Monotonic Optimization Approach to the Network Utility Maximization Problem

Bazarragchaa Barsbold
School of Mathematics and Computer Science
National University of Mongolia
210646 Ulaanbaatar, Mongolia
Email: barsboldb@smcs.num.edu.mn

Rentsen Enkhbat
Institute of Mathematics
National University of Mongolia
210646 Ulaanbaatar, Mongolia
Email: renkhbat46@yahoo.com

Hwang Won Joo
Department of Information and Communication Engineering,
Inje University,
621-749, Gimhae, Korea
Email: ichwang@inje.ac.kr

***Abstract*—We consider the Network Utility Maximization Problem with a non-concave, monotonically increasing objective function. Non-concave utility functions model inelastic traffic flows. Maximization of a non-concave function over a closed, convex set is a non-convex programming problem. In this paper we establish a necessary and sufficient global optimality condition for problems with a monotonic objective function. On the basis of this result we propose a numerical method for solving a class of monotonic problems.**

I. Introduction

A Network Utility Maximization (NUM) Problem is defined as follows.

$$\begin{array}{lll} \text{maximize} & \sum_{s\in S} f_s(x_s) & (1)\\ \text{subject to} & \sum_{s\in S(l)} x_s \le c_l, \quad \forall l\in L & (2)\\ & x\ge 0, & (3)\end{array}$$

where S is a set of sources, L is links connecting sources, x_s for $s\in S$ is a transmission rate of s-th source, c_l for $l\in L$ is a capacity of l-th link, $S(l)$ for $l\in L$ is a number of transmission sources using l-th link. f_s for $s\in S$ is utility function of the s-th source. The problem is to find transmission rate of each sources maximizing total network utility at given capacity of the links.
Utility functions f_s for $s\in S$ are often assumed to be strongly concave. This assumption much simplifies the problem mathematically. Strongly concave functions model only elastic traffic flows. This leads to mainly convex minimization problems with well established theory and algorithms based on Lagrangian functions.

When traffic flows are inelastic, utility functions can be nonconcave. However it has practical importance and this case has received little attention until recent publications by Chiang et al. [2], Lee et al. [3], mainly because of computational complexity. When it deals with a nonconvex problems, particularly convex maximization problems, finding an exact global optimal solution numerically is NP-hard [4]. The standard outer approximation method [4] for convex maximization was adapted for the solving monotonic optimization problem in [9]. This approach is based on the approximation of compact normal sets by simple normal sets called polyblocks.

The problem of maximizing monotonic function under monotonic constraints have been discussed in [8]. In this work Polyblock outer approximation algorithm has been proposed for maximizing increasing functions over intersection of a compact normal set with so called reverse normal set.

In this paper, we propose a global optimality condition for solving the Network Utility Maximization /NUM/ problem with a monotonic increasing utility function. This paper is organized as follows. In the first section we reformulated problem (1)-(3) as a global optimization problem and introduced new lower bound constraints to make the model more realistic. Common types of nonconcave utility functions have been considered and classified from the computational complexity point of view.
In the second section we derived a new necessary and sufficient condition for the network utility maximization problem with a monotonic objective function.
In the third section we constructed a numerical algorithm for the NUM problem based on the global optimality conditions. Some numerical results are provided.

II. Classification of Network Utility Maximization Problem

We can reformulate problem (1)-(3) as follows.

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{j=1}^{S} f_j(x_j) & (4)\\ \text{subject to} & \displaystyle\sum_{j=1}^{S} d_{ij}x_j \le c_i, \quad \forall i=1,...,L & (5)\\ & x\ge 0, & (6)\end{array}$$

where S is number of sources, L is number of links connecting sources, $d_{ij}\in\{0,1\}$ and $d_{ij}=1$ if i-th source transmits signal using j-th link.
However in an initial formulation of problem (1)-(3) transmission rates are assumed to be nonnegative, it is useful to consider a case in which transmission rate x_j for all $j=1,...,S$ is above some fixed level. Otherwise, some transmission rate can take a value equal to zero which means that its source is totally discriminated and will not emit any signal. It is unnatural. That is why we include to the NUM problem additional constraints bounding below transmission

rates. Then problem (4)-(6) takes the following form.

$$\text{maximize} \quad \sum_{j=1}^{S} f_j(x_j) \tag{7}$$

$$\text{subject to} \quad \sum_{j=1}^{S} d_{ij}x_j \leq c_i, \ \ \forall i = 1, ..., L \tag{8}$$

$$x \geq lb > 0, \tag{9}$$

where vector $lb \in R^S$ and lb_j is the lowest possible level of transmission rate for j-th source.

