日本オペレーションズ・リサーチ学会
2004年秋季研究発表会

逆凸制約付き線形計画問題に対する解法

02402070　筑波大学　*永井 秀稔　NAGAI Hidetoshi
01107230　筑波大学　久野 誉人　KUNO Takahito

1　逆凸制約付き線形計画問題

本研究で対象とする逆凸制約付き線形計画問題 (Linear Programs with an additional Reverse Convex constraint) とは，線形計画問題 (LP) に逆凸制約が加えられた問題である．行列 $\mathbf{A} \in \mathbb{R}^{m \times n}$, 列ベクトル $\mathbf{b} \in \mathbb{R}^m$, 行ベクトル $\mathbf{c} \in \mathbb{R}^n$, 凸関数 $g : \mathbb{R}^n \mapsto \mathbb{R}^1$, に対し、以下のように定式化される．

$$(LPRC)\left|\begin{array}{ll} \min\ z = & \mathbf{cx} \\ \text{s.t.} & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0} \\ & g(\mathbf{x}) \geq 0 \end{array}\right.$$

また，$D = \{\mathbf{x} \mid \mathbf{Ax} \leq \mathbf{b},\ \mathbf{x} \geq \mathbf{0}\}$, $G = \{\mathbf{x} \mid g(\mathbf{x}) < 0\}$ とすれば，$\min\{\mathbf{cx} \mid \mathbf{x} \in D \setminus G\}$ とも表現できる．凸関数 g に対し，$g(\mathbf{x}) \leq 0$ であれば実行可能領域は凸集合となるのだが，不等号が逆向きであるが故に，非凸集合となるどころか，非連結となることもあり得る．この問題は，製品開発や規模の経済効果を考慮した制約を持つ問題などをはじめ，多くの分野で見受けられる．本研究の目的は，この逆凸制約付き線形計画問題の大域的最適解を効率的に生成するアルゴリズムを開発することである．よって以降，単に最適解と言えば，大域的最適解を指すものとする．

この問題に対し，先行研究で提案されているアルゴリズムは，概ね以下の 4 種に分けられる．

- Simplex-type pivoting [3,7]: pivot により，G に含まれる D の端点をすべて列挙する．各端点に対し，これと，隣接する端点とを結ぶ辺上に局所解が無いか調べる．
- Alternating local search and concave minimization [4,10]: $\partial G = \{\mathbf{x} \mid g(\mathbf{x}) = 0\}$ 上の局所解を求めることと，大域性確認のための凹最小化問題を解くことを交互に行う．
- Outer approximation [2,5]: concavity cut や facial cut などを用いて，最適解を除いた D の辺上の点をすべて cut する．
- Conical branch-and-bound [8,9]: 錐で分割しながら，凸関数 g を線形緩和した問題を解いていく．

これらに対し本研究では，Simplex-type pivoting に属するアルゴリズムを改良し，記憶量に関して改善を試みた．

2　$(LPRC)$ の性質

$(LPRC)$ に対し，次の仮定を置く．

1. 実行可能領域 $D \setminus G$ は非空有界．
2. $\underset{\min\{\mathbf{cx} \mid \mathbf{x} \in D \setminus G\}}{(LPRC)} > \underset{\min\{\mathbf{cx} \mid \mathbf{x} \in D\}}{(LP)}$.

これらの仮定の下で，$(LPRC)$ の最適解について以下の性質を持つことが知られている．

1. $\partial G = \{\mathbf{x} \mid g(\mathbf{x}) = 0\}$ と D の辺との交点上に最適解が存在する．
2. D の辺 $\mathbf{v}^1$–$\mathbf{v}^2 : \mathbf{cv}^1 \geq \mathbf{cv}^2$ のうち，$g(\mathbf{v}^1) \geq 0$, $g(\mathbf{v}^2) < 0$ を満たすものの中に最適解が存在する．

3　Simplex-type pivoting

上述した性質より，D の辺を列挙することで最適解を生成するアルゴリズムが考えられる．それが Simplex-type pivoting である．すなわち，pivot により G に含まれる D の端点を列挙していき，列挙された端点と，それに隣接する端点とを結ぶ辺上に局所解が無いか調べていくことで，最適解を得る．基本的なアルゴリズムの枠組みは以下のようになる [3].

$N(\mathbf{x})$: $\mathbf{x}$ の隣接端点の集合
S : 列挙した端点の集合
T : S のうち，隣接端点の列挙を終えた端点の集合

Step 0. $\mathbf{x}^* := \max\{\mathbf{cx} \mid \mathbf{x} \in D\}$. $\mathbf{x}^0 := \min\{\mathbf{cx} \mid \mathbf{x} \in D\}$. $S := \{\mathbf{x}^0\}$. $T := \emptyset$.

Step 1. If $S \setminus T = \emptyset$, then $\mathbf{x}^*$ is optimal, terminate. Otherwise, select $\mathbf{x} \in S \setminus T$. $T := T \cup \{\mathbf{x}\}$. Calculate $N(\mathbf{x})$.

