A Non-Convex Distributed Optimization Framework and its Application to Wireless Ad-hoc Networks

Georgios Tychogiorgos, *Student Member, IEEE,* Athanasios Gkelias, *Member, IEEE,*
and Kin K. Leung, *Fellow, IEEE*

***Abstract*—The continuously increasing demand for resources in modern, both wired and wireless, communication networks urges for more efficient resource allocation. Such an allocation of resources to network users can be formulated as an optimization problem. Traditional resource allocation protocols, such as TCP, operate inefficiently in cases that there is competition for resources by multimedia applications and some, or possibly all, links in the network are wireless. In this paper, the performance degradation of TCP in modern networks is quantified to highlight the necessity for a novel optimization-based resource allocation protocol. To this direction, a new optimization framework is presented that can provide the theoretical foundations of such a protocol by proving a sufficient, and in some cases also necessary, condition for distributed solution of non-convex problems. The wide applicability of this general framework is illustrated by considering a resource allocation formulation in TDMA/CDMA ad-hoc networks. The convergence properties to the optimal solution are first identified and a distributed algorithm is proposed. Moreover, a novel heuristic is developed to approximate the optimal solution when the condition does not hold and resolve network oscillations. Finally, the performance of the proposed methodology is evaluated and compared against other approaches in literature by simulation.**

***Index Terms*—Network utility maximization, non-convex optimization, resource allocation, wireless networks.**

I. Introduction

MODERN communication networks must encompass and simultaneously support multiple users, services and applications with diverse demands and requirements that push networks' performance closer to their limit. Therefore, optimum resource allocation between users and/or applications is of paramount importance in order to assure efficient utilization of the network. The *resource allocation* problem is one of the numerous research areas in which *Optimization Theory* has found extensive use, since it can lead to the development of distributed algorithms to assure optimal allocation of resources in a network.

Kelly et al. in their seminal paper [1], and Low et al. [2] later using a different mathematical approach, introduced the *Network Utility Maximization (NUM)* framework, where the *resource allocation* problem is expressed as an optimization problem. Specifically, they formulate the following optimization problem:

$$\textbf{Problem } \mathbf{\Pi_{NUM}}: \quad \begin{array}{ll} \text{maximize} & \sum_{r \in R} U_r\left(x_r\right) \\ \text{subject to} & \boldsymbol{A}x \leq \boldsymbol{C},\ x \geq 0, \end{array}$$

where r, x_r, $U_r\left(x_r\right)$, $\boldsymbol{A}$ and $\boldsymbol{C}$ denote the source node, the data rate of source node r, the utility of node r when transmitting at rate x_r, the routing matrix and the capacity vector respectively. The authors propose an algorithm that determines the optimal way to share the link bandwidths among different traffic flows under the assumption that the utility functions are concave functions of the rate x_r. This algorithm relies on the fact that Problem $\mathbf{\Pi_{NUM}}$ is convex and there is no *duality* gap between the optimal solutions of the primal and dual problems, and manages to solve them both at the same time iteratively.

This convex optimization framework has found numerous applications in network resource allocation in wired networks [3][4]. Lately, there have been some attempts to extend the NUM framework to wireless networks. In [5], TCP is extended so that it can optimize the resource allocation in High-SINR wireless environment. [6] proposes a joint resource allocation and power control problem formulation along with a distributed algorithm to solve it and [7] proves that the NUM formulation with concave utilities for wireless networks has zero duality gap.

The main focus of the above pieces of work are on modeling applications that generate *elastic* traffic [8]. Elastic traffic is the traffic that can adjust to changes in network conditions, such as delay, throughput etc, and still meet the *Quality of Service (QoS)* needs of its applications. These applications include FTP and HTTP, which were the majority of internet traffic until recently. TCP is an example of a protocol designed to perform optimally for this traffic in wired networks. However, modern internet traffic is dominated by real-time applications, such as video and audio streaming, that are considered *inelastic* [8].

The main challenge when attempting to optimize networks shared by inelastic applications is that these applications cannot be modeled using concave utility functions and therefore

Manuscript received May 24, 2012; revised January 13 and May 15, 2013; accepted June 7, 2013. The associate editor coordinating the review of this paper and approving it for publication was T. Hou.

The authors are with the Department of Electrical and Electronic Engineering, Imperial College, London, UK (e-mail: {g.tychogiorgos, a.gkelias, kin.leung}@imperial.ac.uk).

This research was partially sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defense and was accomplished under Agreement Number W911NF-06-3-0001.

The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

Digital Object Identifier 10.1109/TW.2013.072313.120739

the resulting problem turns into a non-convex one, which is difficult to solve. This is because, contrary to what happens in *convex optimization*, the gap between the primal and dual optimal solutions in *non-convex* problems can be positive and then more sophisticated techniques must be employed to solve them [9]. The lack of convexity due to the existence of inelastic traffic in current communication networks, makes TCP operate suboptimally.

Recent work tries to relax the assumption of concave utilities in the context of *NUM*. More specifically, authors in [10] and [11] propose the use of sigmoidal or step functions to model such traffic. In [12] [13], a non-convex NUM formulation in wired networks is examined and the necessary and sufficient conditions are proved so that a gradient-based distributed algorithm can solve this problem formulation optimally. In addition, capacity provisioning is examined as a method of avoiding the appearance of network oscillations. Authors in [14] suggest an approximation of Shannon's capacity formula that takes into account the outage probability of wireless channels and proposes a convexification method that could be applied to some non-concave utilities.

Most of the aforementioned work is restricted only to specific non-concave formulations and do not provide a general optimization framework. **The absence of alternative transport protocols to allow network optimization for inelastic applications is the main motivation behind this work.** This paper makes the following contributions:

- Demonstrates the inability of current resource allocation protocols, such as TCP, to behave optimally in both wired and wireless networks.
- Proposes a non-convex optimization framework that removes the critical assumptions for convexity of the problem formulation and proves a sufficient, and in some cases also necessary, condition so that the framework can solve a non-convex optimization formulation. The significance of this framework is its generality and, therefore, its suitability to a wide range of applications.
- Proposes an efficient resource allocation heuristic to resolve user oscillations that occur when the provided condition does not hold.
- Presents an application of the aforementioned framework in wireless TDMA/CDMA ad-hoc networks. The proposed resource allocation formulation, firstly, incorporates the interference among links, and secondly, introduces a power penalty term in the objective function to ensure convergence and energy efficiency of the power control subproblem.
- Develops a distributed joint rate allocation and power control algorithm, which enables network nodes to optimize their performance, even for the case of inelastic traffic.

The rest of the paper is organized as follows. Section II highlights the shortcomings of the widely used TCP protocol in allocating bandwidth to networks shared by various types of applications. Section III presents the general optimization framework and proves a sufficient condition to assure optimality of the solution. This condition is also shown to also be a necessary condition for some non-convex optimization problems. In Section IV, the framework is applied to the *resource allocation problem* in *wireless* ad-hoc networks and a distributed gradient-based algorithm is proposed. The case of source rate oscillation is discussed and an efficient heuristic is proposed to resolve it efficiently. Then, the performance of the method is evaluated by simulations in Section V, and, finally, Section VI concludes our current work and outlines our future research.

II. TCP in Current Communication Networks

The *Transmission Control Protocol - TCP* [15] is currently the most popular *resource allocation* mechanism. TCP is an end-to-end connection-oriented protocol which relies only on implicit information that is used to estimate the state of the network and adjust the transmission rate of a connection. The *congestion control* in TCP is implemented using a "window", whose size varies based on an implicit measurement of the congestion in the network; the more unacknowledged packets, the more congestion in the network. The size of the window essentially determines the transmission rate of the source with larger window leading to higher bitrate. Over the years, a number of TCP variations have been proposed in order to overcome some of the shortcomings of the initial protocol with TCP Reno [16], TCP Vegas [17] and TCP CUBIC [18] being some of the most popular ones.

