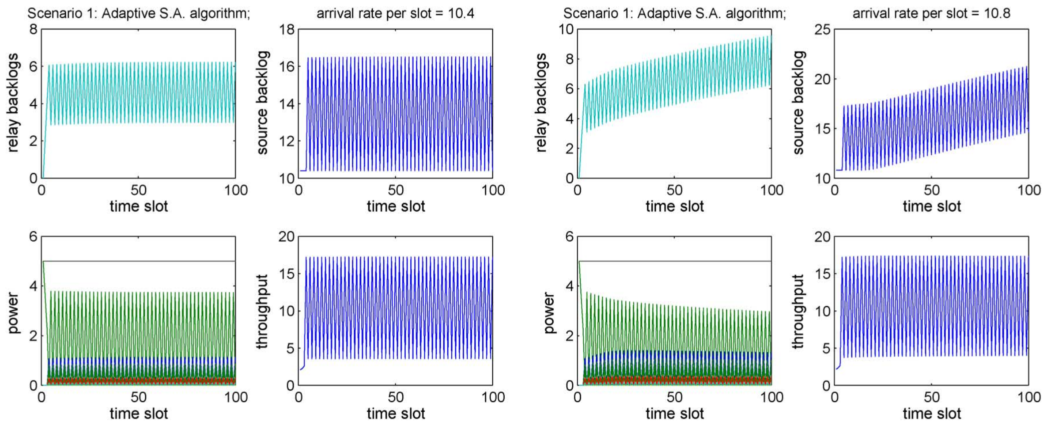

Fig. 6. Scenario 1: Adaptive Successive Approximation, arrival rate $= 10.4$ (left) and 10.8 (right).

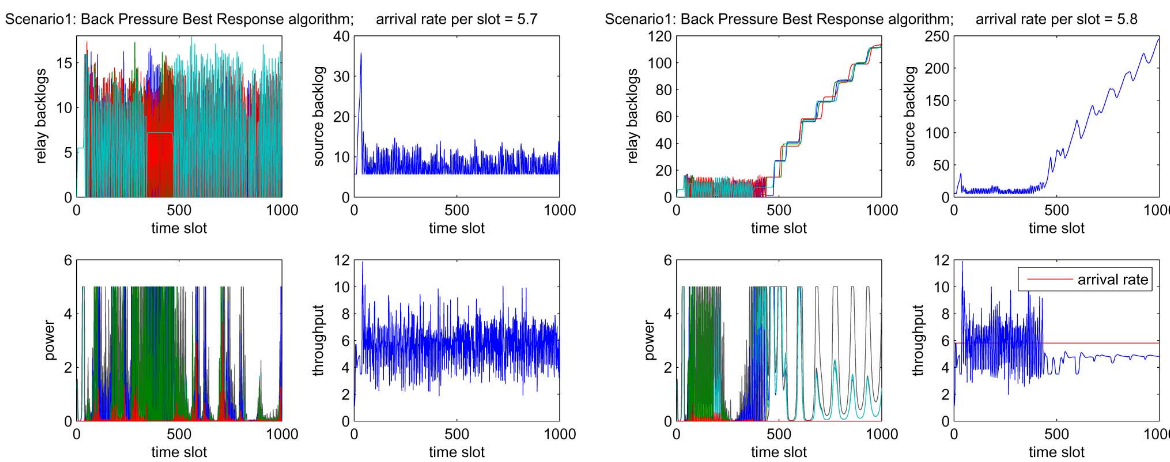

Fig. 7. Scenario 1: Back Pressure Best Response algorithm, arrival rate $= 5.7$ (left) and 5.8 (right).

solution of the Batch high-SINR algorithm, to give it the best possible warm start under the high-SINR assumption. However, it still takes far longer to reach steady-state (when it can stabilize the system) than previously discussed algorithms, hence we simulate it for 1000 slots in the sequel. Experiments showed that the highest arrival rate that Best Response can handle is 5.7 pps. Plots of the evolution of source and relay backlogs, power allocation, and end-to-end throughput, for the arrival rate 5.7 and 5.8 pps, are shown in Fig. 7. Obviously, this algorithm leaves much to be desired in terms of maximum stable throughput, delay (cf. queue backlogs and Little's theorem), and settling time relative to batch and adaptive high-SINR and S.A. On the other hand, Best Response is very cheap in terms of computation. As an indication, its average run-time here was 0.0085 seconds per slot, which is two orders of magnitude less than the best-performing algorithms.

