A Convexity Method for Linear Multiplicative Programming

Chunfeng Wang, Yaping Deng, Xiaodi Wu

***Abstract*—For solving linear multiplicative programming (P_0), a global optimization algorithm is proposed in this paper. First, by using logarithmic transformation, we equivalently transform problem (P_0) into an problem (P). Then, a mixed integer convex programming problem (M) is derived based on piecewise linear approximation for nonconvex part of the objective function in (P), which is a convex approximation of (P). Finally, by solving the problem (M), an ϵ-global optimum of (P_0) can be acquired. The numerical results verify the feasibility and effectiveness of this method.**

***Index Terms*—Linear multiplicative programming, Optimization, Piecewise linearization, Mixed-integer convex programming**

I. Introduction

THIS paper considers linear multiplicative programming problem (P_0) as follows:

$$(\mathrm{P}_0):\begin{cases} \min & f(x)=\prod_{i=1}^{p}(c_i^\top x+d_i)^{\gamma_i}, \\ \text{s.t.} & Ax\le b, \end{cases}$$

where

$$X=\{x\in R^n : Ax\le b\}$$

is nonempty and bounded,

$$A\in R^{m\times n}, b\in R^m, x, c_i\in R^n, d_i\in R,$$

and

$$c_i^\top x+d_i>0, i=1,\cdots,p.$$

The problem (P_0) has many applications in real-world, such as financial problem [1,2], data mining [3], VLSI chip design [4], and so on. In addition, the problem (P_0) also contains a wide category of mathematical programming problems, such as quadratic programming, bilinear programming, linear multiplicative programming, and so on. Thus, it is necessary to propose good algorithms, and it has attracted considerable attention from researchers. However, just as pointed out in [1], the problem (P_0) belongs to a nonconvex problem, and may have multiple local optima, most of which fail to be global optimal optima. In other words, it is hard to solve problem (P_0). During the past years, for solving (P_0), researchers have presented many approaches, such as, outer-approximation method [5]; branch-and-bound method [6,7]; level set method [8]; parameterization based method [9-11]; cutting plane method [12,13]; primal and dual simplex method [14]; heuristic method [15,16]. Although great progress has been made, how to globally solve the (P_0) still is a Gordian knot. On the basis of rectangular branch-and-bound frame, Kuno [17] presented an algorithm for minimizing a product of affine functions over a polytope. For solving a linear program with multiplicative constraints, Benson [18] presented a decomposition branch-and-bound algorithm. To solve a type of generalized linear multiplicative programming, Jiao [19] established a reliable and effective algorithm by utilizing the linear approximation of exponent and logarithmic functions. By combining a suitable deleting technique with the branch and bound scheme, Shen [20] put forward a new accelerating method for generalized linear multiplicative programming. Through using a lower bounding procedure and a new branching scheme, Ryoo et al. [21] presented a global optimization method for a generalized linear multiplicative program. Recently, for a special linear multiplicative optimization problem, Shen et al. [22,23] developed two branch and bound methods.

The goal of this paper is to design a method to globally solve (P_0). In this algorithm, first, by using an equivalent transformation, (P_0) is transformed into (P); then, for the nonconvex part of the objective function in problem (P), a piecewise linearization method is designed. Based on such piecewise linearization method, a mixed-integer convex programming (M) is derived. Finally, through solving (M), an ϵ-global optimum can be acquired. For a given ϵ, compared with other methods, the proposed method only needs to solve a mixed integer convex programming problem, which can be easily solved by LINGO.

The organization of this paper is as follows. In Section II, the equivalent transformation and the piecewise linearization method are introduced. Four specific examples are implemented to verify the efficiency and feasibility of the proposed algorithm in Section III. Conclusion is given in Section IV.

II. Piecewise linear approximation of logarithmic function

In the proposed method, the piecewise linear approximation plays a key role. To derive the piecewise linear approximation, the objective function $f(x)$ in problem (P_0) will be considered first. Without loss of generality, denote

$$T_1=\{i|\gamma_i>0\},$$

and

$$T_2=\{i|\gamma_i<0\}.$$

Manuscript received March 28, 2021; revised September 22, 2021. The research was supported by the National Natural Science Foundation of China(12071133).

