Network Utility Maximization With Nonconcave Utilities Using Sum-of-Squares Method

Maryam Fazel
Control and Dynamical Systems, Caltech
maryam@cds.caltech.edu

Mung Chiang
Electrical Engineering, Princeton University
chiangm@princeton.edu

***Abstract*— The Network Utility Maximization problem has recently been used extensively to analyze and design distributed rate allocation in networks such as the Internet. A major limitation in the state-of-the-art is that user utility functions are assumed to be strictly concave functions, modeling elastic flows. Many applications require inelastic flow models where nonconcave utility functions need to be maximized. It has been an open problem to find the globally optimal rate allocation that solves nonconcave network utility maximization, which is a difficult nonconvex optimization problem.**

We provide a centralized algorithm for off-line analysis and establishment of a performance benchmark for nonconcave utility maximization. Based on the semialgebraic approach to polynomial optimization, we employ convex sum-of-squares relaxations solved by a sequence of semidefinite programs, to obtain increasingly tighter upper bounds on total achievable utility for polynomial utilities. Surprisingly, in all our experiments, a very low order and often a minimal order relaxation yields not just a bound on attainable network utility, but the globally maximized network utility. When the bound is exact, which can be proved using a sufficient test, we can also recover a globally optimal rate allocation. In addition to polynomial utilities, sigmoidal utilities can be transformed into polynomials and are handled. Furthermore, using two alternative representation theorems for positive polynomials, we present price interpretations in economics terms for these relaxations, extending the classical interpretation of independent congestion pricing on each link to pricing for the simultaneous usage of multiple links.

Keywords: Nonconvex optimization, network utility, rate allocation, algebraic geometry, sum of squares method.

I. Introduction

A. Background: Basic network utility maximization

Since the publication of the seminal paper [6] by Kelly, Maulloo, and Tan in 1998, the framework of Network Utility Maximization (NUM) has found many applications in network rate allocation algorithms and Internet congestion control protocols (*e.g.*, [10]), as well as user behavior models and network efficiency-fairness characterization. By allowing nonlinear concave utility objective functions, NUM substantially expands the scope of the classical LP-based Network Flow Problems.

Consider a communication network with L links, each with a fixed capacity of c_l bps, and S sources (*i.e.*, end users), each transmitting at a source rate of x_s bps. Each source s emits one flow, using a fixed set $L(s)$ of links in its path, and has a utility function $U_s(x_s)$. Each link l is shared by a set $S(l)$ of sources. Network Utility Maximization (NUM), in its basic version, is the following problem of maximizing the total utility of the network $\sum_s U_s(x_s)$, over the source rates $\mathbf{x}$, subject to linear flow constraints $\sum_{s:l\in L(s)} x_s \le c_l$ for all links l:

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & \sum_{s\in S(l)} x_s \le c_l, \ \ \forall l, \\ & \mathbf{x} \succeq 0 \end{array} \tag{1}$$

where the variables are $\mathbf{x} \in \mathbf{R}^S$.

There are many nice properties of the basic NUM model due to several simplifying assumptions of the utility functions and flow constraints, which provide the mathematical tractability of problem (1) but also limit its applicability. In particular, the utility functions $\{U_s\}$ are often assumed to be increasing and strictly concave functions. In this paper, we investigate the extension of the basic NUM to maximization of nonconcave utilities.

Assuming that $U_s(x_s)$ becomes concave for large enough x_s is reasonable, because the law of diminishing marginal utility eventually will be effective. However, U_s may not be concave throughout its domain. In his seminal paper published a decade ago, Shenker [18] differentiated inelastic network traffic from elastic traffic. Utility functions for elastic traffic were modeled as strictly concave functions. While *inelastic* flows with nonconcave utility functions represent important applications in practice, they have received little attention and rate allocation among them have scarcely any mathematical foundation, except the very recent publications [9], [3], due to their intrinsic intractability in the utility maximization framework.

B. Review: Canonical distributed algorithm

A reason that the the assumption of utility function's concavity is upheld in almost all papers on NUM is that it leads to three highly desirable mathematical properties of the basic NUM:

- It is a convex optimization problem, therefore the global minimum can be computed (at least in centralized algorithms) in worst-case polynomial-time complexity [2].
- Strong duality holds for (1) and its Lagrange dual problem, *i.e.*, the difference between the optimized value of (1) and that of its dual problem (the optimal duality gap) is zero [1], [2]. Zero duality gap enables a dual approach to solve (1).

- Minimization of a separable objective function over linear constraints can be conducted by distributed algorithms based on the dual approach.

Indeed, the basic NUM (1) is such a 'nice' optimization problem that its theoretical and computational properties have been well studied since the 1960s in the field of monotropic programming, *e.g.*, as summarized in [15]. For network rate allocation problems, a dual-based distributed algorithm has been widely studied (*e.g.*, in [6], [10]), and is summarized below.

Zero duality gap for (1) states that the solving the Lagrange dual problem is equivalent to solving the primal problem (1). The Lagrange dual problem is readily derived. We first form the Lagrangian of (1):

$$L(\mathbf{x}, \boldsymbol{\lambda}) = \sum_s U_s(x_s) + \sum_l \lambda_l \left(c_l - \sum_{s \in S(l)} x_s \right)$$

where $\lambda_l \geq 0$ is the Lagrange multiplier (link congestion price) associated with the linear flow constraint on link l. Additivity of total utility and linearity of flow constraints lead to a Lagrangian dual decomposition into individual source terms:

$$\begin{aligned} L(\mathbf{x}, \boldsymbol{\lambda}) &= \sum_s \left[U_s(x_s) - \left(\sum_{l \in L(s)} \lambda_l \right) x_s \right] + \sum_l c_l \lambda_l \\ &= \sum_s L_s(x_s, \lambda^s) + \sum_l c_l \lambda_l \end{aligned}$$

where $\lambda^s = \sum_{l \in L(s)} \lambda_l$. For each source s, $L_s(x_s, \lambda^s) = U_s(x_s) - \lambda^s x_s$ only depends on local x_s and the link prices λ_l on those links used by source s.