Recall that computing the dual function is NP-hard, and ISB is only able to provide an *approximate* upper bound to the maximum attainable objective of the primal problem. Due to approximation, the power allocation computed by ISB need not (and in fact does not, in many cases) yield a higher primal objective than batch/adaptive S.A. Still, it is reassuring to see that the approximate upper bound derived from ISB is indeed higher and not very far from the value of the objective computed from batch/adaptive S.A. Illustrating this gap is the objective of the next two plots. We consider two arrival rates, one well-within

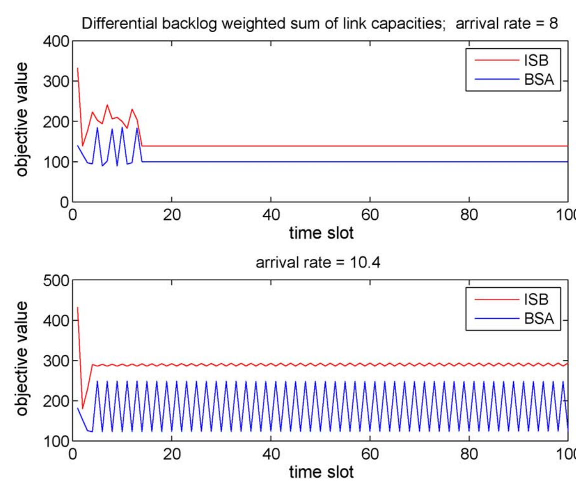

Fig. 8. Scenario 1: ISB versus Batch S.A.: Comparison between objective values attained.

the stable region and another on the margin: 8, and 10.4 pps. The network evolves under control of the batch S.A. algorithm, and ISB is used to (approximately) upper bound the attainable differential backlog weighted sum of link rates for each slot. The link power optimization in step 5 of the ISB algorithm is performed using a grid search over $[0\ P_\ell]$ of accuracy 10^{-4}. For each slot t, all elements of $\boldsymbol{\lambda}$ are initially set to 1, and the update of $\boldsymbol{\lambda}$ in step 7 is performed using the subgradient method with

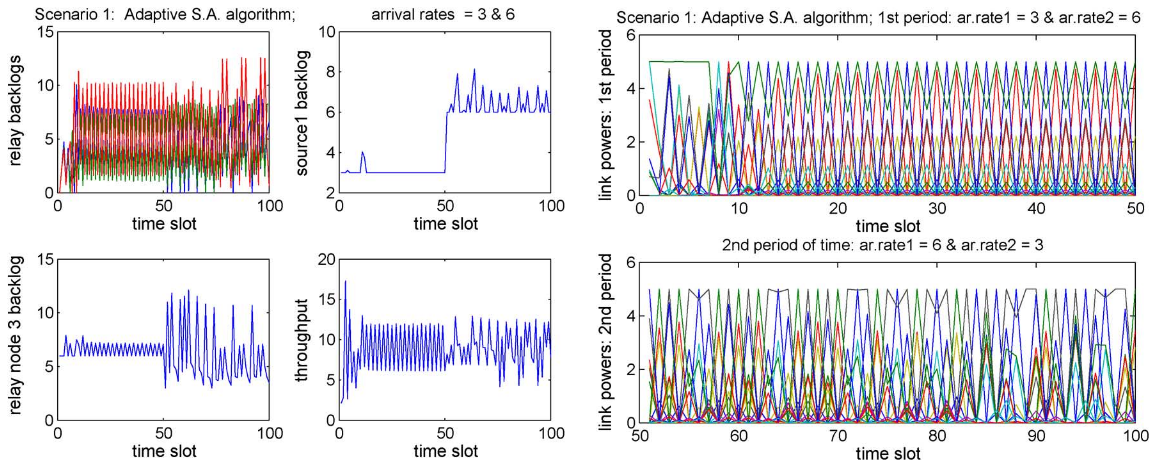

Fig. 9. Scenario 1: Switching injection pattern: Adaptive S.A., backlogs / throughput(left), link powers (right).

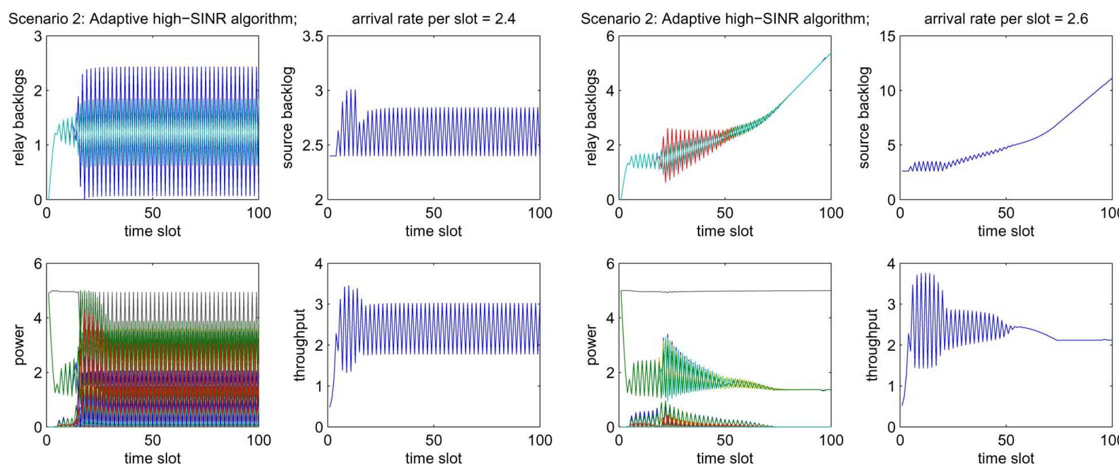

Fig. 10. Scenario 2: Adaptive high-SINR, arrival rate $= 2.4$ (left) and 2.6 (right).

