Distributed Stochastic Power Control in Ad-hoc Networks: A Nonconvex Case

Lei Yang, Yalin E. Sagduyu, Junshan Zhang, and Jason H. Li

Abstract—Utility-based power allocation in wireless ad-hoc networks is inherently nonconvex 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, which is utilized as a basis 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. The proposed distributed power control algorithm can guarantee global optimality at the cost of slow convergence due to simulated annealing involved in the global optimization. The geometric cooling scheme and suitable penalty parameters are 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. Finally, we generalize the above distributed power control algorithms to multicast communications, and show their global optimality for multicast traffic.

Index Terms—Distributed Power Control, Nonconvex Optimization, Extended Duality Theory, Simulated Annealing, Queue Stability, Unicast Communications, Multicast Communications.

1 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 [2], [3], [4], [5]. 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 interference [6].

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, [7], [8] devised distributed power control algorithms to find power allocations that 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 non-cooperative game among transmitters [9], [10]. However, distributed solutions that converge to a Nash equilibrium may be suboptimal in terms of maximizing the total system utility. Different from these approaches, [11] proposed a globally optimal power control scheme, named MAPEL, by exploiting the monotonic nature of the optimization problem. However, the complexity and the centralized nature 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, an interesting work [12] has proposed the SEER algorithm based on Gibbs sampling [13], 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.

A challenging task in distributed power control in ad-hoc networks is to reduce the amount of message passing while preserving the global optimality. In this paper, we tackle this challenge by combining recent advances in extended duality theory (EDT) [14] with simulated annealing (SA) [15]. 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 II,
for the subproblem of each individual user, the extended dual can then be solved through stochastic search using 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 problem, although the convergence rate can be slow (but this is clearly due to the slow convergence nature of SA). Then, to improve the convergence rate of DSPC, we propose an enhanced DSPC (EDSPC) algorithm that employs the geometric cooling schedule 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 utility region. The performance gain is further verified by comparing our distributed algorithms with MAPEL [11], SEER [12], and ADP [7] 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 Signal-to-Interference-plus-Noise (SINR) feedback for adaptation.

Next, we integrate the above distributed power control with the back-pressure algorithm [22] and devise a joint scheduling and power allocation policy for improving the 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 without extra technical conditions [23] [25]. Then, we generalize the study to consider multicast communications, where a single transmission may simultaneously deliver packets to multiple recipients [16]. 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 paper is organized as follows. In Section 2, 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, in Section 3, building on these power control algorithms, we develop a joint scheduling and power allocation policy to stabilize queueing systems. The generalization to multicast communications is presented in Section 4. We conclude the paper in Section 5.

Fig. 1: System model.

2 Power Control for Unicast Communications

2.1 System Model

We consider an ad-hoc wireless network with a set \( \mathcal{L} = \{1, ..., L\} \) of links, where the channel is interference-limited, and all \( L \) links treat interference as noise, as illustrated in Fig. 1. Such a model of communication is also applicable to cellular networks [2]. Each link consists of a dedicated transmitter-receiver pair. 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 \( P_{l}^{\text{max}} \) being its maximum power constraint. For static channels, the received SINR for the \( l \)th user with a matched filter receiver is given by

\[
\gamma_l(p) = \frac{h_{ll}p_l}{n_l + \sum_{k \neq l} h_{lk}p_k},
\]

where \( p = (p_1, ..., p_L) \) is a vector of the users’ transmission powers and \( n_l \) is the noise power. Accordingly, the \( l \)th 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, some notation is listed in Table 1.

2.2 Network Utility Maximization

We seek to find the optimal power allocation \( p^* \) that maximizes the overall system utility subject to the individual power constraints, given by the following optimization problem:

\[
\begin{align*}
\text{maximize} & \quad \sum_{l \in \mathcal{L}} U_l(\gamma_l(p)) \\
\text{subject to} & \quad 0 \leq p_l \leq P_{l}^{\text{max}}, \forall l \in \mathcal{L} \\
\text{variables} & \quad \{p\}.
\end{align*}
\]

1. We use the terms “user” and “link” interchangeably throughout.
2. We use bold symbols (e.g., \( p \)) to denote vectors and calligraphic symbols (e.g., \( \mathcal{L} \)) to denote sets.
### 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 |
| \( H \) | link gain matrix |
| \( p_l \) (in vector \( p \)) | transmission power of link \( l \) |
| \( n_l \) (in vector \( n \)) | noise power for link \( l \) |
| \( \gamma_l \) | SINR of link \( l \) |
| \( \gamma_l(\cdot) \) | SINR function of link \( l \) |
| \( \gamma_l(\cdot) \) | utility function of link \( l \) |
| \( x_l \) (in vector \( x \)) | ratio of link \( l \)'s utility to the total network utility |
| \( r_l \) (in vector \( r \)) | transmission rate function of link \( l \) |
| \( \alpha, \beta \) | penalty multipliers |

### 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] | 3.10 | [1, 0.61] | 0.98 |
| ADP      | [20, 6.46] | 3.10 | [0, 2] | 1.22 |
| MAPEL    | [20, 6.79] | 3.10 | [0, 2] | 1.22 |
| SEER     | [20, 6.90] | 3.10 | [0, 2] | 1.22 |

In general, 2 is a nonconvex problem\(^3\). In particular, if the utility function is the Shannon rate achievable over Gaussian flat fading channels, namely \( U_l(\gamma_l(p)) = w_l \log(1 + \gamma_l(p)) \), 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 \(^6\). Note that the weights can serve as the fairness measures \(^1\) for different scenarios. In particular, in queueing systems, for arrival rates within the stability region, packet queues can be stabilized by solving this weighted sum rate maximization problem, when the instantaneous queue lengths are chosen as the weights. In Section 3 we will discuss how to stabilize the packet queues by integrating our distributed power control algorithms with the back-pressure algorithm.