Problem (1)-(3) has been considered in [2] for the case $f_j(x_j) = x_j^2$, $f_j(x_j) = x_j^3$ and $f_j(x_j) = \frac{1}{1+e^{-ax_j-b_j}}$, and sum-of-squares (SOS) method is applied to solve it.

When function f_j is convex for all $j = 1, ..., S$ then the problem is treated as a convex maximization problem. Convex maximization problem has found many applications in engineering and finding its exact global optimal solution numerically is NP-hard [4]. The problem complexity is still kept unchanged even in the case of simple bound constrained problems [4]. For this type of problems an appropriate global optimality condition and numerical algorithm can be found in [1].

Let us consider some additional properties of the problem for $f_j(x_j) = \frac{1}{1+e^{-ax_j-b_j}}$. We denote the objective function by $f(x) = \sum_{j=1}^{S} f_j(x_j)$. Clearly we have:

$$\nabla f(x) = \left(\frac{df_1}{dx_1}(x_1), ..., \frac{df_S}{dx_S}(x_S) \right), \tag{10}$$

$$\nabla^2 f(x) = \begin{pmatrix} \frac{d^2 f_1}{dx_1^2}(x_1) & 0 & ... & 0 \\ 0 & \ddots & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & ... & 0 & \frac{d^2 f_S}{dx_S^2}(x_S) \end{pmatrix} \tag{11}$$

Proposition 1.

a. *The objective function has zero second order derivative when $x_j = -\frac{b_j}{a_j}$ for all $j = 1, ..., S$.*

b. *It is natural to assume about the network architecture that for all $j = 1, ..., S$ there exists at least one $i = 1, ..., S$ satisfying $d_{ij} = 1$. Otherwise, j-th link has no application. In addition to this assumption, if it holds the following inequality*

$$\min_{i:\begin{cases}1\leq i\leq L\\ d_{ij}=1\end{cases}} c_i \leq -\frac{b_j}{a_j} \quad \textit{for all} \quad j = 1, ..., S \tag{12}$$

then the objective function f is strictly convex.

Proof. a. By solving the following system of S independent equations we get the result.

$$f_j''(x_j) = 0, \text{ for } j = 1, ..., S.$$

b. From constraints (8)-(9) it implies that

$$d_{ij}x_j \leq c_i$$

for all $i = 1, ..., L$ and $j = 1, ..., S$. We can write it equivalently as:

$$x_j \leq \min_{i:\begin{cases}1\leq i\leq L\\ d_{ij}=1\end{cases}} c_i$$

for all $j = 1, ..., S$. Therefore, if we assume that condition (12) is true then we obtain:

$$x_j \leq -\frac{b_j}{a_j}$$

for all $j = 1, ..., S$. Under this condition the second order derivatives of the utility functions are positive, so $f_j(x_j)$ are strictly convex for $j = 1, ..., S$. Thus objective function f is also strictly convex as a sum of strictly convex functions [6]. ■

Based on Proposition 1, we can classify problem (7)-(9) as follows.

A. Concave maximization problem

Let us consider the case when it holds:

$$lb_j \geq -\frac{b_j}{a_j} \ \text{ for } j = 1, ..., L. \tag{13}$$

In this case, the lower bounds lb_j on transmission rates x_j are high enough so that the objective function became strictly concave and canonical algorithms for the NUM problem will find a global optimal solution in polynomial time. It is a well studied part of nonlinear programming. We will omit discussion on it as a trivial case. Brief summary on canonical algorithms can be found in Chiang et al. [2].

B. Convex maximization problem

If link capacities are small enough, i.e. if inequalities (12) hold, then problem (7)-(9) becomes a convex maximization problem. In this case the problem is NP-hard. For global optimality condition and numerical algorithm for this type of problem we refer to [1].

C. General nonconcave maximization problem

If neither conditions (12) nor (13) hold the second order derivative (11) have both positive and negative diagonal entries, so the objective function f is neither concave nor convex. In this case the problem is classified as global optimization problem of general type. Special methods and algorithms will be employed for solving it.

In our further investigations, we will focus on the monotonic maximization problem as a special case of the nonconcave maximization problem.

