韓國ＯＲ學會誌
第６卷 第１號
1981年 4月

線型計劃法에 대한 Khachiyan 方法의 응용연구

(The Application of Khachiyan's Algorithm for Linear Programming: State of the Art)

姜 錫 昊*
朴 河 英*

Abstract

L.G. Khachiyan's algorithm for solving a system of strict (or open) linear inequalities with integral coefficients is described. This algorithm is based on the construction of a sequence of ellipsoids in R^n of decreasing n-dimensional volume and containing feasible region. The running time of the algorithm is polynomial in the number of bits of computer core memory required to store the coefficients. It can be applied to solve linear programming problems in polynomially bounded time by the duality theorem of the linear programming problem. But it is difficult to use in solving practical problems. Because it requires the computation of a square roots, besides other arithmatic operations, it is impossible to do these computations exactly with absolute precision.

I. 序 論

이제까지 線型計劃法問題를 푸는데 사용되어온 Simplex algorithm은 일반적으로 사용되는 entering variable과 dropping variable 선택법을 사용했을 때 最惡의 경우 문제를 푸는데 드는 노력이 문제가 포함하고 있는 變數의 갯수에 대해 指數的으로 증가하는 것으로 알려져 있다. 그리고 最惡의 경우에 문제가 포함하고 있는 變數의 갯수에 대해 多項的으로 제한되는 문제를 푸는 노력을 보장하는 entering variable과 dropping variable의 선택법이 아직 발견되지 않은 상태이다.

과연 "線型計劃法問題를 多項的으로 제한된 노력으로 풀 수 있는 algorithm(polynomially bounded algorithm)이 존재할까?" 하는 문제가 소련의 Leonid Genrikhovich Khachiyan이 Doklady Akademiia Nauk SSSR 244:5(1979)(영어번역 : Soviet-Math-Doklady-Journal)에 발표한 algorithm에 의해 存在하는 것으로 판명되었다.

여기에서 이야기하는 線型計劃法問題의 크기는 문제를 컴퓨터에 저장하는데 필요한 core memory의 bit 수로 한다. 따라서 다음과 같은 線型計劃法問題의 크기는 다음의 L이 된다.

$$\begin{aligned} &\text{maximize } C^T X \\ &\text{subject to } AX \le b \qquad (1) \\ &\qquad X \ge 0 \end{aligned}$$

C : order $n\times 1$
A : order $m\times n$
b : order $m\times 1$
(모든 숫자는 整數, $m\ge 2$, $n\ge 2$)

$$L = \sum_{\substack{1\le i\le m \\ 1\le j\le n}} [\log_2(|a_{ij}|+1)] + \sum_{1\le i\le m} [\log_2(|b_i|+1)] + \log_2 mn + 1 \qquad (2)$$

*서울大學校 工科大學 産業工學科

Ⅱ. Khachiyan의 algorithm

이 algorithm은 整數를 係數로 하는 strict(또는 open) inequality들의 system을 푸는 algorithm이다. ($AX<b$) 여기에서 중요한 사실은 (a) strict inequalities가 사용되었다는 것과 (b) A와 b가 모두 整數 係數를 갖는다는 것이다. (a)는 nonempty feasible region은 반드시 open set이어야 하고 positive volume을 갖는다는 것을 의미한다. (b)에 의해 문제에 不連續性이 부여됨으로 인해 찾으려는 feasible region 안의 점들은 R^n 안의 임의의 점들이 아니고 제한된 分母를 갖는 有理數로 이루어진 점들이라는 사실을 의미하게 된다. 또한 평행하지 않은 hyperplane을 갖는 $AX<b$ 안의 어느 두 制限條件도 서로가 서로의 기울기를 制限하게 된다. 그러므로 feasible region이 存在한다면 이 feasible region 안에 어떤 주어진 體積을 갖고 원점으로부터 어떤 주어진 거리에 있는 집합 S가 存在함을 알 수 있다.