The Lagrange dual function $g(\boldsymbol{\lambda})$ is defined as the maximized $L(\mathbf{x}, \boldsymbol{\lambda})$ over $\mathbf{x}$. This 'net utility' maximization obviously can be conducted distributively by the each source, as long as the aggregate link price $\lambda^s = \sum_{l \in L(s)} \lambda_l$ is available to source s, where source s maximizes a strictly concave function $L_s(x_s, \lambda^s)$ over x_s for a given λ^s:

$$x_s^*(\lambda^s) = \text{argmax} \left[U_s(x_s) - \lambda^s x_s \right], \quad \forall s. \tag{2}$$

The Lagrange dual problem is

$$\begin{array}{ll} \text{minimize} & g(\boldsymbol{\lambda}) = L(\mathbf{x}^*(\boldsymbol{\lambda}), \boldsymbol{\lambda}) \\ \text{subject to} & \boldsymbol{\lambda} \succeq 0 \end{array} \tag{3}$$

where the optimization variable is $\boldsymbol{\lambda}$. Any algorithms that find a pair of primal-dual variables $(\mathbf{x}, \boldsymbol{\lambda})$ that satisfy the KKT optimality condition would solve (1) and its dual problem (3). One possibility is a distributed, iterative subgradient method, which updates the dual variables $\boldsymbol{\lambda}$ to solve the dual problem (3):

$$\lambda_l(t+1) = \left[\lambda_l(t) - \alpha(t) \left(c_l - \sum_{s \in S(l)} x_s(\lambda^s(t)) \right) \right]^+, \quad \forall l \tag{4}$$

where t is the iteration number and $\alpha(t) > 0$ are step sizes. Certain choices of step sizes, such as $\alpha(t) = \beta/t, \ \beta > 0$, guarantee that the sequence of dual variables $\boldsymbol{\lambda}(t)$ will converge to the dual optimal $\boldsymbol{\lambda}^*$ as $t \to \infty$. The primal variable $\mathbf{x}(\boldsymbol{\lambda}(t))$ will also converge to the primal optimal variable $\mathbf{x}^*$. For a primal problem that is a convex optimization, the convergence is towards the global optimum.

The sequence of the pair of algorithmic steps (2,4) forms a *canonical distributed algorithm* that globally solves network utility optimization problem (1) and the dual (3) and computes the optimal rates $\mathbf{x}^*$ and link prices $\boldsymbol{\lambda}^*$.

C. Summary of results

It is known that for many multimedia applications, user satisfaction may assume non-concave shape as a function of the allocated rate. For example, the utility for streaming applications is better described by a sigmoidal function: with a convex part at low rate and a concave part at high rate, and a single inflexion point x^0 (with $U_s''(x^0) = 0$) separating the two parts. The concavity assumption on U_s is also related to the elasticity assumption on rate demands by users. When demands for x_s are not perfectly elastic, $U_s(x_s)$ may not be concave.

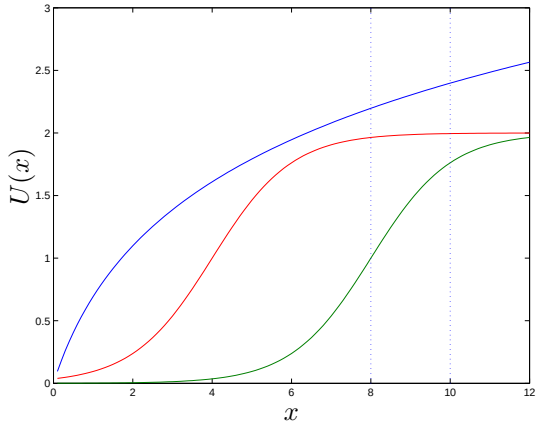

Fig. 1. Some examples of utility functions $U_s(x_s)$: it can be concave or sigmoidal as shown in the graph, or any general nonconcave function. If the bottleneck link capacity used by the source is small enough, *i.e.*, if the dotted vertical line is pushed to the left, a sigmoidal utility function effectively becomes a convex utility function.

Suppose we remove the critical assumption that $\{U_s\}$ are concave functions, and allow them to be any nonlinear functions. The resulting NUM becomes nonconvex optimization and significantly harder to be analyzed and solved, even by centralized computational methods. In particular, a local optimum may not be a global optimum and the duality gap can be strictly positive. The standard distributive algorithms that solve the dual problem may produce infeasible or suboptimal rate allocation. Global maximization of nonconcave functions is an intrinsically difficult problem of nonconvex optimization. Indeed, over the last two decades, it has been widely recognized that "*in fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity*" (Quote from Rockafellar [16]).

Despite such difficulties, there have been two very recent publications on distributed algorithm for nonconcave

utility maximization. In [9], it is shown that, in general, the canonical distributive algorithm that solves the dual problem may produce suboptimal, or even infeasible, rate allocation, and a 'self-regulation' heuristic is proposed to avoid the resulting oscillation in rate allocation. However, the heuristic converges only to a suboptimal rate allocation. In [3], a set of sufficient conditions and necessary conditions is presented under which the canonical distributed algorithm still converges to the globally optimal solution. However, these conditions may not hold in many cases. In summary, currently there is no theoretically polynomial-time and practically efficient algorithm (distributed or centralized) known for nonconcave utility maximization.

