DISTRIBUTED BACK-PRESSURE POWER CONTROL FOR WIRELESS MULTI-HOP NETWORKS

E. Matskani[*] *N.D. Sidiropoulos*[†] *L. Tassiulas*[‡]

Dept. of ECE
Tech. Univ. of Crete
73100 Chania - Crete, Greece

Dept. of ECE
Univ. of Minnesota
Minneapolis, MN 55455, USA

Dept. of CE & T
Univ. of Thessaly
38221 Volos, Greece

ABSTRACT

A key problem in wireless networking is how to choose a link activation schedule and associated powers in concert with routing decisions to optimize throughput. Back-pressure control policies are optimal in this context, but the underlying power control problem is non-convex. Back-pressure power control (BPPC) was recently shown to be NP-hard, yet amenable to successive convex approximation strategies that deliver manifold improvements in end-to-end throughput relative to the prior art in wireless networking. A drawback is that existing implementations are centralized, whereas practical power control has to be distributed across the network. This paper fills this gap by developing a distributed version of the core step of successive convex approximation of the BPPC problem, building upon the Alternating Direction Method of Multipliers (ADMoM). The resulting protocol enjoys favorable properties relative to dual decomposition - based implementations, and allows tight approximation of the BPPC objective in all interference regimes. Judicious simulations reveal that the proposed algorithm matches the performance of its centralized counterpart, as well as pertinent trade-offs in terms of the design parameters.

Keywords: power control, routing, cross-layer, wireless, multi-hop

1. INTRODUCTION

Back-pressure routing is well-appreciated for its throughput optimality and conceptual simplicity, since its introduction in the early 90's [11, 10]. More recently (e.g., see [5] and references therein) it has attracted considerable interest in the context of cross-layer wireless networking. We consider the back-pressure power control problem for maximal end-to-end throughput in a wireless multi-hop network. At the physical layer, for each scheduling slot, the core back-pressure power control (BPPC) problem amounts to maximizing a differential backlog-weighted sum of link rates. This was recently shown to be NP-hard in [6], which also explored effective successive convex approximation strategies. A drawback is that the solution in [6] is centralized. The contribution of this paper is a distributed implementation of the successive convex approximation approach in [6], using the Alternating Direction Method of Multipliers (ADMoM) [1, 2]. Towards this end, the core step is distributed implementation of the convex lower-bounding approximation of BPPC at any operating point.

Distributed approximation of the weighted sum-rate maximization problem has been considered in [3], in the high-SINR (Signal to Interference plus Noise Ratio) regime and using a dual decomposition approach [3]. Very recently, [8] proposed an iteratively re-weighted Minimum Mean Square Error (MMSE) approach to weighted sum rate maximization for the MIMO interference channel. The algorithm in [8] provides a one-shot approximate solution to the weighted sum rate maximization problem - it cannot be tuned to approximate the problem around different operating points, as needed for successive convex approximation.

Instead of employing dual decomposition as in [3], here we use an Alternating Direction Method of Multipliers (ADMoM) approach, in light of its favorable convergence properties [1, 2]. Further employing a consensus-on-the-min algorithm [4, 9] to reach agreement among links regarding the termination of iterations, we come up with a fully decentralized and promising, performance-wise, algorithm.

2. SYSTEM MODEL AND PROBLEM STATEMENT

We consider a wireless multi-hop network comprising N nodes, modeled by the directed graph $(\mathcal{N}, \mathcal{L})$, where $\mathcal{N} := \{1, \ldots, N\}$ and $\mathcal{L} := \{1, \ldots, L\}$ denote the set of nodes and the set of links, respectively. Each link $\ell \in \mathcal{L}$ corresponds to an ordered pair (i, j), where $i, j \in \mathcal{N}$ and $i \neq j$. By $\text{Tx}(\ell)$ and $\text{Rx}(\ell)$ we denote the transmitter and the receiver of link ℓ, thus, for link $\ell = (i, j)$, $\text{Tx}(\ell) = i$ and $\text{Rx}(\ell) = j$. We assume that any node can transmit data to any other node, and may also split its incoming traffic into multiple outgoing links. Crosstalk factors $G_{k\ell}$ denote the aggregate path loss between the transmitter of link k and the receiver of link ℓ; p_ℓ denotes the power transmitted on link ℓ, V_ℓ the background noise power at the receiver of link ℓ, and G the spreading gain, if any (absorbed in the $G_{k\ell}$'s in the sequel, for brevity). Then, the SINR attained at the receiver of link ℓ is