III. The Global Optimality Condition

Consider the following separable optimization problem

$$\max_{x\in D} \ \sum_{j=1}^{S} f_j(x_j), \tag{14}$$

where a constraint set D is a convex compact subset of R^S, functions $f_j : R \to R, \ j = 1, ..., S$ are continuous and strictly increasing. We introduce the following notations

$$f(x) = \sum_{j=1}^{S} f_j(x_j), \tag{15}$$

$$D^+(y) = \{x \in D : \ x_j \geq y_j \ \text{ for all } j = 1, ..., S\}, \tag{16}$$

$$E_\alpha(f) = \{x \in R^n : \ f(x) = \alpha\}, \ \ \alpha \in R, \tag{17}$$

Set $E_\alpha(f)$ is referred as a level set of function f. We formulate necessary and sufficient condition for problem (14) in the following theorem.

Theorem 1. *A point $z \in D$ is the global solution to (14) if and only if*

$$int\, D^+(y) = \emptyset \ \text{ for all } y \in L_{f(z)}(f). \tag{18}$$

Proof. Necessity. Assume that z is a global solution, that is,

$$f(x) \leq f(z) \quad \forall x \in D. \tag{19}$$

Suppose that there exists $\overline{x} \in int\, D^+(\overline{y})$ for some $\overline{y} \in E_{f(z)}(f)$. It implies

$$\overline{x}_j > \overline{y}_j \ \text{ for all } \ j = 1, ..., S.$$

Since f_j is strictly increasing, we have $f_j(\overline{x}_j) > f_j(\overline{y}_j)$ for all $j = 1, ..., S$. Consequently, by (15) we get

$$f(\overline{x}) > f(\overline{y}) = f(z).$$

which contradicts our assumption (19) on z.

Sufficiency. Assume that (18) holds for some $z \in D$ and z is not a global solution. That is

$$\exists u \in D, \ \ f(u) > f(z). \tag{20}$$

Since D is compact, there exist $v_j \in R$ satisfying

$$v_j = \min_{x \in D} x_j$$

for all $j = 1, ..., S$,. Let us define vectors v and h by

$$v = (v_1, ..., v_S)^T \ \text{ and } \ h = u - v.$$

Clearly $h \geq 0$. Consider a line segment given by

$$x(t) = (u_1 - th_1, \, ..., \, u_S - th_S)^T$$

for $t \in [0; 1]$. Since $h_j \geq 0$, each function $x_j(t)$ is decreasing on $[0; 1]$ for all $j = 1, ..., S$.

Define the function Φ by $\Phi(t) = f(x(t))$ for $t \geq 0$. It is clear that

$$\Phi(0) > f(z) > \Phi(1). \tag{21}$$

Since function Φ is continuous, it attains all intermediate values between $\Phi(0)$ and $\Phi(1)$, i.e.

$$\exists \overline{t} \in [0; 1] \ \text{ such that } \ \Phi(\overline{t}) = f(x(\overline{t})) = f(z).$$

Since vector function $x(\cdot)$ is componentwise decreasing, we conclude that $v > \overline{x}$. Therefore, we get

$$v \in int\, D^+(\overline{x}).$$

This contradicts assumption (18). The proof is complete. ■

IV. Numerical Algorithm

Note that the optimality condition (18) requires checking the set $int\ D^+(y)$ for emptiness for each $y \in E_{f(z)}(f)$. It is a hard problem. In order to implement this condition numerically, we need to relax it by checking emptiness for only a finite number of points of the level set. For this purpose, we introduce the following definition.

Definition 1. *Let define set the A^m_z by*

$$A^m_z = \{y^1, y^2, \ldots, y^m : \ \ y^i \in E_{f(z)}(f), \ \ i = 1, 2, \ldots, m\} \tag{22}$$

is called the approximation set to the level set $E_{f(z)}(f)$ at the point z.

Lemma 1. *If there exists $u \in int\ D^+(y)$ for $y \in E_{f(z)}(f)$ and $z \in D$, then $f(u) > f(z)$.*

Proof. The proof follows from Theorem 1.

Based on Theorem 1 and the above lemma we can construct an algorithm for solving problem (14).

Algorithm NUM

Input: A separable and strictly monotonic objective function f, a bounded, closed and convex polyhedral set D.
Output: An approximate solution z to problem (14) i.e., an approximate global maximizer of f over D.