algorithm은 원점에 中心을 갖고 S를 포함할 만큼 충분히 커다란 半經을 갖는 球에서 출발한다. 그후 連續的으로 S를 포함하며 減少하는 體積을 갖는 n-次元의 타원을 만들어 나간다. (이러한 이유로 ellipsoid algorithm이라고도 한다.) 타원의 中心이 infeasible 하면 만족시키지 못한 制限條件에 평행하며 中心을 지나는 hyperplane으로 타원을 半으로 자른다. 한쪽의 半은 완전히 infeasible하며 다른 半을 포함하는 타원을 만든다. 이러한 절차를 계속해 나간다. 끝까지 $AX<b$의 解를 찾지 못한다면 S보다 적은 體積을 갖는 타원으로 끝나게 되며 이것은 S를 포함한다는 가정과 不一致한다. 이러한 不一致는 feasible region이 存在하지 않는다는 것을 의미한다.

algorithm을 설명할 때 $AX<b$를 사용하며 $A\in Z^{m\times n}$, $b\in z^m$이고 하나의 inequality는 $a_i^TX<b_i$로 나타내고 $0\neq a_i\in z^n$, $b_i\in z$이다.

Khachiyan의 algorithm

step 1. (Initialize)

$X^{(0)}$를 0, $B^{(0)}$를 $2^{2L}I$ 그리고 k를 0으로 한다.

step 2. (Terminate?)

$X^{(k)}$가 $AX<b$의 解이면 계산을 끝내고 $X^{(k)}$를 feasible solution으로 한다.

그렇지 않은 경우 $k<4(n+1)^2L$이면 step 3으로 간다.

그렇지 않은 경우 解가 存在하지 않는 것으로 하고 끝낸다.

step 3. (Next iteration)

$X^{(k)}$가 만족되지 않는 inequality를 $AX<b$ 중에서 찾는다. 예를 들어 그 inequality가 $a_i^TX^{(k)}\geq b_i$라고 하면,

$$X^{(k+1)}\leftarrow X^{(k)}-\frac{1}{n+1}\ \frac{B^{(k)}a_i}{\sqrt{a_i^TB^{(k)}a_i}}, \tag{3}$$

$$B^{(k+1)}\leftarrow\frac{n^2}{n^2-1}\left[B^{(k)}-\frac{2}{n+1}\ \frac{(B^{(k)}a_i)(B^{(k)}a_i)^T}{a_i^TB^{(k)}a_i}\right], \tag{4}$$

$k\leftarrow k+1$

로 한 뒤 step 2로 돌아간다.

X'와 對稱의 陽의 定符號(symmetric positive definite) 行列 B가 주어지면 집합

$$E=\{X\,|\,(X-X')^TB^{-1}(X-X')\leq 1\} \tag{5}$$

는 X'를 中心으로 하는 타원을 나타낸다. $\frac{1}{2}E_a$가 반타원을 나타낸다고 하면,

$$\frac{1}{2}E_a=E\cap\{X\,|\,a^T(X-X')\leq 0\} \tag{6}$$

이고 E'를 집합 $\{y\,|\,(y-y')^T(B')^{-1}(y-y')\leq 1\}$이라고 한다.

$$y'=X'-\frac{1}{n+1}\ \frac{Ba}{\sqrt{a^TBa}}, \tag{7}$$

$$B'=\frac{n^2}{n^2-1}\left(B-\frac{2}{n+1}\ \frac{(Ba)(Ba)^T}{a^TBa}\right). \tag{8}$$

E'는 y'를 中心으로 하는 타원이며 $\frac{1}{2}E_a\subseteq E'$이고 E'의 體積 $\lambda(E')$는 $\lambda(E)$보다 적다는 것을 보일 수 있다.

algorithm이 解가 存在하기만 한다면 반드시 解에 도달한다는 것을 보여주는 몇 가지 定理들을 소개하면 다음과 같다. (참고문헌 1, 4, 9 : 증명은 생략)

Lemma 1.

Every vertex V of polyhedron

$AX\leq b$

$X\geq 0$

satisfies $||V||_\infty<2^L/mn$. Furthermore, its cordinates are rational numbers with denominators less than $2^L/mn$ in absolute value.

Corollary 1.