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

and the link rate is

$$c_\ell = \log(1 + \gamma_\ell).$$

We assume a unit time slotted system, indexed by t, and denote by $D_\ell(t)$ the *differential backlog* of link $\ell \in \mathcal{L}$ at the end of slot t, as defined in [11, 6]. In case of multiple flows, $D_\ell(t)$ is the maximum over all flows traversing link ℓ, i.e., with obvious notation, $D_\ell(t) := max_f D_\ell^{(f)}(t)$. Then, the BPPC problem under per link

[*]E-mail: matskani@telecom.tuc.gr. Co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

[†]Contact author. Tel: +16126251242, Fax: +16126254583, E-mail: nikos@umn.edu

[‡]E-mail: leandros@uth.gr

power constraints is [11, 10, 5, 6]

$$\max_{\{0\le\ p_\ell\le P_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} D_\ell(t)c_\ell.$$

In [6], after lower-bounding link rates at slot t (using an idea from [7]) as $\log(1+\gamma_\ell) \ge \alpha_\ell(t)\log(\gamma_\ell)+\beta_\ell(t)$, and applying a logarithmic change of variables, the following convex approximation of BPPC is proposed (see [6] for details)

$$\max_{\{\tilde{p}_\ell\le\tilde{P}_\ell\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} D_\ell(t)\underline{c}_\ell \tag{1}$$

where

$$\underline{c}_\ell := \alpha_\ell(t)\left(\tilde{G}_{\ell\ell}+\tilde{p}_\ell-\log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{G}_{k\ell}+\tilde{p}_k}+e^{\tilde{V}_\ell}\right)\right)+\beta_\ell(t),$$

$\tilde{p}_\ell := \log p_\ell$, $\tilde{P}_\ell := \log(P_\ell)$, $\tilde{G}_{k_\ell} := \log(G_{k_\ell})$, $\tilde{V}_\ell := \log(V_\ell)$, and the parameters $\alpha_\ell(t)$ and $\beta_\ell(t)$ are chosen to make the lower bound tight at a feasible point obtained in the previous iteration. Our aim here is to solve an arbitrary instance of (1), for any t, $D_\ell(t)$, $\alpha_\ell(t)$, and $\beta_\ell(t)$, in a distributed fashion. In the sequel, we omit the index t from all quantities, for brevity.

3. DISTRIBUTED IMPLEMENTATION

In order to solve (1) in a distributed fashion, we would ideally like each link ℓ to be able to optimize its own variable $\tilde{p}_\ell$, relying on as low-rate feedback as possible from other links. The core problem could then split into L subproblems that could be solved in parallel. Yet, this is not directly possible, due to the coupling of the power variables in the objective of (1). As a first step, we may shift this coupling from the objective to the constraints, by introducing auxiliary variables $\{\tilde{i}_{\ell k}\}_{k\neq\ell}$ to represent the interference that link ℓ receives from link $k\neq\ell$, and equality constraints $\tilde{G}_{k\ell}+\tilde{p}_k=\tilde{i}_{\ell k}, \forall k\neq\ell, \forall\ell\in\mathcal{L}$; e.g., cf. [3, 9]. This yields

$$\min_{\tilde{\boldsymbol{p}},\left\{\tilde{\boldsymbol{i}}_\ell\right\}_{\ell\in\mathcal{L}}} \sum_{\ell=1}^{L} -D_\ell\alpha_\ell\left(\tilde{G}_{\ell\ell}+\tilde{p}_\ell\right)$$

$$+D_\ell\alpha_\ell\log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{i}_{\ell k}}+e^{\tilde{V}_\ell}\right)-D_\ell\beta_\ell \tag{2}$$

$$\text{subject to}\quad \tilde{G}_{k\ell}+\tilde{p}_k=\tilde{i}_{\ell k},\quad \forall k\neq\ell,\quad \forall\ell\in\mathcal{L}, \tag{3}$$

$$\text{subject to}\quad \tilde{p}_\ell\le\tilde{P}_\ell,\quad \ell\in\mathcal{L}, \tag{4}$$

where $\tilde{\boldsymbol{p}}$ denotes the vector of variables $\{\tilde{p}_\ell\}_{\ell\in\mathcal{L}}$, $\tilde{\boldsymbol{i}}_\ell$ the vector of auxiliary variables $\left\{\tilde{i}_{\ell k}\right\}_{k\neq\ell}$ of link ℓ. Note that $\tilde{p}_\ell$ and $\left\{\tilde{i}_{\ell k}\right\}_{k\neq\ell}$ are local variables of link ℓ.