- Step 1 Choose $x^0 \in R^S$ and set $k := 0$.
- Step 2 Find a local maximum z^k of problem (14) using a local method starting with x^0 as an initial approximation.
- Step 3 Choose sufficiently large m and randomly generated directions $h^1, ..., h^m \in R^S$ such that $h^i_j \geq 0$ for all $j = 1, ..., S$ and $i = 1, ..., m$. Solve univariate equations $f(\alpha_i h^i) = f(z^k)$ for α_i employing the bisection method for all $i = 1, ..., m$.
- Step 4 Construct an approximation set $A^m_{z^k}$ at the point z^k by selecting $y^i = \alpha_i h^i$ for all $i = 1, ..., m$.
- Step 5 For each $y^i \in A^m_{z^k}$ check set $int\ D^+(y^i)$ for emptiness solving the following linear programming problem.

$$\max_{\substack{x \in D, \\ x \geq y^i}} x_1 + ... + x_S.$$

  If exists i_0 and solution x^{i_0} to the above problem such that $x^{i_0} \neq y^{i_0}$ then select $x^0 = x^{i_0}$ and Goto Step 5, otherwise Go to Step 6.
- Step 6 z^k is an approximate global maximizer and terminate. ■

We can easily see that if the optimality condition (18) is not satisfied at each z^k then the sequence $f(z^k)$ is a strictly monotonic increasing sequence, i.e., $f(z^{k+1}) > f(z^k)$. On the other hand, the solving equation $f(\alpha_i h^i) = f(z^k)$ for given h^i with respect to α_i can be implemented in a polynomial time. Since we can solve linear programming by interior methods at each iteration, the algorithm terminates in a polynomial time finding an approximate global solution for the given number m. If we are not satisfied with the approximate global solution found at a current iteration, there are two ways to improve this

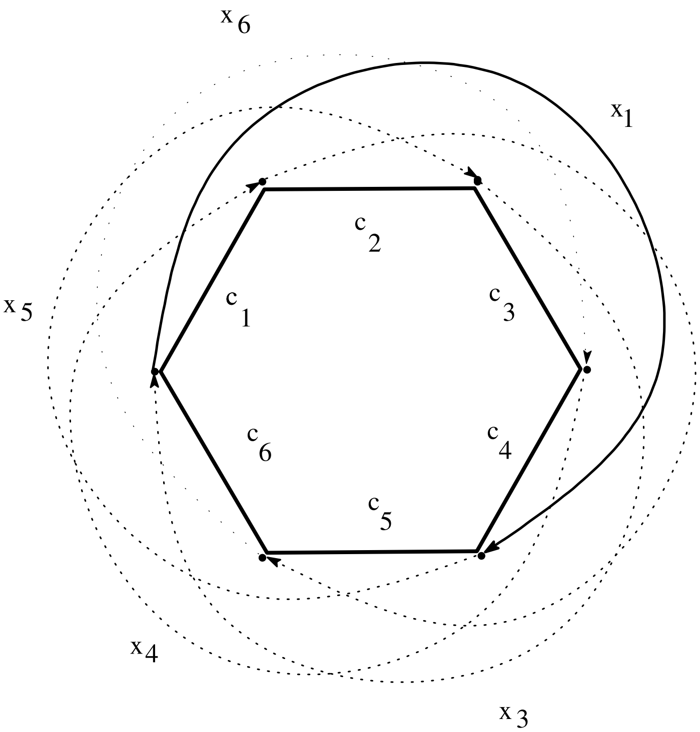

Fig. 1. Ring network with $S = 6$

solution. It can be done in the following way.
1)To increase a number of points of the level set of the function at the current local maximizer.
2) To combine the current level set of the function at the local maximizer z^k with other global search methods such as inner, outer approximations or branch and bound method.
The latter is beyond the scope of our paper and has to be be examined in our further investigations in this direction.