Every vertex V of polytope

$AX\leq b$

$X\geq 0$

$X\leq[2^L/n]e$

$e=(1,\ 1,\ \cdots\cdots,\ 1)^T$

has cordinates that are rational numbers with denominators less than $2^L/mn$ in absolute value.

Lemma 2.

If $AX<b$ has a solution, then the volume of its

solution space inside the sphere $||X||_2<2^L$ is at least $2^{-(n+1)L}$

다음의 세 Lemma들은 affine transformation하에서 invariant하다. 그러므로 $X'=0$, $B=I$ 그리고 $a=(-1, 0, \cdots\cdots, 0)$라고 가정한다. 즉, 타원 E는 원점에서의 unit sphere이며 접하는 half-space는 $x_1\geq 0$이라고 가정한다. 그러면,

$$y'=\left(\frac{1}{n+1},\ 0,\ \cdots\cdots,\ 0\right)^T,$$

$$B'=\mathrm{diag}\left(\frac{n^2}{(n+1)^2},\ \frac{n^2}{n^2-1},\ \cdots\cdots,\ \frac{n^2}{n^2-1}\right)$$

이다. 다음의 3가지 lemma들이 affine transformation 하에서 invariant하다는 사실을 확실히 하기 위해 다음을 소개한다.

Proposition 1.

Let B be an $n\times n$ symmetric positive definite matrix, let $a\neq 0$ and X' be n-vectors and let α, β, and $\gamma<1$ be positive constants.

Define

$$E=\{X|X^TX\leq 1\}$$

$$\overline{E}=\{X|(X-X')^TB^{-1}(X-X')\leq 1\}$$

Let $V=(-1,\ 0,\ \cdots\cdots,\ 0)^T$, and define

$$E'=\{X|(X+\alpha V)^T\beta(I-\gamma(VV^T))^{-1}(X+\alpha V)\leq 1\}$$

$$\overline{E}'=\left\{X|\left(X-X'+\alpha\frac{Ba}{\sqrt{a^TBa}}\right)^T \beta\left(B-\gamma\frac{(Ba)(Ba)^T}{a^TBa}\right)^{-1} X\left(X-X'+\alpha\frac{Ba}{\sqrt{a^TBa}}\right)\leq 1\right\}$$

Then the affine transformation sending E to $\overline{E}$ also sends E' to $\overline{E}'$.

algorithm의 公式들이 일반적인 affine system 내의 球 E와 이로부터 유도된 타원 E'를 나타낸다는 것을 알 수 있다.

Lemma 3.

If the matrix B is symmetric and positive definite, so is the matrix B' (i.e, the set E' defines an ellipsoid).

Lemma 4.

The semiellipsoid $\frac{1}{2}E_a$ is completely contained in the ellipsoid E'.

Lemma 5.

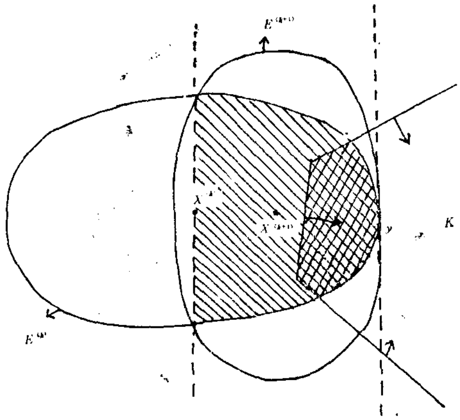
The volumes of the two ellipsoids E' and E satisfy $\lambda(E')=c(n)\lambda(E)$, where

$$c(n)=\frac{n}{n+1}\left(\frac{n^2}{n^2-1}\right)^{(n-1)/2}<2^{-1/(2(n+1))}$$

Theorem 1.

Algorithm turns a feasible vector $X^{(k)}$ if and only if $AX<b$ is satisfiable.

이 algorithm의 기하학적 설명은 〈그림 1〉에서 볼 수 있다.