We will use the ADMoM with the dual ascent method [1], in order to solve (2)–(4) in a decentralized fashion. A key step is to exploit the decomposable structure of the augmented Lagrangian corresponding to (2)–(4). For link ℓ, we let $\{\gamma_{\ell k}\}_{k\neq\ell}$ denote the Lagrange multipliers associated with its local equality constraints $\tilde{G}_{k\ell}+\tilde{p}_k=\tilde{i}_{\ell k}, \forall k\neq\ell$, and λ_ℓ the Lagrange multiplier associated with the inequality $\tilde{p}_\ell\le\tilde{P}_\ell$. The augmented Lagrangian with *penalty parameter* ρ, is given by

$$L_\rho=\sum_{\ell=1}^{L}\left(-D_\ell\alpha_\ell\left(\tilde{G}_{\ell\ell}+\tilde{p}_\ell\right)+D_\ell\alpha_\ell\log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{i}_{\ell k}}+e^{\tilde{V}_\ell}\right)\right.$$

$$-D_\ell\beta_\ell+\lambda_\ell\left(\tilde{p}_\ell-\tilde{P}_\ell\right)+\sum_{\substack{k=1\\k\neq\ell}}^{L}\gamma_{k\ell}\tilde{G}_{\ell k}+\tilde{p}_\ell\left(\sum_{\substack{k=1\\k\neq\ell}}^{L}\gamma_{k\ell}\right)$$

$$\left.-\sum_{\substack{k=1\\k\neq\ell}}^{L}\gamma_{\ell k}\tilde{i}_{\ell k}\right)+\frac{\rho}{2}\sum_{\ell=1}^{L}\sum_{\substack{k=1\\k\neq\ell}}^{L}\left(\tilde{G}_{k\ell}+\tilde{p}_k-\tilde{i}_{\ell k}\right)^2. \tag{5}$$

Note that $\tilde{p}_\ell$ and $\left\{\tilde{i}_{\ell k}\right\}_{k\neq\ell}$ are *local primal* variables for link ℓ, while, λ_ℓ and $\{\gamma_{\ell k}\}_{k\neq\ell}$ are its *local dual* variables. Denoting by s the iteration index, and applying ADMoM's optimization steps for the update of variables $\tilde{\boldsymbol{p}}$, $\left\{\tilde{\boldsymbol{i}}_\ell\right\}_{\ell\in\mathcal{L}}$, and $\{\gamma_{\ell k}\}_{\ell\in\mathcal{L},k\neq\ell}$, and a projected gradient step for $\boldsymbol{\lambda}$, the iterates to be carried out at each link $\ell\in\mathcal{L}$ boil down to

$$\tilde{p}_\ell(s) := \arg\min_{\tilde{p}_\ell} -D_\ell\alpha_\ell\tilde{p}_\ell+\lambda_\ell(s-1)\tilde{p}_\ell+\tilde{p}_\ell\left(\sum_{\substack{k=1\\k\neq\ell}}^{L}\gamma_{k\ell}\,(s-1)\right)$$

$$+\frac{\rho}{2}\sum_{\substack{k=1\\k\neq\ell}}^{L}\left(\tilde{G}_{\ell k}+\tilde{p}_\ell-\tilde{i}_{k\ell}(s-1)\right)^2 \tag{6}$$

$$\left\{\tilde{i}_{\ell k}\right\}_{k\neq\ell}(s) := \arg\min_{\left\{\tilde{i}_{\ell k}\right\}_{k\neq\ell}} D_\ell\alpha_\ell\log\left(\sum_{\substack{k=1\\k\neq\ell}}^{L} e^{\tilde{i}_{\ell k}}+e^{\tilde{V}_\ell}\right)$$