C.F. Wang is an associate professor of the School of Mathematics and Statistics, Xianyang Normal University, Xianxiang, 712099, PR China; College of Mathematics and Information, Henan Normal University, Xinxiang, 453007, PR China. Email:wangchunfeng09@126.com

Y.P. Deng is a graduate student of the College of Mathematics and Information, Henan Normal University, Xinxiang, 453007, PR China. E-mail:1514349724@qq.com

X.D. Wu is a lecturer of the Faculty of Mathematical and Physics, Guangxi University for Nationalities, Nanning, 530006, PR China. E-mail:1710703068@qq.com

By using logarithmic transformation, $f(x)$ can be transformed into the following form:

$$\begin{aligned} g(x) &= \ln(f(x)) \\ &= \sum_{i=1}^{p} \gamma_i \ln({c_i}^\top x + d_i) \\ &= \sum_{i \in T_1} \gamma_i \ln({c_i}^\top x + d_i) + \sum_{i \in T_2} \gamma_i \ln({c_i}^\top x + d_i). \end{aligned}$$

Thus, the problem (P_0) can be equivalently transformed into problem (P) as follows:

$$\begin{cases} \min\ g(x) = \sum_{i \in T_1} \gamma_i \ln({c_i}^\top x + d_i) + \sum_{i \in T_2} \gamma_i \ln({c_i}^\top x + d_i), \\ \text{s.t. } Ax \le b. \end{cases}$$

The equivalence between problems (P_0) and (P) is given by the following theorem.

Theorem 1. If x^* is an optimum of problem (P_0), then x^* is an optimum of problem (P). Conversely, if x^* is an optimum of problem (P), then x^* is an optimum of problem (P_0).

Proof. According the process of transformation, it is not difficult to derive the equivalence between (P_0) and (P).

According to Theorem 1, to solve (P_0), we can solve the problem (P) instead. Next, will show how to solve (P).

For the function $g(x)$ in (P), there are two parts

$$\sum_{i \in T_1} \gamma_i \ln({c_i}^\top x + d_i)$$

and

$$\sum_{i \in T_2} \gamma_i \ln({c_i}^\top x + d_i).$$

We discuss them separately.

For the first part: since

$$\ln({c_i}^\top x + d_i)$$

is a concave function, and $\gamma_i > 0$,

$$\sum_{i \in T_1} \gamma_i \ln({c_i}^\top x + d_i)$$

is a concave function.

For the second part: since

$$\ln({c_i}^\top x + d_i)$$

is a concave function, and $\gamma_i < 0$,

$$\sum_{i \in T_2} \gamma_i \ln({c_i}^\top x + d_i)$$

is a convex function.

Obviously, the nonconvexity of problem (P) is caused by

$$\sum_{i \in T_1} \gamma_i \ln({c_i}^\top x + d_i).$$

Therefore, to solve problem (P), for the nonconvex part

$$\sum_{i \in T_1} \gamma_i \ln({c_i}^\top x + d_i),$$

this paper propose a convex technique based on piecewise linear approximation of logarithmic function.

The specific process is as follows: for $i \in T_1$, let

$$y_i = {c_i}^\top x + d_i.$$

Through solving $2|T_1|$ linear programming problems, it is easy to obtain the lower bound and upper bound of y_i,

$$l_i = \min_{x \in X} {\mathrm{c_i}}^\top x + d_i,$$

$$u_i = \max_{x \in X} {c_i}^\top x + d_i,$$

that is

$$l_i \le y_i \le u_i,\ i \in T_1.$$

Thus, problem (P) can be transformed into the following form:

$$\begin{cases} \min & g(x) = \sum_{i \in T_1} \gamma_i \ln(y_i) + \sum_{i \in T_2} \gamma_i \ln({c_i}^\top x + d_i), \\ \text{s.t.} & Ax \le b, \\ & y_i = {c_i}^\top x + d_i, \\ & l_i \le y_i \le u_i,\ i \in T_1. \end{cases}$$

A. Linearization of single logarithm function

To derive the piecewise linear approximation function of

$$\sum_{i \in T_1} \gamma_i \ln({c_i}^\top x + d_i),$$

first of all, we consider one term, i.e. $\gamma_i \ln y_i$. Furthermore, the piecewise linear approximation function of multinomial is given.