In order to implement our proposed algorithm, we considered a network with ring topology with S users and S links connecting $\lfloor \frac{S}{2} \rfloor + 1$ subsequent users, where $\lfloor \alpha \rfloor$ denotes an integer part of α(Figure 1). We have tested our approach for 12 test problems for maximizing sigmoidal utility functions with randomly generated parameters a_j and b_j for all $j = 1, ..., S$. Link capacities c_j are also chosen randomly so that the objective function stays neither convex nor concave over the constraint set. The proposed algorithm was implemented in MATLAB 7 on PC with Pentium IV 2.4GHz processor and 256 MB RAM. In all cases, the global solutions were found by the algorithm. Results of numerical experiments are given in Table I. In the table by N we denote an example number, by S a number of variables of the test problem. The column labeled by f_0 contains an initial local optimal values. The approximate global optimal values obtained by the proposed algorithm is listed in the column labeled by $\tilde{f}$. Columns marked with $Imp.\ \%$ shows improvement in percent given by the formula:

$$\frac{\tilde{f} - f_0}{\tilde{f}} * 100\%.$$

A number of level set approximations required during computation are shown in the column labeled by $Appx$. A column with label $Imp.\ /times/$ contains a number of improvements encountered by computation. A column marked

TABLE I.

N	S	f_0	$\tilde{f}$	Imp. %	Appx.	Imp. /times/	Time /h:m:s/
1	5	0.4964	0.68	27.00	150	1	0:00:2.363
2	5	1.0985	1.3054	15.85	150	1	0:00:2.504
3	5	1.5932	1.9993	20.31	150	2	0:00:3.044
4	9	3.6193	3.9577	8.55	81	2	0:00:2.063
5	10	1.0002	5.0001	80.00	100	2	0:00:2.063
6	10	2.0244	6.0049	66.29	100	2	0:00:2.494
7	11	1.061	1.0746	1.26	127	1	0:00:6.148
8	11	0.59376	1.1208	47.02	1210	1	0:00:6.099
9	11	0.38051	0.62501	39.12	1210	1	0:00:6.629
10	11	0.44026	1.3867	68.25	1210	1	0:00:5.939
11	15	4.0768	5.5082	25.99	225	2	0:00:3.515
12	27	0.17581	0.31421	44.05	7290	1	0:00:45.445

by $Time/h:m:s/$ contains computation time in $hour:minute:second$ format.

Example 1.

a=(0.10129, 0.041534, 12.152, 4.3175, 14.135)^T,
b=(-17, -4.2046, -0.07851, -6.7597, -9.5621)^T,
c=(336.28, 202.52, , 0.062225, 3.3959, 1.4018)^T
$\tilde{x}$=(0, 1.4018, 0.0622, 0, 0)^T.

Example 2.

a=(11.237, 7.4969, 2.336, 0.34878, 4.2607,)^T,
b=(-13.637, -6.7609, -12.834, -2.6612, -0.81216,)^T,
c=(0.74312, 1.1961, 4.6924, 3.616, 2.2573,)^T,
$\tilde{x}$=(0, 0.7431, 0, 0.4530, 1.5142)^T.

Example 3.

a=(4.6136, 0.59055, 4.226, 0.14369, 2.9238,)^T,
b=(-1.9121, -12.408, -12.624, -6.2084, -9.8421,)^T,
c=(1.4886, 102.05, 14.46, , 206.94, 13.96,)^T,
$\tilde{x}$=(1.4886, 0, 0, 8.3439, 6.1165)^T.

Example 4.

$a = ($1.6823, 2.4037, 0.64806, 0.39055, 0.016966, 10.192, 9.8806, 4.7697, 8.5486,)^T,
b=(-19.455, -0.051552 , -7.8483, -0.3408, -8.0184, -0.13108, -5.038, -0.022219, -1.1896)^T,
c=(58.254, 0.79151, 60.642, 5.2232, 2363.8, 1.0301, 2.7945, 0.69306, 1.3808)^T,
$\tilde{x}$=(1.6823, 2.4037, 0.64806, 0.39055 0.016966, 10.192, 9.8806, 4.7697, 8.5486,)^T.

Example 5.

a=(13.262, 14.346, 0.06523, 9.7665, 12.935, 5.4189, 12.995, 2.5366, 5.8157, 1.0793,)^T,
b=(-14.491, -26.1, -27.137, -10.561, -8.9898, -11.064, -29.4, -5.6099, -16.605)^T,
c=(3.4182, 54.018, 58.754, 8.3209, 70.275, 93.594, 31.453, 85.115, 86.378,
35.158)^T,
$\tilde{x}$=(0, 1.6538, 0, 0, 1.5085, 0.2558, 1.6463, 0, 4.6237, 23.5294,)^T.