In this paper, we remove the concavity assumption on utility functions, thus turning NUM into a nonconvex optimization problem with a strictly positive duality gap. Such problems in general are NP hard, thus extremely unlikely to be polynomial-time solvable even by centralized computations. Using a family of convex semidefinite programming (SDP) relaxations based on the sum-of-squares (SOS) relaxation and the Positivstellensatz Theorem in real algebraic geometry, we apply a centralized computational method to bound the total network utility in polynomial-time. A surprising result is that for all the examples we have tried, wherever we could verify the result, the tightest possible bound (*i.e.*, the globally optimal solution) of NUM with nonconcave utilities is computed with a very low order relaxation. This efficient numerical method for off-line analysis also provides the benchmark for distributed heuristics. We also examine two forms of sigmoidal utilities, and use a change of variables to transform the original problem into one that involves only polynomials. The sum-of-squares approach mentioned above can then be applied.

Our focus has been not only on calculating numerical bounds for the problem, but also on understanding the inner workings of the relaxations, and the mechanism behind the tightening of the upper bound, in the context of NUM problems. In this regard, we have examined two polynomial representations that are particularly suited for an economics/price interpretation of NUM. One result is that the classical pricing of congestion on a link is (partially) extended to pricing of the usage of multiple links.

These three different approaches: proposing distributed but suboptimal heuristics (for sigmoidal utilities) in [9], determining optimality conditions for the canonical distributed algorithm to converge globally (for all nonlinear utilities) in [3], and proposing efficient but centralized method to compute the global optimum (for a wide class of utilities that can be transformed into polynomial utilities) in this paper, are complementary in the study of distributed rate allocation by nonconcave NUM, a difficult class of nonlinear optimization.

II. Global maximization of nonconcave network utility

A. Sum-of-squares method

First consider a NUM with polynomial utilities, such as $U_s(x_s) = x_s^2$. Sigmoidal utilities will be considered in subsection III.B. For notational simplicity, we assume the domain of definition of the $U_s(x_s)$ implies $x_s \geq 0$.

$$\begin{array}{ll} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & \sum_{s \in S(l)} x_s \leq c_l, \ \ \forall l. \end{array} \tag{5}$$

We would like to bound the maximum network utility by γ in polynomial time and search for a tight bound. Had there been no link capacity constraints, maximizing a polynomial is already an NP hard problem, but can be relaxed into a SDP [19]. This is because testing if the following bounding inequality holds $\gamma \geq p(\mathbf{x})$, where $p(\mathbf{x})$ is a polynomial of degree d in n variables, is equivalent to testing the positivity of $\gamma - p(\mathbf{x})$, which can be relaxed into testing if $\gamma - p(\mathbf{x})$ can be written as a sum of squares (SOS): $p(\mathbf{x}) = \sum_{i=1}^{r} q_i(\mathbf{x})^2$ for some polynomials q_i, where the degree of q_i is less than or equal to $d/2$. This is referred to as the SOS relaxation (for unconstrained minimization/maximization). If a polynomial can be written as a sum of squares, it must be non-negative, but not vice versa. Conditions under which this relaxation is tight were studied since Hilbert, and it is known that, for example, the relaxation is tight for quadratic polynomials. Determining if a sum of squares decomposition exists can be formulated as an SDP feasibility problem, thus polynomial-time solvable.

Constrained nonconcave NUM can be relaxed by a generalization of the Lagrange duality theory, which involves *nonlinear* combinations of the constraints instead of linear combinations in the standard duality theory, as discussed in the next section. The key result is the Positivstellensatz, due to Stengle [20], in real algebraic geometry, which states that for a system of polynomial inequalities, either there exists a solution in $\mathbf{R}^n$ or there exists a polynomial which is a certificate that no solution exists. This infeasibility certificate is recently shown to be also computable by an SDP of sufficient size [12], [11], a process that is referred to as the sum-of-squares method [1] and automated by the software SOSTOOLS [13].

Furthermore, as will be leveraged in the next section, the bound γ itself can become an optimization variable in the SDP and can be directly minimized. A nested family of SDP relaxations, each indexed by the degree of the certificate polynomial, is guaranteed to produce the exact global maximum. Of course, given the problem is NP hard, it is not surprising that the worst-case degree of certificate (thus the number of SDP relaxations needed) is exponential in the number of variables. What is interesting is the observation that in applying SOSTOOLS to nonconcave utility maximization, a very low order, often the minimum order relaxation already produces the globally optimal solution.

B. Application of SOS method to nonconcave NUM

Using sum-of-squares and the Positivstellensatz, we set up the following problem whose objective value converges

[1] For a complete theory and many applications of SOS methods, see [12] and references therein.

to the optimal value of problem (5), as the degree of the polynomials involved is increased.

$$\begin{array}{l}\text{minimize} \quad \gamma \\ \text{subject to} \\ \gamma - \sum_s U_s(x_s) - \sum_l \lambda_l(\mathbf{x})(c_l - \sum_{s\in S(l)} x_s) \\ - \sum_{j,k} \lambda_{jk}(\mathbf{x})(c_j - \sum_{s\in S(j)} x_s)(c_k - \sum_{s\in S(k)} x_s) - \\ \ldots - \lambda_{12\ldots n}(\mathbf{x})(c_1 - \sum_{s\in S(1)} x_s)\ldots(c_n - \sum_{s\in S(n)} x_s) \\ \text{ is SOS,} \\ \lambda_l(\mathbf{x}), \lambda_{jk}(\mathbf{x}), \ldots, \lambda_{12\ldots n}(\mathbf{x}) \text{ are SOS.}\end{array} \tag{6}$$

The optimization variables are γ and all of the *coefficients* in polynomials $\lambda_l(\mathbf{x}), \lambda_{jk}(\mathbf{x}), \ldots, \lambda_{12\ldots n}(\mathbf{x})$. Note that $\mathbf{x}$ is *not* an optimization variable; the constraints hold for all $\mathbf{x}$, therefore imposing constraints on the coefficients. This formulation uses Schmüdgen's representation of positive polynomials over compact sets [17].[2] Two alternative representations are discussed in section IV.