$$-\sum_{\substack{k=1\\k\neq\ell}}^{L}\gamma_{\ell k}(s-1)\tilde{i}_{\ell k}+\frac{\rho}{2}\sum_{\substack{k=1\\k\neq\ell}}^{L}\left(\left(\tilde{G}_{k\ell}+\tilde{p}_k\right)(s)-\tilde{i}_{\ell k}\right)^2 \tag{7}$$

$$\gamma_{\ell k}(s) := \gamma_{\ell k}(s-1)+\rho\left(\left(\tilde{G}_{k\ell}+\tilde{p}_k\right)(s)-\tilde{i}_{\ell k}(s)\right), k\neq\ell \tag{8}$$

$$\lambda_\ell(s) := \left[\lambda_\ell(s-1)+\delta_s\left(\tilde{p}_\ell(s)-\tilde{P}_\ell\right)\right]_0^+. \tag{9}$$

Steps (6)–(9) are executed in parallel at all links, as long as certain feedback requirements are satisfied. Step (6) for link ℓ involves $\{\gamma_{k\ell}(s-1)\}_{k\neq\ell}$ and $\left\{\tilde{i}_{k\ell}(s-1)\right\}_{k\neq\ell}$, i.e., dual and auxiliary variables of all its interfering links $k\neq\ell$, as computed in the previous iteration $(s-1)$. This information can be communicated via message passing. Then, (6) is a convex quadratic in $\tilde{p}_\ell$, whose minimum can be found analytically. The unconstrained minimization in (7) with respect to $\tilde{\boldsymbol{i}}_\ell$ is carried out at link ℓ, which is assumed to have knowledge of the interference $\left(\tilde{G}_{k\ell}+\tilde{p}_k\right)(s)$ received from link $k\neq\ell$ at iteration s. Depending on the operational setup, this can either be estimated by ℓ, or communicated to ℓ. Then, variables $\tilde{\boldsymbol{i}}_\ell$ are updated by solving (7) using e.g., damped Newton's method. Next, step (8) is straightforward. Notice that ADMoM requires a step size for (8) equal to the parameter ρ, in order for its convergence properties to hold, along with other assumptions; cf. [1, 2]. The update in (9) is carried out according to the dual ascent method. The associated step-size sequence δ_s can be chosen as $\delta_s=\delta_1/s$, or a sufficiently small constant $\delta_s=\delta$ can be employed - convergence of the iterates to the optimal centralized solution is guaranteed in both cases. We chose a small constant step size in our simulations, because it resulted in faster convergence. The resulting distributed algorithm is summarized as follows

Algorithm 1 Distributed convex approximation of BPPC*:*
Given D_ℓ, α_ℓ, β_ℓ, $\forall\ell\in\mathcal{L}$, and $s :=$ iteration counter,

- ***Initialization:*** *For $s = 0$ set: $\rho > 0$, $\delta_0 > 0$, $\{\lambda_\ell(0)\}_{\ell=1}^{L} > 0$, $\{\gamma_{\ell k}(0)\}_{\ell\in\mathcal{L},k\neq\ell} > 0$, and $\{\tilde{i}_{\ell k}(0)\}_{\ell\in\mathcal{L},k\neq\ell}$ random.*
- *$\forall \ell \in \mathcal{L}$: transmit initial $\gamma_{\ell k}(0)$ and $\tilde{i}_{\ell k}(0)$ to link k, $\forall k \neq \ell$.*
- ***Repeat:*** *Set $s := s + 1$*
  1. *$\forall \ell \in \mathcal{L}$: update $\tilde{p}_\ell(s)$ by solving* (6).
  2. *$\forall \ell \in \mathcal{L}$: update $\{\tilde{i}_{\ell k}(s)\}_{k\neq\ell}$ by solving* (7).
  3. *$\forall \ell \in \mathcal{L}$: update $\{\gamma_{\ell k}(s)\}_{k\neq\ell}$ according to* (8).
  4. *$\forall \ell \in \mathcal{L}$: update $\lambda_\ell(s)$ according to* (9). *Transmit $\gamma_{\ell k}(s)$ and $\tilde{i}_{\ell k}(s)$ to link k, $\forall k \neq \ell$.*
- ***Until:*** *convergence (within ϵ-accuracy); then $\tilde{p}_\ell^{opt} := \tilde{p}_\ell(s)$, $\forall \ell \in \mathcal{L}$.*