TCP was designed based on a set of practical algorithms to adjust the size of the transmission window without any optimization theory considerations. However, recently, Low et al. [19][20] proved that TCP implicitly solves a resource allocation optimization problem and that the various TCP variations differ in the utilities comprising the objective function of the problem. More specifically, TCP Reno solves **Problem $\mathbf{\Pi_{NUM}}$** with utility function $U_r\left(x_r\right) = \frac{1}{D_r}\log\frac{x_r D_r}{2x_r D_r+3}$, where D_r is the round trip delay, and TCP Vegas solves the same problem but with utility function $U_r\left(x_r\right) = \alpha_r d_r \log x_r$, where α_r is a positive calibration parameter and d_r is the round trip propagation delay of source r.

It is evident from the above that the resource allocation mechanism of TCP assigns the same concave utility function to all flows in the network independently of the nature of the application generating the traffic. When TCP was designed, the majority of the traffic over the Internet was elastic but the capacity of current communication networks is mainly used for real-time applications. For example, Cisco [21] [22] foresee that by 2015 62% of the Internet traffic and nearly 66% of the mobile traffic will be video. With such significant amount of traffic generated by inelastic applications, the use of TCP can lead to significantly suboptimal resource allocations in both wired and wireless networks.

An optimization-based algorithm, such as Algorithm 1 presented later in this paper, can allocate the resources of current networks more efficiently. The use of such an algorithm to allocate network resources would have two advantages over TCP. First, each application in the network will be modeled using a different utility function based on the user quality perception for this application. This implies that elastic applications will be modeled using concave utilities and inelastic using non-concave ones.

To illustrate the performance improvement that can be achieved using an optimization-based resource allocation algo-

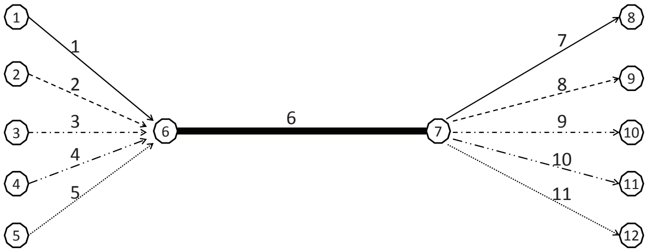

Fig. 1. Example of a single-bottleneck network.

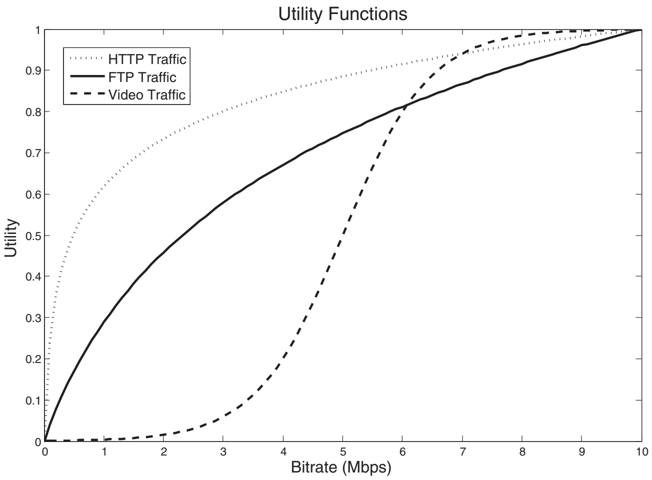

Fig. 2. Example of utility functions.

rithm, consider the single bottleneck wired network topology of Fig. 1, which consists of five traffic flows that share the capacity of link 6. The capacity of links $1-5$ and $7-11$ is assumed to be sufficiently large to serve any transmission rate of source nodes $1-5$ while the capacity of link 6 is assumed to be insufficient to accommodate all flows at their maximum transmission rate, thus creating a bottleneck in the network. For the comparison shown in this section the bottleneck link was set to $28Mb/s$. The applications sharing the network included HTTP, FTP and video streaming. The utilities that were used are shown in Fig. 2. Fig. 3 shows the improvement that can be achieved if the resource allocation is carried out by an optimization-based algorithm as opposed to the congestion control mechanism in TCP. These quantitative results refer to TCP Vegas but similar performance is expected for other TCP variations [19] when applied to networks serving a combination of elastic and inelastic applications. Note that the optimization-based approach described in this paper is not compared against TCP as a complete protocol but only compared against the resource allocation mechanism of TCP, which has be described as an optimization problem in several pieces of work in literature [19][20].

The two methods were compared while the number of real-time applications varied. The x-axis shows the number of real-time applications out of five applications that compete for resources in the network. The rest were either HTTP or FTP applications. For example, the performance comparison for two real-time applications corresponds to a scenario with two sigmoidal utilities, one FTP concave utility and two HTTP concave utilities. The red and black lines, corresponding to the left y-axis, show the total network utility that each method achieved, while the bars, corresponding to the right y-axis, show the percentage of improvement that the optimization-based algorithm achieved.

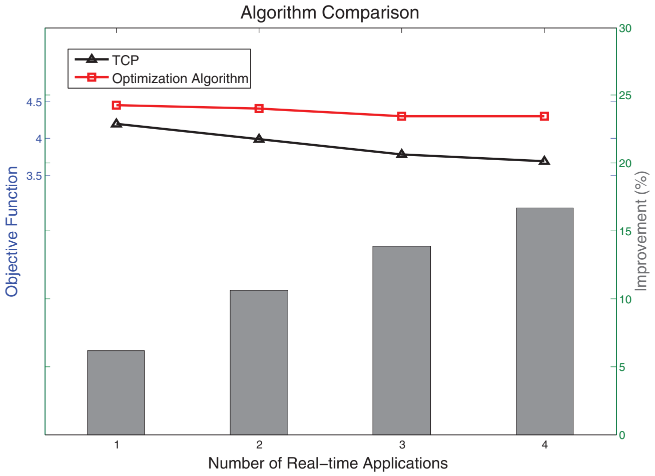

Fig. 3. Improvement of the optimization algorithm over TCP vegas.

It is evident that the more real-time applications share the network, the worse TCP performs by modelling all applications with the same concave utility. On the other hand, an optimization-based algorithm can allocate network bandwidth efficiently since it uses a different utility for each application. Moreover, the improvement in performance can be even larger in networks with a number of wireless links since Algorithm 1 takes into account the interference in order to calculate the link capacities while TCP does not.

Motivated by these results, the next section will focus on the development of an optimization framework that can offer the foundations of future optimization-based resource allocation protocols.

III. An Optimization Framework for Non-convex Problems

The *NUM* framework as presented in [1] and [2] is restricted by the need for concave utilities and the fact that the capacity of all links is fixed. However, as explained above, such assumptions are not valid for the majority of current communication networks. Any prior work that attempts to remove any of them refers to very specific applications, thus lacking generality. This highlights the need for a general non-convex optimization framework that will be able to solve optimization problems resulting from any non-convex network application.

Not all non-convex optimization problems are difficult to solve. In fact, there are cases that can be solved as easy as a convex optimization problem. *Therefore, our main consideration is to develop an optimization framework that can first identify such non-convex problems and then solve them in a distributed way, while being generic enough in order to cover as many applications as possible.* Towards the development of such a framework, first, consider the maximization problem

$\mathbf{\Pi_p}$ over the vector of variables $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]$:

$$\textbf{Problem } \mathbf{\Pi_p}: \quad \max_{\boldsymbol{x}} \quad f(\boldsymbol{x}) \qquad \text{s. t.} \quad h_i(\boldsymbol{x}) \leq 0,\, \boldsymbol{x} \geq \mathbf{0}. \quad \forall\, i$$

To form the *dual* problem, we first define the *Langrangian* function $L(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \sum_{i=0}^{M} \lambda_i h_i(\boldsymbol{x})$, where M is the number of constraints of the optimization problem, λ_i is the dual variable associated with the i^{th} constraint and $\boldsymbol{\lambda}$ is the vector containing all dual variables. According to *Duality Theory*, the *dual* objective function is defined as $d(\boldsymbol{\lambda}) = \sup_x L(\boldsymbol{x}, \boldsymbol{\lambda}) = \sup_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}) + \sum_{i=0}^{M} \lambda_i h_i(\boldsymbol{x}) \right\}$ and the dual optimization problem is:

$$\textbf{Problem } \mathbf{\Pi_d}: \quad \min_{\boldsymbol{\lambda}} \quad d(\boldsymbol{\lambda}) = L(\boldsymbol{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}) \qquad \text{s. t.} \quad \boldsymbol{\lambda} \geq 0,$$

where $\boldsymbol{x}^*(\boldsymbol{\lambda})$ is a function that maximizes the Lagrangian for a given vector $\boldsymbol{\lambda}$, i.e.