$\beta = 0.1$. The desired accuracy for the convergence of $\mathbf{p}$ and $\boldsymbol{\lambda}$ was set to $\epsilon = 10^{-2}$. The final objective values attained by ISB and the batch S.A. algorithm are plotted together as functions of t in Fig. 8, in separate panels for the two arrival rates considered. Notice that ISB consistently hovers above batch S.A. in both cases considered, which is satisfying.

We next consider a tracking experiment to illustrate the ability of the adaptive S.A. algorithm to follow changes in the operational environment. Packets are now injected not only at the original source node (node 1), but also at an intermediate relay (node 3). There is still one destination, and all packets are treated the same way—there is only one buffer per node. For the first 50 slots, traffic is injected at 3 pps through node 1, and at 6 pps through node 3. For the next 50 slots the injection pattern reverses: 6 pps through node 1, and 3 pps through node 3. Fig. 9 shows simulation results for the adaptive S.A. algorithm, which evidently responds swiftly to the change in the traffic pattern. The Best Response algorithm cannot stabilize the network for this pair of rates; but even at lower, sustainable rates, its response to the change of the injection pattern was very slow and hard to discern. We skip associated plots for brevity.

Scenario 2—Small Network, More Interference: We next consider a scenario that is more interference-limited (less power-limited) than before: using $G = 8$ instead of 128. ISB takes disproportional time to converge in this scenario, thus we drop it from consideration. Due to stronger interference, the high-SINR algorithms can now stabilize the system for arrival rates only up to 2.4 pps. Fig. 10 plots results for adaptive high-SINR for arrival rate 2.4 pps (left) and 2.6 pps (right). The S.A. algorithms proved much better, supporting *three times higher* stable throughput (7.5 pps). It turned out that, in order to do so, they both converged to the same solution: keeping only the direct link from source to destination, always on and transmitting at maximum power. Simulation results are shown in Fig. 11 for arrival rate 7.5 pps (left), and 7.9 pps (right). The complexity advantage of the adaptive algorithms relative to the batch ones remains similar to scenario 1. Best Response managed to stabilize the system for rates up to 2.1 units per slot. The respective plots are shown in Fig. 12 for the 2.1 pps (left) and 2.3 pps (right).

Scenario 3—Larger Network, Moderate Interference: A larger network comprising $N = 12$ nodes ($L = 121$ links) is considered in our third scenario. The setup is otherwise identical to scenario 1. Simulation showed that the high-SINR algorithms managed to stabilize the network for arrival rates up to 12.6 pps. The adaptive S.A., on the other hand, managed to stabilize the system for up to 15.7 pps. Unlike adaptive S.A., the batch S.A. algorithm was too slow to include in this

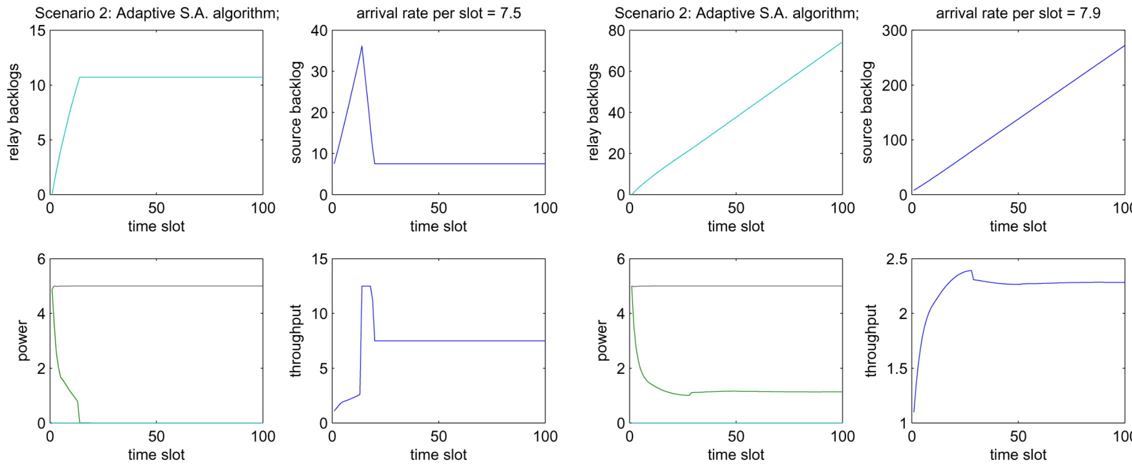

Fig. 11. Scenario 2: Adaptive Successive Approximation, arrival rate $= 7.5$ (left) and 7.9 (right).