Let D be the degree of the expression in the first constraint in (6). We refer to problem (6) as the SOS relaxation of order D for the constrained NUM. For a fixed D, the problem can be solved via SDP. As D is increased, the expression includes more terms, the corresponding SDP becomes larger, and the relaxation gives tighter bounds. An important property of this nested family of relaxations is guaranteed convergence of the bound to the global maximum.

To see the relation of SOS relaxation with the Lagrange dual, consider the simplest case of (6) where λ_l are nonnegative constants and all other multipliers are zero,

$$\begin{array}{l}\text{minimize} \quad \gamma \\ \text{subject to} \\ \gamma - \sum_s U_s(x_s) - \sum_l \lambda_l(c_l - \sum_{s\in S(l)} x_s) \text{ is SOS,} \\ \lambda_l \geq 0, \ \ \forall l.\end{array} \tag{7}$$

Comparing this with the Lagrange dual of (5),

$$\begin{array}{ll}\text{minimize} & \max_{\mathbf{x}}\{\sum_s U_s(x_s) + \sum_l \lambda_l(c_l - \sum_{s\in S(l)} x_s)\} \\ \text{subject to} & \lambda_l \geq 0, \ \ \forall l,\end{array} \tag{8}$$

or

$$\begin{array}{l}\text{minimize} \quad \gamma \\ \text{subject to} \\ \gamma - \sum_s U_s(x_s) - \sum_l \lambda_l(c_l - \sum_{s\in S(l)} x_s) \geq 0, \ \ \forall \mathbf{x} \\ \lambda_l \geq 0, \ \ \forall l,\end{array}$$

shows that (7) is an SOS relaxation of (8). There are several special cases (namely, Hilbert's conditions) where problems (7) and (8) are equivalent, *e.g.*, when the utilities are quadratic. [3]

[2]Schmüdgen's representation applies when $\gamma - \sum U_s(x_s)$ is strictly positive on the feasible set. Therefore the convergence is asymptotic in theory, however in practice finite convergence is observed most of the time. If we were to use Stengle's Positivstellensatz, we would have finite convergence but could not have γ as an optimization variable and at each relaxation level would have to use a bisection on γ. For computational convenience, we choose Schmüdgen's form.

[3]In fact, in the quadratic case, this relaxation coincides with the well-known S-procedure.

There is a standard *price interpretation* for the Lagrange dual. For the case of concave utilities, the dual variables $\boldsymbol{\lambda}$ can be interpreted as link prices, and the bound γ from (8) is exact.

In the non-concave utility case the gap between the dual (8) and the original problem (5) (known as the duality gap), and also the gap between (7) and (5) are in general nonzero. So the γ obtained from (7) is only an upper bound; however $\boldsymbol{\lambda}$ can still be interpreted as link prices, in the following sense. If the lth capacity constraint is violated, users incur an extra charge proportional to the amount of violation, with price λ_l (since $c_l - \sum_{s\in S(l)} x_s$ is negative and subtracts from the total utility). Similarly, users are rewarded proportional to the amount of under-used capacity. In sharp contrast to the concave utility case, for nonconcave utilities, these are not equilibrium prices and do not result in optimal or even feasible rate allocation, unless the relaxation is exact. In section IV, we discuss this interpretation for higher order relaxations.

Higher order relaxations can improve the upper bound. For example, consider allowing products of constraints such that $D = 2$ (note that in this case the multiplier for the product of two constraints has to be a constant). We have

$$\begin{array}{l}\text{minimize} \quad \gamma \\ \text{subject to} \\ \gamma - \sum_s U_s(x_s) - \sum_l \lambda_l(c_l - \sum_{s\in S(l)} x_s) - \\ \sum_{j,k} \lambda_{jk}(c_j - \sum_{s\in S(j)} x_s)(c_k - \sum_{s\in S(k)} x_s) \text{ is SOS,} \\ \lambda_l \geq 0, \ \ \lambda_{jk} \geq 0, \ \ \forall l, j, k.\end{array} \tag{9}$$

This problem is in fact the SOS relaxation of the Lagrange dual for problem (5) with some added redundant constraints; namely, the pairwise product of every two non-negative terms $(c_j - \sum_{s\in S(j)} x_s)(c_k - \sum_{s\in S(k)} x_s)$. As mentioned before, this problem can be solved via SDP, and yields a bound that is at least as strong as the first-order relaxation (7).

Regarding the choice of degree D for each level of relaxation, clearly a polynomial of odd degree cannot be SOS, so we need to consider only the cases where the expression has even degree. Therefore, the degree of the first non-trivial relaxation is the largest even number greater than or equal to degree of $\sum_s U_s(x_s)$, and the degree is increased by 2 for the next level.

A key question now becomes: How do we find out, after solving an SOS relaxation, if the bound happens to be exact? Fortunately, there is a *sufficient test* that can reveal this, using the properties of the SDP and its dual solution. In [5], [7], a parallel set of relaxations, equivalent to the SOS ones, is developed in the dual framework. The dual of checking the nonnegativity of a polynomial over a semi-algebraic set turns out to be finding a sequence of *moments* that represent a probability measure with support in that set. To be a valid set of moments, the sequence should form a positive semidefinite moment matrix. Then, each level of relaxation fixes the size of this matrix, *i.e.*, considers moments up a certain order, and therefore solves an SDP. This is equivalent to fixing the order of the polynomials appearing in SOS relaxations. The

sufficient rank test checks a rank condition on this moment matrix and recovers (one or several) optimal $\mathbf{x}^*$, as discussed in [5].

