Yang *et al. EURASIP Journal on Wireless Communications and Networking* 2012, **2012**:231
http://jwcn.eurasipjournals.com/content/2012/1/231

EURASIP Journal on Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

Distributed stochastic power control in ad hoc networks: a nonconvex optimization case

Lei Yang[1*], Yalin E Sagduyu[2], Junshan Zhang[1] and Jason H Li[2]

Abstract

Signal-to-interference-plus-noise-based power allocation in wireless ad hoc networks is inherently a nonconvex optimization problem because of the global coupling induced by the co-channel interference. To tackle this challenge, we first show that the globally optimal point lies on the boundary of the feasible region. This property is utilized to transform the utility maximization problem into an equivalent max–min problem with more structure. By using extended duality theory, penalty multipliers are introduced for penalizing the constraint violations, and the minimum weighted utility maximization problem is then decomposed into subproblems for individual users to devise a distributed stochastic power control algorithm, where each user stochastically adjusts its target utility to improve the total utility by simulated annealing (SA). The proposed distributed power control algorithm can guarantee global optimality at the cost of slower convergence due to SA involved in the global optimization. The geometric cooling scheme with suitable choice of penalty parameters is then used to improve the convergence rate. Next, by integrating the stochastic power control approach with the back-pressure algorithm, we develop a joint scheduling and power allocation policy to stabilize the queueing systems under random packet traffic. Finally, we generalize the above distributed power control algorithms to multicast communications, and show their global optimality for multicast traffic.

Keywords: Distributed power control, Nonconvex optimization, Extended duality theory, Simulated annealing, Queue stability, Unicast communications, Multicast communications

Introduction

The broadcast nature of wireless transmissions makes wireless networks susceptible to interference, which deteriorates quality of service (QoS) provisioning. Power control is considered as a promising technique to mitigate interference. One primary objective of power control is to maximize the system utility that can achieve a variety of fairness objectives among users [1-4]. However, maximizing the system utility, under the physical interference model, often involves nonconvex optimization and it is known to be NP-hard, due to the complicated coupling among users through mutual interference effects [5].

Due to the nonconvex nature of the power control problem, it is challenging to find the globally optimal power allocation in a distributed manner. Notably, the authors of [6-9] devised distributed power control algorithms to find power allocations that can only satisfy the local optimality conditions, but global optimality could not be guaranteed in general, except for some special convexifiable cases (e.g., with strictly increasing log-concave utility functions). Another thread of work applied game-theoretic approaches to power control by treating it as a noncooperative game among transmitters [10,11]. However, distributed solutions that converge to a Nash equilibrium may be suboptimal in terms of maximizing the total system utility. Different from these approaches, the authors of [12] transformed the power control problem into a DC (difference of convex functions) optimization problem [13]. Then, the global optimal solution can be solved in a centralized manner with the branch-and-bound algorithm. Recent study [14] proposed a globally optimal power control scheme, named MAPEL, by exploiting the monotonic nature of the underlying optimization problem. However, the complexity and the centralized nature

*Correspondence: lyang55@asu.edu
[1] School of ECEE, Arizona State University, Tempe, AZ 85287, USA
Full list of author information is available at the end of the article

of MAPEL hinder its applicability in practical scenarios, and thus it can be treated rather as a benchmark for performance evaluation in distributed networks.

To find the globally optimal power allocation in a distributed setting, recent study [15] has proposed the SEER algorithm based on Gibbs sampling [16], which can approach the globally optimal solution in an asymptotic sense when the control parameter in Gibbs sampling tends to infinity. Notably, for each iteration in the SEER algorithm, each user utilizes Gibbs sampling to compute its transition probability distribution for updating its transmission power, where the requirement for message passing and computing the transition probability distribution in each iteration can be demanding when applied to ad hoc communications without centralized control.

A challenging task in distributed power control in ad hoc networks is to reduce the amount of message passing while preserving the global optimality. To tackle this challenge, we first show that the globally optimal point lies on the boundary of the feasible region. This property is utilized to transform the utility maximization problem into an equivalent max–min problem with more structure, which can be solved by combining recent advances in extended duality theory (EDT) [17] with simulated annealing (SA) [18]. Compared with the classical duality theory with nonzero duality gap for nonconvex optimization problems, EDT can guarantee zero duality gap between the primal and dual problems by utilizing nonlinear Lagrangian functions. This property allows for solving the nonconvex problem by its *extended dual* while preserving the global optimality with distributed implementation. Furthermore, as will be shown in Section “Power control for unicast communications”, for the subproblem of each individual user, the extended dual can then be solved through stochastic search with SA. In particular, we first transform the original utility maximization problem into an equivalent max–min problem. This step is based on the key observation that in the case with continuous and strictly increasing utility functions, the globally optimal solution is always on the *boundary* of the feasible (utility) region. Then, appealing to EDT and SA, we develop a distributed stochastic power control (DSPC) algorithm that stochastically searches for the optimal power allocation in the neighborhood of the feasible region’s boundary, instead of bouncing around in the entire feasible region.

Specifically, we first show that DSPC can achieve the global optimality in the underlying nonconvex optimization problem, although the convergence rate can be slow (but this is clearly due to the slow convergence nature of SA with logarithmic cooling schedule). Then, to improve the convergence rate of DSPC, we propose an enhanced DSPC (EDSPC) algorithm that employs the geometric cooling schedule [19] and performs a careful selection of penalty parameters. As a benchmark for performance evaluation, we also develop a centralized algorithm to search for the globally optimal solution over simplices that cover the feasible region. The performance gain is further verified by comparing our distributed algorithms with MAPEL [14], SEER [15], and ADP [6] algorithms. Worth noting is that the proposed DSPC and EDSPC algorithms do not require any knowledge of channel gains, which is typically needed in existing algorithms, and instead they need only the standard feedback of signal-to-interference-plus-noise (SINR) for adaptation.

Next, we integrate the proposed distributed power control approach with the back-pressure algorithm [20] and devise a joint scheduling and power allocation policy for improving the queue stability in the presence of dynamic packet arrivals and departures. This policy fits into the dynamic back-pressure and resource allocation framework and enables distributed utility maximization under stochastic packet traffic [21,22]. Then, we generalize the study to consider multicast communications, where a single transmission may simultaneously deliver packets to multiple recipients [23,24]. Specifically, we extend DSPC and EDSPC algorithms to multicast communications with distributed implementation, and show that these algorithms can also achieve the global optimality in terms of jointly maximizing the minimum rates on bottleneck links in different multicast groups.

The rest of the article is organized as follows. In the following section, we first introduce the system model, establish the equivalence between the utility maximization problem and its max–min form, and then develop both centralized and distributed algorithms for the max–min problem. Next, building on these power control algorithms, we develop in Section “Joint scheduling and power control for stability of queueing systems” a joint scheduling and power allocation policy to stabilize queueing systems. The generalization to multicast communications is presented in Section “Power control for multicast communications”. We conclude the article in “Conclusion” Section.

Power control for unicast communications

System model

We consider an ad hoc wireless network with a set $\mathcal{L} = \{1, \ldots, L\}$ of links, where the channel is interference-limited, and all L links treat interference as noise, as illustrated in Figure 1. Such a model of communication is readily applicable to cellular networks [1]. Each link consists of a dedicated transmitter-receiver pair.[a] We denote by h_{lk} the fixed channel gain between user l’s transmitter and user k’s receiver, and by p_l the transmission power of link l with $P_l^{\max}$ being its maximum power constraint. It follows that the received SINR for the lth user with a matched filter receiver is given by

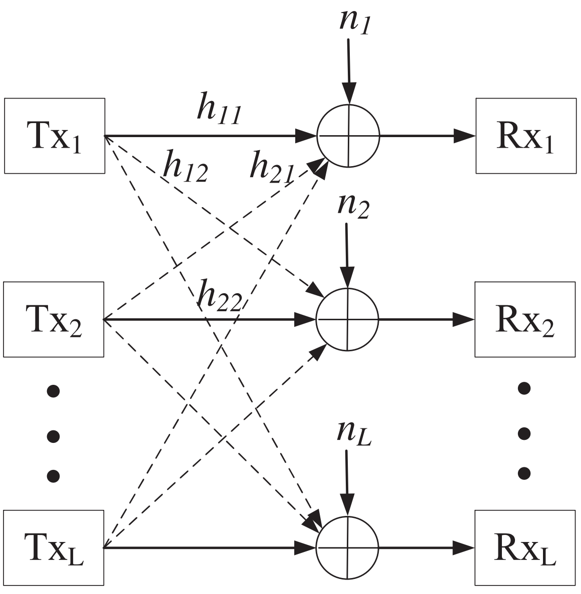

Figure 1 System model.

$$\gamma_l(\mathbf{p}) = \frac{h_{ll}p_l}{n_l + \sum\limits_{k \neq l} h_{kl}p_k}, \tag{1}$$

where $\mathbf{p} = (p_1, \ldots, p_L)$ is a vector of the users' transmission powers and n_l is the noise power. Accordingly, the lth user receives the utility $U_l(\gamma_l)$, where $U_l(\cdot)$ is continuous and strictly increasing. We assume that each user l's utility is zero when $\gamma_l = 0$, i.e., $U_l(0) = 0$. For ease of reference, the key notation of this article is listed in Table 1.[b]

Table 1 Summary of the notations and definitions

Notation	Definition
$\mathcal{L}$	Set of links
L	Total number of links
h_{lk}	Channel gain from link l's transmitter to link k's receiver
$\mathbf{H}$	Link gain matrix
p_l (in vector $\mathbf{p}$)	Transmission power of link l
n_l (in vector $\mathbf{n}$)	Noise power for link l
γ_l	SINR of link l
$\gamma_l(\cdot)$	SINR function of link l
$U_l(\cdot)$	Utility function of link l
x_l (in vector $\mathbf{x}$)	Ratio of link l's utility to the total network utility
r_l (in vector $\mathbf{r}$)	Transmission rate of link l
$r_l(\cdot)$	Transmission rate function of link l
α, β	Penalty multipliers