$$\boldsymbol{x}^*(\boldsymbol{\lambda}) = \arg\max L(\boldsymbol{x}, \boldsymbol{\lambda}). \tag{1}$$

Each of the dual variables λ_i corresponds to a specific inequality constraint that are often referred as shadow prices. In addition, $\boldsymbol{x}^*(\boldsymbol{\lambda})$ is the optimal solution of problem $\mathbf{\Pi_p}$ for the particular price vector $\boldsymbol{\lambda}$. The dual function $d(\boldsymbol{\lambda})$ is always convex as a point-wise supremum of a family of affine functions of $\boldsymbol{\lambda}$ and problem $\mathbf{\Pi_d}$ is always convex even if the primal problem $\mathbf{\Pi_p}$ is not concave [23]. Therefore, it is possible to solve the dual problem using the iterative equation:

$$\lambda_i(t+1) = \lambda_i(t) - \delta_\lambda \frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda})}{\partial \lambda_i} \tag{2}$$

where δ_λ is the step size and $\frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda})}{\partial \lambda_i}$ is the partial derivative of lagrangian function with respect to λ_i. The uniqueness of the optimal vector $\boldsymbol{\lambda}$ is not guaranteed in all cases but prior work in literature can provide necessary and sufficient condition for its uniqueness [24].

Equations (1) and (2) constitute an iterative primal-dual optimization algorithm which would converge to the optimal solution if problem $\mathbf{\Pi_p}$ had been concave. However, convergence to the optimal is not guaranteed otherwise. Nonetheless, there are non-concave problems where the duality gap is zero and (1) and (2) can converge to the optimal solution. To identify these cases, one can use the sufficient condition of Theorem 1.

Theorem 1 (Sufficient Condition)**.** *If the price based function $\boldsymbol{x}^*(\boldsymbol{\lambda})$ is continuous around at least one of the optimal lagrange multiplier vectors $\boldsymbol{\lambda}^*$ then the iterative algorithm consisting of equations* (1) *and* (2) *converges to the globally optimal solution.*

Proof: See Appendix A. ■

The aforementioned condition is also a necessary condition for convergence of the distributed gradient-based algorithm for some non-convex optimization problems as the following theorem suggests.

Theorem 2. *If at least one constraint of problem $\mathbf{\Pi_p}$ is active at the optimal solution, the condition in Theorem 1 is also a necessary condition.*

Proof: See Appendix B. ■

The condition of theorems 1 and 2 constitutes a significant contribution to optimization theory in general. Compared to other pieces of work, such as [10]-[13], that refer to specific non-convex *NUM* formulations in wired networks, this work provides a far more general optimization formulation and therefore can be widely applicable. The applicability of the framework to a specific problem relies on the continuity properties of the price-based function $\boldsymbol{x}^*(\boldsymbol{\lambda})$. Until now, the convexity of an optimization problem was considered the indication whether a problem can be solved in a distributed way. Problems that did not satisfy the convexity condition were considered difficult to be solved. Therefore, this is a significant result that shows that a family of non-convex problems can be solved distributedly using a gradient based method. Even though the development of a general procedure to determine continuity of $\boldsymbol{x}^*(\boldsymbol{\lambda})$ for any optimization problem is an open research issue, there are cases that either the calculation of a closed form solution is possible or the continuity properties of $\boldsymbol{x}^*(\boldsymbol{\lambda})$ are known.

IV. Resource Allocation in Wireless Ad-hoc Networks

The non-convex optimization framework presented in the previous section can be applied to the resource allocation problem in wireless networks in order to identify and solve non-convex problem formulations that stem from the incorporation of inelastic traffic and the existence of wireless links in the network. The analysis of such a non-convex formulation is the focus of this section.

A. Problem Formulation

Consider a multi-hop wireless network where each node can operate either as traffic source, destination or relay that just forwards traffic to its neighbors. We define the transmission rate vector $\boldsymbol{r} = [r_1, r_2, \ldots, r_M]^T$ which includes the transmission rates of all M source nodes in the wireless network. Moreover, we define the link l as the tuple (T_l, R_l), where T_l is the transmitting and R_l the receiving node, respectively. We also define $\boldsymbol{p} = [p_1, p_2, \ldots, p_L]^T$ as the vector which includes the transmission powers of the L links. The wireless channel is modelled as follows. Let $\boldsymbol{G}$ be a matrix of size $L \times L$, where G_{km}, with $k, m \in 1, 2, \ldots, L$, represents the path loss coefficient for the path between the transmitter of link k and the receiver of link m. The elements of the path loss matrix $\boldsymbol{G}$ depend on the physical characteristics of the wireless links.

The network performance optimization can be formulated as a maximization problem of the form:

$$\textbf{Problem } \mathbf{\Pi_{MWN}}: \max_{\boldsymbol{r}, \boldsymbol{p}} \quad \sum_{i=1}^{M} U_i(r_i) - \gamma \sum_{l=1}^{L} V_l(p_l) \qquad \text{s. t.} \quad \sum_{i=1}^{M} \alpha_{il} r_i \leq C_l(\boldsymbol{p}), \qquad \forall\, l$$

where parameter α_{il} is one if the traffic of user i is passing through link l, and zero otherwise. The parameters α_{il}, with $i \in \{1, 2, \ldots, M\}$ and $l \in \{1, 2, \ldots, L\}$, form the routing matrix $\boldsymbol{A}$ of the network, which is considered to be fixed and known a priori for the duration of the optimization process. The rates r_i and powers p_l are positive quantities and γ is a positive weighting parameter.

In order to account for the half duplex limitations of wireless transceivers and avoid excessive interference, a hybrid TDMA/CDMA scheme is assumed to operate in the network. More specifically, we consider Orthogonal-CDMA (OCDMA) for transmissions towards the same receiver, and pseudo-noise-CDMA (PN-CDMA) between different receivers. This means that the transmitted signal is first spread through multiplication by a Welsh-Hadamard (WH) sequence with N chips per symbol. Then a PN sequence is overlayed either without further spreading (i.e., with the same chip rate) or with further spreading by a factor K (i.e., number of chips per WH chip). All users transmitting towards the same receiver employ the same PN sequence, and N orthogonal sequences are reused at each receiver. Moreover, TDMA is employed throughout the multihop routes. This implies that time is divided into frames, each of them comprises of two equally sized slots, where transceivers alter from transmitting to receiving mode.

Based on this channel model, the capacity of a link follows Shannon's capacity formula, $C_l\left(\boldsymbol{p}\right) = B \cdot \log_2\left(1 + SINR_l\right)$ and is a function of the Signal to Noise plus Interference Ratio (SINR) at the receiver of the link and the channel bandwidth B. This formula is a non-concave function of powers and this might prevent any gradient based algorithm from converging to the optimal power vector. However, under the assumption that $SINR_l \gg 1$, the concave formula $C_l\left(\boldsymbol{p}\right) = B\log_2\left(SINR_l\right)$ can provide a sufficiently accurate approximation of link capacity [25]. Such a *high SINR* environment can be easily achievable for the aforementioned TDMA/CDMA channel model. For the remainder of this paper, the link capacity $C_l\left(\boldsymbol{p}\right)$ will be calculated using this approximation.

The choices for utility $U_i(r_i)$ in problem $\mathbf{\Pi_{MWN}}$ are not restricted to concave functions, as in the traditional NUM framework, so that the problem formulation can be applied to networks with various types of traffic. This makes problem $\mathbf{\Pi_{MWN}}$ non-convex and therefore can be solved in an optimal way using a gradient-based distributed algorithm only if Theorem 1 holds. However, as with most of the relevant work in literature[1], a utility function is not unrestricted, but is assumed to have some generic properties that stem from the physical properties of user satisfaction. These properties include:

- The utility function is an increasing function of rates.
- The utility function is normalized to take values within the range $[0, 1]$.
- The utility function is a continuous function of rate in the range $(0, r^{max})$, where r^{max} is the rate for which the utility function has value 1.