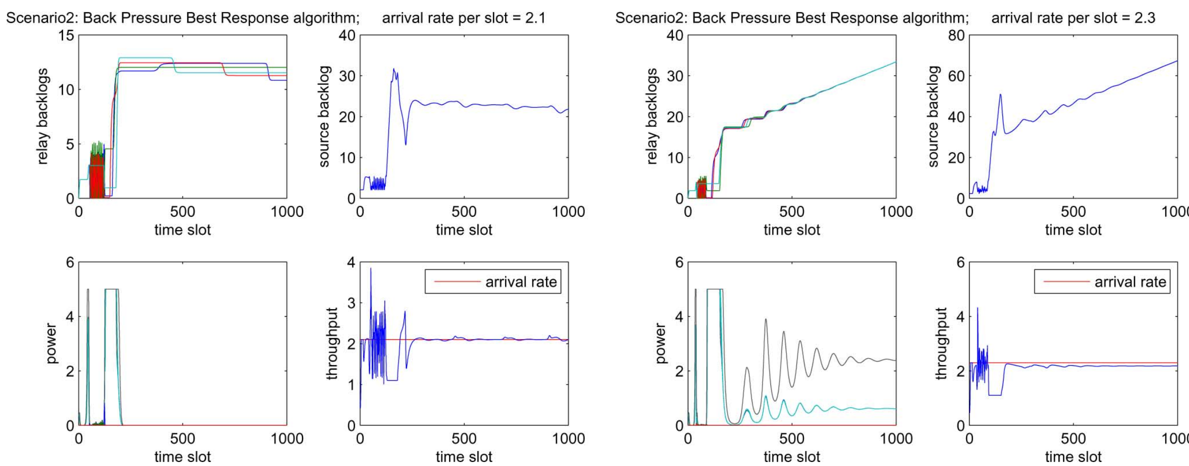

Fig. 12. Scenario 2: Back Pressure Best Response algorithm, arrival rate $=$ 2.1 (left) and 2.3 (right).

TABLE I
ATTAINABLE STABLE ARRIVAL RATES IN PACKETS PER SLOT

Scenario	Batch high-SINR	Adaptive high-SINR	Batch S.A.	Adaptive S.A.	Best Response
Scenario 1	9.7	9.7	10.4	10.4	5.7
Scenario 2	2.4	2.4	7.5	7.5	2.1
Scenario 3	12.6	12.6	N/A	15.7	4.4

comparison. The maximum rate that Best Response could support was 4.4 pps. We skip associated plots for brevity, and instead gather all results concerning attainable rates in Table I.

IX. Conclusion

We have considered the power control problem in wireless multihop networks. From the viewpoint of maximizing stable end-to-end throughput, the objective is to maximize a differential backlog-weighted sum of link rates, subject to power constraints. Following physical layer optimization at the beginning of each transmission round, back-pressure routing is used for packet forwarding over the network. This two-step approach is optimal from a throughput perspective, and, for this reason, the back-pressure power control (BPPC) problem is central in wireless multihop networks.

BPPC was known to be nonconvex [6]; we established that it is in fact NP-hard. This precludes optimal solution at worst-case polynomial complexity, and motivates the pursuit of appropriate approximation algorithms. Drawing from related problems in the DSL literature (in particular the SCALE algorithm of [18], [19]), we proposed two ways to approximate the BPPC problem which far outperform the previous state of art in [6]. Most importantly, recognizing the high computational burden arising from the need to solve BPPC on a per-slot (transmission round) basis, and capitalizing on quasi-periodicity of the power allocation in stable setups due to the push-pull nature of the solution, we proposed two custom adaptive approximation algorithms that offer excellent throughput performance at reasonable computational complexity, which remains worst-case polynomial. In addition to throughput margin, the proposed algorithms feature shorter backlogs / queueing delays, and faster transient response.

An interesting research direction is to consider means of implementing the proposed algorithms—adaptive S.A. in particular—in a distributed fashion. Recent progress in distributing the Newton step using Gaussian belief propagation [28], and related work in distributed network utility maximization [8] may be useful towards this end.