Convergence of the algorithm can be determined based on local computation and communication. Each link ℓ keeps track of a local metric, such as the value of its local Lagrange function, and/or the norm of its residual local equality constraint violation vector $\mathbf{r}_\ell(s)$, with elements $r_{\ell k}(s) := \tilde{G}_{k\ell} + \tilde{p}_k(s) - \tilde{i}_{\ell k}(s)$, $\forall k \neq \ell$, evaluated at s. Each link maintains a binary flag taking the value 1 whenever convergence with respect to its local metric has been achieved, within a given accuracy. Then, a distributed consensus-on-the-min algorithm [9, 4] can be employed among links, so that iterates terminate once all links reach convergence.

4. SIMULATION RESULTS

The convergence properties and quality of approximation of the proposed distributed solution relative to the centralized one in [6] are examined in this section through simulations. Due to space limitations, we only report selected indicative results, summarizing our experience from a more comprehensive set of simulations.

We consider the same network as in [6]. There is one flow (source - destination pair) and a total of $N = 6$ nodes, thrown randomly on a 100×100 square; the lower-left node is taken as source, and the upper-right one as destination. There are $L = 25$ possible links, since the destination is a sink and no node transmits to itself. We take direct and crosstalk power losses $\sim \frac{1}{\text{distance}^4}$. If Rx($\ell$) = Tx($k$), then $G_{k,\ell}$ is set to $1/eps$, where eps is machine precision, to model a no-listen-while-you-talk scenario. A spreading gain $G = 128$ is assumed to moderate interference; $V_\ell = 10^{-12}$, and $P_\ell = 5$, $\forall \ell$. We simulate the network for 100 packet/control slots, assuming deterministic arrivals at a rate of 9 packets/slot, under control of the centralized batch high-SINR algorithm in [6], which corresponds to setting $\alpha_\ell(t) = 1$ and $\beta_\ell(t) = 0$, $\forall \ell$ and $\forall t$. The resulting differential backlogs $D_\ell(t), \forall \ell$, at each time slot $t \in \{1, 100\}$, are then used to solve 100 corresponding problem instances via the distributed algorithm.

Parameter choices for the distributed algorithm are as follows. A constant step-size $\delta_s = \delta = 0.01, \forall s$ was used for the gradient step in (9). For initialization, the dual parameters $\{\lambda_\ell\}_{\ell\in\mathcal{L}}$ and $\{\gamma_{\ell k}\}_{k\neq\ell}^{\ell\in\mathcal{L}}$ and the auxiliary variables $\left\{\tilde{\boldsymbol{i}}_\ell\right\}_{\ell\in\mathcal{L}}$ were all set equal to 1. For the termination criterion, each link keeps track of i) the norm of its residual vector and ii) successive differences of the value of its local augmented Lagrangian. Both must drop under $\epsilon = 10^{-2}$ for the protocol to terminate. The reason is that for higher ρ the residual vector is forced quickly to zero, however convergence towards an optimal (vis-a-vis admissible) solution can be slow, because the $\gamma_{\ell k}$ update in (8) is coarse in the earlier stages.

Convergence of ADMoM to a solution of (1) is assured for any ρ, however the choice of ρ affects both the number of iterations required to drop below a given tolerance threshold in terms of the aforementioned convergence metrics, and the value of (1) attained at termination. To appreciate this, we present indicative results for the 30^{th} time slot in Table 1. Solving the problem for various values of ρ, we examine its impact on convergence speed and the finally attained value of (1). Notice that, as ρ is increased from 0.002 to 5, there is an initial decrease in the required number of iterations, reaching a 'sweet spot' at $\rho \sim 0.1$, followed by an increase. This notch-type behavior is typical. Also note that the higher ρ is the (slightly) lower the final objective value of (1), because the quadratic regularizing term is more heavily weighted for a given tolerance ϵ.