Based on the aforementioned properties of the utility function, at the optimal solution either all the sources will be transmitting at their maximum rate or there will be at least one bottleneck link in the network , which will be restricting transmission rates from being further increased. The former case is beyond the focus of this paper and beyond the focus of the network resource allocation research area in general. In the latter case, at least one constraint will be active, those corresponding to the bottleneck links, and hence Theorem 2 will hold.

Comparing Problem $\mathbf{\Pi_{MWN}}$ with other pieces of work in literature, this formulation extends NUM for *wireless* networks by allowing non-concave utility functions while considering mutual interference among links and by using a power penalty term to ensure energy efficiency and convergence of the distributed power control algorithm.

B. Distributed Algorithm

Problem $\mathbf{\Pi_{MWN}}$ optimizes the allocation of resources in an ad-hoc network and therefore the applicability of any solution relies on the ability to develop a distributed algorithm with minimum message overhead among nodes. *Duality Theory* provides the means to develop such a distributed algorithm, and to this purpose, we first define the *Lagrangian* function as:

$$L(\boldsymbol{r}, \boldsymbol{p}, \boldsymbol{\lambda}) = \sum_{i=1}^{M}\left\{U_i(r_i) - r_i \cdot \left(\sum_{l=1}^{L}\alpha_{il}\lambda_l\right)\right\} - \gamma\sum_{l=1}^{L}V_l(p_l) + \sum_{l=1}^{L}\lambda_l B\log\left(\frac{NKp_lG_{ll}}{\sum\limits_{k\neq l}p_kG_{kl} + n_l}\right). \quad (3)$$

Regarding the physical meaning of the major terms on the lagrangian function, $U_i\left(r_i\right)$ is the "profit" that source i will make for sending its traffic at rate r_i and quantity $r_i \cdot \left(\sum_{l=1}^{L}\alpha_{il}\lambda_l\right)$ represents the total cost for source i in order to send r_i b/s of traffic through the network. Then, term $\sum_{l=1}^{L}\lambda_l B\log\left(\frac{NKp_lG_{ll}}{\sum_{k\neq l}p_kG_{kl}+n_l}\right)$ represents the total "profit" that the links will make by charging each unit of their capacity with λ_l and term $\gamma\sum_{l=1}^{L}V_l(p_l)$ represents the *weighted* cost for the links to achieve a capacity of $B \cdot \log\left(\frac{NKp_lG_{ll}}{\sum_{k\neq l}p_kG_{kl}+n_l}\right)$ for $l = 1, \ldots, L$. After a careful observation of the Lagrangian function, one can see that the optimization process consists of two subproblems of the primal variables $\boldsymbol{r}$ and $\boldsymbol{p}$ coupled by the dual optimization variable vector $\boldsymbol{\lambda}$. The first subproblem is the *rate allocation*, maximizing the net revenue of each *source*, and the second is a *power control* problem, determining the optimal transmission power of the *links*.

Based on the lagrangian function, every source i can calculate its optimal rate $r_i^*\left(\boldsymbol{\lambda}\right)$ using:

$$r_i^*\left(\boldsymbol{\lambda}\right) = \arg\max\left[U_i(r_i) - r_i \cdot \lambda^i\right], \quad (4)$$

where $\lambda^i = \sum_{l=1}^{L}\alpha_{il}\lambda_l$ is the aggregate price for user i and it represents the cost of sending a unit of traffic through the network. There are several methods to solve the optimization problem of (4). First, it is known that the optimal solution will be at the point where the first derivative of the objective function diminishes and therefore

$$r_i^*\left(\boldsymbol{\lambda}\right) = {U'}_i^{-1}\left(\lambda^i\right), \quad (5)$$

where ${U'}_i^{-1}\left(\cdot\right)$ is the inverse function of the first derivative of the utility function. It is evident that (5) can only be

[1] such as [11], [13] and [10]

used if $U'_i(\cdot)$ is an one-to-one function and its inverse can be calculated. In cases that this is not possible, one should use alternative methods, such as the gradient based iterative equation:

$$r_i(t+1) = r_i(t) + \delta_r(t)\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial r_i} \tag{6}$$

where $\delta_r(t)$ is a positive step size and the gradient of the lagrangian function is given by:

$$\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial r_i} = U_i^{'}(r_i) - \sum_{l=1}^{L}\alpha_{il}\lambda_l. \tag{7}$$

The aforementioned equations assume that the Lagrangian function is differentiable with respect to the rate, power and price vectors. Based on the definition of the utility function above, the differentiability of the Lagrangian depends on the differentiability of the constraints and the cost function $V_j(p_j)$. In case they are not differentiable, one could use a similar algorithm based on the sub-gradient method [9][3]. However, since the focus of this paper is the more often case of differentiable Lagrangian function, for the remainder of this work, the constraints are assumed to be differentiable with respect to the optimization variables and the cost function is assumed to be a continuous and convex function of the transmission powers. In general, iterative gradient-based equations such as (6) should be used with care as they can converge to local optima instead of global. However, knowledge of the shape of the optimal rate allocation function can be used in some cases, such as in the case of Problem $\mathbf{\Pi_{MWN}}$ to assure that (6) will converge to the globally optimal solution. Nonetheless, the distributed Algorithm 1 uses the general equation (4) to allow the implementation of the most appropriate method for $r_i^*(\boldsymbol{\lambda})$.

A similar approach can be used to calculate the power and price variables, p_l and λ_l respectively:

$$\lambda_l(t) = \lambda_l(t-1) - \delta_\lambda(t)\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial \lambda_l} \tag{8}$$

$$p_l(t) = p_l(t-1) + \delta_p(t)\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial p_l}, \tag{9}$$

where $\delta_\lambda(t)$ and $\delta_p(t)$ are small positive step sizes and the gradients are given by:

$$\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial \lambda_l} = B\cdot\log_2\left(\frac{NKp_lG_{ll}}{\sum\limits_{k\neq l}p_kG_{kl}+n_l}\right) - \sum_{i=1}^{M}\alpha_{il}r_i \tag{10}$$

$$\begin{aligned}\frac{\partial L(\boldsymbol{r},\boldsymbol{p},\boldsymbol{\lambda})}{\partial p_l} =& \frac{1}{p_l\ln(2)}\left[\lambda_l - \sum_{m\neq l}\lambda_m\frac{G_{lm}P_l}{\sum_{k\neq m}G_{km}P_k+n_m}\right]\\ &-\gamma V_l^{'}(p_l).\end{aligned} \tag{11}$$

As explained analytically in [9][23], the step size has a dominant effect in the complexity of the method as opposed to the network size, which has a smaller (in fact linear) effect on the complexity of the method [26]. There are several step size selection methods for either constant or variable step sizes. Methods with variable step size, such as Newton Method, can calculate in practice the theoretical optimal solution as opposed to constant step size methods, where there is always a (small) distance between the theoretical and practical optimal solution. However, variable step size methods are computationally much more demanding (matrix inverse calculations are included) which can make their use impractical in some applications. On the other hand, constant step sizes are less computationally demanding but there is a trade-off between the convergence rate and the approximation error. A comparison of the various step size methods is beyond the scope of this paper and therefore a constant step size method was used during our simulation results for simplicity. Other methods can be also used depending on the requirements of each individual application of the framework. Regarding the complexity of the algorithm, it is evident by examining (4), (8) and (9) that the rates, link prices and transmission powers respectively are calculated using simple mathematical operations. This fact along with the selection of the appropriate step size to achieve fast convergence can make the proposed iterative algorithm a method of relatively low complexity.

Equations (4), (8) and (9) constitute an iterative distributed algorithm, which is summarized in Algorithm 1. The calculation of optimal rate at each iteration can be implemented with either (5), if the inverse utility derivative can be calculated, or using (6) and taking into account that the optimal solution is always in the concave region of the utility function and thus restricting the rate values within this interval. At every iteration, each link and each source node are updating their power, price and rate according to the feedback they get from the network. Regarding the stopping criterion of the algorithm, one could stop the optimization process when all derivatives have diminished or when the value of the objective function has not changed significantly for a number of consecutive iterations [9]. In any case, the values of the step sizes δ_λ and δ_p constitute an important trade-off between convergence speed and accuracy. The initial vectors of $\boldsymbol{r}$, $\boldsymbol{\lambda}$ and $\boldsymbol{p}$ for $t=0$ can be set to any feasible value.