〈그림 1〉 Open feasible region K, 알고리즘에 의해 얻어지는 두개의 타원 $E^{(k)}$, $E^{(k+1)}$

L은 $AX<b$의 input size이다. Khachiyan algorithm의 각 iteration마다 $0((m+n)n)$의 arithmatic operation을 필요로 한다. 그리고 最惡의 경우 $4(n+1)^2L$번의 iteration을 하게 된다. 따라서 $AX<b$를 푸는데 드는 노력은 총괄적으로 order $0(n^4L)$을 갖게 된다. 그러므로 Khachiyan의 algorithm이 $AX<b$를 푸는데 多項的으로 制限된 노력을 필요로 하는 algorithm인 것이다.

Ⅲ. 線型計劃法에의 應用

duality theorem에 의해 式 (1)의 線型計劃法問題를 푸는 것은 다음의 closed linear inequalities의 system을 푸는 것과 동일하다.

$$\begin{aligned} &AX\leq b\\ &A^TY\geq c\\ &X\geq 0 \qquad\qquad (9)\\ &Y\geq 0\\ &cX-b^TY\geq 0 \end{aligned}$$

fundamental duality theorem에 의해 式 (9)의 어떤 feasible solution이나 마지막 制限條件에서는 等式이 성립된다. 그러므로 마지막 制限條件을 이용하여 變數 하나를 소거하면 式 (9)는 $m+n-1$개의 變數로 이루어진 $2(m+n)$개의 closed inequality constraints의 system이 된다. $(X^T, Y^T)^T$가 式 9의 feasible solution이면 X^T는 式 (1)의 최적해이며 Y^T는 dual문제의 최적해이다.

式 (9)의 closed inequalities system에 khachiyan algorithm을 사용하려면 式 (9)의 data를 perturbing시켜 새로운 *open inequalities system*을 구성해야 한다. 다음 Lemma에 의해 perturb된 system이 infeasible이면 式 (9)도 infeasible이고, khachiyan algorithm에 의해 perturb된 system의 feasible solution을 얻으면 그것이 式 (9)의 feasible solution이 됨을 알 수 있다. (참고문헌 I, 4, 9 : 증명은 생략)

Lemma 6.

The system of linear inequalities

$$a_i^T X \leq b_i, \quad i=1, \cdots\cdots, m \tag{10}$$

has a solution if and only if the system of strict linear inequalities

$$a_i^T X < b_i + 2^{-L}, \quad i=1, \cdots\cdots, m \tag{11}$$

has a solution.

$AX \leq b$의 inequalities system을 Khachiyan algorithm에 적용하기 위하여 $2^L AX < 2^L b + 1$의 system으로 변형시켰을 경우 最惡의 경우 解를 얻기 위하여 행해지는 arithmatic operation의 갯수는 order $0((m+n)mn^4L)$을 갖는다. 線型計劃法問題를 푸는 경우도 같은 범주에 속하게 된다.

다음과 같은 線型計劃法問題를 Khachiyan algorithm에 의해 IBM 360-125로 풀어 본 결과 정확한 解에 도달하기 전에 B行列이 對稱의 陽의 定符號라는 성질을 잃어, 즉, $a_i^T B^{(k)} a_i < 0$이 되어 더 이상 계산을 할 수 없었다.

$$\begin{aligned} \text{maximize} \quad & 12x_1 + 15x_2 \\ \text{subject to} \quad & 4x_1 + 3x_2 \leq 12 \\ & 2x_1 + 5x_2 \leq 10 \\ & x_1, x_2 \geq 0 \end{aligned}$$

이 문제를 참고문헌 9의 version 2(Khachiyan의 algorithm 중 〈그림 1〉의 ▩부분을 포함하며 가능한 가장 적은 體積을 갖는 타원을 만들도록 변형시켜 解로 빨리 수렴하도록 한 algorithm)에 의해 푼 결과 최적해 근처에서 振動하며, 매우 느린 속도로 최적해에 접근하여 정확한 최적해와 10^{-4} 정도의 오차를 갖는 解를 구할 수 있었다.