Network utility maximization

We seek to find the optimal power allocation $\mathbf{p}^*$ that maximizes the overall system utility subject to the individual power constraints, given by the following optimization problem[c]:

$$\begin{array}{ll} \text{maximize} & \sum\limits_{l \in \mathcal{L}} U_l(\gamma_l(\mathbf{p})) \\ \text{subject to} & 0 \leq p_l \leq P_l^{\max}, \forall l \in \mathcal{L} \\ \text{variables} & \{\mathbf{p}\}. \end{array} \tag{2}$$

In general, (2) is a nonconvex optimization problem.[d] In particular, if the utility function is the Shannon rate achievable over Gaussian flat fading channels, namely $U_l(\gamma_l(\mathbf{p})) = w_l \log(1 + \gamma_l(\mathbf{p}))$, where $w_l > 0$ is a weight associated with user l, (2) boils down to the weighted sum rate maximization problem, which is known to be nonconvex and NP-hard [5]. Note that the weights in (2) can serve as the fairness measures [25] for different scenarios. In particular, in queueing systems, packet queues for arrival rates within the stability region can be stabilized by solving this weighted sum rate maximization problem, where the instantaneous queue lengths are chosen as the weights. In Section "Joint scheduling and power control for stability of queueing systems", we will discuss how to stabilize the packet queues by integrating our distributed power control algorithms with the back-pressure algorithm.

Let $\mathcal{F}$ denote the feasible utility region, where for each point $\mathbf{U} = (U_1, \ldots, U_L)$ in $\mathcal{F}$, there exists a power vector $\mathbf{p}$ such that $U_l = U_l(\gamma_l(\mathbf{p}))$ for all $l \in \mathcal{L}$. The feasible utility region $\mathcal{F}$ is nonconvex, and in general, finding the globally optimal solution to (2) in $\mathcal{F}$ is challenging. In the following example, we illustrate the geometry of $\mathcal{F}$ for the utility function $U_l(\gamma_l(\mathbf{p})) = w_l \log(1 + \gamma_l(\mathbf{p}))$ and evaluate the solutions to (2) given by some existing power control approaches discussed in Section "Introduction".

Example. For the case with two links, Figure 2 illustrates the nonconvex feasible utility region $\mathcal{F}$ for different system parameters. We compare the performance of the existing approaches [3,6,14,15] in Table 2.

Remarks. The solutions to (2) given by the authors of [3,6,14] are either distributed but suboptimal or optimal but centralized. In particular, Chiang et al. [3] solve (2) by using geometric programming (GP) under the high-SINR assumption, which yields a suboptimal solution to (2) when this assumption does not hold (e.g., this is the case in the example above). The ADP algorithm [6] can guarantee only local optimality[e] in a distributed manner. The MAPEL algorithm [14] can achieve the globally optimal solutions but it is centralized with high computational complexity. Compared with these algorithms,

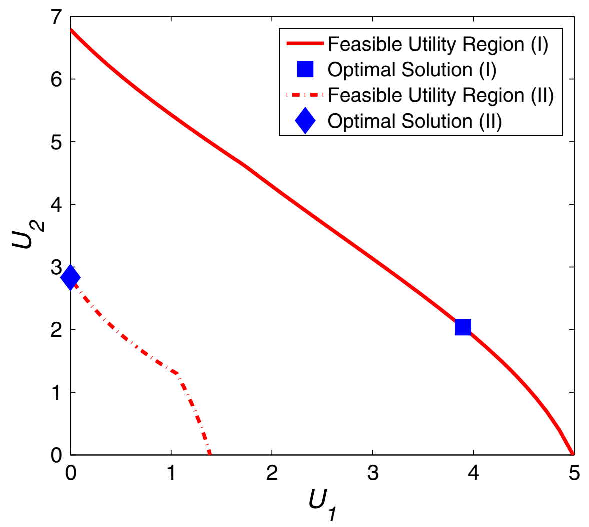

Figure 2 The feasible utility region $\mathcal{F}$. Case (I): the channel gains are given by $h_{11} = 0.73$, $h_{12} = 0.04$, $h_{21} = 0.03$, and $h_{22} = 0.89$, and the maximum power are $P_1^{\max} = 20$, $P_2^{\max} = 100$; Case (II): the channel gains are given by $h_{11} = 0.30$, $h_{12} = 0.50$, $h_{21} = 0.03$, and $h_{22} = 0.80$, and the maximum power are $P_1^{\max} = 1$, $P_2^{\max} = 2$. In both cases, the noise power is 0.1 for each link, and the weights are $w_1 = 0.57$, $w_2 = 0.43$.

the SEER algorithm [15] can guarantee global optimality in a distributed manner but message passing needed in each iteration can be demanding, i.e., each link needs the knowledge of the channel gains, the receiver SINR and the signal power of all the other links. It is worth noting that the performance of SEER hinges heavily on the control parameter that can be challenging to choose on the fly.

From network utility maximization to minimum weighted utility maximization

In order to devise low-complexity distributed algorithms that can guarantee global optimality, we first study the basic properties for the solutions to (2), before transforming (2) into a more structured max–min problem.

Lemma 1. *The optimal solution to (2) is on the boundary of the feasible utility region $\mathcal{F}$.*

Table 2 The performance of the existing approaches for Case I and II

Approach	Case I		Case II	
	Power	Sum rate	Power	Sum rate
GP	[20, 7.68]	3.02	[1, 0.61]	0.98
ADP	[20, 6.46]	3.10	[1, 2]	1.16
MAPEL	[20, 6.79]	3.10	[0, 2]	1.22
SEER	[20, 6.90]	3.10	[0, 2]	1.22

Proof. Let $\mathbf{U}^*$ denote a globally optimal solution to (2) over $\mathcal{F}$, and γ^* denote the corresponding SINR that supports $\mathbf{U}^*$. Since $U_l(\cdot)$ is continuous and strictly increasing, proving that $\mathbf{U}^*$ is on the boundary of $\mathcal{F}$ is equivalent to showing that γ^* is on the boundary of the feasible SINR region. Suppose that γ^* is not on the boundary of the feasible SINR region, which indicates that there exists some point $\hat{\gamma}$ such that $\hat{\gamma}_l \geq \gamma_l^*$ for all $l \in \mathcal{L}$ and $\hat{\gamma}_{l'} > \gamma_{l'}^*$ for some $l' \in \mathcal{L}$. Since $U_l(\cdot)$ for any $l \in \mathcal{L}$ is strictly increasing in γ_l, we have $U_l(\hat{\gamma}_l) \geq U_l(\gamma_l^*)$ for all $l \in \mathcal{L}$ and $U_l(\hat{\gamma}_{l'}) > U_l(\gamma_{l'}^*)$ for some $l' \in \mathcal{L}$, which contradicts the fact that γ^* is a globally optimal solution. Hence, Lemma 1 follows. □

Based on Lemma 1, if we can characterize the boundary of $\mathcal{F}$, then it is possible to solve (2) efficiently. Thus motivated, we first establish, by introducing a "contribution weight" for each user, the equivalence between (2) and the minimum weighted utility maximization problem.

Lemma 2. *Problem (2) is equivalent to the following minimum weighted utility maximization:*

$$\begin{array}{ll} \text{maximize} & \min\limits_{l \in \mathcal{L}} \dfrac{U_l(\gamma_l(\mathbf{p}))}{x_l} \\ \text{subject to} & 0 \leq p_l \leq P_l^{\max}, \forall l \in \mathcal{L} \\ & 0 \leq x_l \leq 1, \forall l \in \mathcal{L} \\ & \sum\limits_{l \in \mathcal{L}} x_l = 1 \\ \text{variables} & \{\mathbf{p}, \mathbf{x}\}. \end{array} \tag{3}$$

Proof. Let $t = \sum\limits_{l \in \mathcal{L}} U_l(\gamma_l(\mathbf{p}))$ denote the total utility. Since $U_l(\cdot)$ is nonnegative, we define $x_l \in [0, 1]$ as a ratio for the contribution of user l's utility to t. Therefore, $U_l(\gamma_l(\mathbf{p})) = tx_l$ and $\sum\limits_{l \in \mathcal{L}} x_l = 1$. Then (2) can be rewritten as

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & t = \dfrac{U_l(\gamma_l(\mathbf{p}))}{x_l}, \forall l \in \mathcal{L} \\ & 0 \leq p_l \leq P_l^{\max}, \forall l \in \mathcal{L} \\ & 0 \leq x_l \leq 1, \forall l \in \mathcal{L} \\ & 0 \leq t, \sum\limits_{l \in \mathcal{L}} x_l = 1 \\ \text{variables} & \{\mathbf{p}, \mathbf{x}, t\}. \end{array} \tag{4}$$

In order to maximize t, it suffices then to relax $t = \frac{U_l(\gamma_l(\mathbf{p}))}{x_l}$ in (4) as $t \leq \frac{U_l(\gamma_l(\mathbf{p}))}{x_l}$, $\forall l \in \mathcal{L}$, which is equivalent to $t \leq \min\limits_{l \in \mathcal{L}} \frac{U_l(\gamma_l(\mathbf{p}))}{x_l}$. Therefore, (4) can be treated as the hypograph form of (3), i.e., (4) and (3) are equivalent [26], thereby concluding the proof. □

By transforming (2) to this more structured max–min problem (3), the problem is reduced to finding a globally optimal $\mathbf{x}^*$, given which we can efficiently obtain a globally optimal solution, i.e., the tangent point of the hyperplane and $\mathcal{F}$, as illustrated in Figure 3. Intuitively speaking, $\mathbf{x}$ represents a search direction. Once we find the best search direction $\mathbf{x}^*$, $\mathbf{p}^*$ can be obtained efficiently by searching along the direction of $\mathbf{x}^*$. Actually, for given $\mathbf{x}$, (3) is quasi-convex.[f] By introducing an auxiliary variable t, we obtain the following equivalent formulation:

$$\begin{array}{ll} \text{maximize} & t \\ \text{subject to} & U_l^{-1}(tx_l)(n_l + \sum_{k \neq l} h_{kl}p_k) \leq h_{ll}p_l \\ & 0 \leq p_l \leq P_l^{\max}, \forall l \in \mathcal{L},\ 0 \leq t \\ \text{variables} & \{\mathbf{p}, t\}, \end{array} \tag{5}$$

which can be solved in polynomial time through binary search on t [26]. However, the optimal search direction $\mathbf{x}^*$ is difficult to find due to the nonconvex nature of the problem. In the following section, we study how to find the globally optimal search direction $\mathbf{x}^*$.