In summary, we have the following **Algorithm** for centralized computation of a globally optimal rate allocation to nonconcave utility maximization, where the utility functions can be written as or converted into polynomials (details about such conversions are in the next section):

1) Formulate the relaxed problem (6) for a given degree D.
2) Use SDP to solve the Dth order relaxation, which can be conducted using SOSTOOLS [13].
3) If the resulting dual SDP solution satisfies the sufficient rank condition, the Dth order optimizer $\gamma^*(D)$ is the globally optimal network utility, and a corresponding $\mathbf{x}^*$ can be obtained. Otherwise, $\gamma^*(D)$ may still be the globally optimal network utility but is only provably an upper bound.
4) Increase D to $D+2$, *i.e.*, the next higher order relaxation, and repeat.

In the following section, we give examples of the application of SOS relaxation to the nonconcave NUM. We also apply the above sufficient test to check if the bound is exact, and if so, we recover the optimum rate allocation $\mathbf{x}^*$ that achieve this tightest bound.

III. Numerical Examples and Sigmoidal Utilities

A. Polynomial utility examples

First, consider quadratic utilities, *i.e.*, $U_s(x_s) = x_s^2$ as a simple case to start with (this can be useful, for example, when the bottleneck link capacity limits sources to their convex region of a sigmoidal utility). We can also handle weights on the utilities, cubic or higher order polynomials as utilities, or U_s of different orders for different users, in a similar fashion. We present examples that are typical, in our experience, of the performance of the relaxations.

Example 1. *Small illustrative example*. Consider the simple 2 link, 3 user network shown in Figure 2, with $\mathbf{c} = [1,\ 2]$. The optimization problem is

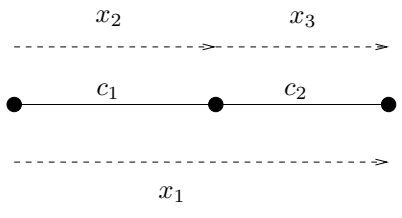

Fig. 2. Network topology for example 1.

$$\begin{array}{ll} \text{maximize} & \sum_s x_s^2 \\ \text{subject to} & x_1 + x_2 \le 1 \\ & x_1 + x_3 \le 2 \\ & x_1, x_2, x_3 \ge 0. \end{array} \tag{10}$$

The first level relaxation with $D = 2$ is

$$\begin{array}{l} \text{minimize } \gamma \\ \text{subject to} \\ \gamma - (x_1^2 + x_2^2 + x_3^2) - \lambda_1(-x_1 - x_2 + 1) - \lambda_2(-x_1 \\ -x_3 + 2) - \lambda_3 x_1 - \lambda_4 x_2 - \lambda_5 x_3 - \lambda_6(-x_1 - x_2 + 1) \\ (-x_1 - x_3 + 2) - \lambda_7 x_1(-x_1 - x_2 + 1) - \lambda_8 x_2(-x_1 \\ -x_2 + 1) - \lambda_9 x_3(-x_1 - x_2 + 1) - \lambda_{10} x_1(-x_1 - x_3 + 2) \\ -\lambda_{11} x_2(-x_1 - x_3 + 2) - \lambda_{12} x_3(-x_1 - x_3 + 2) - \\ \lambda_{13} x_1 x_2 - \lambda_{14} x_1 x_3 - \lambda_{15} x_2 x_3 \text{ is SOS}, \\ \lambda_i \ge 0,\ i = 1, \ldots, 15. \end{array} \tag{11}$$

The first constraint above can be written as $x^T Q x$ for $x = [1,\ x_1,\ x_2,\ x_3]^T$ and an appropriate Q. For example, the (1,1) entry which is the constant term reads $\gamma - \lambda_1 - 2\lambda_2 - 2\lambda_6$, the (2,1) entry, coefficient of x_1, reads $\lambda_1 + \lambda_2 - \lambda_3 + 3\lambda_6 - \lambda_7 - 2\lambda_{10}$, and so on. The expression is SOS if and only if $Q \ge 0$. The optimal γ is 5, which is achieved by, *e.g.*, $\lambda_1 = 1,\ \lambda_2 = 2,\ \lambda_3 = 1,\ \lambda_8 = 1,\ \lambda_{10} = 1,\ \lambda_{12} = 1,\ \lambda_{13} = 1,\ \lambda_{14} = 2$ and the rest of the λ_i equal to zero. Using the sufficient test (or in this example, by inspection) we find the optimal rates $\mathbf{x}_0 = [0,\ 1,\ 2]$.

In this example, many of the λ_i could be chosen to be zero. This means not all product terms appearing in 11 are needed in constructing the SOS polynomial. Such information is valuable from the decentralization point of view, and can help determine to what extent our bound can be calculated in a distributed manner. This is a topic for future work.

Example 2. Consider the 4 link, 4 user network shown in Figure 3, with quadratic utilities $U_s = x_s^2$.

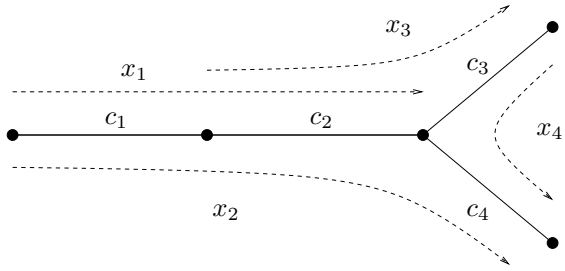

Fig. 3. Network topology for example 2.

If we set all link capacities $\{c_l\}$ to 1, using an SOS relaxation with $D = 2$, we obtain the upper bound $\gamma = 2$. Either by using the sufficient test or by inspection, we find that the rate vector $\mathbf{x}_0 = [1,\ 0,\ 0,\ 1]$ achieves this bound and the bound is exact. As another example, with $\mathbf{c} = [2,\ 3,\ 4,\ 1]$, we obtain $\gamma = 10$. Again, we find that $\mathbf{x}_0 = [0,\ 0,\ 3,\ 1]$ achieves this upper bound, which is therefore exact.