Another interesting direction is to consider time-varying channels. If the channel variation is far slower than the rate of power re-optimization, and the channels can be tracked at the central scheduler, then our methods remain operational. If the channels change (perhaps abruptly) only at certain points in time, but otherwise remain fixed between such changes and known to the central scheduler, and the BPPC problem is *exactly* solved before each such 'dwell', then throughput optimality still holds [23]. This scenario is plausible when several different networks are time-division multiplexed (similar to multiplexing ALOHA protocols, for example). We have shown that exact solution of BPPC is NP-hard; but we have also shown that our approximate solutions deliver substantial throughput improvement relative to the prior art in [6]. Still, much work is needed to figure out good policies for general time-varying scenarios, including the interplay with distributed implementation.

APPENDIX

Extension to Multiple Flows: In the case of multiple flows, each node is assumed to maintain a separate queue for each flow (this is easy to implement and it does not require additional storage). Let there be F flows (i.e., F destinations) in the network. For every scheduling time slot $t \in \{1, 2, \cdots\}$, the following algorithm is executed.

Let $W_{i,f}(t)$ denote the queue length of node i for flow f at the end of slot t. The differential backlog of link $\ell = (i, j)$ for flow f at the end of slot t is defined as

$$D_{\ell,f}(t) := \begin{cases} \max\{0, W_{i,f}(t) - W_{j,f}(t)\}, & j \neq \mathrm{dest}(f) \\ W_{i,f}(t), & j = \mathrm{dest}(f) \end{cases}$$

where $\mathrm{dest}(f)$ is the destination node for flow f. Let

$$\bar{D}_\ell(t) := \max_{f \in \{1,\cdots,F\}} D_{\ell,f}(t)$$

be the maximum differential backlog for link ℓ at the end of slot t, and

$$\bar{f}_\ell(t) := \arg \max_{f \in \{1,\cdots,F\}} D_{\ell,f}(t)$$

be a flow that attains maximum differential backlog for link ℓ at the end of slot t. Then link ℓ is *dedicated* to flow $\bar{f}_\ell(t)$ for the next scheduling period (if $\bar{D}_\ell(t) = 0$, then link ℓ remains idle), and transmission powers are set according to

$$\max_{\{p_\ell\}_{\ell \in \mathcal{L}}} \sum_{\ell=1}^{L} \bar{D}_\ell(t) c_\ell$$
$$\text{subject to} \quad 0 \leq \sum_{\ell:\mathrm{Tx}(\ell)=i} p_\ell \leq P_i, \quad i \in \mathcal{N}.$$

That is, the same type of BPPC problem is solved as in the single-flow case, but this time using $\bar{D}_\ell(t)$'s in place of $D_\ell(t)$'s. The other difference with the case of a single flow is that now a decision has to be made as to which flow(s) will be routed on any given link and how to multiplex them—but it turns out that simple winner-takes-all routing combined with BPPC is in fact optimal from a throughput perspective, as shown in [24].

Proof: (Claim 1) We will show that the problem contains that of determining the size of the maximum independent set in a graph, which is NP-hard. The proof draws heavily from the proof of Theorem 1 in [13], which deals with the multiuser sum-rate maximization problem for the interference channel. Back-pressure power control looks like the same problem, but there is a catch: the $G_{k,\ell}$ parameters are completely free in the multiuser sum-rate maximization problem in [13], but in back-pressure power control the $G_{k,\ell}$'s are subject to certain restrictions. Consider the case of two links k, ℓ stemming from the same node. Clearly, $G_{k,\ell} = G_{\ell,\ell}$, and $G_{\ell,k} = G_{k,k}$. Likewise, consider two links k, ℓ with common receiving node. Then $G_{k,k} = G_{k,\ell}$, and $G_{\ell,\ell} = G_{\ell,k}$. This seems to suggest that back-pressure power control is a *restriction* of the multiuser sum-rate maximization problem, and restriction of an NP-hard problem is not necessarily NP-hard. Another important difference is that back-pressure power control may be subject to per-node (instead of per-link) power constraints, which couple the transmission power across multiple links.

The key to a simple and clear proof is *backlog reduction*: realizing that there is freedom to choose the backlogs in such a way that we peel off these complicating factors to reveal the multiuser sum-rate maximization problem as a special case.

The *conflict graph* is a familiar concept in network scheduling. Each *directed link* in the network corresponds to a node in the conflict graph. Nodes k, ℓ in the conflict graph are connected by an *undirected edge* if $G_{k,\ell} \neq 0$, or $G_{\ell,k} \neq 0$, or both—i.e., when links k and ℓ can interfere with each other. An *independent set* of nodes (not connected by an edge) in the conflict graph corresponds to a set of network links that can be simultaneously activated without causing interference to one another. Given an undirected connected graph $\Gamma = (V, E)$ with $|V| = L$ nodes, construct an instance of a wireless multihop network whose conflict graph is Γ as follows:

- Choose $2L$ network nodes, split them in L pairs, draw a directed link between each pair, and set the differential backlogs of these links to 1. Set the differential backlogs of all remaining links to 0. The L drawn links are the only ones that can be activated in the next slot. This is important because it effectively reduces the network to a set of cochannel links that do not share transmitters or receivers; see Fig. 1. Since no transmitter is shared, there is no distinction between per-link or per-node power constraints in so far as this proof is concerned.
- For the links with nonzero differential backlog, set:
  - — $G_{\ell,\ell} = 1$, $P_\ell = 1$, $G = 1$ (no spreading), and $V_\ell = M$, where $M > L$ is a constant;
  - — For every edge in E connecting nodes k and ℓ, set $G_{k,\ell} = G_{\ell,k} = ML^2$ (notice that we enforce symmetry, which will be used later in the proof).