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

**Example:** For the case with two links, Fig. 2 illustrates the nonconvex feasible utility region \( F \) for different system parameters. We compare the performance of the existing approaches 2, 7, 11, 12 in Table 2.

**Remarks:** The solutions to 2 given by 2, 7, 11 are either distributed but suboptimal or optimal but centralized. In particular, 2 solves 2 by using geometric programming (GP) under the high-SINR assumption, which yields a suboptimal solution to 2 when the assumption does not hold (e.g., this is the case in this example above). The ADP algorithm 2 can guarantee only local optimality\(^4\) in a distributed manner. The MAPEL algorithm 11 can achieve the globally optimal solutions but it is centralized with high computational complexity. Compared with these algorithms, the SEER algorithm 12 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 on the control parameter that can be challenging to choose on the fly.

### 2.3 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, and then convert 2 into a more structured max-min problem.

**Lemma 2.1:** The optimal solution to 2 is on the boundary of the feasible utility region \( F \).

3. For some special utility functions \( U_l(\cdot) \), 2 can be transformed into a convex problem \(^4\). In this paper, we focus on the nonconvex case that cannot be transformed to a convex problem by change of variables.

4. The local optimal solution found by ADP happens to be globally optimal only in one of the cases that are illustrated in Table 2.
Proof: Let $U^*$ denote a globally optimal solution to (2) over $F$, and $\gamma^*$ denote the corresponding SINR that supports $U^*$. Since $U_l(\cdot)$ is continuous and strictly increasing, proving that $U^*$ is on the boundary of $F$ is equivalent to showing that $\gamma^*$ is on the boundary of the feasible SINR region. Suppose that $\gamma^*$ is not on the boundary of the feasible SINR region, which indicates that there exists some point $\tilde{\gamma}$ such that $\tilde{\gamma}_l \geq \gamma^*_l$ for all $l \in L$ and $\gamma_l > \gamma^*_l$ for some $l$. Since $U_l(\cdot)$ for any $l \in L$ is strictly increasing in $\gamma_l$, we have $U_l(\tilde{\gamma}_l) \geq U_l(\gamma^*_l)$ for all $l \in L$ and $U_l(\tilde{\gamma}_l) > U_l(\gamma^*_l)$ for some $l$, which contradicts the fact $\gamma^*$ is a globally optimal solution. Hence, Lemma 2.1 follows.

Based on Lemma 2.1 if we can characterize the boundary of $F$, 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 2.2:** Problem (2) is equivalent to the following minimum weighted utility maximization:

\[
\begin{align*}
\text{maximize} & \quad \min_{l \in L} \frac{U_l(\gamma(p))}{x_l} \\
\text{subject to} & \quad 0 \leq p_l \leq P_l^{\max}, \forall l \in L \\
& \quad 0 \leq x_l \leq 1, \forall l \in L \\
& \quad \sum_{l \in L} x_l = 1 \\
\text{variables} & \quad \{p, x\}.
\end{align*}
\]

Proof: Let $t = \sum_{l \in L} U_l(\gamma(p))$ 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, $U_l(\gamma(p)) = tx_l$ and $\sum_{l \in L} x_l = 1$. Then (2) can be rewritten as

\[
\begin{align*}
\text{maximize} & \quad t \\
\text{subject to} & \quad t = \frac{U_l(\gamma(p))}{x_l}, \forall l \in L \\
& \quad 0 \leq p_l \leq P_l^{\max}, \forall l \in L \\
& \quad 0 \leq x_l \leq 1, \forall l \in L \\
& \quad 0 \leq t, \sum_{l \in L} x_l = 1 \\
\text{variables} & \quad \{p, x\}.
\end{align*}
\]

Then, in the context of maximizing $t$, it suffices to relax $t = \frac{U_l(\gamma(p))}{x_l}$ in (3) as $t \leq \frac{U_l(\gamma(p))}{x_l}$, $\forall l \in L$, which is equivalent to $t \leq \min_{l \in L} \frac{U_l(\gamma(p))}{x_l}$. Therefore, (4) can be treated as the hypograph form of (3), i.e., (4) and (3) are equivalent [18], thereby concluding the proof.

For given $x$, (3) is quasi-convex. By introducing an auxiliary variable $t$, we obtain the following equivalent formulation:

\[
\begin{align*}
\text{maximize} & \quad t \\
\text{subject to} & \quad U_l^{-1}(tx_l)(n_l + \sum_{k \neq l} h_{kl}p_k) \leq h_lp_l \\
& \quad 0 \leq p_l \leq P_l^{\max}, \forall l \in L, 0 \leq t \\
\text{variables} & \quad \{p, t\},
\end{align*}
\]

which can be solved in polynomial time through binary search on $t$ [18]. By transforming (2) to this more structured max-min problem (3), we are able to find each boundary point efficiently. Then, the problem is reduced to finding a globally optimal $x^*$, given which we can obtain a globally optimal solution, i.e., the tangent point of the hyperplane $F$, as illustrated in Fig. 3. Intuitively speaking, $x$ represents a search direction. Once we find the best search direction $x^*$, $p^*$ can be obtained efficiently by searching along the direction of $x^*$.