Regarding the information exchange of the algorithm, users need to know the aggregate link price λ^i along the path that they are using but not each individual link price. This can be either stored in the ACK packets sent by the destination to the source node, or, if the link price is viewed as the link delay in a manner similar to TCP [19][20], it can be implicitly measured by the packet queuing delay in the network. Regarding the calculation of the transmission power at the next iteration, it is indeed necessary to know the channel condition and prices at the receivers in the network. Although equation (11) indicates that the interference (i.e., transmitting power and path loss) from all network terminals should be considered, in practical implementations the interference from terminals more than one hop away is negligible and can be omitted from the calculation. In other words, one can define an interference threshold below which the interference caused by a transmitter is considered negligible and is not taken into account in the calculation of the gradient of the Lagrangian function. Therefore the power calculation process requires that a terminal knows only the channel conditions of its neighbouring nodes which can be easily obtained from the lower layers of the protocol stack.

C. Convergence and Oscillation Resolving Heuristic

The cost function $V_l\left(p_l\right)$ assures that the optimization problem will have a finite optimal power vector. In the absence of this cost function, i.e. when $\gamma = 0$, it is possible to fall in a situation where equation (11) is always positive, leading to infinite powers. On the other hand, when $\gamma > 0$, there will be a finite power vector $\boldsymbol{p'}$ at which any further increase would lead to a decrease in the network utility and thus the algorithm will converge to a finite power value. In literature, this case of infinitely increasing power is often prevented by assuming a maximum transmission power value p_l^{max}. Such an assumption, even though is reasonable in a practical system, causes distortion in the theoretical analysis since it creates artificial convergence points. Specifically, according to the *Brouwer Fixed Point Theorem* [27], a continuous mapping of the power vector within a closed range $\left[p_l^{min}, p_l^{max}\right]$ creates fixed points of an algorithm that might otherwise never converge. Therefore, the use of the penalty function $V_l\left(p_l\right)$ is a more natural way of assuring both energy efficiency and convergence of the distributed power control algorithm.

Algorithm 1 is an extension of the standard gradient algorithm to solve any convex optimization problem and whose convergence properties have been extensively studied in prior work [9]. According to Theorem 1, the convergence of the algorithm to the optimal solution relies on the continuity of (4) around at least one of the optimal price vectors $\boldsymbol{\lambda}^*$. The continuity properties of (4) rely on the shape of the utility function $U_i\left(r_i\right)$. More specifically, if $U_i\left(r_i\right)$ has a concave shape, i.e. it is modelling an elastic application, (4) is a continuous function of the aggregate price λ^i. If, however, $U_i\left(r_i\right)$ models an *inelastic* application, (4) can be discontinuous at one or more points and user oscillations can occur when the optimal price vector λ^i leads to an aggregate price (for that specific user) equal to a discontinuity point. While a generic procedure to determine the continuity properties of $r_i\left(\lambda^i\right)$ for any utility function is an open research problem, these properties have been extensively studied for single-sigmoidal utilities [10]. For such utility function, $r_i^*\left(\boldsymbol{\lambda}\right)$ is discontinuous at only one point, which represents the user's maximum willingness to pay, λ_{max}^i, and there is an analytical methodology to be calculated. For example, the optimal rate for a given price vector can be calculated using (6) due to the fact that the optimal solution can only exist in the concave region of the utility function.

Fig. 4 shows an example of the rate allocation function $r_i\left(\lambda^i\right)$ for a single-sigmoidal utility which is discontinuous for $\lambda^i = \lambda_{max}^i = 0.7385$. In the remainder of this paper, we will assume that inelastic applications will be modelled by single-sigmoidal utility shapes, such as the one in Fig. 2, which is the most widely used shape to model real-time multimedia applications. The use of single-sigmoidal utilities allows us also to compare our approach with other approaches in literature and highlight the wide applicability of our framework. The examination of formulations with different non-concave utility shapes (such as multi-sigmoidal utilities) is one of our future research targets. Moreover, we assume that the cost function $V_l\left(p_l\right)$ is a convex function of the transmission powers. This assumption ensures that the power control sub-problem is convex and the iterative algorithm will converge to the optimal power vector. The investigation of other shapes for the utility and cost functions, and the consideration of their effect on the development of more energy efficient distributed algorithms will be the focus of our future research.

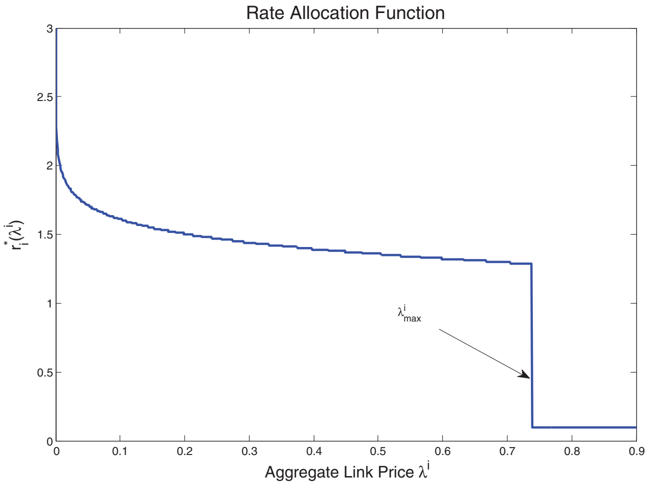

Fig. 4. Example of the rate allocation function $r_i\left(\lambda^i\right)$ for sigmoidal utilities.

The phenomenon of oscillation occurs when the optimal rate function $r_i^*\left(\boldsymbol{\lambda}\right)$ of a specific user i is a discontinuous function of the aggregate price and the optimal price vector λ^i leads to an aggregate price (for that specific user) equal to the discontinuity point. As explained earlier, the existence of discontinuity points in $r_i^*\left(\boldsymbol{\lambda}\right)$ depends only on the shape of the utility function. Specifically, for sigmoidal utilities, $r_i^*\left(\boldsymbol{\lambda}\right)$ is discontinuous only for aggregate price $\lambda^i = \lambda_{max}^i$ and when the optimal price vector leads to that aggregate price, the rate of user i oscillates and the distributed algorithm can not converge.

Practically, a user oscillation occurs when the user transmits at an excessive data rate (compared to the available capacity) in an iteration of the algorithm, and in the next iteration, the user transmits at an exceedingly low rate. An oscillation is formed as the repetition of these two events continues indefinitely, which prevents the user from converging to the optimal transmission rate. In this case, user i needs to resolve this oscillation and approximate the optimal solution. To this purpose, Algorithm 2 describes an efficient heuristic that ensures convergence to the optimal solution, when users do not oscillate, and stability when one or more users oscillate. Note that Algorithm 2 is carried out distributedly by each source node to determine the most appropriate rate at time t after an updated aggregate price is received and, in essence, replaces the initial rate update mechanism in line 6 of Algorithm 1.