Centralized versus distributed algorithms

In this section, we study algorithms achieving global optimality for (3). First, we propose a centralized algorithm for (3), which will serve as a benchmark for performance comparison. Then, by using EDT and SA, we propose a distributed algorithm, DSPC, for the problem (3). Building on this, we propose an EDSPC algorithm to improve the convergence rate of DSPC.

A centralized algorithm

Based on Lemmas 1 and 2, we develop a centralized algorithm (Algorithm 1) to solve the max–min optimization problem (3) under consideration. Roughly speaking, by dividing the simplex $\mathcal{S} = \{\mathbf{x} | \sum_{l \in \mathcal{L}} x_l = 1, 0 \leq x_l \leq 1, \forall l \in \mathcal{L}\}$ into many small simplices, the algorithm can find the optimal point on the boundary of $\mathcal{F}$. Figure 4 illustrates how the simplex cutting is performed for the case with three links. Compared with the MAPEL algorithm [14], Algorithm 1 directly computes the points on the boundary, instead of constructing a series of polyblocks to approximate the boundary of the feasible region.

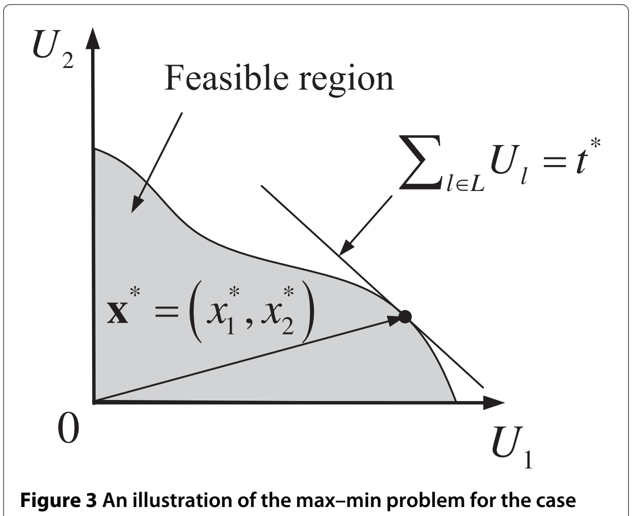

Figure 3 An illustration of the max–min problem for the case with two links.

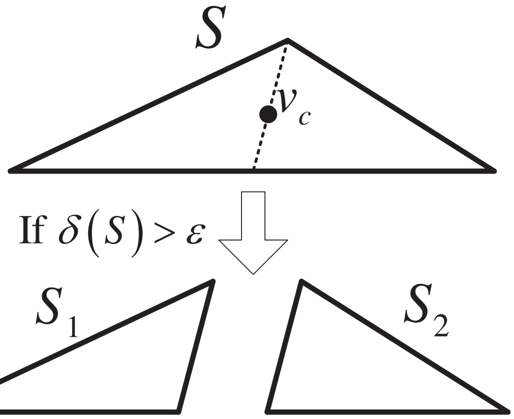

Figure 4 An illustration of the simplex cutting for the case with three links.

Algorithm 1

Initialization: Choose the approximation factor $\epsilon > 0$, and construct the initial simplex $\mathcal{S}$ with the vertex set $V = \{v_1, \ldots, v_L\}$, where $v_l = e_l$ and e_l is the lth unit coordinate vector. Let $v_c = \frac{1}{L}\sum_{l \in \mathcal{L}} v_l$ be the center of S. Compute $\mathbf{p}^*$ by solving (5) at the point $\mathbf{x} = v_c$. Denote $\delta(\mathcal{S}) = \max_{v \in V} |v_c - v|$ as the diameter of $\mathcal{S}$.

Repeat

1. Divide each simplex $\mathcal{S}_i$ by using *bisection* method, which chooses the midpoint of one of the longest edges of the simplex $\mathcal{S}_i$, i.e., $v_m = \frac{1}{2}(v_r + v_s)$, where v_r and v_s are the end points of a longest edge of the simplex. In this case, the simplex $\mathcal{S}_i$ is subdivided into two simplices $\mathcal{S}_{i_1}$ and $\mathcal{S}_{i_2}$.
2. For each new simplex $\mathcal{S}_{i_j}$, compute the diameter $\delta(\mathcal{S}_{i_j})$ and $\mathbf{p}^*$ by solving (5) at $\mathbf{x}$ given by the center point of the simplex.
3. Find the current best solution to (3) and the maximal diameter δ^m in these new subdivided simplices.

Until $\delta^m < \epsilon$.

Proposition 1. *Algorithm 1 converges monotonically to the globally optimal solution to (3) as the approximation factor ϵ approaches zero.*

Proof. For given ϵ, Algorithm 1 divides the simplex $\mathcal{S} = \{\mathbf{x} | \sum_{l \in \mathcal{L}} x_l = 1, 0 \leq x_l \leq 1, \forall l \in \mathcal{L}\}$ until the maximal diameter of the subdivided simplices δ^m is less than ϵ.

Let $d(\epsilon)$ denote the maximum distance between the optimal solution $\mathbf{x}^*$ and the center point of the simplex that contains $\mathbf{x}^*$. Obviously, $d(\epsilon)$ is bounded by δ^m. Since δ^m decreases with the decreasing of ϵ, therefore $d(\epsilon)$ decreases monotonically with the decreasing of ϵ, i.e., the solution given by Algorithm 1 monotonically converges to $\mathbf{x}^*$. As ϵ approaches zero, Algorithm 1 exhaustively searches every point in the simplex $\mathcal{S}$, thereby concluding the proof. □

Remarks. Algorithm 1 can be used to obtain an ϵ-optimal solution with $|\mathbf{x} - \mathbf{x}^*| \leq \epsilon$. That is to say, by controlling ϵ, one can strike a balance between the optimality and the computation time. Since finding the globally optimal solution requires centralized implementation, Algorithm 1 will be used only as a benchmark for performance evaluation of distributed algorithms.

DSPC algorithm

Next, we devise a DSPC algorithm based on EDT [17] and SA [18]. Compared to the classical duality theory with nonzero duality gap for nonconvex optimization problems, EDT can guarantee zero duality gap between the primal and dual problems by utilizing nonlinear Lagrangian functions. This property allows for solving the nonconvex problem by its *extended dual* while preserving the global optimality in distributed implementation. To this end, we first introduce auxiliary variables and use EDT to transform (3) with the auxiliary variables into an unconstrained problem. Then, we solve the unconstrained problem by using the SA mechanism. Specifically, we define $t_l = \frac{U_l(\gamma_l(\mathbf{p}))}{x_l}$ and rewrite (3) as

$$\begin{array}{ll} \text{minimize} & -\min\limits_{l \in \mathcal{L}} t_l \\ \text{subject to} & t_l x_l \leq U_l(\gamma_l(\mathbf{p})), \forall l \in \mathcal{L} \\ & \sum\limits_{l \in \mathcal{L}} x_l = 1 \\ & 0 \leq p_l \leq P_l^{\max}, \forall l \in \mathcal{L} \\ & 0 \leq t_l, 0 \leq x_l \leq 1, \forall l \in \mathcal{L} \\ \text{variables} & \{\mathbf{p}, \mathbf{x}, \mathbf{t}\}. \end{array} \tag{6}$$

Next, we use EDT to write the Lagrangian function for (6) as

$$\begin{aligned} L(\mathbf{p}, \mathbf{x}, \mathbf{t}, \alpha, \boldsymbol{\beta}) = & -\min_{l \in \mathcal{L}} t_l + \alpha \Big| \sum_{l \in \mathcal{L}} x_l - 1 \Big| \\ & + \sum_{l \in \mathcal{L}} \beta_l (t_l x_l - U_l(\gamma_l(\mathbf{p})))^+, \end{aligned} \tag{7}$$

where $(y)^+ = \max(0, y)$, and $\alpha \in \mathbb{R}$ and $\boldsymbol{\beta} \in \mathbb{R}^L$ are the penalty multipliers for penalizing the constraint violations. Based on EDT [17], there exist finite $\alpha^* \geq 0$ and $\beta_l^* \geq 0$, for all $l \in \mathcal{L}$, such that, for any $\alpha > \alpha^*$ and $\beta_l > \beta_l^*$, $\forall\ l \in \mathcal{L}$, the solution to (7) is the same as (6). Note that (7) does not include the constraints of p_l, x_l, and t_l for each user, and there will be no constraint violation when each user updates these variables locally. Therefore, the minimization of (7) with respect to the primal variables ($\mathbf{p}$, $\mathbf{x}$, and $\mathbf{t}$) can be carried out individually by each user in a distributed fashion.