Notice that the choice of the coupling factors in the above has implications for other links in the network: For example, strong crosstalk between two links implies that there is a strong direct channel gain between the transmitter of one link and the receiver of the other. If the two transmitters could switch receivers they would set up more favorable links. This possibility is excluded, however, by our selection of differential backlogs: all except the L chosen links have zero differential backlog.

Let $\mathbf{p}^* \in [0,1]^{L\times 1}$ be an optimum solution of the back-pressure power control problem for this network instance, and let r^* be the corresponding optimal value of the (sum-rate) objective. Let I be a maximum independent set of Γ. By activating only the network links corresponding to conflict graph nodes in I, we obtain sum-rate $|I|\log\left(1+\frac{1}{M}\right)$, hence $r^* \geq |I|\log\left(1+\frac{1}{M}\right)$. Conversely, consider $\mathbf{p}^*$ and split it in two parts: those elements that are positive, and the rest that are zero. It has been shown in [13] that, under our working assumptions (in particular, $M > L$) the sum-rate objective is convex-U with respect to each component of $\mathbf{p}$, albeit not jointly convex in $\mathbf{p}$ as a whole. Since the maximum of a convex function over a polytope can always be attained at a vertex, and the constraint is $\mathbf{p}^* \in [0,1]^{L\times 1}$, it follows that we may assume, without loss of generality, that $\mathbf{p}^* \in \{0,1\}^{L\times 1}$. This is important, because it implies that if two or more interfering links are simultaneously active, the interference level will be lower bounded by ML^2. Let $A^* := \{\ell | \mathbf{p}^*(\ell) = 1\}$, and let J be a maximum independent subset of A^* in Γ. Clearly, $|J| \leq |I|$. We have

$$r^* \leq |J|\log\left(1+\frac{1}{M}\right)+(|A^*|-|J|)\log\left(1+\frac{1}{M+ML^2}\right)$$

since $\log\left(1+\frac{1}{M}\right)$ is an upper bound on rate for each link in J, in the best-case scenario that no link in $A^* - J$ interferes with it; and links outside J must have at least one interferer in J, for otherwise J is not a maximum independent subset of A^* in Γ. We can further bound r^* as follows:

$$r^* \leq |J|\log\left(1+\frac{1}{M}\right)+L\log\left(1+\frac{1}{M+ML^2}\right).$$

The term $L\log\left(1+\frac{1}{M+ML^2}\right)$ is decreasing in L, and equal to $\log\left(1+\frac{1}{2M}\right) < \log\left(1+\frac{1}{M}\right)$ for $L=1$; it follows that $r^* < (|J|+1)\log\left(1+\frac{1}{M}\right)$, and since $|J| \leq |I|$, we have $r^* < (|I|+1)\log\left(1+\frac{1}{M}\right)$. Putting everything together

$$|I|\log\left(1+\frac{1}{M}\right) \leq r^* < (|I|+1)\log\left(1+\frac{1}{M}\right).$$

and solving for $|I|$, we obtain

$$|I| \in \left(\frac{r^*}{\log\left(1+\frac{1}{M}\right)}-1, \frac{r^*}{\log\left(1+\frac{1}{M}\right)}\right].$$

This determines the exact (integer) value of I. It follows that if we could efficiently solve the back-pressure power control problem in polynomial time, we would be in position to determine the size of the maximum independent set in an arbitrary graph in polynomial time. ■

Gradient and Hessian Computation: The first and second order derivatives of

$$\tilde{f} = -\tau\left(\sum_{\ell=1}^{L} D_\ell(t)\left(a_\ell(t)\left(\tilde{G}_{\ell\ell}+\tilde{p}_\ell - \log\left(\sum_{\substack{\ell=1\\ \ell\neq n}}^{L} e^{\tilde{G}_{k\ell}+\tilde{p}_k}+e^{\tilde{V}_\ell}\right)\right)+\beta_\ell(t)\right)\right) - \sum_{\ell=1}^{L}\log(\tilde{P}_\ell - \tilde{p}_\ell)$$

which are needed to compute the gradient and Hessian for the adaptive algorithms, are given by