Algorithm 2 is based on the idea that an oscillating user will be allocated some rate, and will be removed from the rest of the optimization process to allow stability of the network. User oscillations indicate that the optimal rate allocation is non-zero, but due to the discontinuity at λ_s^{max}, the optimal rate can not be calculated. More specifically, user i is associated with a parameter θ_i, the maximum number of consecutive oscillations before the oscillation resolving mechanism is evoked (line 2). As long as an oscillation is not detected, user i calculates its

Algorithm 1 – Iterative Distributed Algorithm

```
1: $t = 1$;
2: repeat
3:     Links calculate $p_j^t$ using (9) based on $\boldsymbol{\lambda}^{t-1}$ and channel state information;
4:     Links calculate $\lambda_j^t$ using (8), based on $\lambda_j^{t-1}$, $p_j^t$, channel state information and aggregate rate traversing link $j$;
5:     New prices $\boldsymbol{\lambda}$ are sent to sources;
6:     Sources calculate $r_i^t$ using (4) based on received $\lambda^i$ and $r_i^{t-1}$;
7:     $t = t + 1$;
8: until $algorithm\_converges$
```

Algorithm 2 – Source Rate Calculation

```
1: if $user\_takes\_part\_in\_optimization$ then
2:     if $user\_is\_oscillating\,(\boldsymbol{r}, \theta_i)$ then
3:         $r_i^t = inflection\_point\_rate$;
4:         $user\_takes\_part\_in\_optimization =$ false;
5:     else
6:         Calculate $r_i^t$ using (4);
7:     end if
8: else
9:     $r_i^t = r_i^{t-1}$;
10: end if
```

rate based on the aggregate price (line 6). Once oscillations are detected, user i starts transmitting at rate equal to the inflection point of its sigmoidal utility and leaves the optimization process (lines 2-5 and 9). The reason behind this selection is the following. The cause of oscillation is that the optimal rate is between the inflection point of the utility function and the rate that corresponds to aggregate price λ_s^{max}. However, due to the fact that the rate allocation function is discontinuous between these two values, the distributed algorithm cannot converge. Therefore, a sensible choice of rate for the heuristic is the minimum of the two values, which is the infection point of the utility function.

Removing oscillating users from the optimization process is an obvious decision to ensure stability of the network but the question lies in the allocated rate to oscillating users. An alternative choice for resolving oscillations is to not allocate any rate to this user. Authors in [10] attempt to solve the oscillation problem by removing them without allocating any rate. However, such approaches lack fairness because they unnecessarily prevent some users from accessing the network resources. Algorithm 2 has the following advantages against this approach. The self-regulating heuristic has been proven to be optimal for wired networks with infinite number of users/data sources. If the number of users is finite though, by completely removing an oscillating user from the optimization problem, there is a non-zero probability that the remaining users will not be able to exploit the remaining available resources and therefore the resulting resource allocation is significantly suboptimal. The oscillation resolving heuristic presented in this paper can accommodate more users since it allocates some rate even to oscillating ones. In addition, allowing more users to transmit in a high SINR environment makes a better use of the capacity of the wireless medium and ultimately leads to higher aggregate utility in the network for practical applications. This will be shown by an example in Section V.

V. Numerical Results

Algorithms 1 and 2 were simulated in MATLAB for various network scenarios. For illustration purposes, in this section let us consider the network topology shown in Fig. 5. The wireless network consists of four source nodes, four intermediate nodes and one destination node. Source nodes 1 and 4 serve real-time applications with sigmoidal utilities while source nodes 2 and 3 serve HTTP applications with concave utilities.

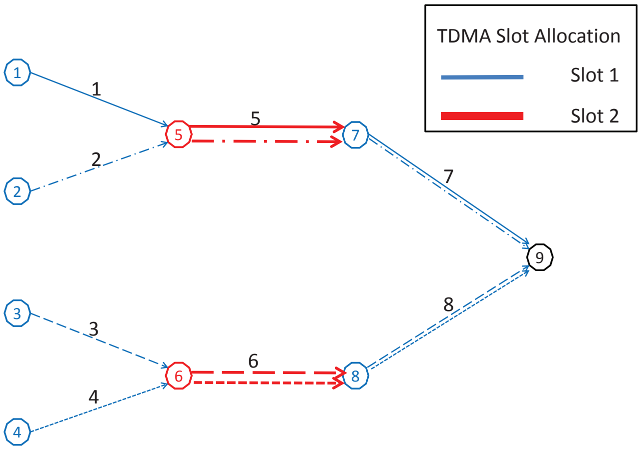

Fig. 5. Example network topology.

In the topology example of Fig. 5, the two time slots are designated with blue and red color. In other words, nodes $1-4$, 7 and 8 transmit only at the first time slot while nodes 5 and 6 only during time slot 2. The hybrid TDMA/CDMA scheme described in Section IV was deployed with $N = 2$ chips per symbol, a spreading gain $K = 4$ and channel bandwidth of $B = 2$MHz. Finally, the utility functions of the four sources where defined as $U_i\,(r_i) = \frac{1}{1+\exp(\alpha(r_i-\beta))}$ [10], with $\alpha = -1.38$ and $\beta = 5$ for $i \in \{1, 4\}$, $U_2\,(r_2) = \frac{\log(r_2+1)}{\log(\alpha+1)}$ [28], with $\alpha = 6$, and $U_3\,(r_3) = \alpha \cdot log(\beta \cdot r_2 + \gamma)$ [10], with $\alpha = 0.417$, $\beta = 0.417$ and $\gamma = 1$. Then, the exact value of the step sizes in our simulations varied between 0.001 and 0.01 in order to achieve an acceptable balance between approximation error and convergence speed. Regarding the feasible power vectors, it is assumed that there is a feasible power vector to achieve capacity adequate to accommodate the non-concave utilities when transmitting at rate equal to their inflection point.

The performance of Algorithms 1 and 2 is compared against that of the standard gradient algorithm when the self-regulating heuristic [10] is applied to resolve oscillations. Figs. 6, 7 and 8 illustrate their performance. Soon after the optimization process starts, the aggregate price for users 1 and 4 exceeds their maximum "willingness to pay" and they start oscillating. As shown clearly in Fig. 6, the rate oscillation of users 1 and 4 cause oscillations of smaller degree to other users

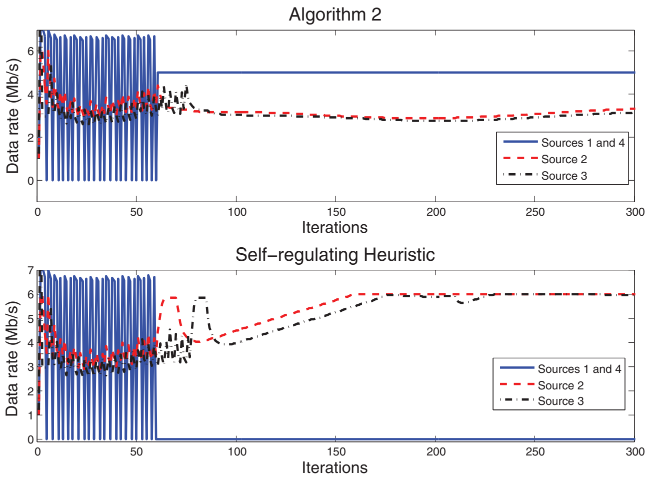

Fig. 6. Convergence of rate allocation.

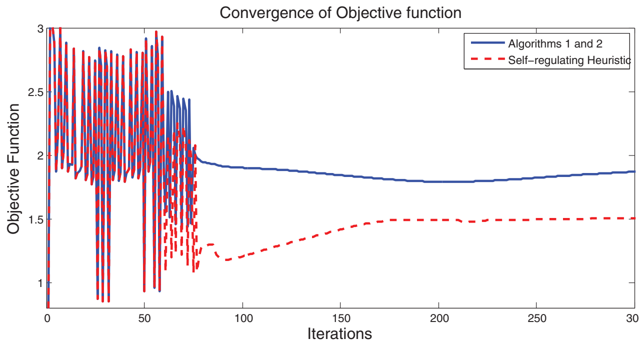

Fig. 7. Convergence of objective function.

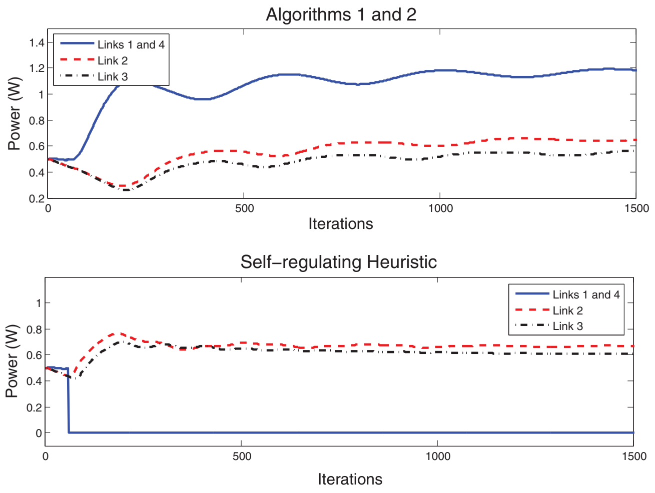

Fig. 8. Convergence of transmission power allocation.

as well. This happens since oscillations cause abrupt changes in the competition for resources in the network. When such an oscillation is observed, a heuristic is evoked to resolve it. Algorithm 2 sets the rate to the non-zero value of the inflection point (in this case to 5 Mb/s) and continues the optimization process. In the self-regulating heuristic case the rate is set to zero.