Example 6.

a=(10.318, 10.855, 5.0737, 5.4268, 0.5133, 13.592, 13.03, 0.35046, 4.9926, 5.4885, $)^T$,
b=(-16.314, -24.878, -26.05, -22.997, -13.936, -17.718, -4.8455, -5.7467,
-6.4728, $)^T$,
c=(39.697, 71.747, 72.97, 5.9242, 81.472, 99.219, 57.065, 58.073, 60.385,
70.791, $)^T$,
$\tilde{x}$=(3.4536, 30.9639, , 2.7597, 0, 0, 1.5693, 1.9239, 0, 2.4310, 18.0718$)^T$.

Example 7.

a=(6.18415, 10.1737, 12.057, 3.99671, 2.48311, 3.45971, 7.98772, 10.3249, 6.6653, 12.8748, 12.2667$)^T$,
b=(-8.24553, -13.5649, -16.076, -5.32895, -3.31081, -4.61294, -10.6503, -13.7665, -8.88706, -17.1664, -16.3556$)^T$,
c=(2.0614, 3.3912, 4.019, 1.3322, 0.8277, 1.1532, 2.6626, 3.4416, 2.2218, 4.2916, 4.0889$)^T$,
$\tilde{x}$=(-5.9398e-021, 1.7293, 8.3477e-020, -6.5621e-020, 0.33207, 1.2828e-020,
2.9099e-020, 0.48183, 0.013795, -1.8691e-019, 0.40528$)^T$.

Example 8.

a=(14.5972, 11.677, 8.63457, 5.46258, 3.80776, 6.92992, 5.15274, 5.32616, 1.15266, 7.94597, 5.78247$)^T$,

b=(-19.463, -15.5694, -11.5128, -7.28345, -5.07701,
-9.2399, -6.87032,
-7.10155, -1.53689, -10.5946, -7.70996$)^T$,
c=(4.8657, 3.8923, 2.8782, 1.8209, 1.2693, 2.31, 1.7176, 1.7754, 0.38422, 2.6487, 1.9275$)^T$,
$\tilde{x}$=(0, 1.5793e-032, 3.6978e-032, 1.8209, 1.438e-032, -1.5407e-033, 2.8273e-033, 1.0785e-032, -2.4074e-032, 0.04809, 0.10663$)^T$.

Example 9.

a=(13.1485, 2.74604, 0.61563, 11.0912, 3.86477, 4.68572, 5.06805, 9.43611, 3.05993, 11.0927, 4.85062$)^T$,
b=(-17.5314, -3.66139, -0.820841, -14.7883, -5.15303, -6.24763, -6.75739, -12.5815, -4.07991, -14.7903, -6.46749$)^T$,
c=(4.3828, 0.91535, 0.20521, 3.6971, 1.2883, 1.5619, 1.6893, 3.1454, 1.02, 3.6976, 1.6169$)^T$,
$\tilde{x}$=(1.6105e-034, -5.7778e-034, -3.8519e-034, 0, 0.20521, 0, 0, -7.7037e-034, 1.02, 4.6689e-035, -4.5613e-034$)^T$.

Example 10.

a=(8.61067, 14.6349, 14.1092, 12.0477, 7.73585, 4.98874, 1.39492, 10.2659 7.27186, 14.5349, 0.164902$)^T$,
b=(-3.05486, -10.4141, -12.3622, -5.88926, -19.7598, -11.352, -12.1663, -18.5229,
-1.88139, -4.38441, -18.861$)^T$,
c=(4.4414, 4.5866, 3.5612, 4.2102, 2.0438, 2.715, 4.4867, 0.16875, 0.7962, 0.84141, 2.549$)^T$,
$\tilde{x}$=(4.6222e-033, 6.5482e-033, 2.157e-032, 0.84141, 1.7076, 0.16745, 1.3482e-033, -6.5482e-033, 0.16875, -1.5407e-033, -5.7778e-033$)^T$.

Example 11.

a=(4.0459, 3.4301, 0.25348, 2.7333, 0.69564,
0.69773, 3.5446, 0.0075284, 8.7518, 8.4621, 2.9769,
11.106, 8.6258, 11.255, 0.052249$)^T$,
b=(-3.7016, -1.9931, -16.817, -4.6917, -2.9847, -15.612, -11.311, -0.055569, -11.985, -4.4391, -1.4239, -7.6251, -6.9061, -7.8518, -6.1528$)^T$,
c=(4.5771, 3.1429, 332.37, 9.5166, 21.478, 111.94, 16.87, 36.953, 6.8877, 3.5423, 3.1726, 4.0753, 4.7842, 3.9894, 589.34$)^T$,
$\tilde{x}$=(0, 0, 0, 1.4896 0, 0, 0, 0, 1.6533, 0.8239, 0.8242, 0.9427, 0, 0.9516, 0$)^T$.