The next key step is to perform a stochastic local search by each user based on SA. Let t_l, x_l, and p_l denote the primal values of the lth user, and t_l' and x_l' denote the new values randomly chosen by the lth user. Accordingly, $t_l' x_l'$ can be treated as a new target utility for the lth user. To achieve this target utility, the lth user updates p_l' by

$$p_l' = \min\left(\frac{U_l^{-1}(t_l' x_l')}{\gamma_l} p_l, P_l^{\max} \right), \tag{8}$$

where γ_l is the current SINR measured at the lth user's receiver. Note that (8) does not need any information of channel gains, except the feedback of SINR γ_l. Since (8) corresponds to the distributed power control algorithm of standard form as described in [27],[g] it converges geometrically fast to the target utility. Thus, we assume that each user l updates p_l at a faster time-scale than t_l and x_l such that p_l always converges before the next update of t_l and x_l. Next, we use SA to update t_l and x_l in a stochastic operation. By using the analogy with annealing in metallurgy, SA was proposed in [18] to mimic the behavior of the microscopic constituents in heating and controlled cooling of a material. By allowing occasional uphill moves, SA is able to escape from the local optimal points. In particular, let Δ denote the difference between $L(p_l, x_l, t_l | \mathbf{p}_{-l}, \mathbf{x}_{-l}, \mathbf{t}_{-l}, \alpha, \boldsymbol{\beta})$ and $L(p_l', x_l', t_l' | \mathbf{p}_{-l}', \mathbf{x}_{-l}, \mathbf{t}_{-l}, \alpha, \boldsymbol{\beta})$, where $\mathbf{y}_{-l}$ is the vector $\mathbf{y}$ without the lth user's variable. If $\Delta \geq 0$, i.e., t_l', x_l' and p_l' reduce Lagrangian (7), then they are accepted with probability 1; otherwise, they are accepted with probability $\exp\left(\frac{\Delta}{T}\right)$, where T is a control parameter and sometimes it is called temperature. Note that, as T decreases, the acceptance of uphill move becomes less and less probable, and therefore a fine-grained search is needed. It has been shown that, as T approaches 0 according to a *logarithmic cooling schedule*, SA converges to a globally optimal point [16,19]. To compute Δ locally by each user l, user l needs to broadcast the terms t_l, x_l and $\beta_l(t_l x_l - U_l(\gamma_l(\mathbf{p})))^+$, whenever any of these terms changes.

Note that the target utility $t_l x_l$ may not be feasible, i.e., the target utility cannot be achieved even though the user transmits at the maximum power. In this case, it can be shown that the power of those users with feasible target utilities will converge to a feasible solution, whereas the other users that cannot achieve the target utility will continue to transmit at maximum power [1]. If some target utility is not feasible as T tends to 0, based on EDT, the current values of α and $\boldsymbol{\beta}$ do not satisfy $\alpha > \alpha^*$ or $\beta_l > \beta_l^*$ for all $l \in \mathcal{L}$. Therefore, each user l also needs to update α and β_l. In particular, if any constraint is violated, α and β_l are updated as follows:

$$\begin{aligned} \alpha &\leftarrow \alpha + \sigma \Big| \sum_{l \in \mathcal{L}} x_l - 1 \Big|, \\ \beta_l &\leftarrow \beta_l + \varrho_l (t_l x_l - U_l(\gamma_l))^+, \ \forall l \in \mathcal{L}, \end{aligned} \tag{9}$$

where σ and ϱ_l are used to control the rate of updating α and β_l. A detailed description of DSPC algorithm is given in Algorithm 2.

Remarks:

(1) In Algorithm 2, each user randomly picks $t_l' \in [U_l^{\max}, U_{tot}^{\max}]$, where $U_l^{\max}$ denotes the maximum utility of the user l when the user l transmits at the maximum power while the other users do not transmit, and $U_{tot}^{\max} = \sum_{l \in \mathcal{L}} U_l^{\max}$. Note that $U_l^{\max}$ can be computed by each user locally. Further, we assume that each user broadcasts $U_l^{\max}$ before running the algorithm so that $U_{tot}^{\max}$ is also known by each user.

(2) In practice, after initialization, α and β_l increase in proportion to the violation of the corresponding constraint, which may lead to excessively large penalty values. Since it is beneficial to periodically scale down the penalty values to ease the unconstrained optimization, α and β_l are scaled down by multiplying with a random value (it can be chosen between 0.7 and 0.95 according to [17]), if the *penalty decrease condition* is satisfied, i.e., the maximum violation of constraints is not decreased after consecutively running Step 1 in Algorithm 2 several times, e.g., five times in [17].

(3) In Algorithm 2, each user requires the knowledge of T and time epochs $\{\tau_1, \tau_2, \ldots\}$ to update t_l and x_l, which can be determined and informed to each user offline.

Algorithm 2 DSPC

```
Initialization: Choose $\epsilon > 0$. Let $\alpha = 0$, $\beta_l = 0$, $\forall l \in \mathcal{L}$, and randomly choose $\mathbf{p}$, $\mathbf{x}$ and $\mathbf{t}$.
Step 1: update primal variables
Set $T = T_0$, and select a sequence of time epochs $\{\tau_1, \tau_2, \ldots\}$ in continuous time.
  Repeat for each user $l$
    1. Randomly pick $t_l' \in [U_l^{\max}, U_{tot}^{\max}]$ and $x_l' \in [0, 1]$, and update $p_l'$ according to (8).
    2. Keep sensing the change of $\beta_l(t_l x_l - U_l(\gamma_l(\mathbf{p})))^+$ broadcast by other users.
    3. Compute $\Delta$, and accept $t_l'$, $x_l'$, and $p_l'$ with probability 1, if $\Delta \geq 0$, or with probability $\exp(\frac{\Delta}{T})$, otherwise.
    4. Broadcast $t_l'$ and $x_l'$, if $t_l'$ and $x_l'$ are updated.
    5. For each time epoch $\tau_i$, update $T = T_0 / \log(i+1)$.
  Until $T < \epsilon$.
Step 2: update penalty variables
  For each user $l$,
    1. Update $\alpha$ and $\beta_l$ according to (9), and scale down $\alpha$ and $\beta_l$, if the penalty decrease condition is satisfied.
    2. Go to Step 1 until no constraint is violated.
```

Proposition 2. *The DSPC algorithm (Algorithm 2) converges almost surely to a globally optimal solution to (3), as temperature T in SA decreases to zero.*

Proof. To show that Algorithm 2 converges almost surely to a globally optimal solution to (3), we only need to show that when $\alpha > \alpha^*$ and $\beta_l > \beta_l^*$ for all $l \in \mathcal{L}$, Algorithm 2 can converge almost surely to a globally optimal solution to (6), since (3) is equivalent to (6), and if the solution does not satisfy the constraints of (6), α and β_l will increase in proportion to the violation of the corresponding constraint. Since Algorithm 2 uses SA with the logarithmic cooling schedule, based on [16,19] it can converge almost surely to a globally optimal solution to (7), which is also a globally optimal solution to (6) based on EDT when $\alpha > \alpha^*$ and $\beta_l > \beta_l^*$ for all $l \in \mathcal{L}$ [17]. Hence, Proposition 2 follows. □

Remarks. The DSPC algorithm can guarantee global optimality in a distributed manner without the need of channel information. In particular, it needs the information of t_l and x_l, and can adapt to channel variations by utilizing the SINR feedback. However, the convergence rate of DSPC is slow due to the use of logarithmic cooling schedule.

EDSPC algorithm

It can be seen from Algorithm 2 that it is critical to find the optimal penalty variables α and $\boldsymbol{\beta}$ for computing (7). Moreover, a logarithmic cooling schedule is used to ensure convergence to a global optimum. To improve the convergence rate, we propose next an EDSPC algorithm by empirically choosing the initial penalty values α_0 and $\boldsymbol{\beta}_0$ and employing a *geometric cooling schedule* [18], which reduces the temperature T in SA by $T = \xi T$, $0 < \xi < 1$, at each time epoch. Compared with the logarithmic cooling schedule, T converges to 0 much faster under the geometric cooling schedule, which in turn improves the convergence rate beyond DSPC. The resulting solution is given in Algorithm 3.