Example 3. *Mixed utilities*. Consider the example above, with utilities x_s^2 for users 1 and 2, and x_s^3 for users 3 and 4. With capacity $\mathbf{c} = [1,\ 5,\ 4,\ 3]$, we obtain the exact bound $\gamma = 65$, and using the sufficient test we recover two optimal rate allocations $[1,\ 0,\ 4,\ 0]$ and $[0,\ 1,\ 4,\ 0]$ that achieve this bound.

Example 4. As a larger example, consider the network shown in Figure III-A with 7 links. We allow 9 users, with

the following routing table that lists the links on each user's path.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1,2	1,2,4	2,3	4,5	2,4	6,5,7	5,6	7	5

For $\mathbf{c} = [5,\ 10,\ 4,\ 3,\ 7,\ 3,\ 5]$, we obtain the bound $\gamma = 116$ with $D = 2$, which turns out to be globally optimal, and the globally optimal rate vector can be recovered: $\mathbf{x}_0 = [5,\ 0,\ 4,\ 0,\ 1,\ 0,\ 0,\ 5,\ 7]$. In this example, exhaustive search is too computationally intensive, and the sufficient condition test plays an important role in proving the bound was exact and in recovering $\mathbf{x}_0$.

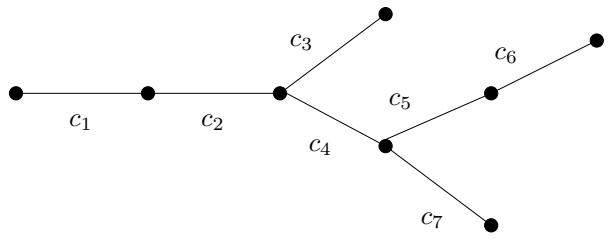

Fig. 4. Network topology for example 4.

Example 5. *Large m-hop ring topology*. Consider a ring network with n nodes, n users and n links where each user's flow starts from a node and goes clockwise through the next m links, as shown in figure III-A for $n = 6$, $m = 2$. As a large example, with $m = 2$, $n = 25$ and capacities chosen randomly for a uniform distribution on $[0,\ 10]$, using relaxation of order $D = 2$ we obtain the exact bound $\gamma = 321.11$ and recover an optimal rate allocation. For $m = 2$, $n = 30$, and capacities randomly chosen from $[0,\ 15]$, it turns out that $D = 2$ relaxation yields the exact bound 816.95 and a globally optimal rate allocation.

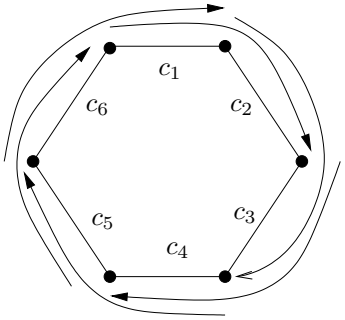

Fig. 5. Network topology for example 5.

B. Sigmoidal utility examples

Now consider sigmoidal utilities in a standard form:

$$U_s(x_s) = \frac{1}{1 + e^{-(a_s x_s + b_s)}},$$

where $\{a_s, b_s\}$ are constant integers. Even though these sigmoidal functions are not polynomials, we show the problem can be cast as one with polynomial cost and constraints, with a change of variables.

Example 6. Consider the simple 2 link, 3 user example shown in Figure 2 for $a_s = 1$ and $b_s = -5$.

The NUM problem is to

$$\begin{array}{ll} \text{maximize} & \sum_s \frac{1}{1+e^{-(x_s-5)}} \\ \text{subject to} & x_1 + x_2 \le c_1 \\ & x_1 + x_3 \le c_2 \\ & \mathbf{x} \ge 0. \end{array} \tag{12}$$

Let $y_s = \frac{1}{1+e^{-(x_s-5)}}$, then $x_s = -\log(\frac{1}{y_s} - 1) + 5$. Substituting for $x_1,\ x_2$ in the first constraint, arranging terms and taking exponentials, then multiplying the sides by $y_1 y_2$ (note that $y_1, y_2 > 0$), we get

$$(1 - y_1)(1 - y_2) \ge e^{(10 - c_1)} y_1 y_2,$$

which is polynomial in the new variables $\mathbf{y}$. This applies to all capacity constraints, and the non-negativity constraints for x_s translate to $y_s \ge \frac{1}{1+e^5}$. Therefore the whole problem can be written in polynomial form, and SOS methods apply. This transformation renders the problem polynomial for general sigmoidal utility functions, with any a_s and b_s.

We present some numerical results, using a small illustrative example. Here SOS relaxations of order 4 ($D = 4$) were used. For $c_1 = 4,\ c_2 = 8$, we find $\gamma = 1.228$, which turns out to be a global optimum, with $\mathbf{x}_0 = [0,\ 4,\ 8]$ as the optimal rate vector. For $c_1 = 9, c_2 = 10$, we find $\gamma = 1.982$ and $\mathbf{x}_0 = [0,\ 9,\ 10]$. Now place a weight of 2 on y_1, while the other y_s have weight one, we obtain $\gamma = 1.982$ and $\mathbf{x}_0 = [9,\ 0,\ 1]$.