To intuitively illustrate the piecewise linear approximation method, Fig. 1 below shows piecewise linear lower estimation of $p^i(y_i)$ with three segments, where s_i $(i = 0, 1, 2, 3)$ are segmentation points.

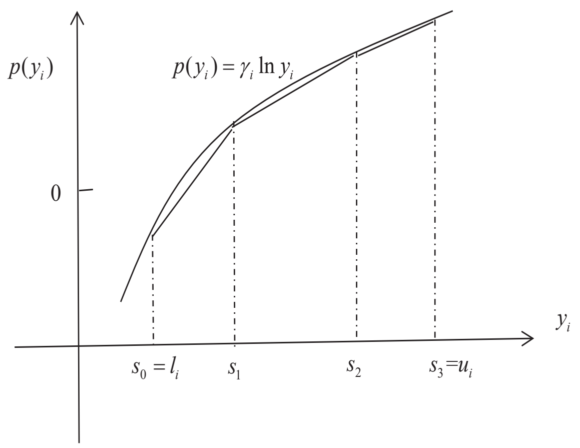

Fig.1 Piecewise linear lower estimation with three segments

From Fig.1, it can be seen that, each linear lower estimator has error with the original function. Meanwhile, we can see that the error between the linear lower estimator and the original function can be further reduced by adding more segmentation points. Thus, for ensuring the approximation degree of the piecewise linear approximation method, it can be controlled by a given error ϵ. For the given ϵ, we first need to determine the segmentation points.

To this end, let

$$p^i(y_i) = \gamma_i \ln y_i,$$

where $l_i \le y_i \le u_i$.

Without loss of generality, consider interval $[s_1^i, s_2^i]$. Obviously, the slope of secant over this interval is

$$k_2^i = \frac{p^i(s_2^i) - p^i(s_1^i)}{s_2^i - s_1^i}.$$

In this paper, to determine the segmentation points, the maximum error between the secant and the original function is used. The process is as follows: assume that the maximum error between secant and original function is achieved at the point $y_i = t$, then we have

$$(p^i(t))' = k_2^i.$$

Furthermore, we have

$$t = \frac{r_i(s_2^i - s_1^i)}{p^i(s_2^i) - p^i(s_1^i)}.$$

Let the maximum error at $y_i = t$ be ϵ, that is:

$$\epsilon = k_2^i(t - s_1^i) + p^i(s_1^i) - p^i(t).$$

By solving this equation, we can get the next segmentation point s_2^i.

For a given ϵ, the following Algorithm 1 gives the detailed process to determine these segmentation points.

Algorithm 1: Determining segmentation points

01: Initialization: Given $\epsilon > 0$, let $j = 0, s_j^i = l_i, n = 0$.
02: Using the method of root to determine $s_{j+1}^i > s_j^i$, such that the maximum error between secant and original function is ϵ over the interval $[s_j^i, s_{j+1}^i]$.
03: Set $j = j + 1$.
04: If $s_j^i > u_i$
05: set $s_j^i = u_i, n = j$, then stop;
06: else
07: return to step 1.
08: End if
09: Output segmentation points.

According to Algorithm 1, when it terminates, these segmentation points can be obtained, and the maximum error between secant and original function is no more than ϵ.

Based on these segmentation points, the construction process of piecewise linear function is given below. Let

$$k_j^i = \frac{p^i(s_j^i) - p^i(s_{j-1}^i)}{s_j^i - s_{j-1}^i}$$

be the slope of secant over the j-th interval,

$$\Delta s_j^i = s_j^i - s_{j-1}^i, j = 1, \cdots, n,$$

be the length of this interval. By introducing binary variables $u_1^i, u_2^i, \cdots, u_n^i$, we can get piecewise linear function of $p^i(y_i)$ as follows:

$$\underline{p}^i(y_j) = p^i(s_0^i) + \sum_{j=1}^{n} k_j^i p_j^i,$$

where

$$y_j = s_0^j + \sum_{j=1}^{n} p_j^i,$$

$$\Delta s_j^i u_{j+1}^i \le p_j^i \le \Delta s_j^i u_j^i, j = 1, ..., n,$$

$$0 \le p_n^i \le \Delta s_n^i u_n^i.$$

Based on the above, it is easy to know that

$$p^i(y_i) - \underline{p}^i(y_i) \le \epsilon,$$

i.e.