Comparing the values of the objective function throughout the application of the algorithms in Fig. 7, one should note that the relatively high values at the early iterations can not be compared with the value of the objective function that the algorithm converged to for the following reason. The values of the objective function at the first few iterations, roughly until iteration 80, do not correspond to feasible solutions. This is typical for a gradient-based algorithm since the algorithm tries to converge to a solution where all the derivatives are zero and during this process some of the intermediate results can be infeasible solutions of the optimization problem [9].

As illustrated in Fig. 7, the decision for non zero rate for the oscillating users yields higher value of the objective function compared to the self-regulating heuristic, for the reasons explained earlier. Note that, as Theorem 1 states, when the optimal vector $\boldsymbol{\lambda}$ does not lead to oscillations, the optimization process comprised of Algorithms 1 and 2 converges to the globally optimal solution. Fig. 8 shows the convergence of the transmission powers allocation of the first 4 links in the network, the ones initiated from the 4 source nodes. The difference in dealing with oscillations between the two heuristics is illustrated in the power vectors as well. The self-regulating heuristic leads to zero powers for the oscillating users while Algorithm 2 gives non zero powers to achieve the necessary channel capacity. Finally, since we have assumed the operation at a high SINR environment, we should mention that the SINR ranges from $7dB$ to $18dB$, and therefore the error introduced by our capacity approximation in the worst case is less than 10% (note that for $SINR > 10dB$ the error is less than 4%). It is important to mention here that the approximation provides an underestimation of the link capacity, and therefore the upper bound of the Shannon capacity formula is not violated. This justifies the valid use of the approximated capacity formula.

VI. Concluding Remarks

Motivated by the non-convex resource allocation problems in *Network Utility Maximization* and the necessity for a novel optimization-based resource allocation protocol, this paper presents a general optimization framework for non-convex problems and provides a sufficient condition to assure that a distributed gradient-based algorithm converges to the optimal solution and can be also a necessary condition in some non-convex problem formulations. The optimization framework is applied on an optimization problem formulation in wireless ad-hoc networks. This formulation includes a power penalty function to assure convergence and energy efficiency of the power allocation. Consequently, a distributed algorithm to solve this problem is developed and an oscillation resolving heuristic is presented to assure network stability in non-convex problems whose optimal solution can not be calculated distributedly.

The focus of our future research will be twofold. First, we plan to investigate the applicability of the optimization framework to a wider selection of utility functions, and, then, work towards a generic procedure to identify whether the necessary and sufficient condition of Theorem 1 is satisfied for any non-convex optimization problem.

Appendix

A. Proof of Theorem 1

To prove its sufficiency, we start by showing that continuity of $\boldsymbol{x}^*(\boldsymbol{\lambda})$ around the optimal dual variables λ_i^* implies that *complementary slackness* is satisfied for problem $\mathbf{\Pi_p}$. Recall that the *complementary slackness* condition states that $\lambda_i^* h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^*)) = 0$, $\forall i$ at the optimal solution $x^*(\boldsymbol{\lambda}^*)$.

First, the case where $\lambda_i^* > 0$ for an arbitrary chosen i is examined. A very small positive constant $\epsilon > 0$ and a new vector $\boldsymbol{\lambda}^-$ are defined where

$$\lambda_j^- = \begin{cases} \lambda_j^* - \epsilon & \text{, if } j = i \\ \lambda_j^* & \text{, if } j \neq i \end{cases} \tag{12}$$

In other words, vectors $\boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^-$ differ only at one element, which has been reduced by the constant ϵ. Then, by definition of the sub-gradient, we have that

$$\begin{aligned} d(\boldsymbol{\lambda}^*) &\geq d(\boldsymbol{\lambda}^-) + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^-)^T \boldsymbol{\Lambda}(\boldsymbol{\lambda}^-) \Leftrightarrow \\ d(\boldsymbol{\lambda}^*) &\geq d(\boldsymbol{\lambda}^-) + \epsilon \frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda}^-)}{\partial \lambda_i} \Leftrightarrow \\ d(\boldsymbol{\lambda}^*) - d(\boldsymbol{\lambda}^-) &\geq \epsilon h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^-)). \end{aligned} \tag{13}$$

where $\boldsymbol{\Lambda}$ is a vector containing the partial derivatives of the lagrangian with respect to the dual variables, i.e. $\boldsymbol{\Lambda} = [\frac{\partial L(\boldsymbol{x},\boldsymbol{\lambda})}{\partial \lambda_i}, i \in [1, M]]$. But since the dual problem is a minimization problem and $\boldsymbol{\lambda}^*$ is its optimal solution, it follows that $d(\boldsymbol{\lambda}^*) \leq d(\boldsymbol{\lambda}^-)$ and hence by (13)

$$h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^-)) \leq 0. \tag{14}$$

Working at the same way, a second vector $\boldsymbol{\lambda}^+$ is defined as

$$\lambda_j^+ = \begin{cases} \lambda_j^* + \epsilon & \text{, if } j = i \\ \lambda_j^* & \text{, if } j \neq i \end{cases} \tag{15}$$

Again, by definition of the sub-gradient, it follows that

$$\begin{aligned} d(\boldsymbol{\lambda}^*) &\geq d(\boldsymbol{\lambda}^+) + (\boldsymbol{\lambda}^* - \boldsymbol{\lambda}^+)^T \boldsymbol{\Lambda}(\boldsymbol{\lambda}^+) \Leftrightarrow \\ d(\boldsymbol{\lambda}^*) &\geq d(\boldsymbol{\lambda}^+) - \epsilon \frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda}^+)}{\partial \lambda_i} \Leftrightarrow \\ d(\boldsymbol{\lambda}^*) - d(\boldsymbol{\lambda}^+) &\geq -\epsilon h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^+)). \end{aligned} \tag{16}$$

But for the same reason as before, $d(\boldsymbol{\lambda}^*) \leq d(\boldsymbol{\lambda}^+)$ and hence by (16), we conclude that

$$h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^+)) \geq 0. \tag{17}$$

From (14) and (17) we get to the conclusion that as long as $\boldsymbol{x}^*(\boldsymbol{\lambda})$ is continuous around $\boldsymbol{\lambda}^*$, then

$$h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^-)) = h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^+)) = h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^*)) = 0 \tag{18}$$

and hence *complementary slackness* is satisfied and the solution $\boldsymbol{x}^*(\boldsymbol{\lambda}^*)$ is primal feasible.

Then, the case where $\lambda_i^* = 0$ is examined. In this case, it is obvious that *complementary slackness* is satisfied. Primal feasibility of the solution can be shown using the positive constant ϵ and the price vector $\boldsymbol{\lambda}^+$ are defined as before. Equation (17) is reached again and under the continuity condition it follows that $h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^+)) \geq 0$. Hence, the *complementary slackness* condition is satisfied under the condition that the price-based function $\boldsymbol{x}^*(\boldsymbol{\lambda})$ is continuous at the optimal price vector $\boldsymbol{\lambda}^*$.

By definition of the dual problem, its optimal solution is given by $d^* = f(\boldsymbol{x}^*(\boldsymbol{\lambda}^*)) + \sum_{i=0}^m \lambda_i^* h_i(\boldsymbol{x}^*(\boldsymbol{\lambda}^*))$, and since *complementary slackness* holds, it reduces to $d^* = f(\boldsymbol{x}^*(\boldsymbol{\lambda}^*))$, which by definition of the primal problem is $f(\boldsymbol{x}^*(\boldsymbol{\lambda}^*)) \leq f^*(\boldsymbol{x})$ and hence $d^* \leq p^*$. But by *weak duality* it is known that $d^* \geq p^*$ and therefore it follows that $d^* = p^*$, where p^* and d^* are the optimal values of the primal and the dual problem respectively.