$$\frac{\partial \tilde{f}}{\partial \tilde{p}_n} = -\tau D_n(t)a_n(t) + \frac{1}{\tilde{P}_n - \tilde{p}_n} + \tau\sum_{\substack{\ell=1\\ \ell\neq n}}^{L}\frac{D_\ell(t)a_\ell(t)e^{\tilde{G}_{n\ell}+\tilde{p}_n}}{I_\ell}, \quad \forall n \in \mathcal{L}.$$

where

$$I_\ell = \sum_{\substack{k=1\\ k\neq \ell}}^{L} e^{\tilde{G}_{k\ell}+\tilde{p}_k}+e^{\tilde{V}_\ell}$$

is the total interference to link ℓ from all other links. The second-order partial derivatives are

$$\frac{\partial^2 \tilde{f}}{\partial \tilde{p}_n^2} = \frac{1}{(\tilde{P}_n - \tilde{p}_n)^2} + \tau\sum_{\substack{\ell=1\\ \ell\neq n}}^{L}\frac{D_\ell(t)a_\ell(t)e^{\tilde{G}_{n\ell}+\tilde{p}_n}(I_\ell - e^{\tilde{G}_{n\ell}+\tilde{p}_n})}{I_\ell^2}$$

and $$\frac{\partial^2 \tilde{f}}{\partial \tilde{p}_n \partial \tilde{p}_m} = -\tau\sum_{\substack{\ell=1\\ \ell\neq m,n}}^{L}\frac{D_\ell(t)a_\ell(t)e^{\tilde{G}_{n\ell}+\tilde{p}_n}e^{\tilde{G}_{m\ell}+\tilde{p}_m}}{I_\ell^2}. \quad \forall m,n \in \mathcal{L}$$

References

[1] S. Boyd and L. Vandenberghe*, Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[2] R. Bro, N. D. Sidiropoulos, and A. K. Smilde, "Maximum likelihood fitting using ordinary least squares algorithms," *J. Chemometrics*, vol. 16, pp. 387–400, Sep. 2002.

[3] R. Cendrillon, W. Yu, M. Moonen, J. Verliden, and T. Bostoen, "Optimal multi-user spectrum management for digital subscriber lines," *IEEE Trans. Commun.*, vol. 54, no. 5, pp. 922–933, May 2006.

[4] P. Chaporkar, K. Kar, and S. Sarkar, "Throughput guarantees in maximal scheduling in wireless networks," in *Proc. 43rd Annu. Allerton Conf. Commun., Contr., Comput.*, Allerton,, IL, Sep. 2005, pp. 1557–1567.

[5] L. Georgiadis, M. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Found. Trends Netw.*, vol. 1, pp. 1–147, 2006.

[6] A. Giannoulis, K. Tsoukatos, and L. Tassiulas, "Maximum throughput power control in CDMA wireless networks," in *Proc. IEEE Int. Conf. Commun.*, Instabul, Turkey, Jun. 2006, pp. 4457–4462.

[7] M. Grant and S. Boyd, CVX: Matlab Software for Disciplined Convex Programming web page and software: [Online]. Available: http://stanford.edu/ boyd/cvx

[8] A. Jadbabaie, A. Ozdaglar, and M. Zargham, "A distributed newton method for network optimization," in *Proc. IEEE Conf. Decision Contr.*, Shanghai, China, Dec. 2009, pp. 2736–2741.

[9] C. Joo, X. Lin, and N. B. Shroff, "Performance limits of greedy maximal matching in multi-hop wireless networks," in *Proc. IEEE CDC*, New Orleans, LA, Dec. 2007, pp. 1128–1133.

[10] E. Karipidis, "Transmit Beamforming to Multiple Cochannel Multicast Groups," Ph.D. dissertation, Dept. ECE, Technical Univ. Crete, Crete, Greece, Aug. 2008, [chapter 4].

[11] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 302–315, Apr. 2006.

[12] Z.-Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1426–1438, Aug. 2006.

[13] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 57–73, Feb. 2008.

[14] Z.-Q. Luo, private communication, Univ. Minnesota, Minneapolis, MN, Jul. 2009.

[15] E. Matskani, N. D. Sidiropoulos, and L. Tassiulas, "Convex approximation algorithms for back-pressure power control of wireless multi-hop networks," in *Proc. IEEE ICASSP*, Prague, Czech Republic, May 22–27, 2011, pp. 3032–3035.

[16] M. J. Neely, "Energy optimal control for time varying wireless networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 7, pp. 2915–2934, Jul. 2006.

[17] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 13–17, 2005, pp. 1723–1734.

[18] J. Papandriopoulos and J. S. Evans, "Low-complexity distributed algorithms for spectrum balancing in multi-user DSL networks," in *Proc. IEEE Int. Conf. Commun.*, Instabul, Turkey, Jun. 2006, pp. 3270–3275.