$$p^i(y_i) \le \underline{p}^i(y_i) + \epsilon.$$

Set

$$\overline{p}^i(y_i) = \underline{p}^i(y_i) + \epsilon.$$

Obviously, it is an upper estimation function of the original function.

B. Piecewise linear approximation function of multinomial

In this subsection, we consider the piecewise linear approximation function for the sum term

$$\sum_{i \in T_1} \gamma_i \ln({c_i}^\top x + d_i).$$

Based on the previous subsection, let

$$p(y) = \sum_{i \in T_1} \gamma_i \ln(y_i),$$

$$p^i(y_i) = \gamma_i \ln(y_i),$$

and

$$s_0^i = l_i, \ s_{n_i}^i = u_i,$$

we can get a convex approximation (M) for (P), which is given as follows:

$$(\mathrm{M}): \begin{cases} \min \sum\limits_{i \in T_1} (p^i(s_0^i) + \sum\limits_{j=1}^{n_i} k_j^i p_j^i) + \sum\limits_{i \in T_2} \gamma_i \ln({c_i}^\top x + d_i), \\ \text{s.t. } \Delta s_j^i u_{j+1}^i \le p_j^i \le \Delta s_j^i u_j^i, \\ \quad 0 \le p_{n_i}^i \le \Delta s_{n_i}^i u_{n_i}^i, \ j = 1, ..., n_i, \\ \quad Ax \le b, \end{cases}$$

where

$$k_j^i = \frac{p^i(s_j^i) - p^i(s_{j-1}^i)}{s_j^i - s_{j-1}^i}$$

is the slop of the i-th logarithmic function over the j-th interval, and

$$\Delta s_j^i = s_j^i - s_{j-1}^i$$

is the length of the i-th logarithmic function, u_j^i is a binary variable. Obviously, (M) is a mixed-integer convex programming.

The following Theorem 2 gives the relationship between the optima of the problem (M) and the problem (P).

Theorem 2. If x^* is a global optimum of problem (M), then x^* is an ϵ-global optimum of problem (P).

Proof. The conclusion can be derived easily from the derivation process.

According to theorems 1 and 2, to obtain an ϵ-global optimal solution of (P_0), we can solve the mixed-integer convex programming (M) instead.

III. Numerical experiments

In this section, to test the feasibility of the proposed algorithm, four numerical experiments are carried on. In these four examples, the termination error $\epsilon = 0.01$. The algorithm is coded in Matlab (2018a), and the test examples are implemented on the microcomputer with dual process Intel(R) Core(TM) i5-4200M CPU (2.5GHz), LINGO is used to solve mixed-integer convex programming problem.

Example 1.

$$\begin{array}{ll} \min & (x_1+x_2)\times(x_1-x_2+7) \\ \text{s.t.} & 2x_1+x_2\le 14, \\ & x_1+x_2\le 10, \\ & -4x_1+x_2\le 0, \\ & 2x_1+x_2\ge 6, \\ & x_1+2x_2\ge 6, \\ & x_1-x_2\le 3, \\ & x_1+x_2\ge 0, \\ & x_1-x_2+7\ge 0, \\ & x_1,x_2\ge 0. \end{array}$$

The optimal solution is (2,8), and the optimal value is 10.

Example 2.

$$\begin{array}{ll} \min & (x_1+x_2+1)^{1.01}\times(2x_1+x_2+1) \\ & \times(x_1+2x_2+1)^{1.03} \\ \text{s.t.} & x_1+2x_2\le 6, \\ & 2x_1+2x_2\le 8, \\ & 1\le x_1\le 3, \\ & 1\le x_2\le 3. \end{array}$$

The optimal solution is (1,1), and the optimal value is 50.5911.