We note that although EDSPC converges much faster than DSPC, it may yield only near-optimal solutions. Based on EDT, we choose $\alpha_0 > \alpha^*$ and $\beta_{0l} > \beta_l^*$, $\forall\, l \in \mathcal{L}$, to satisfy the optimality conditions for penalty variables. Obviously, by choosing large α_0 and β_{0l}, these conditions can be always satisfied. Nevertheless, very large penalties introduce heavy costs for constraint violations such that EDSPC may end up with a feasible but suboptimal solution. Therefore, the selection of initial penalty values plays a critical role in the performance of EDSPC and deserves more attention in future work. In practice, we can choose the initial penalties based on the maximum value of the constraint that is associated with each of the penalty variables. This choice performs well in the simulations. For example, we can choose $\beta_{0l} = U_l^{\max}$ for the constraint $t_l x_l \le U_l(\gamma_l(\mathbf{p}))$.

```
Algorithm 3 EDSPC
Initialization: Choose $\epsilon > 0$. Let $\alpha = \alpha_0$, $\beta_l = \beta_{0l}$,
$\forall l \in \mathcal{L}$, and randomly choose $\mathbf{p}$, $\mathbf{x}$ and $\mathbf{t}$.
Set $T = T_0$, and select a sequence of time epochs
$\{\tau_1, \tau_2, \ldots\}$ in continuous time.
Repeat for each user $l$
    1. Randomly pick $t_l' \in [\,U_l^{\max}, U_{tot}^{\max}]$ and $x_l' \in [\,0, 1]$, and
       update $p_l'$ according to (8).
    2. Keep sensing the change of $\beta_l(t_l x_l - U_l(\gamma_l(\mathbf{p})))^+$
       broadcast by other users.
    3. Compute $\Delta$, and accept $t_l'$, $x_l'$, and $p_l'$ with probability
       1, if $\Delta \ge 0$, or with probability $\exp(\frac{\Delta}{T})$, otherwise.
    4. Broadcast $t_l'$ and $x_l'$, if $t_l'$ and $x_l'$ are updated.
    5. For each time epoch $\tau_i$, update $T = \xi T$.
Until $T < \epsilon$.
```

Performance evaluation

In this section, we evaluate the utility and convergence performance of Algorithms 2 and 3 (DSPC[h] and EDSPC). We consider a wireless network with six links randomly distributed on a 10-by-10 m^2 square area. The channel gains h_{lk} are equal to d_{lk}^{-4}, where d_{lk} represents the distance between the transmitter of user l and the receiver of user k. We assume $U_l(\gamma_l(\mathbf{p})) = \log(1 + \gamma_l(\mathbf{p}))$, $P_l^{\max} = 1$ and $n_l = 10^{-4}$ for all $l \in \mathcal{L}$, and consider one randomly generated realization of channel gains given by

$$\mathbf{H} = \begin{bmatrix} 0.3318 & 0.0049 & 0.0141 & 0.0021 & 0.0016 & 0.0007 \\ 0.0031 & 0.9554 & 0.0063 & 0.0140 & 0.0012 & 0.0025 \\ 0.0155 & 0.0042 & 0.6166 & 0.0046 & 0.0108 & 0.0018 \\ 0.0017 & 0.2188 & 0.0340 & 0.6754 & 0.0062 & 0.0215 \\ 0.0020 & 0.0017 & 0.2216 & 0.0042 & 0.2955 & 0.0028 \\ 0.0007 & 0.0079 & 0.0254 & 0.2553 & 0.0404 & 0.3025 \end{bmatrix}.$$

Figure 5 shows how the total utility in the EDSPC algorithm converges over time, when we choose all the initial penalty values equal to 10. Also, we choose $\xi = 0.9$, $\rho = 1$, and $\varrho = 1$, and use Algorithm 1 as a benchmark to evaluate the optimal performance. As shown in Figure 5, the EDSPC algorithm approaches the optimal utility, when the initial penalty values are carefully chosen. Moreover, the convergence rate of the EDSPC algorithm is much faster than DSPC, since DSPC continues updating the penalty values even after the optimal solution is found for the current penalty values. Figure 6 illustrates the average performance (with confidence interval) of DSPC, EDSPC, SEER, and ADP under 100 random initializations, with the same system parameters as used in Figure 5. As shown in Figure 6, both DSPC and EDSPC are robust against the variations of initial values.

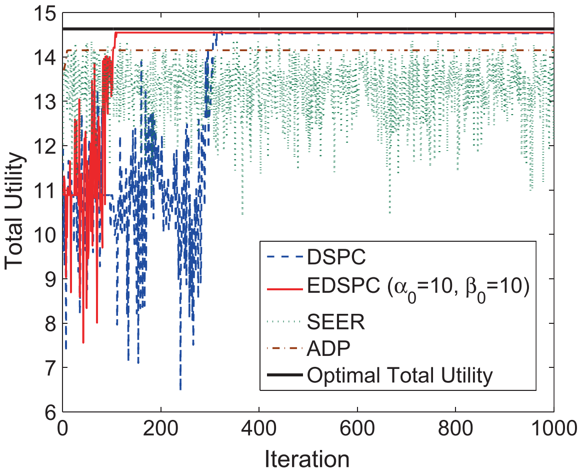

Figure 5 Convergence performance of DSPC, EDSPC, SEER, and ADP.

Figures 5 and 6 compare the proposed algorithms with the SEER and ADP. As mentioned in Section "Introduction", ADP can only guarantee local optimality. Therefore, for nonconvex problems (e.g., in this example), ADP may converge to a suboptimal solution. As noted in [15], the performance of SEER heavily hinges on the control parameter that can be challenging to choose in online

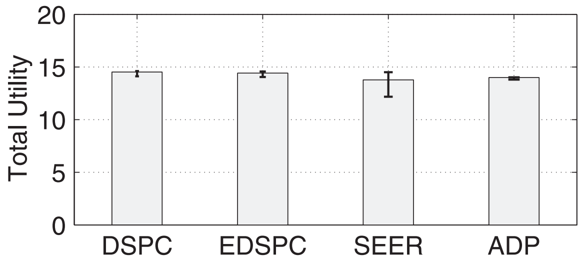

Figure 6 Comparison of the average utility performance (with confidence interval) of DSPC, EDSPC, SEER, and ADP.

operation. In contrast, DSPC can approach the globally optimal solution regardless of the initial parameter selection, but the convergence rate may be slower. Furthermore, EDSPC improves the convergence rate, but in this case the initial penalty values would impact how close it can approach the optimal point. In terms of message passing, our algorithms do not require individual links to know the channel gains (including its own channel gain), the receiver SINR of the other links and the signal power of the other links, which are all needed in the SEER algorithm.

Joint scheduling and power control for stability of queueing systems

In Section "Power control for unicast communications", we studied the distributed power allocation, by using DSPC and EDSPC, for utility maximization in the saturated case with uninterrupted packet traffic. In this section, we generalize the study by considering a queueing system with dynamic packet arrivals and departures. Specifically, we develop a joint scheduling and power allocation policy to stabilize packet queues by integrating our power control algorithms with the celebrated back-pressure algorithm [20].

Stability region and throughput optimal power allocation policy

Consider the same wireless network model with L links as in Section "Power control for unicast communications". We assume that there are S classes of users in the system, and that the traffic brought by users of class s follows $\{A_{sl}(t)\}_{t=1}^{\infty}$, which are *i.i.d.* sequences of random variables for all $l = 1, \ldots, L$ and $s = 1, \ldots, S$, where $A_{sl}(t)$ denotes the amount of traffic generated by users of class s that enters the link l in slot t. We assume that the second moments of the arrival process $\{A_{sl}(t)\}_{t=1}^{\infty}$ are finite. Let $Q^s_{T(l)}(t)$ and $Q^s_{R(l)}(t)$ denote the current backlog in the queue of class s in slot t on the transmitter and receiver sides of link l, respectively. The queue length $Q^s_{T(l)}(t)$ evolves over time as

$$Q^s_{T(l)}(t+1) = \max(Q^s_{T(l)}(t) - r^s_l(t), 0) + A_{sl}(t) + \sum_{\{m|T(l)=R(m), m\in\mathcal{L}\}} r^s_m(t), \qquad (10)$$

where $r^s_l(t)$ denotes the transmission rate of link l for users of class s. The third term in (10) denotes the traffic from the other links. The queue length process $\{Q^s_{T(l)}(t)\}_{t=1}^{\infty}$ forms a Markov chain.

Let ψ_s denote the first moment of $\{A_{sl}(t)\}_{t=1}^{\infty}$, i.e., the load brought by users of class s. As is standard [20,21,28], the stability region is defined as follows.

Definition 1. *The stability region Λ is the closure of the set of all $\{\psi_s\}_{s=1}^{S}$ for which there exists some feasible power allocation policy under which the system is stable, i.e., $\Lambda = \bigcup_{\mathbf{p}\in\mathcal{P}} \Lambda(\mathbf{p})$, where $\Lambda(\mathbf{p}) = \{\{\psi_s\}_{s=1}^{S} | \sum_{s=1}^{S} E_{sl}\psi_s < r_l(\mathbf{p}), \forall l\}$, $\mathcal{P}$ denotes the set of feasible power allocation, $r_l(\mathbf{p})$ denotes the rate of link l under power allocation $\mathbf{p}$, and $E_{sl} = 1$ is the indicator that the path of users of class s uses link l, and $E_{sl} = 0$, otherwise.*

For the sake of comparison, the throughput region[i] $\mathcal{F}$ of the corresponding saturated case is defined as the set of all feasible link rates, i.e., $\mathcal{F} = \{\mathbf{r}|r_l = r_l(\mathbf{p}), \mathbf{p} \in \mathcal{P}\}$. In general, the throughput region $\mathcal{F}$ may be different from the stability region Λ, except for some special cases (e.g., in slotted ALOHA systems the throughput region and the stability region are the same [29] for two links and in a multiple-access channel the information theoretic capacity region is equivalent to its stability region under specific feedback assumptions [30]).

The queueing system is stable if the arrival rates of packet queues are less than the service rates such that the queue lengths do not grow to infinity [31]. In order to stabilize packet queues, it is critical to find the optimal scheduling and power allocation policy that maximizes the weighted sum rate given by (11). By integrating our power control algorithms with the back-pressure algorithm, we propose a joint scheduling and power allocation policy presented in Algorithm 4 to stabilize the queueing system.

Proposition 3. *The joint scheduling and power allocation policy (Algorithm 4) can stabilize the system such that $\limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1}\sum_{l,s} E\{Q^s_l(\tau)\} < \infty$, when the traffic load $\{\psi_s\}_{s=1}^{S}$ is strictly interior to the stability region Λ, i.e., there exists some $\epsilon > 0$ such that $\{\psi_s + \epsilon\}_{s=1}^{S} \in \Lambda$.*

The proof is similar to that in [21,32], and is omitted for brevity.