**2.4 Algorithms for Global Optimization**

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 enhanced algorithm EDSPC to improve the convergence rate of DSPC.

1) A Centralized Algorithm

Based on Lemma 2.1 and Lemma 2.2, we develop a centralized algorithm (Algorithm 1) to solve the max-min optimization problem (3) under consideration. Roughly speaking, by dividing the simplex $S = \{x|\sum_{l \in L} x_l = 1, 0 \leq x_l \leq 1, \forall l \in L\}$ into many small simplices, the algorithm can find the optimal point on the boundary of $F$.

**Proposition 2.1:** Algorithm 1 converges monotonically to a globally optimal solution to (3) as the approximation factor $\epsilon$ approaches zero.

Proof: For given $\epsilon$, Algorithm 1 divides the simplex $S = \{x|\sum_{l \in L} x_l = 1, 0 \leq x_l \leq 1, \forall l \in L\}$ into $\lceil 1/\epsilon \rceil$ simplices. Then Algorithm 1 computes the power allocation $p^*$ by solving (5) at $x$ given by the center point of the simplex. Since the optimal search direction $x^*$ is in $S$, when the approximation factor $\epsilon$ approaches
zero, Algorithm 1 exhaustively searches every point in the simplex $S$. Therefore, Algorithm 1 can converge monotonically to a globally optimal solution to (3).

Remarks: In Algorithm 1 by controlling $\epsilon$, one can obtain a solution arbitrarily close to a globally optimal one. Accordingly, Algorithm 1 can guarantee global optimality. However, the complexity of this algorithm can be high and this is possible only with centralized implementation. Algorithm 1 will be used only as a benchmark for performance evaluation of distributed algorithms.

Algorithm 1

Initialization: Choose the approximation factor $\epsilon > 0$, and construct the initial simplex $S$ with the vertex set $V = \{v_1, ..., v_L\}$, where $v_l = e_l$ and $e_l$ is the $l$th unit coordinate vector. Let $v_c = \frac{1}{L} \sum_{l \in L} v_l$ be the center of $S$. Compute $p^*$ by solving (3) at the point $x = v_c$.

Repeat

1) Each simplex is divided into $L$ sub simplices $S_1, ..., S_L$. Let $S$ be a simplex with the vertex set $V = \{v_1, ..., v_L\}$, and let $v \in S \setminus V$. Choose $v = \frac{1}{L} \sum_{l \in L} v_l$. Then, each simplex $S_l$ is defined as having vertex set $V \setminus v \cup \{v\}$.

2) For each new simplex, compute $p^*$ by solving (3) at $x$ given by the center point of the simplex.

3) Find the current best solution to (3).

Until The number of simplices is greater than $1/\epsilon$.

2) DSPC Algorithm

Next, we devise a distributed stochastic power control (DSPC) algorithm based on EDT [14] and SA [15]. 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, define $t_l = \frac{U_l(\gamma_l(p))}{\gamma_l}$ and rewrite (3) as

\[
\begin{align*}
\text{minimize} & \quad -\min_{l \in \mathcal{L}} t_l \\
\text{subject to} & \quad t_l x_l \leq U_l(\gamma_l(p)), \forall l \in \mathcal{L} \\
& \quad \sum_{l \in \mathcal{L}} x_l = 1 \\
& \quad 0 \leq p_l \leq P^\text{max}, \forall l \in \mathcal{L} \\
& \quad 0 \leq t_l, 0 \leq x_l \leq 1, \forall l \in \mathcal{L} \\
\text{variables} & \quad (p, x, t).
\end{align*}
\]

Next, we use EDT to write Lagrangian for (5) as

\[
L(p, x, t, \alpha, \beta) = -\min_{l \in \mathcal{L}} t_l + \alpha \left| \sum_{l \in \mathcal{L}} x_l - 1 \right| + \sum_{l \in \mathcal{L}} \beta_l (t_l x_l - U_l(\gamma_l(p)))^+,
\]

where $(y)^+ = \max(0, y)$, and $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^L$ are the penalty multipliers for penalizing the constraint violations. Based on EDT [14], 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 power constraints, due to the fact that there is no coupling among the user powers. Therefore, the minimization of (7) with respect to the primal variables $(p, x, t)$ 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 $l$th user, and $t'_l$ and $x'_l$ denote the new values randomly chosen by the $l$th user. Accordingly, $t'_l, x'_l$ can be treated as a new target utility for the $l$th user. To achieve this target utility, the $l$th user updates $p'_l$ by

\[
p'_l = \min \left( \frac{U_l^{-1}(U_l(p'_l))}{\gamma_l}, P^\text{max}_l \right),
\]