Example 3.

$$\begin{array}{ll} \min & (-4x_1-2x_4+3x_5+21)\times(4x_1+2x_2+3x_3 \\ & -4x_4+4x_5-3)\times(3x_1+4x_2+2x_3-2x_4 \\ & +2x_5-7)\times(-2x_1+x_2-2x_3+2x_5+11) \\ \text{s.t.} & 4x_1+4x_2+5x_3+3x_4+x_5\le 25, \\ & -x_1-5x_2+2x_3+3x_4+x_5\le 2, \\ & x_1+2x_2+x_3-2x_4+2x_5\ge 6, \\ & 4x_2+3x_3-8x_4+11x_5\ge 8, \\ & x_1+x_2+x_3+x_4+x_5\le 6, \\ & x_1,x_2,x_3,x_4,x_5\ge 1. \end{array}$$

The optimal solution is (1,2,1,1,1), and the optimal value is 9504.

Example 4.

$$\begin{array}{ll} \min & (0.813396\mathrm{x}_1+0.6744\mathrm{x}_2+0.305038\mathrm{x}_3 \\ & +0.129742\mathrm{x}_4+0.217796)\times(0.224508\mathrm{x}_1 \\ & +0.063458\mathrm{x}_2+0.93223\mathrm{x}_3+0.528736\mathrm{x}_4 \\ & +0.091947) \\ \text{s.t.} & 0.488509\mathrm{x}_1+0.063458\mathrm{x}_2+0.945686\mathrm{x}_3 \\ & +0.210704\mathrm{x}_4\le 3.562809, \\ & -0.324014\mathrm{x}_1-0.501754\mathrm{x}_2-0.719204\mathrm{x}_3 \\ & +0.099562\mathrm{x}_4\le -0.052215, \\ & 0.445225\mathrm{x}_1-0.346896\mathrm{x}_2+0.637939\mathrm{x}_3 \\ & -0.257623\mathrm{x}_4\le 0.42792, \\ & -0.202821\mathrm{x}_1+0.647361\mathrm{x}_2+0.920135\mathrm{x}_3 \\ & -0.983091\mathrm{x}_4\le 0.84095, \\ & -0.886420\mathrm{x}_1-0.802444\mathrm{x}_2-0.305441\mathrm{x}_3 \\ & -0.180123\mathrm{x}_4\le -1.353686, \\ & -0.515399\mathrm{x}_1-0.424820\mathrm{x}_2+0.897498\mathrm{x}_3 \\ & +0.187268\mathrm{x}_4\le 2.137251, \\ & -0.591515\mathrm{x}_1+0.060581\mathrm{x}_2-0.427365\mathrm{x}_3 \\ & +0.579388\mathrm{x}_4\le -0.290987, \\ & 0.423524\mathrm{x}_1+0.940496\mathrm{x}_2-0.437944\mathrm{x}_3 \\ & -0.742941\mathrm{x}_4\le 0.37362, \\ & \mathrm{x}_1,\mathrm{x}_2,\mathrm{x}_3,\mathrm{x}_4\ge 0. \end{array}$$

The optimal solution is (1.3148,0.1396,0,0.4233), and the optimal value is 0.8902.

IV. CONCLUSION

For problem (P_0), through using piecewise linear approximation function, a mixed-integer convex programming problem (M) is derived. By solving (M), we can obtain an ϵ-global optimum of (P_0). Numerical experiments show that this method is feasible and effective. As pointed out above, the advantage of the proposed method is that the algorithm only needs to solve a mixed integer linear programming problem with a given calculation error ϵ. The disadvantage is that a large number of binary variables may be introduced for large-scale problem algorithm. In future work, we will study whether there are other better methods to generate segmentation points.

REFERENCES

[1] Avriel M, Diewert W E, Schaible S. *Generalized Convexity*, New York: Plenum Press,1988.

[2] Maranas C D, Androulakis I P, Floudas C A, et al. "Solving long-term financial planning problems via global optimization", *Journal of Economic Dynamics and Control*, vol. 21, no. 8-9, pp. 1405-1425, 1997.