Note that Algorithm 4 can be viewed as a dynamic back-pressure and resource allocation policy [32], crafted towards solving the weighted sum rate maximization problem (11). Specifically, by using the DSPC algorithm, Algorithm 4 can be implemented distributively to find the globally optimal resource allocation. We should caution that EDSPC can be applied to improve the convergence rate of Stage 2 in Algorithm 4 but it may render a suboptimal schedule (i.e., it can not stabilize all possible $\{\psi_s\}_{s=1}^{S}$ within Λ), due to the fact that EDSPC may not always find the global optimal power allocation.

To reduce the complexity, we can consider a policy that computes (11) periodically every few slots, and it can be shown that this policy can also stabilize the system, when $\{\psi_s\}_{s=1}^{S}$ is strictly interior to the stability region Λ [33,34].

Algorithm 4 Joint scheduling and power allocation policy

Stage 1: For each link l, select a link weight according to $w_l(t) = \max_{s=1,\ldots,S} D_l^s(t)$, where the difference of queue lengths of class s is $D_l^s(t) = \max(Q^s_{T(l)}(t) - Q^s_{R(l)}(t), 0)$, if the receiver of link l is not the destination of class s's traffic, and $D_l^s(t) = Q^s_{T(l)}$, otherwise.

Stage 2: Compute the optimal power allocation $\mathbf{p}^*$ in each slot t by solving the following problem with DSPC algorithm

$$\mathbf{p}^* = \arg\max_{\mathbf{p}} \sum_{l=1}^{L} w_l(t) r_l(\mathbf{p}). \tag{11}$$

Thus, the transmission rate of link l in slot t is given by $r_l(\mathbf{p}^*) = \log(1 + \gamma_l(\mathbf{p}^*))$.

Stage 3: Let $s_l^* = \arg\max_{s=1,\ldots,S} D_{sl}(t)$ denote the class scheduled in slot t; if multiple classes satisfy this condition, then s_l^* is randomly chosen as one of these classes. Then, schedule these classes according to the solution given by Stage 2.

Performance evaluation

In this section, we present numerical results to illustrate the use of Algorithm 4 for stabilizing a queueing system. We consider a one-hop network (i.e., $\mathbf{E} = \{E_{sl}\}$ is the identity matrix) with two users (classes), where the channel gains are $h_{11} = 0.3$, $h_{12} = 0.5$, $h_{21} = 0.03$, and $h_{22} = 0.8$, and the noise power is 0.1 for each link. The maximum transmission power is set to 1 and 2 for links 1 and 2, respectively. Besides, we assume that the users of class s arrive at the network according to a Poisson process with rate λ_s, and that the size of packet batch for users of class s follows an exponential distribution with mean ν_s. The load brought by users of class s is then $\psi_s = \lambda_s \nu_s$. Figure 7 shows the stability region Λ and compares it with the throughput region $\mathcal{F}$ of the corresponding saturated case. The stability region follows from the union of link rates that are conditioned on whether the other link is backlogged or not [29,30]. First, we derive the stability region for the given power allocation. Then, we vary power allocation in the feasible region, and by taking the envelope of these regions, we obtain the overall stability region shown in Figure 7. However, different from the previous cases, where the throughput region is the same as the stability region, e.g., in a slotted ALOHA system with two links [29] and in a multiple-access channel [30], the throughput region $\mathcal{F}$ under the SINR model is strictly smaller than the stability region (due to the underlying nonconvex optimization problem), as observed from Figure 7, which is the convex hull of $\mathcal{F}$, i.e., $Co(\mathcal{F})$, achievable by timesharing across different transmission modes.[j]

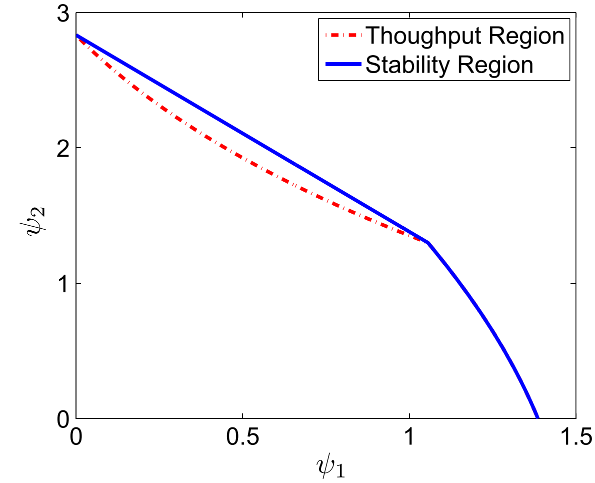

Figure 7 Comparison of the stability region and the throughput region.

Then, we vary the arrival rate λ and the average batch size ν to change the traffic intensity $\psi = \lambda\nu$. Assuming that the arrival rate and the average batch size of each user are the same, we compare in Figure 8 the sample paths of each user's queue length for $\psi = 1$ ($\lambda = 1$, $\nu = 1$) with $\psi = 1.5$ ($\lambda = 1.5$, $\nu = 1$). When $\psi = 1$, which falls in the stability region shown in Figure 7, the system is stabilized by using Algorithm 4. On the other hand, the system becomes unstable when $\psi = 1.5$, which is outside the stability region. Figure 9 illustrates the average delay of the system as a function of the arrival rates. The delay is finite for small loads and grows unbounded when the loads are outside the stability region.

Power control for multicast communications

Due to wireless multicast advantage [23], multicasting enables efficient data delivery to multiple recipients with a single transmission. In this section, we extend the DSPC

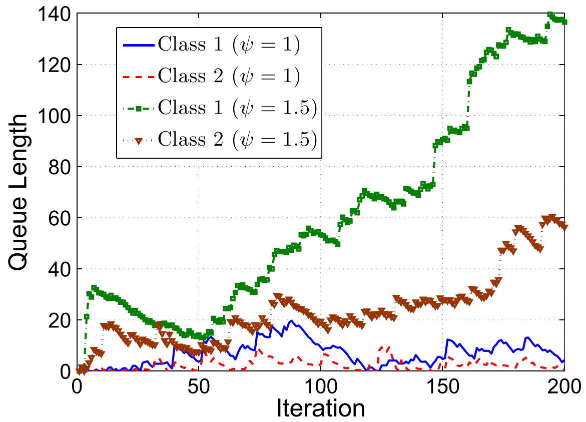

Figure 8 Comparison of sample paths of a user's queue length for different traffic loads.

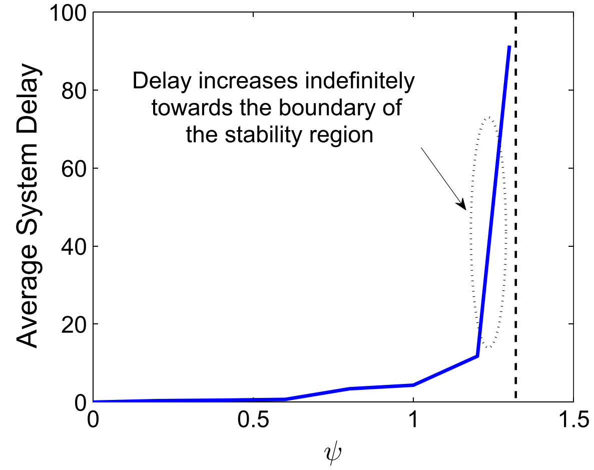

Figure 9 Average delay of the system versus system loads .

algorithms in Section "Power control for unicast communications" to support multicast communications.

System model

Beyond the model described in Section "Power control for unicast communications", we consider that each user l has one transmitter and a set $\mathcal{M}_l$ of receivers. The corresponding transmission rate, r_l, is determined by the bottleneck link among these transmitter–receiver pairs, i.e., $r_l = \min_{m \in \mathcal{M}_l} r_{lm}$, where r_{lm} denotes the link rate between the transmitter of user l and its receiver m, and it is calculated from the Shannon rate $\log(1+\gamma_{lm}(\mathbf{p}))$ for Gaussian, flat fading channels. Here, we do not consider the general broadcast capacity region but rather focus on maximizing the bottleneck link rates.

Network utility maximization

We seek to find the optimal power allocation $\mathbf{p}^*$ that maximizes the overall system utility subject to the power constraints in multicast communications, as follows:

$$\begin{array}{ll} \text{maximize} & \sum_{l\in\mathcal{L}} U_l(r_l) \\ \text{subject to} & r_l = \min_{m\in\mathcal{M}_l} r_{lm}, \forall\, l \in \mathcal{L} \\ & r_{lm} = \log(1+\gamma_{lm}(\mathbf{p})), \forall\, l \in \mathcal{L}, m \in \mathcal{M}_l \\ & 0 \le p_l \le P_l^{\max}, \forall l \in \mathcal{L} \\ \text{variables} & \{\mathbf{p}, \{r_l\}, \{r_{lm}\}\}. \end{array} \tag{12}$$

Similar to (2), (12) is nonconvex due to the complicated interference coupling between individual links. Different from the techniques used in Section "Power control for unicast communications", we relax $r_l = \min_{m\in\mathcal{M}_l} r_{lm}$ in (12) as $r_l \le \log(1+\gamma_{lm}(\mathbf{p})), \forall\, l \in \mathcal{L}, m \in \mathcal{M}_l$, in order to devise distributed algorithms solving (12). Thus, (12) can be rewritten as

$$\begin{array}{ll} \text{maximize} & \sum_{l\in\mathcal{L}} U_l(r_l) \\ \text{subject to} & r_l \le \log(1+\gamma_{lm}(\mathbf{p})), \forall\, l \in \mathcal{L}, m \in \mathcal{M}_l \\ & 0 \le p_l \le P_l^{\max}, \forall l \in \mathcal{L} \\ \text{variables} & \{\mathbf{p}, \mathbf{r}\}. \end{array} \tag{13}$$