where $\gamma_l$ is the current SINR measured at the $l$th user's receiver. Note that (5) does not need any information of channel gains except the SINR feedback, i.e., $\gamma_l$. Since (3) corresponds to the distributed power control algorithm of standard form as described in [19], it converges geometrically fast to the target utility. Thus, we assume that each user $l$ updates $p_l$ in a faster timescale than $t_l$ and $x_l$ such that $p_l$ always converges before the next update of $t_l$ and $x_l$. Let $\Delta$ denote the difference between $L(p_l, x_l, t_l ; p_{-l}, x_{-l}, t_{-l}, \alpha, \beta)$ and $L(p'_l, x'_l, t'_l ; p_{-l}', x_{-l}', t_{-l}', \alpha, \beta)$, where $y_{-l}$ is the vector $y$ without the $l$th 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 (it is also called temperature). Note that, as $T$ decreases, the acceptance of uphill move becomes less and less probable, and therefore a fine-grained search takes place. It has been shown that, as $T$ tends to 0 according to a logarithmic cooling schedule, SA converges to a globally optimal point [13, 20]. To compute $\Delta$ locally by each user $l$, we assume that user $l$ broadcasts the terms $t_l, x_l$ and $\beta_l(t_l x_l - U_l(\gamma_l(p)))^+$, whenever any of these terms changes.

Besides updating the primal variables, each user $l$ also needs to update $\alpha$ and $\beta_l$ to satisfy $\alpha > \alpha^*$ and $\beta_l > \beta_l^*$. Here, we apply the method given by [14] to update $\alpha$ and $\beta_l$. In particular, if any constraint is violated, $\alpha$ and $\beta_l$ are updated as follows:

\[
\alpha \leftarrow \alpha + \sigma \left| \sum_{l \in \mathcal{L}} x_l - 1 \right|,
\]

\[
\beta_l \leftarrow \beta_l + \sigma_l (t_l x_l - U_l(\gamma_l(p)))^+,
\]

where $\sigma$ and $\sigma_l$ are used to control the rate of updating $\alpha$ and $\beta_l$. Thus, after initialization, $\alpha$ and $\beta_l$ 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, $\alpha$ and $\beta_l$ are scaled down by multiplying with a random value (it is chosen empirically between 0.7 to 0.95) if the penalty decrease condition is satisfied, i.e., the maximum violation
of constraints is not decreased after running Step 1 in Algorithm 2 several times consecutively, e.g., five times. A detailed description of DSPC algorithm is given in Algorithm 2.

**Proposition 2.2:** The distributed stochastic power control algorithm (Algorithm 2) converges monotonically to a globally optimal solution to (3), as temperature $T$ in SA decreases to zero.

**Proof:** For a given pair of $\alpha$ and $\beta$, Algorithm 2 converges to a globally optimal solution to (7) by using the logarithmic cooling schedule [15], [16]. If the solution satisfies the constraints of (3), it is also a globally optimal solution to (6) based on EDT, i.e., current $\alpha$ and $\beta$ satisfy $\alpha > \alpha^*$ and $\beta_l > \beta_l^*$ for all $l \in \mathcal{L}$ [14]. By iteratively updating $\alpha$ and $\beta$, Algorithm 2 will converge to a globally optimal solution to (3), when $\alpha$ and $\beta$ satisfy $\alpha > \alpha^*$ and $\beta_l > \beta_l^*$ for all $l \in \mathcal{L}$. $\blacksquare$

**Remarks:** The DSPC algorithm can guarantee global optimality in a distributed manner without the need of channel information. In particular, it 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.

### Algorithm 2 Distributed Stochastic Power Control (DSPC)

**Initialization:** Choose $\epsilon > 0$. Let $\alpha = 0$, $\beta_l = 0$, $\forall l \in \mathcal{L}$, and randomly choose $p_l$, $x_l$, and $t_l$.

**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 $t_l'$ and $x_l'$ in the feasible region, and update $p_l'$ according to (8).
2. Keep sensing the change of $\beta_l(t_l x_l - U_l(\gamma_l(p)))^+$ 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 $\tau_l$, 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. Goto Step 1 until no constraint is violated.

3) **Enhanced DSPC Algorithm**

It can be seen from Algorithm 2 that it is critical to find the optimal penalty variables $\alpha$ and $\beta$ for computing $\nabla_f$. Moreover, a logarithmic cooling schedule is used to ensure convergence to a global optimum. To improve the convergence rate, we propose next an enhanced algorithm for DSPC (EDSPC) by empirically choosing the initial penalty values $\alpha_0$ and $\beta_0$, and employing a geometric cooling schedule [15], 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 of 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 $\beta_0 > \beta_l^*, \forall l \in \mathcal{L}$, to satisfy the optimality conditions for penalty variables. Obviously, by choosing large $\alpha_0$ and $\beta_0$, 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.

### Algorithm 3 Enhanced Distributed Stochastic Power Control (EDSPC)

**Initialization:** Choose $\epsilon > 0$. Let $\alpha = \alpha_0$, $\beta_l = \beta_0$, $\forall l \in \mathcal{L}$, and randomly choose $p_l$, $x_l$, and $t_l$.

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 $t_l'$ and $x_l'$ in the feasible region, and update $p_l'$ according to (8).
2. Keep sensing the change of $\beta_l(t_l x_l - U_l(\gamma_l(p)))^+$ 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 $\tau_l$, update $T = T_0 / \log(i + 1)$.