[3] Benett K P, Mangasarian O L. "Bilinear separation of two sets in n-space", *Computational Optimization and Applications*, vol. 2, no. 3, pp. 207-227, 1994.

[4] Dorneich M C, Sahinidis N V. "Global optimization algorithms for chip layout and compaction", *Engineering Optimization*, vol. 25, no. 2, pp. 131-154, 1995.

[5] Gao Y L, Xu C X, Yang Y T. "An outcome-space finite algorithm for solving linear multiplicative programming", *Applied Mathematics and Computation*, vol. 179, no. 2, pp. 494-505, 2006.

[6] Wang C F, Liu S Y. "A new linearization method for generalized linear multiplicative programming", *Computers and Operations Research*, vol. 38, no. 7, pp. 1008-1013, 2011.

[7] Wang C F, Chu X Y. "A new branch and bound method for solving sum of linear ratios problem", *IAENG International Journal of Applied Mathmatics*, vol. 47, no. 3, pp. 276-281, 2017.

[8] Liu S Y, Zhao Y F. "An efficient algorithm for globally solving generalized linear multiplicative programming", *Journal of Computational and Applied Mathematics*, vol. 296, no. 1, pp. 840-847, 2016.

[9] Konno H, Yajima Y, Matsui T. "Parametric simplex algorithms for solving a special class of nonconvex minimization problems", *Journal of Global Optimization*, vol. 1, no. 1, pp. 65-81, 1991.

[10] Konno H, Kuno T, Yajima Y. "Global minimization of a generalized convex multiplicative function", *Journal of Global Optimization*, vol. 4,no. 1, pp. 47-62, 1994.

[11] Van Thoai N. "A global optimization approach for solving the convex multiplicative programming problem", *Journal of Global Optimization*, vol. 1, no. 4, pp. 341-357, 1991.

[12] Konno H, Kuno T, Yajima Y. "Global minimization of a generalized convex multiplicative function", *Journal of Global Optimization*, vol. 4, no. 1, pp. 47-62, 1994.

[13] Benson H P, Boger G M. "Outcome-space cutting-plane algorithm for linear multiplicative programming", *Journal of Optimization Theory and Applications*, vol. 104, no.2, pp. 301-322, 2000.

[14] Schaible S, Sodini C. "Finite algorithm for generalized linear multiplicative programming", *Journal of Optimization Theory and Applications*, vol. 87, no.2, pp. 441-455, 1995.

[15] Benson H P, Boger G M. "Multiplicative programming problems: Analysis and efficient point search heuristic", *Journal of Optimization Theory and Applications*, vol. 94, no. 2, pp. 487-510, 1997.

[16] Liu X J, Umegaki T, Yamamoto T. "Heuristic methods for linear multiplicative programming", *Journal of Global Optimization*, vol. 15, no. 4, pp. 433-447, 1999.

[17] Kuno T. "A finite branch-and-bound algorithm for linear multiplicative programming", *Computational Optimization And Applications*, vol. 20, no. 2, pp. 119-135, 2001.

[18] Benson H P. "Decomposition branch and bound based algorithm for linear programs with additional multiplicative constraints", *Journal of Optimization Theory and Applications*, vol. 126, no. 1, pp. 41-46, 2005.

[19] Jiao H W. "A branch and bound algorithm for globally solving a class of nonconvex programming problems", *Nonlinear Analysis*, vol. 70, no. 2, pp. 1113-1123, 2008.

[20] Shen P P, Bai X D, Li W M. "A new accelerating method for globally solving a class of nonconvex programming problems", *Nonlinear Analysis*, vol. 71, no. 7-8, pp. 2866-2876, 2009.

[21] Ryoo H S, Sahinidis N V. “Global optimization of multiplicative programs”, *Journal of Global Optimization*, vol. 26, no. 4, pp. 387-418, 2003.
[22] Shen P P, Huang B D. “Global algorithm for solving linear multiplicative programming problems”, *Optimization Letters*, vol. 14, no. 3, pp. 1-18, 2020.
[23] Shen P P, Wang K M, Lu T. “Global optimization algorithm for solving linear multiplicative programming problems”, *Optimization*, DOI:10.1080/02331934.2020.1812603, 2020.