Distributed global optimization algorithms

We develop next distributed algorithms that can find the globally optimal solutions to (13) based on EDT and SA. To this end, we first rewrite the optimization problem (13) as

$$\begin{array}{ll} \text{minimize} & -\sum_{l\in\mathcal{L}} U_l(r_l) \\ \text{subject to} & r_l \le \log(1+\gamma_{lm}(\mathbf{p})), \forall\, l \in \mathcal{L}, m \in \mathcal{M}_l \\ & 0 \le p_l \le P_l^{\max}, \forall l \in \mathcal{L} \\ \text{variables} & \{\mathbf{p}, \mathbf{r}\}. \end{array} \tag{14}$$

Next, we use EDT to write the Lagrangian function for (14) as

$$\begin{aligned} L(\mathbf{p}, \mathbf{r}, \{\alpha_{lm}\}) = & -\sum_{l\in\mathcal{L}} U_l(r_l) \\ & + \sum_{l\in\mathcal{L}, m\in\mathcal{M}_l} \alpha_{lm}(r_l - \log(1+\gamma_{lm}(\mathbf{p})))^+, \end{aligned} \tag{15}$$

where $\alpha_{lm} \in \mathbb{R}$ are the penalty multipliers. From EDT, there exist finite $\alpha_{lm}^* \ge 0$ for all $l \in \mathcal{L}, m \in \mathcal{M}_l$ such that, for any $\alpha_{lm} > \alpha_{lm}^*, \forall\, l \in \mathcal{L}, m \in \mathcal{M}_l$, the solution to (15) is the same as (14) [17]. Note that (15) does not include the constraint of p_l for each user. Therefore, there will be no constraint violation when each user updates the transmission power locally, while minimizing (15) in a distributed operation.

As in Section "Power control for unicast communications", the key step is to let each user perform a local stochastic search based on SA. Let r_l and p_l denote the primal values of the lth user, and r_l' denote the new value randomly chosen by the lth user, which is treated as a new target transmission rate for the lth user. Different from the unicast communications case, the lth user updates p_l' by

$$p_l' = \min\left(\frac{e^{r_l'}-1}{\min_{m\in\mathcal{M}_l} \gamma_{lm}} p_l, P_l^{\max}\right), \tag{16}$$

where γ_{lm} is the current SINR measured at the receiver m of user l. Note that (16) does not need any information of the channel gains, except the feedback of SINR

γ_{lm} from the intended receivers. Since (16) is in standard form as described in [27], it converges geometrically fast to the target transmission rate. The steps to update r_l and α_{lm} are similar to DSPC Algorithm 2 in Section "Power control for unicast communications". Note that the target transmission rate r_l may not be feasible, i.e., the target utility cannot be achieved even though the user transmits at the maximum power. In this case, it can be shown that the power of those users with feasible target transmission rates will converge to a feasible solution, whereas the other users that cannot achieve the target transmission rate will continue to transmit at maximum power [1]. A detailed description of DSPC algorithm for multicast communications is presented in Algorithm 5.

Remarks. In Algorithm 5, each user randomly picks $r'_l \in [0, r_l^{\max}]$, where $r_l^{\max} = \min_{m \in \mathcal{M}_l} r_{lm}^{\max}$, and $r_{lm}^{\max}$ is the maximum link rate between the transmitter of the user l and its receiver m, when the user l transmits at the maximum power while the other users do not transmit.

Proposition 4. *The DSPC algorithm for multicast communications (Algorithm 5) converges almost surely to a globally optimal solution to (13), as temperature T in SA approaches zero.*

Proof. The proof is based on EDT and SA arguments, and follows similar steps used in the proof of Proposition 2, and it is omitted here for brevity. □

To improve the convergence rate, we also propose an enhanced algorithm for Algorithm 5 by empirically choosing the initial penalty values and employing a geometric cooling schedule. The resulting algorithm is given in Algorithm 6. Similar to the unicast case, Algorithms 5 and 6 do not need any knowledge of channel information (or the bottleneck link) and they are dynamically updated by the SINR feedback from the intended receivers.

Algorithm 5 DSPC for multicast communications

```
Initialization: Choose $\epsilon > 0$. Let $\alpha_{lm} = 0, \forall\, l \in \mathcal{L}, m \in \mathcal{M}_l$ and randomly choose $\mathbf{r}$ and $\mathbf{p}$.

Step 1: update primal variables
Set $T = T_0$, and select a sequence of time epochs $\{\tau_1, \tau_2, \ldots\}$ in continuous time.
  Repeat for each user $l$
    1. Randomly pick $r'_l \in [0, r_l^{\max}]$, and update $p'_l$ according to (16).
    2. Keep sensing the change of
       $\sum_{m \in \mathcal{M}_l} \alpha_{lm}(r_l - \log(1 + \gamma_{lm}(\mathbf{p})))^+$ broadcast by other users.
    3. Let $\Delta$ be the difference between $L(\mathbf{p}, r_l | \mathbf{r}_{-l}, \{\alpha_{lm}\})$
       and $L(\mathbf{p}', r'_l | \mathbf{r}_{-l}, \{\alpha_{lm}\})$, and accept $r'_l$ and $p'_l$ with
       probability 1, if $\Delta \geq 0$, or with probability $\exp(\frac{\Delta}{T})$,
       otherwise.
    4. Broadcast $U_l(r'_l)$, if $r'_l$ is accepted.
    5. For each time epoch $\tau_i$, update $T = T_0 / \log(i + 1)$.
  Until $T < \epsilon$.

Step 2: update penalty variables
  For each user $l$,
    1. Update $\alpha_{lm} \leftarrow \alpha_{lm} + \varrho_{lm}(r_l - \log(1 + \gamma_{lm}(\mathbf{p})))^+$,
       and scale down $\alpha_{lm}$, if the condition of penalty
       decrease is satisfied.
    2. Go to Step 1 until no constraint is violated.
```

Algorithm 6 EDSPC for multicast communications

```
Initialization: Choose $\epsilon > 0$. Let $\alpha_{lm} = \alpha_{lm}^0, \forall\, l \in \mathcal{L}, m \in \mathcal{M}_l$ and randomly choose $\mathbf{r}$ and $\mathbf{p}$.
Set $T = T_0$, and select a sequence of time epochs $\{\tau_1, \tau_2, \ldots\}$ in continuous time.

Repeat for each user $l$
  1. Randomly pick $r'_l \in [0, r_l^{\max}]$, and update $p'_l$ according to (16).
  2. Keep sensing the change of
     $\sum_{m \in \mathcal{M}_l} \alpha_{lm}(r_l - \log(1 + \gamma_{lm}(\mathbf{p})))^+$ broadcast by other users.
  3. Let $\Delta$ be the difference between $L(\mathbf{p}, r_l | \mathbf{r}_{-l}, \{\alpha_{lm}\})$
     and $L(\mathbf{p}', r'_l | \mathbf{r}_{-l}, \{\alpha_{lm}\})$, and accept $r'_l$ and $p'_l$ with
     probability 1, if $\Delta \geq 0$, or with probability $\exp(\frac{\Delta}{T})$,
     otherwise.
  4. Broadcast $U_l(r'_l)$, if $r'_l$ is accepted.
  5. For each time epoch $\tau_i$, update $T = \xi T$.
Until $T < \epsilon$.
```

Performance evaluation

In this section, we evaluate the performance of Algorithms 5 and 6 for multicast communications. We consider a wireless network with four transmitters and each transmitter has two receivers. These transmitters and receivers are randomly placed on a 10-by-10 m^2 square area. The channel gains h_{lm} are equal to d_{lm}^{-4}, where d_{lm} represents the distance between the transmitter l and the receiver m. The channel gains h_{lm} are equal to d_{lm}^{-4}, where d_{lm} represents the distance between the transmitter l and the receiver m. We assume $U_l(r_l) = r_l$, $P_l^{\max} = 1$, and $n_{lm} = 10^{-4}$ for all $l \in \mathcal{L}$ and $m \in \mathcal{M}_l$. Figure 10 illustrates the fast convergence performance of Algorithms 5 and 6 in multicast communications.[k] Besides, we examine the average performance (with confidence interval) of DSPC and EDSPC for multicast communications under 100 random initializations with the same system parameters as in

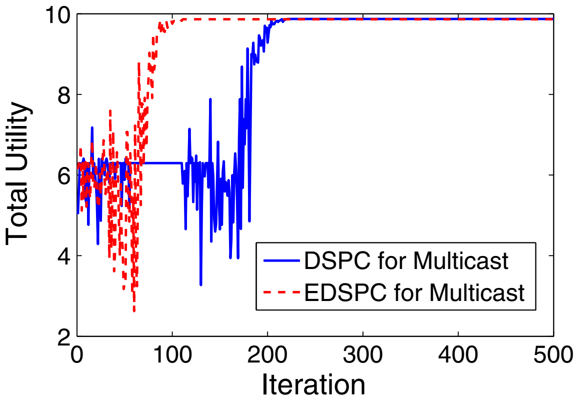

Figure 10 Convergence performance of DSPC and EDSPC for multicast communications, where $\alpha_{lm}^{0} = 20, \forall l \in \mathcal{L}, m \in \mathcal{M}_l$.

the case shown in Figure 10. As illustrated in Figure 11, both Algorithms 5 and 6 are robust against the initial value variations.