**Until** $T < \epsilon$.

#### 2.5 Numerical Examples

In this section, we evaluate the utility and convergence performance of Algorithms 2 and 3 (DSPC and EDSPC). We consider a wireless network with six links randomly distributed on a 10m-by-10m 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 $U_l(\gamma_l(p)) = \log(1 + \gamma_l(p))$, $P_{\text{max}}^{\text{dB}} = 1$ and $n_l = 10^{-4}$ for all $l \in \mathcal{L}$, and consider one randomly generated realization of channel gains given by

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

9. The geometric cooling schedule is employed to accelerate the convergence rate of DSPC in the simulation. D5SPC updates penalty values until they satisfy the threshold-based optimality condition.
be challenging to choose on the fly. In contrast, DSPC can
of SEER heavily hinges on the control parameter that can
suboptimal solution. As noted in [12], the performance

guarantee local optimality. Therefore, for nonconvex
initializations, with the same system parameters as in
interval) of DSPC, EDSPC, and SEER under 100 random
solution is found for the current penalty values. Fig.
continues updating the penalty values after the optimal
EDSPC algorithm is much faster than DSPC, since DSPC

As shown in Fig. 4, the EDSPC algorithm approaches
as a benchmark to evaluate the optimal performance.

ξ
algorithm converges over time, where we choose all
ADP .

Fig. 4: Convergence performance of DSPC, EDSPC, SEER
and ADP.

Fig. 5: Comparison of the average utility performance
(with confidence interval) of DSPC, EDSPC, SEER and
ADP.

Fig. 4 shows how the total utility in the EDSPC
algorithm converges over time, where we choose all
the initial penalty values equal to 10. Also, we choose
ξ = 0.9, ρ = 1 and q = 1, and use Algorithm 1
as a benchmark to evaluate the optimal performance.
As shown in Fig. 4, 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 after the optimal
solution is found for the current penalty values. Fig.
illustrates the average performance (with confidence
interval) of DSPC, EDSPC, and SEER under 100 random
initializations, with the same system parameters as in
Fig. 4. As shown in Fig. 5, both DSPC and EDSPC are
robust against the initial value variations.

A comparison with the SEER and ADP is also depicted
in Fig. 4 and 5. As mentioned in Section 2, ADP can only

approach the globally optimal solution regardless of the
initial parameter selection, but the convergence rate may
be slower. Further, EDSPC improves the convergence
rate, but the initial penalty values would impact how
close it can approach the optimal point. From the point of
view of reducing the amount of message passing, in
our algorithms each link does not need any knowledge
of 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 used in the SEER algorithm.

3 Joint Scheduling and Power Control for Stability of Queueing Systems

In Section 2 we studied the distributed power allocation,
by using DSPC and EDSPC, for utility maximization
in the saturated case. 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 sta-
bilize packet queues by integrating our power control
algorithms with the celebrated back-pressure algorithm [22].

3.1 Stability Region and Throughput Optimal Power Allocation Policy

Consider the same wireless network model with L links
as in Section 2. We assume that there are S classes of
users in the system, and that the traffic brought by users
of class s follows \( \{A_{st}(t)\}_{t=1}^{\infty} \), which are i.i.d. sequences
of random variables for all \( l = 1, \ldots, L \) and \( s = 1, \ldots, S \),
where \( A_{st}(t) \) denotes the amount of traffic generated by
users of class \( s \) that enters the link \( l \) in slot \( t \). Let \( Q_{T(l)}^s(t) \)
and \( Q_{R(l)}^s(t) \) 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 \( Q_{T(l)}^s(t) \) evolves over
time as

\[
Q_{T(l)}^s(t + 1) = \max(Q_{T(l)}^s(t) - r_l^s(t), 0) + A_{st}(t) + \sum_{m \in \mathcal{L}} r_m(t).
\]

where \( r_l^s(t) \) denotes the transmission rate of link \( l \)
for users of class \( s \). The third term in (10) denotes the traffic
rate from the other links. Assuming that the second moments
of the arrival process \( \{A_{st}(t)\}_{t=1}^{\infty} \) are finite, the queue
length process \( \{Q_{T(l)}^s(t)\}_{t=1}^{\infty} \) forms a Markov chain.

Let \( E_{st} = 1 \) be the indicator that the path of users of
class \( s \) uses link \( l \), and \( E_{st} = 0 \), otherwise. As is standard
[21], [22], [23], the stability region is defined as follows.

Definition 3.1: The stability region \( \Lambda \) 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 sys-
tem is stable, i.e., \( \Lambda = \bigcup_{\mathcal{P} \in \mathcal{P}} \Lambda(\mathcal{P}) \), where \( \Lambda(\mathcal{P}) = \{\psi_s\}_{s=1}^S \) \( \sum_{s=1}^S E_{st}\psi_s < r_l(\mathcal{P}), \forall l \} \) and \( \mathcal{P} \) denotes the set
of feasible power allocation. Here \( \psi_s \) denotes the first
moment of \( \{A_{st}(t)\}_{t=1}^{\infty} \), i.e., the load brought by users
of class \( s \), and \( r_l(\mathcal{P}) \) denotes the rate of link \( l \) under power
allocation \( \mathcal{P} \).
For the sake of comparison, the throughput region$^10$ $\mathcal{F}$ of the corresponding saturated case is defined as the set of all feasible link rates, i.e., $\mathcal{F} = \{ r | r = n(p), p \in \mathcal{P} \}$. In general, the throughput region $\mathcal{F}$ may be different from the stability region $\Lambda$, except for some special cases (e.g., in slotted ALOHA systems the throughput region and the stability region are the same$^{28}$ for two links and in a multiple-access channel the information theoretic capacity region is equivalent to its stability region$^{29}$).