Our experience from a rather comprehensive set of simulations is that the distributed algorithm consistently yields essentially the same solution as the centralized one - not only in terms of the set of activated links (whose SINR turns out above machine precision), but also in terms of numerical values of the corresponding SINRs. As an illustration, Table 2 summarizes results for the 30^{th} time slot. Fig. 1 depicts the progress of the sum of augmented Lagrangian functions for all links (left) versus iterations, and the average, over all links, norm of the residual vector $\mathbf{r}_\ell(s)$ (right), for $\rho = 0.01$ (top), and $\rho = 0.1$ (bottom). As can be seen, convergence is achieved even at a moderate number of iterations.

Fig. 2(left) depicts the progress of the objective function in (1) versus iterations, together with the value of (1) attained by the centralized solution, for $\rho = 0.01$ (top), and $\rho = 0.1$ (bottom). For better visualization, the difference of the two curves is plotted in Fig. 2(right), for $\rho = 0.01$ (top), and $\rho = 0.1$ (bottom). Clearly, performance comes closer to the centralized one for $\rho = 0.01$, at the cost of a higher number of iterations till ϵ-convergence (notice the different scaling of the x-axes).

Finally, the value of the objective in (1) obtained from the distributed and the centralized algorithm, for all 100 problem instances, is plotted in Fig. 3(top), for $\rho = 0.01$. For better appreciation of the difference, Fig. 3(bottom) plots the absolute value of the difference of the two curves in 3(top). Respective results for $\rho = 0.1$, are shown in Fig. 4. For the case of $\rho = 0.01$, the average number of iterations was 437, while for $\rho = 0.1$, it was 174.

5. CONCLUSIONS

We have provided the core of an ADMoM-based distributed implementation of the successive convex approximation approach to backpressure power control [6]. Our ADMoM-based solution is not restricted to the high-SINR regime, and is faster and more reliable than dual decomposition. Simulations suggest that the distributed algorithm keeps up with its centralized counterpart; tuning of ϵ and ρ can be used to trade-off solution accuracy for convergence speed, realizing favorable trade-offs. The choice of ρ is particularly important in this context, as there appears to be a sweet spot that minimizes the number of iterations without much loss in terms of accuracy.

In the journal version we will elaborate on feedback requirements relating not only to primal and dual parameter exchanges during iterations, but also to distributed warm start and consensus-on-termination issues, which are very interesting but cannot be included in this conference version due to space limitations.

6. REFERENCES

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, 2nd ed. Belmont, MA: Athena Scientific, 1999.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations & Trends in Machine Learning*, Vol. 3, No 1, pp.1-122, 2011.

Table 1. Objective value and ♯ of iterations, for various ρ, at given tolerance $:= 10^{-2}$. Objective value for Batch high-SINR: 97.6388.

ρ value	0.002	0.003	0.01	0.05
♯ iterations	919	746	184	143
Objective value	97.6519	97.6761	97.6384	97.5964
ρ value	0.1	0.5	1	5
♯ iterations	84	351	566	1088
Objective value	97.5118	97.4771	97.4388	96.9563

Table 2. SINR in dB attained at dominant links, $\rho = 0.01$.

Algorithm	ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5
Centralized	18.23	10.76	11.1	10.78	10.92
Distributed	18.24	10.75	11.05	10.78	10.96

[3] M. Chiang, C. W. Tan, D. P. Palomar, D. O'Neill, and D. Julian, "Power Control By Geometric Programming," *IEEE Trans. on Wireless Communications*, vol. 6, no. 7, pp. 2640-2651, 2007.

[4] J. Cortes, "Distributed algorithms for reaching consensus on general functions," *Automatica*, vol. 44, no. 3, pp. 726-737, 2008.

[5] L. Georgiadis, M. Neely, L. Tassiulas, "Resource Allocation and Cross-Layer Control in Wireless Networks," *Foundations & Trends in Networking*, Vol. 1, pp. 1-147, 2006.

[6] E. Matskani, N. D. Sidiropoulos, L. Tassiulas, "Convex Approximation Algorithms for Back-pressure Power Control of Wireless Multi-hop Networks," in *Proc. ICASSP 2011*, May 22-27, Prague, Czech Republic.

[7] J. Papandriopoulos and J.S. Evans, "SCALE: A Low-Complexity Distributed Protocol for Spectrum Balancing in Multi-User DSL Networks," *IEEE Trans. Inf. Theory*, vol. 55, no.8, pp. 3711-3724, 2009.