Step 2. If $N(\mathbf{x}) = \emptyset$, then go to Step 1. Otherwise, select $\mathbf{v} \in N(\mathbf{x})$. $N(\mathbf{x}) := N(\mathbf{x}) \setminus \{\mathbf{v}\}$.

Step 3. If $\mathbf{cv} > \mathbf{cx}$ and $g(\mathbf{v}) \geq 0$, then go to Step 4. If $\mathbf{cx}^* > \mathbf{cv} > \mathbf{cx}$ and $g(\mathbf{v}) < 0$ and $\mathbf{v} \notin S$, then $S := S \cup \{\mathbf{v}\}$. Go to Step 2.

Step 4. $\alpha^* := \min\{\alpha \mid g(\mathbf{x} + \alpha(\mathbf{v} - \mathbf{x})) = 0\}$. $\mathbf{x}' := \mathbf{x} + \alpha^*(\mathbf{v} - \mathbf{x})$. If $\mathbf{cx}' < \mathbf{cx}^*$, then $\mathbf{x}^* := \mathbf{x}'$. Go to Step 2.

4 本研究のアルゴリズム

前節で述べたアルゴリズムでは，問題サイズが大きくなるにつれ，S や T の要素数が爆発的に増大することが問題点である．[7] では，深さ優先探索を用いることにより，S を必要としないアルゴリズムを提案しているが，T を用いる以上，この問題点に関しての解決策にはなっていない．そこで本研究では，S や T を用いずに重複なく全端点を列挙できる別の方法を利用する．その方法とは，Reverse Bland pivoting [1] である．

4.1 Reverse Bland pivoting

LP を解く際に Bland rule(最小添字規則) を用いると，どの端点も，pivot で次に移る端点が一意に定まる．よって全端点について，その pivot する経路を繋げば，LP の最適解を根とする木が構成される．この木を探索することで，重複なく全端点を列挙できるので，今いる端点の基底情報のみ保持すればよい．

4.2 Concavity cut

G に含まれる D のある端点 $\mathbf{x}$ に対し，$\mathbf{x}$ からその隣接端点 $\mathbf{v}^i$ 方向への半直線と ∂G との交点を $\mathbf{y}^i$ とすると，関数 g の凸性より，$\{\mathbf{x}, \mathbf{y}^1, \ldots, \mathbf{y}^n\}$ から得られる単体に含まれる点は，その端点を除いてすべて実行不能である．よって，$\{\mathbf{y}^1, \ldots, \mathbf{y}^n\}$ を通る超平面で cut を入れても，$(LPRC)$ の実行可能領域が削られることはない．これを concavity cut あるいは Tuy-cut と呼ぶ [6].

そこで，$\min\{\mathbf{cx} \mid \mathbf{x} \in D \cap H\}$, H : concavity cut を満たす領域，の最適解 $\mathbf{x}^*$ を求め，$\mathbf{x}^*$ が $(LPRC)$ に対し実行不能であれば，再び $\mathbf{x}^*$ を用いて concavity cut を生成することを繰り返すアルゴリズムが考えられる．残念なことに，このアルゴリズムは収束性が示されていないが，多くの問題に対し，非常に少ない繰り返し回数で最適解を得られることが実験により確かめられた．これを踏まえ本研究では，繰り返し回数に上限 ($2n$ 回) を設け，上限に達するまでに実行可能解が得られなければ，このアルゴリズムで解くことを諦め，Reverse Bland pivoting により最適解を求める，というアルゴリズムを提案する．

参考文献

[1] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry*, 8:295–313, 1992.

[2] J. Fülöp. A finite cutting plane method for solving linear programs with an additional reverse convex constraint. *European Journal of Operational Research*, 44:395–409, 1990.

[3] R. J. Hillestad. Optimization problems subject to a budget constraint with economies of scale. *Operations Research*, 23:1091–1098, 1975.

[4] R. J. Hillestad and S. E. Jacobsen. Linear programs with an additional reverse convex constraint. *Applied Mathematics and Optimization*, 6:257–269, 1980.

[5] R. J. Hillestad and S. E. Jacobsen. Reverse convex programming. *Applied Mathematics and Optimization*, 6:63–78, 1980.

[6] R. Horst and H. Tuy. *Global Optimization: Deterministic Approaches*. Springer-Verlag, Berlin, 3 edition, 1996.

[7] S. E. Jacobsen and K. Moshirvaziri. Computational experience using an edge search algorithm for linear reverse convex programs. *Journal of Global Optimization*, 9:153–167, 1996.

[8] K. Moshirvaziri and M. A. Amouzegar. A subdivision scheme for linear programs with an additional reverse convex constraint. *Asia-Pacific Journal of Operations Research*, 15:179–192, 1998.

[9] L. D. Muu. A convergent algorithm for solving linear programs with an additional reverse convex constraint. *Kybernetika*, 21:428–435, 1985.

[10] N. V. Thuong and H. Tuy. A finite algorithm for solving linear programs with an additional reverse convex constraint. *Lecture Note in Economics and Mathematical Systems*, 255:291–302, 1984.