In general, if $a_s \ne 1$ for some s, however, the degree of the polynomials in the transformed problem may be very high. If we write the general problem as

$$\begin{array}{ll} \text{maximize} & \sum_s \frac{1}{1+e^{-(a_s x_s + b_s)}} \\ \text{subject to} & \sum_{s \in S(l)} x_s \le c_l,\ \ \forall l, \\ & \mathbf{x} \ge 0, \end{array} \tag{13}$$

each capacity constraint after transformation will be

$$\prod_s (1 - y_s)^{r_{ls} \Pi_{k \ne s} a_k} \ge$$
$$\exp(-\textstyle\prod_s a_s (c_l + \sum_s r_{ls}/a_s b_s)) \prod_s y_s^{r_{ls} \prod_{k \ne s} a_k},$$

where $r_{ls} = 1$ if $l \in L(s)$ and equals 0 otherwise. Since the product of the a_s appears in the exponents, $a_s > 1$ significantly increases the degree of the polynomials appearing in the problem and hence the dimension of the SDP in the SOS method.

It is therefore also useful to consider alternative representations of sigmoidal functions such as the following rational function:

$$U_s(x_s) = \frac{x_s^n}{a + x_s^n},$$

where the inflection point is $x^0 = (\frac{a(n-1)}{n+1})^{1/n}$ and the slope at the inflection point is $U_s(x^0) = \frac{n-1}{4n}(\frac{n+1}{a(n-1)})^{1/n}$. Let $y_s = U_s(x_s)$, the NUM problem in this case is equivalent to

$$\begin{array}{ll} \text{maximize} & \sum_s y_s \\ \text{subject to} & x_s^n - y_s x_s^n - a y_s = 0 \\ & \sum_{s \in S(l)} x_s \le c_l,\ \forall l \\ & \mathbf{x} \ge 0 \end{array} \tag{14}$$

which again can be accommodated in the SOS method and be solved by the proposed Algorithm.

The benefit of this choice of utility function is that the largest degree of the polynomials in the problem is $n+1$, therefore growing linearly with n. The disadvantage compared to the exponential form for sigmoidal functions is that the location of the inflection point and the slope at this point cannot be set independently.

IV. Alternative representations for convex relaxations to nonconcave NUM

The SOS relaxation we used in the last two sections is based on Schmüdgen's representation for positive polynomials over compact sets described by other polynomials. In this section, we briefly discuss two other representations of relevance to the NUM, that are interesting from both theoretical (*e.g.*, interpretation) and computational (*e.g.*, efficiency) points of view.

A. LP relaxation

Exploiting linearity of the constraints in NUM and with the additional assumption of nonempty interior for the feasible set (which holds for NUM), we can use Handelman's representation [4] and refine the Positivstellensatz condition to obtain the following convex relaxation of nonconcave NUM problem:

$$\begin{aligned}&\text{minimize } \gamma\\ &\text{subject to}\\ &\gamma - \textstyle\sum_s U_s(x_s) = \displaystyle\sum_{\alpha\in N^L} \lambda_\alpha \prod_{l=1}^{L} (c_l - \textstyle\sum_{s\in S(l)} x_s)^{\alpha_l},\ \forall \mathbf{x}\\ &\lambda_\alpha \geq 0,\ \forall \alpha,\end{aligned} \tag{15}$$

where the optimization variables are γ and λ_α, and α denotes an ordered set of integers $\{\alpha_l\}$.

Fixing D where $\sum_l \alpha_l \leq D$, and equating the coefficients on the two sides of the equality in (15), yields a linear program (LP). (Note that there are no SOS terms, therefore no semidefiniteness conditions.) As before, increasing the degree D gives higher order relaxations and a tighter bound.

We provide a (partial) price interpretation for problem (15). First, normalize each capacity constraint as $1 - u_l(x) \geq 0$, where $u_l(x) = \sum_{s\in S(l)} x_s/c_l$. We can interpret $u_l(x)$ as *link usage*, or the probability that link l is used at any given point in time. Then, in (15), we have terms linear in u such as $\lambda_l(1-u_l(x))$, in which λ_l has a similar interpretation as in concave NUM, as the price of using link l (at full capacity, due to the normalization). We also have product terms such as $\lambda_{jk}(1-u_j(x))(1-u_k(x))$, where $\lambda_{jk}u_j(x)u_k(x)$ indicates the probability of *simultaneous* usage of links j and k, for links whose usage probabilities are independent (*e.g.*, they do not share any flows). Products of more terms can be interpreted similarly.

While the above price interpretation is not complete and does not justify all the terms appearing in (15) (*e.g.*, powers of the constraints; product terms for links with shared flows), it does provide some useful intuition: this relaxation results in a pricing scheme that provides better incentives for the users to observe the constraints, by putting additional reward (since the corresponding term adds positively to the utility) for simultaneously keeping two links free. Such incentive helps tighten the upper bound and eventually achieve a feasible (and optimal) allocation.

This relaxation is computationally attractive since we need to solve an LPs instead of the previous SDPs at each level. However, significantly more levels may be required [8].

B. Relaxation with no product terms

Putinar [14] showed that a polynomial positive over a compact set (with an extra assumption that always holds for linear constraints as in NUM problems) can be represented as an SOS-combination of the constraints. This yields the following convex relaxation for nonconcave NUM problem:

$$\begin{aligned}&\text{minimize} \quad \gamma\\ &\text{subject to}\\ &\gamma - \textstyle\sum_s U_s(x_s) = \lambda_0(\mathbf{x}) + \sum_{l=1}^{L} \lambda_l(\mathbf{x})(c_l - \sum_{s\in S(l)} x_s),\ \forall \mathbf{x}\\ &\lambda_l(\mathbf{x}) \text{ is SOS}, \quad l = 0,\ldots,L,\end{aligned} \tag{16}$$

where the optimization variables are the coefficients in $\lambda_l(\mathbf{x})$. Similar to the SOS relaxation (6), fixing the order D of the expression in (16) results in an SDP. This relaxation has the nice property that no product terms appear: the relaxation becomes exact with a high enough D without the need of product terms. However, this degree might be much higher than what the previous SOS method requires.