[8] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An Iteratively Weighted MMSE Approach to Distributed Sum-Utility Maximization for a MIMO Interfering Broadcast Channel," *IEEE Trans. on Signal Processing*, vol. 59, no. 9, pp. 4331-4340, Sep. 2011.

[9] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in *Ad Hoc* WSNs With Noisy Links- Part I: Distributed Estimation of Deterministic Signals," *IEEE Trans. Signal Processing*, vol. 56, no.1, pp. 350-364, 2008.

[10] L. Tassiulas, "Scheduling and Performance Limits of Networks with Constantly Changing Topology," *IEEE Trans. Inf. Theory*, vol. 43, no. 3, pp. 1067–1073, 1997.

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

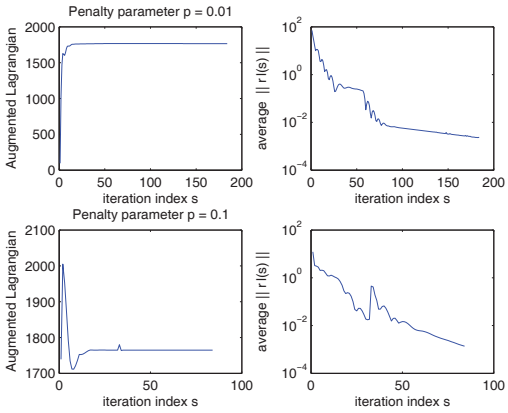

Fig. 1. Augmented Lagrangian and average norm of residual vector versus iterations, for $\rho = 0.01$ (top), and $\rho = 0.1$ (bottom).

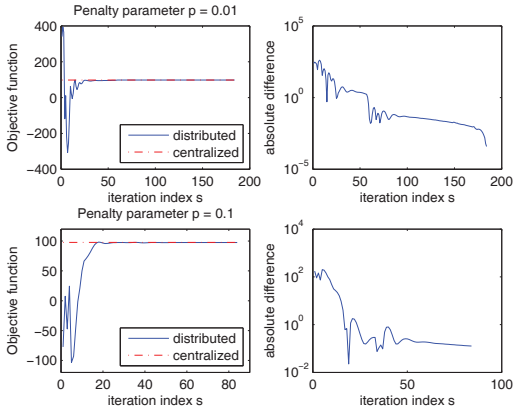

Fig. 2. Objective function (left) and approximation error (right) versus iterations, for $\rho = 0.01$ (top), and $\rho = 0.1$ (bottom).

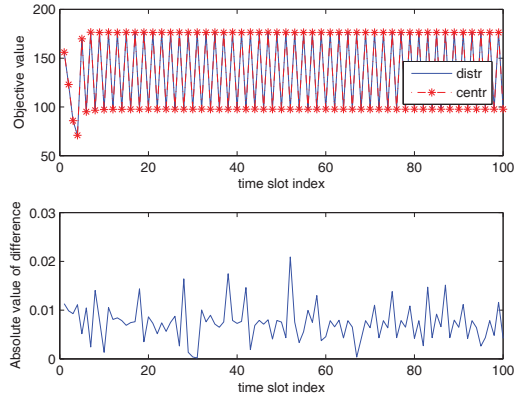

Fig. 3. Comparison between distributed and centralized algorithm: Objective value attained at 100 problem instances (top), absolute value of difference (bottom), for $\rho = 0.01$.

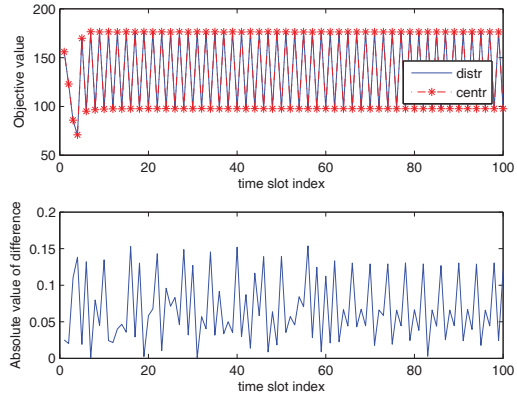

Fig. 4. Comparison between distributed and centralized algorithm: Objective value attained at 100 problem instances (top), absolute value of difference (bottom), for $\rho = 0.1$.