Example 12.

a=(1.95768, 8.03563, 8.69044, 4.57666, 6.45133,
6.90871, 8.53513, 7.77198, 6.0139, 6.12716, 14.539,
7.18362, 4.46336, 3.71163, 8.8661, 3.6463, 3.14693,
6.00674, 3.66249, 3.08932, 11.4875, 6.58181, 5.22211,
8.32127, 7.02947, 7.13808, 5.37541$)^T$,
b=(-2.61024, -10.7142, -11.5873, -6.10221, -8.60177, -9.21162, -11.3802, -10.3626 -8.01853, -8.16955,
-19.3853, -9.57816, -5.95115, -4.94884, -11.8215, -4.86173,
-4.1959, -8.00898, -4.88332, -4.1191, -15.3167, -8.77575,
-6.96281, -11.095, -9.37263, -9.51744, -7.16722$)^T$,
c=(0.65256, 2.6785, 2.8968, 1.5256, 2.1504, 2.3029, 2.845, 2.5907, 2.0046, 2.0424, 4.8463, 2.3945, 1.4878, 1.2372,
2.9554, 1.2154, 1.049, 2.0022, 1.2208, 1.0298,
3.8292, 2.1939, 1.7407, 2.7738, 2.3432, 2.3794, 1.7918$)^T$,
$\tilde{x}$=(0.65256 5.5931e-034, -3.9719e-034, 5.7778e-034, -6.9396e-034, 3.6323e-034, -2.0704e-033, 1.2467e-034,
2.2479e-034, -7.7037e-034, 2.0222e-033, -3.2972e-034,
1.8296e-033, -2.8889e-034, 0.67434, 0.16646, 0.39642,
4.6097e-034 -4.1982e-034, -1.9259e-034, -2.1667e-034,
-6.1389e-034, -6.0185e-034, -3.135e-034, 1.9259e-033,
6.9127e-034, 1.8432e-034$)^T$.

V. Conclusion

We have considered the network utility maximization problem with a monotonic increasing objective function. In order to solve it numerically, we derived a new global optimality condition for the problem. Based on the global optimality conditions, we proposed a numerical algorithm which uses linear programming as subproblems. Efficiency of the algorithm depends on a number of points of the level set of the function. The algorithm generates a sequence of local maximizers. The proposed method can be combined with other global optimization methods. Further investigations has to be done in this direction in the near future. This paper is a preliminary step towards use of global optimality conditions in nonconcave utility maximization problems. Results of numerical experiments are provided.

REFERENCES

[1] Enkhbat, R., Barsbold, B. and Kamada, M. *A Numerical Approach for Solving Some Convex Maximization Problems.* Journal of Global Optimization, 35:85–101, Springer, 2006.

[2] Chiang, M. S. *Network Utility Maximization With Nonconcave Utilities Using Sum-of-Squares Method*, Proceedings of the MoC14.6 44th IEEE Conference on Decision and Control, and the European Control Conference 2005, Seville, Spain, December 12-15, 2005

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

[4] Horst, R. and Pardalos, P.M. (1995), *Handbook of Global Optimization,* Kluwer Academic Publishers, London.

[5] Pardalos, P.M. and S.A. Vavasis, *Quadratic Programming with One Negative Eigenvalue is NP-hard*, Journal of Global Optimization, 1, 15-22. 1990, 4.

[6] Rockafellar, R. T., *Convex analysis*, Princeton Universty Press, Princeton, NJ, 1970.

[7] Strekalovskiǐ, A., *On the problem of the global extremum.* Dokl. Akad. Nauk SSSR, 292(5):1062–1066, 1987.

[8] Tuy,H., *Monotonic Optimization: Problems and Solution Approaches*, SIAM J.OPTIM. Vol.11, No.2, 464-494, 2000.

[9] Tuy,H.,*Normal Sets,Polyblocks and Monotonic Optimization*,Vietnam J.Math., 27, 277-300, 1999.

[10] Tuy,H.,*Convex Analysis and Global Optimization*, Kluwer Academic Publishers, Dordrecht, 1998.