Therefore, it has been proven that continuity of the price based function (1) around at least one of the optimal price vectors implies that the *duality gap* is zero and that by solving the dual optimization problem the optimal solution $\boldsymbol{x}^*$ is also obtained.

B. Proof of Theorem 2

According to Complementary Slackness, which is a necessary condition for optimality, the fact that at least one constraint is active at the optimal solution implies that at least one of the optimal lagrange multipliers is non-zero and therefore the algorithm cannot converge unless (18) holds. Hence, continuity of $x^*(\boldsymbol{\lambda})$ around at least one of the optimal Lagrange multiplier vectors is a necessary condition.

References

[1] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control in communication networks: shadow prices, proportional fairness and stability," *J. Operational Research Soc.*, pp. 237–252, 1998.

[2] S. Low and D. Lapsley, "Optimization flow control—I: basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[3] D. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[4] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: framework and applications," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2254–2269, Dec. 2007.

[5] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control," *IEEE/ACM Trans. Commun.*, vol. 23, no. 1, pp. 104–116, Jan. 2005.

[6] G. Tychogiorgos, K. Leung, A. Misra, and T. LaPorta, "Distributed network utility optimization in wireless sensor networks using power control," in *Proc. 2008 IEEE PIMRC.*

[7] A. Ribeiro and G. Giannakis, "Layer separability of wireless networks," in *Proc. 2008 CISS*, pp. 821–826.

[8] W. Stallings, *Data and Computer Communications*, 9th ed. Pearson Custom Publishing, 2010.

[9] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 1999.

[10] J.-W. Lee, R. R. Mazumdar, and N. B. Shroff, "Non-convex optimization and rate control for multi-class services in the Internet," *IEEE/ACM Trans. Netw.*, vol. 13, no. 4, pp. 827–840, Aug. 2005.

[11] M. Fazel and M. Chiang, "Network utility maximization with nonconcave utilities using sum-of-squares method," in *Proc. 2005 IEEE CDC-ECC*, pp. 1867–1874.

[12] P. Hande, S. Zhang, and M. Chiang, "Distributed rate allocation for inelastic flows," *IEEE/ACM Trans. Netw.*, vol. 15, no. 6, pp. 1240–1253, Dec. 2007.

[13] M. Chiang, S. Zhang, and P. Hande, "Distributed rate allocation for inelastic flows: Optimization frameworks, optimality conditions, and optimal algorithms," in *Proc. 2005 IEEE INFOCOM*, vol. 4, pp. 2679–2690.

[14] J. Papandriopoulos, S. Dey, and J. S. Evans, "Optimal and distributed protocols for cross-layer design of physical and transport layers in manets," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1392–1405, 2008.

[15] J. Postel, "Transmission control protocol," Internet Engineering Task Force, RFC 793, Sep. 1981. Available: http://www.rfc-editor.org/rfc/rfc793.txt

[16] V. Jacobson, "Congestion avoidance and control," in *Proc. 1988 ACM Symp. Commun. Architectures Protocols*, pp. 314–329. Available: http://doi.acm.org/10.1145/52324.52356

[17] L. S. Brakmo and L. L. Peterson, "TCP vegas: end to end congestion avoidance on a global Internet," *IEEE J. Sel. Areas Commun.*, vol. 13, pp. 1465–1480, 1995.

[18] I. Rhee and L. Xu, "Cubic: a new TCP-friendly high-speed TCP variant," in *Proc. 2005 Int. Workshop Protocols Fast Long-Distance Netw.*

[19] S. H. Low, L. L. Peterson, and L. Wang, "Understanding TCP vegas: a duality model," *J. ACM*, vol. 49, no. 2, pp. 207–235, 2002.

[20] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Trans. Netw.*, vol. 11, no. 4, pp. 525–536, 2003.

[21] Cisco, "Cisco visual networking index: global mobile data traffic forecast update, 2010–2015," Cisco Systems Inc., Tech. Rep., Feb. 2011. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf

[22] ——, "Cisco visual networking index: Forecast and methodology, 2010–2015," Cisco Systems Inc., Tech. Rep., June 2011. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.pdf

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[24] J. Kyparisis, "On uniqueness of Kuhn-Tucker multipliers in nonlinear programming," *Mathematical Programming*, vol. 32, pp. 242–246, 1985.

[25] M. Chiang, "To layer or not to layer: balancing transport and physical layers in wireless multihop networks," in *Proc. 2004 IEEE INFOCOM*, vol. 4, pp. 2525–2536.

[26] W. Tsai, J. K. Antonio, and G. Huang, "Complexity of gradient projection method for optimal routing in data networks," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 897–905, 1999.

[27] J. S. P. V. Hutson and M. J. Cloud, *Applications of Functional Analysis and Operator Theory*. Academic Press, 1980.

[28] C. Liu, L. Shi, and B. Liu, "Utility-based bandwidth allocation for triple-play services," in *Proc. 2007 European Conf. Universal Multiservice Netw.*

Georgios Tychogiorgos received a Diploma in computer, networks, & telecommunications engineering from the University of Thessaly, Greece in 2006, and an M.Sc. in communications & signal processing and a Ph.D. in electrical engineering from Imperial College London in 2007 and 2013, respectively. In the period May 2009–January 2013, he worked as a Research Assistant at the Communications & Signal Processing Group at Imperial College London, working on network optimization and operational research. During summer 2011 and 2010, he worked for IBM at the T.J. Watson Research Center, NY, USA, where he worked in the areas of Internet of Things (IoT) and Quality of Information (QoI). During November 2007–April 2009, he was with British Telecom in Ipswich, UK, working on network design for multimedia applications. In 2012, he received the Best Student Paper award at the IEEE PIMRC Conference. In July 2011, he received the IBM First Patent Award for submitting a patent application to the US Patent Office. In November 2007, he received the Ericsson Award of Excellence in Telecommunications for the best undergraduate final project.

Athanasios Gkelias received his Ph.D. and M.Sc. from King's College London, UK, in 2005 and 2001, respectively, and his Diploma in electrical and computer engineering from Aristotle University of Thessaloniki, Greece, in 2000. Currently, he is a post-doctoral researcher at Imperial College London. In the past, he served as the project manager of the University Defense Research Centre (UDRC) in Signal Processing at Imperial College, sponsored by the UK Ministry of Defence (MoD). He also participated in several ICT funded projects such as Mobile-VCE, IBM-ITA, MEMBRANE, e-SENSE, and MIND. In the summer of 2008, he was at the Bell-Labs Research Centre, Alcatel-Lucent, UK, working as a visiting researcher on wireless mesh networks. He has published more than 40 peer-reviewed journal and conference papers and has been a TPC member and on the organizing committee of various international conferences.

Kin K. Leung received his B.S. degree from the Chinese University of Hong Kong in 1980, and his M.S. and Ph.D. degrees from the University of California, Los Angeles, in 1982 and 1985, respectively.

He joined AT&T Bell Labs in New Jersey in 1986 and worked at its successor companies, AT&T Labs and Bell Labs of Lucent Technologies, until 2004. Since then, he has been the Tanaka Chair Professor in the Electrical and Electronic Engineering (EEE) and Computing Departments at Imperial College in London. He serves as the Head of the Communications and Signal Processing Group in the EEE Department at Imperial. His research interests focus on networking, protocols, optimization, and modeling issues of wireless broadband, sensor, and ad-hoc networks. He also works on multi-antenna and cross-layer designs for the physical layer of these networks.

He received the Distinguished Member of Technical Staff Award from AT&T Bell Labs in 1994, and was a co-recipient of the 1997 Lanchester Prize Honorable Mention Award. He was elected as an IEEE Fellow in 2001. He received the Royal Society Wolfson Research Merits Award from 2004 to 2009 and became a member of Academia Europaea in 2012. He has actively served on many conference committees. He serves as a member (2009–2011) and the chairman (2012–2013) of the IEEE Fellow Evaluation Committee for the Communications Society. He was a guest editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC), *IEEE Wireless Communications*, and the *MONET* journal, and as an editor for the JSAC: Wireless Series, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and the IEEE TRANSACTIONS ON COMMUNICATIONS. Currently, he is an editor for *ACM Computing Survey* and the *International Journal on Sensor Network*.