The 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. 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 and the back-pressure algorithm, we propose the following joint scheduling and power allocation policy (presented in Algorithm$^4$) to stabilize the queueing system.

**Proposition 3.1:** The joint scheduling and power allocation policy (Algorithm 4) can stabilize the system when the load $\{ \psi_s \}_{s=1}^S$ is strictly interior to the stability region $\Lambda$, i.e., there exists some $\epsilon > 0$ such that $\{ \psi_s + \epsilon \}_{s=1}^S \in \Lambda$.

The proof is similar to that in$^{23, 24}$, and is omitted for brevity.

Note that Algorithm 4 can be viewed as a single-hop dynamic back-pressure and resource allocation policy$^{24}$, 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 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 $\Lambda$, 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$^{11}$ 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 $\Lambda$.$^{26, 27}$

### 3.2 Numerical Examples

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., $\mathbf{E} = \{ E_{st} \}$ is the identity matrix) with two users (classes), where the channel gains are given by $h_{11} = 0.3$, $h_{12} = 0.5$, $h_{21} = 0.03$, 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 $\lambda_s$, and that the size of file brought by each user follows an exponential distribution.

---

10. Note that the feasible utility region $\mathcal{F}$ defined in Section 3 is the throughput region, when the utility function is the same as the rate function.

---

$^{11}$ The transmission mode is defined as the transmission rate pair within the throughput region $\mathcal{F}$. 

---

| Algorithm 4 Joint Scheduling and Power Allocation Policy |
|----------------------------------------------------------|
| **Stage 1:** For each link $l$, select a link weight according to $w_l(t) = \max_{\{ r_l \}} D_l^*(t)$, where the difference of queue lengths of class $s$ is $D_l^*(t) = \max(Q_L^t(t) - Q_R^t(t))$, if the receiver of link $l$ is not the destination of class $s$'s traffic, and $D_l^*(t) = Q_R^t(t)$, otherwise. |
| **Stage 2:** Compute the optimal power allocation $p^*$ in each slot $t$ by solving the following problem with DSPC algorithm |
| $p^* = \arg \max_{p} \sum_{l=1}^{L} w_l(t) r_l(p)$. |
| The transmission rate of link $l$ in slot $t$ is given by $r_l(p^*) = \log(1 + \gamma_l(p^*))$. |

Then, we vary the arrival rate $\lambda$ and the average file size $\nu$ to change the traffic intensity $\psi = \lambda \nu$. Assuming that the arrival rate and the average file size of each user are the same, we compare the sample paths of each user’s queue length for $\psi = 1$ ($\lambda = 1$, $\nu = 1$) with $\psi = 1.5$ ($\lambda = 1.5$, $\nu = 1$) in Fig. 4. When $\psi = 1$, which falls in the stability region shown in Fig. 5, the system is stabilized by using Algorithm 4, while, when $\psi = 1.5$, which is outside the stability region, the system becomes unstable. Fig. 8 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.

4 POWER CONTROL FOR MULTICAST COMMUNICATIONS

Due to wireless multicast advantage [16], multicasting enables efficient data delivery to multiple recipients with a single transmission. In this section, we extend the distributed stochastic power control algorithms in Section 2 to support multicast communications.

4.1 System Model

Beyond the model described in Section 2, we consider that each user $l$ has one transmitter and a set $M_l$ of receivers. The corresponding transmission rate, $r_l$, is determined by the bottleneck link among these transmitter-receiver pairs, i.e., $r_l = \min_{m \in M_l} r_{lm}$, where $r_{lm}$ denotes the link rate between the transmitter of user $l$ and its receiver $m$, and it is calculated based on the Shannon rate $\log(1 + \gamma_{lm}(p))$ for Gaussian, flat fading channels. Here, we do not consider the general broadcast capacity region but rather focus on maximizing the bottleneck link rates.

4.2 Network Utility Maximization

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

$$\begin{align*}
\text{maximize} & \quad \sum_{l \in \mathcal{L}} U_l(r_l) \\
\text{subject to} & \quad r_l = \min_{m \in M_l} r_{lm}, \forall l \in \mathcal{L} \\
& \quad r_{lm} = \log(1 + \gamma_{lm}(p)), \forall l \in \mathcal{L}, m \in M_l \\
& \quad 0 \leq p_l \leq P_{\text{max}}, \forall l \in \mathcal{L} \\
\text{variables} & \quad \{p, \{r_l\}, \{r_{lm}\}\}.
\end{align*}$$

(12)

Similar to (2), (12) is nonconvex due to the complicated interference coupling between individual links. In order to devise distributed algorithms to solve (12), it suffices to relax $r_l = \min_{m \in M_l} r_{lm}$ in (12) as $r_l \leq \log(1 + \gamma_{lm}(p)), \forall l \in \mathcal{L}, m \in M_l$. Thus, (12) can be rewritten as

$$\begin{align*}
\text{maximize} & \quad \sum_{l \in \mathcal{L}} U_l(r_l) \\
\text{subject to} & \quad r_l \leq \log(1 + \gamma_{lm}(p)), \forall l \in \mathcal{L}, m \in M_l \\
& \quad 0 \leq p_l \leq P_{\text{max}}, \forall l \in \mathcal{L} \\
\text{variables} & \quad \{p, r_l\}.
\end{align*}$$

(13)