Conclusion

We studied the distributed power control problem of optimizing the system utility as a function of the achievable rates in wireless ad hoc networks. Based on the observation that the global optimum lies on the boundary of the feasible region for unicast communications, we focused on the equivalent but more structured problem in the form of maximizing the minimum weighted utility. Appealing to EDT, we decomposed the minimum weighted utility maximization problem into subproblems by using penalty multipliers for constraint violations. We then proposed a DSPC algorithm to seek a globally optimal solution, where each user stochastically announces its target utility to improve the total system utility via SA. In spite of the nonconvexity of the underlying problem, the DSPC algorithm can guarantee global optimality, but only with a slow convergence rate. Therefore, we proposed an EDSPC to improve the convergence rate with geometric cooling schedule in SA. We then compared DSPC and EDSPC with the existing power control algorithms and verified the optimality and complexity reduction.

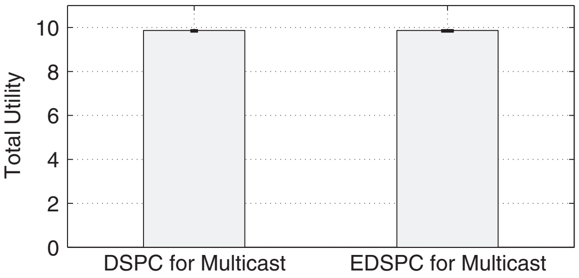

Figure 11 Comparison of average performance (with confidence interval) of DSPC and EDSPC for multicast.

Next, we proposed the joint scheduling and power allocation policy for queueing systems by integrating our distributed power control algorithms with the backpressure algorithm. The stability region was evaluated, which is shown to be strictly greater than the throughput region in the corresponding saturated case. Beyond unicast communications, we generalized our power control algorithms to multicast communications by jointly maximizing the minimum rates on bottleneck links in different multicast groups. Our DSPC approach guarantees global optimality without the need of channel information, while reducing the computation complexity, in general systems with unicast and multicast communications, and applies to both backlogged and random traffic patterns.

Endnotes

[a] We use the terms “user” and “link” interchangeably throughout the article.

[b] We use bold symbols (e.g., **p**) to denote vectors and calligraphic symbols (e.g., $\mathcal{L}$) to denote sets.

[c] The QoS constraint for each link can be incorporated in (2), and the proposed algorithms in the following section can easily be adapted to this case at the cost of added notational complexity.

[d] For some special utility functions $U_l(\cdot)$, (2) can be transformed into a convex problem [3]. In this article, we focus on the nonconvex case that cannot be transformed to a convex problem by change of variables.

[e] The local optimal solution found by ADP matches the globally optimal solution only in one of the cases that are illustrated in Table 2.

[f] By definition, a function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is quasi-convex, if its domain $\mathbf{dom}f$ and all its sublevel sets $\mathcal{S}_c = \{x \in \mathbf{dom}f | f(x) \leq c\}$, for $c \in \mathbb{R}$, are convex [26].

[g] A power control algorithm is of standard form, if the interference function (the effective interference each link must overcome) is positive, monotonic and scalable in power allocation [27].

[h] The geometric cooling schedule is employed to accelerate the convergence rate of DSPC in the simulation. DSPC updates penalty values until they satisfy the threshold-based optimality condition.

[i] Note that the feasible utility region $\mathcal{F}$ defined in Section “Power control for unicast communications” is the throughput region, when the utility function is the same as the rate function.

[j] The transmission mode is defined as the transmission rate pair within the throughput region $\mathcal{F}$.

[k] The other existing algorithms have specifically been designed for unicast communications; therefore, they are excluded here from the performance comparison.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

This study was supported in part by the DoD MURI Project FA9550-09-1-0643 and the AFOSR grants FA9550-10-C-0026 and FA9550-11-C-0006.

Author details

[1]School of ECEE, Arizona State University, Tempe, AZ 85287, USA. [2]Intelligent Automation, Inc., Rockville, MD 20855, USA.

Received: 15 February 2012 Accepted: 4 July 2012
Published: 24 July 2012

References

1. M Chiang, P Hande, T Lan, CW Tan, Power control in wireless cellular networks. Found. Trends Netw. **2**(4), 381–553 (2008)
2. D Julian, M Chiang, D O'Neill, S Boyd, Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks. in *Proc. IEEE INFOCOM,* vol. 2 (New York, USA, 2002), pp. 477–486
3. M Chiang, CW Tan, D Palomar, D O'Neill, D Julian, Power control by geometric programming. IEEE Trans. Wirel. Commun. **1**(7), 2640–2651 (2007)
4. M Xiao, NB Shroff, EKP Chong, A utility-based power control scheme in wireless cellular systems. IEEE/ACM Trans. Netw. **11**(2), 210–221 (2003)
5. Z-Q Luo, S Zhang, Dynamic spectrum management: complexity and duality. IEEE J. Sel. Topics Signal Process. **2**(1), 57–73 (2008)
6. J Huang, R Berry, M Honig, Distributed interference compensation for wireless networks. IEEE J. Sel. Areas Commun. **24**(5), 1074–1084 (2006)
7. P Hande, S Rangan, M Chiang, X Wu, Distributed uplink power control for optimal SIR assignment in celluar data networks. IEEE/ACM Trans. Netw. **16**(6), 1420–1433 (2008)
8. J Papandriopoulos, S Dey, J Evans, Optimal and distributed protocols for cross-layer design of physical and transport layers in MANETs. IEEE/ACM Trans. Netw. **16**(6), 1392–1405 (2008)
9. J Papandriopoulos, JS Evans, SCALE: a low-complexity distributed protocol for spectrum balancing in multiuser DSL networks. IEEE Trans. Inf. Theory. **55**(8), 3711–3724 (2009)
10. CU Saraydar, NB Mandayam, DJ Goodman, Efficient power control via pricing in wireless data networks. IEEE Trans. Commun. **50**(2), 291–303 (2002)
11. T Alpcan, T Basar, R Srikant, E Altman, Wirel. Netw. **8**(6), 659–670 (2002)
12. Y Xu, T Le-Ngoc, S Panigrahi, Global concave minimization for optimal spectrum balancing in multi-user DSL networks. IEEE Trans. Signal Process. **56**(7), 2875–2885 (2008)
13. R Horst, H Tuy, *Global Optimization—Deterministic Approaches* (Springer, New York, 2006)
14. L Qian, YJ Zhang, J Huang, MAPEL: achieving global optimality for a non-convex power control problem. IEEE Trans. Wirel. Commun. **8**(3), 1553–1563 (2009)
15. L Qian, YJ Zhang, M Chiang, Globally optimal distributed power control for nonconcave utility maximization. in *Proc. IEEE GLOBECOM* (Miami, USA, 2010), pp. 1–6
16. S Geman, D Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. **6**(6), 721–741 (1984)
17. Y Chen, M Chen, Extended duality for nonlinear programming. Comput. Optim. Appl. **47**(1), 33–59 (2010)
18. S Kirkpatrick, CD Gelatt, JMP Vecchi, Optimization by simulated annealing. Science. **220**(4598), 671–680 (1983)
19. B Hajek, Cooling schedules for optimal annealing. Math. Oper. Res. **13**(2), 311–329 (1988)
20. L Tassiulas, A Ephremides, Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. IEEE Trans. Autom. Control. **37**(12), 1936–1948 (1992)
21. MJ Neely, E Modiano, CE Rohrs, Dynamic power allocation and routing for time varying wireless networks. IEEE J. Sel. Areas Commun. **23**(1), 89–103 (2005)
22. H-W Lee, E Modiano, LB Le, Distributed throughput maximization in wireless networks via random power allocation. IEEE Trans. Mob. Comput. **11**(4), 577–590 (2011)
23. JE Wieselthier, GD Nguyen, A Ephremides, On construction of energy-efficient broadcast and multicast trees in wireless networks. in *Proc. IEEE INFOCOM,* vol. 2 (Tel Aviv, Israel, 2000), pp. 585–594
24. K Wang, CF Chiasserini, JG Proakis, RR Rao, Joint scheduling and power control supporting multicasting in wireless ad hoc networks. **4**(4), 532–546 (2006). Ad Hoc Netw
25. J Mo, J Walrand, Fair end-to-end window-based congestion control. IEEE/ACM Trans. Netw. **8**(5), 556–567 (2000)
26. S Boyd, L Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, UK, 2004)
27. RD Yates, A framework for uplink power control in cellular radio systems. IEEE J. Sel. Areas Commun. **13**(7), 1341–1347 (1995)
28. X Lin, NB Shroff, R Srikant, On the connection-level stability of congestion-controlled communication networks. IEEE Trans. Inf. Theory. **54**(5), 2317–2338 (2008)
29. R Rao, A Ephremides, On the stability of interacting queues in a multiple-access system. IEEE Trans. Inf. Theory. **34**(5), 918–930 (1988)
30. A ParandehGheibi, M Medard, A Ozdaglar, A Eryilmaz, Information theory vs. queueing theory for resource allocation in multiple access channels. in *Proc. IEEE Pers. Ind. Mob. Radio Commun* (Nice, France, 2008), pp. 1–5
31. R Loynes, The stability of a queue with non-interdependent inter-arrival and service times. Proc. Camb. Philos. Soc. **58**, 497–520 (1962)
32. L Georgiadis, MJ Neely, L Tassiulas, Resource allocation and cross-layer control in wireless networks. Found. Trends Netw. **1**(1), 1–149 (2006)
33. P Chaporkar, S Sarkar, Stable scheduling policies for maximizing throughput in generalized constrained queueing systems. IEEE Trans. Autom. Control. **53**(8), 1913–1931 (2008)
34. Y Yi, J Zhang, M Chiang, Delay and effective throughput of wireless scheduling in heavy traffic regimes: vacation model for complexity. in *Proc. ACM Mobihoc* (New York, USA, 2009), pp. 55–64

doi:10.1186/1687-1499-2012-231
Cite this article as: Yang *et al.*: **Distributed stochastic power control in ad hoc networks: a nonconvex optimization case.** *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:231.