[19] J. Papandriopoulos and J. S. Evans, "SCALE: A low-complexity distributed protocol for spectrum balancing in multiuser DSL networks," *IEEE Trans. Inf. Theory*, vol. 55, no. 8, pp. 3711–3724, Aug. 2009.

[20] L. P. Qian, Y. J. Zhang, and J. Huang, "Mapel: Achieving global optimality for a non-convex wireless power control problem," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1553–1563, Mar. 2009.

[21] S. Sarkar and L. Tassiulas, "A framework for routing and congestion control for multicast information flows," *IEEE Trans. Inf. Theory*, vol. 48, no. 10, pp. 2690–2708, Oct. 2002.

[22] A. L. Stolyar, "Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm," *Queueing Syst.*, vol. 50, pp. 401–457, 2005.

[23] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 1067–1073, Mar. 1997.

[24] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 12, pp. 1936–1948, Dec. 1992.

[25] L. Tassiulas, "Adaptive back-pressure congestion control based on local information," *IEEE Trans. Autom. Control*, vol. 40, no. 2, pp. 236–250, Feb. 1995.

[26] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," in *Proc. IEEE INFOCOM*, San Francisco, CA, 1998, pp. 533–539.

[27] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Trans. Commun.*, vol. 54, no. 7, pp. 1310–1322, Jul. 2006.

[28] D. Dolev, A. Zymnis, S. P. Boyd, D. Bickson, and Y. Tock, "Distributed large scale network utility maximization," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT) 2009*, Seoul, Korea, Jun. 28–Jul. 3 2009, pp. 829–833.

Evaggelia Matskani received the Dipl. degree in electrical and computer engineering from the Aristotle University, Thessaloniki, Greece, in 2005, and the M.Sc. degree in electronic and computer engineering from the Technical University of Crete, Crete, Greece, in 2007, where she is currently working towards the Ph.D degree.

Her research interests include signal processing for communications, with emphasis on convex optimization and cross-layer design of wireless networks.

Nicholas D. Sidiropoulos (S'90–M'92–SM'99–F'09) received the Dipl. degree in electrical engineering from the Aristotelian University, Thessaloniki, Greece, in 1988, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, in 1990 and 1992, respectively.

He served as Assistant Professor at the University of Virginia from 1997 to 1999; Associate Professor at the University of Minnesota, Minneapolis, MN, from 2000 to 2002; Professor at the Technical University of Crete, Greece from 2002 to 2011; and Professor in the Department of Electrical and Computer Engineering at the University of Minnesota—Minneapolis since 2011. His current research interests include signal processing for communications, cross-layer wireless networking, convex optimization/ approximation, and multiway analysis/multilinear algebra.

Dr. Sidiropoulos received the NSF/CAREER Award in 1998 and the IEEE Signal Processing Society (SPS) Best Paper Award in 2001 and 2007. He received the 2010 IEEE Signal Processing Society Meritorious Service Award. He served as IEEE SPS Distinguished Lecturer (2008–2009), and as Chair of the IEEE Signal Processing for Communications and Networking Technical Committee (2007–2008). He served as Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 2000 to 2006, IEEE SIGNAL PROCESSING LETTERS from 2000 to 2002, and currently serves on the editorial board of *IEEE Signal Processing Magazine*. He also served as TPC Chair for IEEE CAMSAP 2005 and IEEE SAM 2008, as General Cochair for IEEE CAMSAP 2007, and currently serves on the organizing committees for IEEE ICASSP 2011 in Prague and ICASSP 2015 in Brisbane.

Leandros Tassiulas (S'89-M'92–SM'05–F'07) was born in 1965 in Katerini, Greece. He received the Dipl. degree in electrical engineering from the Aristotelian University, Thessaloniki, Greece, in 1987, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, MD, in 1989 and 1991, respectively.

He is currently a Professor in the Department of Computer and Telecommunications Engineering, University of Thessaly, Greece, and a Research Professor in the Department of Electrical and Computer Engineering and the Institute for Systems Research, University of Maryland, College Park, since 2001. He has held positions as Assistant Professor at Polytechnic University New York (1991–1995), Assistant and Associate Professor at University of Maryland, College Park (1995–2001), and Professor at University of Ioannina, Greece (1999–2001). His research interests include computer and communication networks with emphasis on fundamental mathematical models, architectures and protocols of wireless systems, sensor networks, high-speed internet, and satellite communications.

Dr. Tassiulas received a National Science Foundation (NSF) Research Initiation Award in 1992, an NSF CAREER Award in 1995, an Office of Naval Research (ONR) Young Investigator Award in 1997, a Bodosaki Foundation Award in 1999, the IEEE INFOCOM '94 Best Paper Award, and the inaugural INFOCOM Achievement award (2007) for Fundamental contributions to resource allocation in communication networks.