4.3 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

$$\begin{align*}
\text{minimize} & \quad -\sum_{l \in \mathcal{L}} U_l(r_l) \\
\text{subject to} & \quad r_l \leq \log(1 + \gamma_{lm}(p)), \forall l \in \mathcal{L}, m \in M_l \\
& \quad 0 \leq p_l \leq P_{\text{max}}, \forall l \in \mathcal{L} \\
\text{variables} & \quad \{p, r_l\}.
\end{align*}$$

(14)
Next, we use EDT to write Lagrangian for (14) as
\[
L(p, r; \{a_{lm}\}) = -\sum_{l \in \mathcal{L}} U_l(r_l) + \sum_{l \in \mathcal{L}, m \in \mathcal{M}_l} a_{lm} (r_l - \log(1 + \gamma_{lm}(p)))^+, \tag{15}
\]
where \(a_{lm} \in \mathbb{R}\) are the penalty multipliers. Based on EDT, there exist finite \(a_{lm} \geq 0\) for all \(l \in \mathcal{L}, m \in \mathcal{M}_l\) such that, for any \(a_{lm} > a_{lm}^*, \forall l \in \mathcal{L}, m \in \mathcal{M}_l\), the solution to (15) is the same as (14) [14]. Since there is no coupling among the power constraints of the individual users, (15) does not include the power constraints. Thus, each user satisfies its own power constraint while minimizing (15) in a distributed operation.

As in Section 2, 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 \(l\)th user, and \(r_l'\) denote the new values randomly chosen by the \(l\)th user, which is treated as a new target transmission rate for the \(l\)th user. Different from the unicast communications case, the \(l\)th user updates \(p_l'\) by
\[
p_l' = \min\left(\frac{r_l' - 1}{\min_{m \in \mathcal{M}_l} \gamma_{lm} p_l}, P_l^{\max}\right), \tag{16}
\]
where \(\gamma_{lm}\) 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 SINR feedback from the intended receivers, i.e., \(r_l\). Since (16) is in standard form as described in [19], it converges geometrically fast to the target utility. The steps to update \(r_l\) and \(a_{lm}\) are similar to DSPC Algorithm 2 in Section 2. A detailed description of DSPC algorithm for multicast communications is presented in Algorithm 5.

**Proposition 4.1.** The distributed stochastic power control algorithm for multicast communications (Algorithm 5) converges to a globally optimal solution to (13), as temperature \(T\) in SA approaches zero.

**Proof.** The proof is based on EDT and SA, and follows similar steps used in the proof of Proposition 2.2 and it is omitted here for brevity. 

Likewise, 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.

### 4.4 Numerical Examples

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 in a 10m-by-10m 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 \(m \in \mathcal{M}_l\). Fig. 9 illustrates the fast convergence performance of Algorithms 5 and 6 in multicast communications.12 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 Fig. 9. As illustrated

**Algorithm 5 DSPC for Multicast Communications**

**Initialization:** Choose \(\epsilon > 0\). Let \(a_{lm} = 0, \forall l \in \mathcal{L}, m \in \mathcal{M}_l\) and randomly choose \(r\) and \(p\).

**Step 1: update primal variables**

1. Randomly pick \(r_l'\) and update \(p_l'\) according to (16).
2. Keep sensing the change of \(\sum_{m \in \mathcal{M}_l} a_{lm} (r_l - \log(1 + \gamma_{lm}(p)))^+\) broadcast by other users.
3. Let \(\Delta\) be the difference between \(L(p, r_l; \{a_{lm}\})\) and \(L(p', r_l; \{a_{lm}\})\), and accept \(r_l'\) and \(p_l'\) with probability 1, if \(\Delta \geq 0\), or with probability \(\exp(\frac{\Delta}{\epsilon})\), otherwise.
4. Broadcast \(U_l(r_l')\), if \(r_l'\) is accepted.
5. For each time epoch \(\tau_l\), update \(T = T_0 / \log(i + 1)\).

**Until** \(T < \epsilon\).

**Step 2: update penalty variables**

For each user \(l\),

1. Update \(a_{lm} \leftarrow a_{lm} + \frac{\partial}{\partial a_{lm}} (r_l - \log(1 + \gamma_{lm}(p)))^+\), and scale down \(a_{lm}\), 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 \(a_{lm} = a_{lm}', \forall l \in \mathcal{L}, m \in \mathcal{M}_l\) and randomly choose \(r\) and \(p\).

**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 \(r_l'\), and update \(p_l'\) according to (16).
2. Keep sensing the change of \(\sum_{m \in \mathcal{M}_l} a_{lm} (r_l - \log(1 + \gamma_{lm}(p)))^+\) broadcast by other users.
3. Let \(\Delta\) be the difference between \(L(p, r_l; \{a_{lm}\})\) and \(L(p', r_l; \{a_{lm}\})\), and accept \(r_l'\) and \(p_l'\) with probability 1, if \(\Delta \geq 0\), or with probability \(\exp(\frac{\Delta}{\epsilon})\), otherwise.
4. Broadcast \(U_l(r_l')\), if \(r_l'\) is accepted.
5. For each time epoch \(\tau_l\), update \(T = \xi T\).

**Until** \(T < \epsilon\).
in Fig. 10 both algorithms 5 and 6 are robust against the initial value variations.

5 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 extended duality theory, we decomposed the minimum weighted utility maximization problem into subproblems by using penalty multipliers for constraint violations. We then proposed a distributed stochastic power control (DSPC) algorithm to seek a globally optimal solution, where each user stochastically announces its target utility to improve the total system utility via simulated annealing. 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 enhanced distributed algorithm (EDSPC) to improve the convergence rate with geometric cooling schedule in simulated annealing. We then compared DSPC and EDSPC with the existing power control algorithms and verified the optimality and complexity reduction.