We note yet another price interpretation: this time the link price is given by an SOS polynomial multiplier that depends on the rates. The physical meaning of such prices, and the computational aspects of this relaxation remain to be explored.

V. Conclusions and further extensions

We consider the NUM problem in the presence of inelastic flows, *i.e.*, flows with nonconcave utilities. Despite its practical importance, this problem has not been studied widely, mainly due to the fact it is a nonconvex, NP-hard problem. There has been no effective mechanism, centralized or distributed, to compute the globally optimal rate allocation for nonconcave utility maximization problems in networks. This limitation has made performance assessment and design of networks that include inelastic flows very difficult.

To address this problem, we employed convex SOS relaxations, solved by a sequence of SDPs, to obtain high quality, increasingly tighter upper bounds on total achievable utility. In practice, the performance of our SOSTOOLS-based algorithm was surprisingly good, and bounds obtained using a polynomial-time (and indeed a low-order and often minimal order) relaxation were found to be exact, achieving the global optimum of nonconcave NUM problems. Furthermore, a dual-based sufficient test, if successful, detects the exactness of the bound, in which case the optimal rate allocation can also be recovered. This surprisingly good performance of the proposed algorithm brings up a fundamental question on whether there is any particular property or structure in

nonconcave NUM that makes it especially suitable for SOS relaxations.

We further examined the use of two more specialized polynomial representations, one that uses products of constraints with constant multipliers, resulting in LP relaxations; and at the other end of spectrum, one that uses a 'linear' combination of constraints with SOS multipliers. We expect these relaxations to give higher order certificates, thus their potential computational benefits need to be examined further. We also show they admit economics interpretations (*e.g.*, prices, incentives) that provide some insight on how the SOS relaxations work in the framework of link congestion pricing for the simultaneous usage of multiple links.

An important research issue to be further investigated is decentralization methods for rate allocation among sources with nonconcave utilities. The proposed algorithm here is not easy to decentralize, given the products of the constraints or polynomial multipliers that destroy the separable structure of the problem. However, when relaxations become exact, the sparsity pattern of the coefficients can provide information about partially decentralized computation of optimal rates. For example, if after solving the NUM off-line, we obtain an exact bound, then if the coefficient of the cross-term $x_i x_j$ turns out to be zero, it means users i and j do not need to communicate to each other to find their optimal rates. An interesting next step in this area of research is to investigate distributed version of the proposed algorithm through limited message passing among clusters of network nodes and links.

It is also worth continuing to explore other types of nonconcave functions that can be transformed into polynomials and handled by SOS methods, in addition to the two sigmoidal forms we already examined in this paper.

ACKNOWLEDGMENT

We would like to thank very helpful discussions with Pablo Parrilo, Steven Low, John Doyle, and Daniel Palomar.

REFERENCES

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

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

[3] M. Chiang, S. Zhang, and P. Hande, "Distributed rate allocation for inelastic flows: Optimization framework, optimality conditions, and optimal algorithms," *Proc. IEEE Infocom*, Miami, FL, March 2005.

[4] D. Handelman, "Representing polynomials by positive linear functions on compact convex polyhedra," *Pacific J. Math.*, vol. 132, pp. 35-62, 1988.

[5] D. Henrion, J.B. Lasserre, "Detecting global optimality and extracting solutions in GloptiPoly," Research report, LAAS-CNRS, 2003.

[6] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of Operations Research Society*, vol. 49, no. 3, pp.237-252, March 1998.

[7] J.B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, no. 3, pp. 796-817, 2001.

[8] J.B. Lasserre, "Polynomial programming: LP-relaxations also converge," *SIAM J. Optimization*, vol. 15, no. 2, pp. 383-393, 2004.

[9] J. W. Lee, R. R. Mazumdar, and N. Shroff, "Non-convex optimization and rate control for multi-class services in the Internet," *Proc. IEEE Infocom*, Hong Kong, China, March 2004.

[10] S. H. Low, "A duality model of TCP and queue management algorithms," *IEEE/ACM Tran. Networking*, vol. 11, no. 4, pp. 525-536, Aug. 2003.

[11] P. A. Parrilo, Structured semidefinite programs and semi-algebraic geometry methods in robustness and optimization," PhD thesis, Caltech, May 2002.

[12] P. A. Parrilo, "Semidefinite programming relaxations for semi-algebraic problems," *Math. Program.*, vol. 96, pp.293-320, 2003.

[13] S. Prajna, A. Papachristodoulou, P. A. Parrilo, "SOSTOOLS: Sum of squares optimization toolbox for Matlab, available from `http://www.cds.caltech.edu/sostools`, 2002-04.

[14] M. Putinar, "Positive polynomials on compact semi-algebraic sets," *Indiana University Mathematics Journal*, vol. 42, no. 3, pp. 969-984, 1993.

[15] R. T. Rockafellar, *Network Flows and Monotropic Programming*, Athena Scientific, 1998.

[16] R. T. Rockafellar, "Lagrange multipliers and optimality," *SIAM Review*, vol. 35, pp. 183-283, 1993.

[17] K. Schmüdgen, "The k-moment problem for compact semialgebraic sets," *Math. Ann.*, vol. 289, pp. 203-206, 1991.

[18] S. Shenker, "Fundamental design issues for the future Internet," *IEEE J. Sel. Area Comm.*, vol. 13, no. 7, pp. 1176-1188, Sept. 1995.

[19] N. Z. Shor, "Quadratic optimization problems," *Soviet J. Comput. Systems Sci.*, vol 25, pp. 1-11, 1987.

[20] G. Stengle, "A Nullstellensatz and a Positivstellensatz in semialgebraic geometry," *Math. Ann.*, vol. 207, pp.87-97, 1974.