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 distributed stochastic power control 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.

References

[1] L. Yang, Y. E. Sagduyu, J. Zhang, and J. H. Li, “Distributed power control for ad-hoc communications via stochastic nonconvex utility optimization,” Proc. IEEE ICC, 2011.
[2] M. Chiang, P. Hande, T. Lan, and C. W. Tan, “Power control in wireless cellular networks,” Foundations and Trends in Networking, vol. 2, no. 4, pp. 381-553, 2008.
[3] D. Julian, M. Chiang, D. O. Neill, and S. Boyd, “Qos and fairness constrained convex optimization of resource allocation for wireless cellular and ad hoc networks,” Proc. IEEE INFOCOM, 2002.
[4] M. Chiang, C. W. Tan, D. Palomar, D. O. Neill, and D. Julian, “Power control by geometric programming,” IEEE Trans. Wireless Commun., vol. 1, no. 7, pp. 2640-2651, 2007.
[5] M. Xiao, N. B. Shroff, and E. K. P. Chong, “A utility-based power control scheme in wireless cellular systems,” IEEE/ACM Trans. Networking, vol. 11, no. 2, pp. 210-221, 2003.
[6] 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, 2008.
[7] J. Huang, R. Berry, and M. Honig, “Distributed interference compensation for wireless networks,” IEEE J. Sel. Areas Commun., vol. 24, no. 5, pp. 1074-1084, 2006.
[8] P. Hande, S. Rangan, M. Chiang, and X. Wu, “Distributed uplink power control for optimal SINR assignment in cellular data networks,” IEEE/ACM Trans. Networking, vol. 15, no. 6, pp. 1420-1433, 2008.
[9] I. T. Saraydar, N. B. Mandayam, D. J. Goodman, “Efficient power control via pricing in wireless data networks,” IEEE Trans. Commun., vol. 50, no. 2, pp. 291-303, 2002.
[10] T. Alpcan, T. Basar, R. Srikant, and E. Altman, “CDMA uplink power control as a noncooperative game,” Wireless Networks, vol. 8, no. 6, pp. 659-670, 2002.
[11] L. Qian, Y. J. Zhang, and J. W. Huang, “MAPEL: achieving global optimality for a non-convex power control problem,” IEEE Trans. Wireless Commun., vol. 8, no. 3, pp. 1553-1563, 2009.
[12] L. Qian, Y. J. Zhang, and M. Chiang, “Globally optimal distributed power control for nonconcave utility maximization,” Proc. IEEE GLOBECOM, 2010.
[13] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 6, no. 6, pp. 721-741, 1984.
[14] Y. Chen and M. Chen, “Extended duality for nonlinear programming,” Comput. Optim. Appl., vol. 47, no. 1, pp. 33-59, 2010.
[15] S. Kirkpatrick, C. D. Gelatt, and J. M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671-680, 1983.

[16] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “On construction of energy-efficient broadcast and multicast trees in wireless networks,” *Proc. IEEE INFOCOM*, 2000.

[17] J. Mo and J. Walrand, “Fair end-to-end window-based congestion control,” IEEE/ACM Trans. Netw., vol. 8, no. 5, pp. 556-567, 2000.

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

[19] R. D. Yates, “A framework for uplink power control in cellular radio systems,” *IEEE J. Sel. Areas Commun.*, vol. 13, no. 7, pp. 1341-1347, 1995.

[20] B. Hajek, “Cooling schedules for optimal annealing,” *Math. Oper. Res.*, vol. 13, no. 2, pp. 311-329, 1988.

[21] X. Lin, N. B. Shroff, and R. Srikant, “On the connection-level stability of congestion-controlled communication networks,” *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2317-2336, 2008.

[22] 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, 1992.

[23] M. J. Neely, E. Modiano, and C. E. Rohrs, “Dynamic power allocation and routing for time varying wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89-103, 2005.

[24] L. Georgiadis, M. J. Neely, L. Tassiulas, “Resource allocation and cross-layer control in wireless networks,” *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1-149, 2006.

[25] H.-W. Lee, E. Modiano, and L. B. Le, “Distributed throughput maximization in wireless networks via random power allocation,” *IEEE Trans. Mobile Comput.*, 2011.

[26] P. Chaporkar, S. Sarkar, “Stable scheduling policies for maximizing throughput in generalized constrained queueing systems,” *IEEE Trans. Autom. Control*, vol. 53, no. 8, pp. 1913-1931, 2008.

[27] Y. Yi and J. Zhang and M. Chiang, “Delay and effective throughput of wireless scheduling in heavy traffic regimes: vacation model for complexity,” *Proc. ACM Mobihoc*, 2009.

[28] R. Rao and A. Ephremides, “On the stability of interacting queues in a multiple-access system,” *IEEE Trans. Inform. Theory*, vol. 34, no. 5, pp. 918-930, 1988.

[29] A. ParandehGheibi, M. Medard, A. Ozdaglar, A. Eryilmaz, “Information theory vs. queueing theory for resource allocation in multiple access channels,” *Proc. IEEE Pers. Ind. Mob. Radio Commun.*, 2008.