Ⅳ. 討 論

Khachiyan의 algorithm은 線型計劃法問題가 포함하고 있는 變數 갯수에 대해 多項的으로 증가하는 노력으로 풀릴 수 있다는 것을 명백히 해준 것에 큰 의의를 갖는다. Khachiyan algorithn의 개념을 이용하여 對稱의 陽의 定符號(symmetric positive definite)의 LCP(linear complementarity problem)와 convex quadratic program이 多項的으로 制限된 시간안에 풀릴 수 있다는 것을 밝힌 論文도 발표되어 있다. (참고문헌 2, 3)

그러나 이 algorithm은 公式중 나눗셈이나 제곱근을 계산해야 하기 때문에 컴퓨터를 이용하든 손으로 계산하든 절대적인 정확도를 유지하며 이러한 계산들을 해나가기가 거의 不可能하였다. exp$(-10nL)$까지 정확히 계산되었을 경우 이 algorithm이 有用한 것으로 알려져 있다. 그렇지 않은 경우 roundoff error가 쌓여 algorithm에서의 B行列의 對稱의 陽의 定符號라는 성질을 잃게 되어 몇 단계 후에는 더 이상 계산을 할 수 없게 되었다.

後記

本 硏究는 1980년도 文敎部 학술연구 조성비에 의한 硏究이다.

附錄

참고문헌 9의 version 2에 의해 coding한 것임.

```
      DIMENSION LEG (25)
      DOUBLE PRECISION A(50, 50), X(50), B
     (50, 50), SUM(50), BA(50)
      DOUBLE PRECISION PRO, DEN, AX, U
      DATA IR, IW/5, 6/
      READ(IR, 1) M, N
    1 FORMAT (2I4)
      DO 97 I=1, 50
      DO 97 J=1, 50
   97 A(I, J)=0.
      MM=M+1
      NN=N+1
      DO 100 I=1, MM
  100 READ(IP, 3) (A(I, J), J=1, NN), LFG(I)
    3 FORMAT(3F4.0, I2)
      M1=MM
      IF(LEG(1). EQ.1) GO TO 1000
      DO 103 J=1, NN
  103 A(1,J)=-1. *A(1, J)
 1000 Do 112 I=2, MM
      IF(LEG(I). EQ. -1) GO TO 112
      IF(LEG(I). EQ. 0) GO TO 1003
```

```
      DO 106 J=1, NN
106   A(I, J)=-1. *A(I, J)
      GO TO 112
1003  M1=M1+1
      DO 109 J=1, NN
109   A(M1, J)=-1. *A(I, J)
112   CONTINUE
      KK=M+N
      JL=KK+1
      LL=KK+M1+N
      DO 115 J=1, N
115   A(1, J)=-1. *A(1, J)
      DO 118 J=2, M1
      J1=N+J-1
118   A(1, J1)=A(J, NN)
      DO 124 I=2, M1
      A(I, JL)=A(I, NN)
124   A(I, NN)=0.
      DO 133 I=1, N
      I1=M1+I
      DO 130 J=2, M1
      J1=N+J-1
130   A(I1, J1)=-1. *A(J, I)
133   A(I1, JL)=A(1, I)
      DO 34 I=1, KK
      I1=M1+N+I
134   A(I1, I)=-1.
      RL=1.
      DO 136 I=1, LL
      DO 136 J=1, JL
136   RL=RL*(DABS(A(I, J))+1.)
      RL=RL*LL*KK*2.
      RL1=ALOG(RL)/ALOG(2.)
      DO 139 I=1, LL
139   A(I, JL)=A(I, JL)+1./RL
      WRITE(IW, 4995) ((A(I, J), J=1, JL), I=
       1, LL)
4995  FORMAT (//, 9(/, 5X, 5F10.5))
      DO 142 I=1, KK
      X(I)=0.
      DO 142 J=1, KK
142   B(I, J)=0.
      DO 145
145   B(I, I)=RL
      K=0
      NL=4. *JL*JL*RL1
      WRITE(IW, 5000) RL, RL1, NL
5000  FORMAT (///, 5X, 2E20.10, I7)
1006  DO 151 I=1, LL
      SUM(I)=0.
      DO 148 J=1, KK
148   SUM(I)=SUM(I)+A(I, J)*X(J)
151   SUM(I)=SUM(I)-A(I, JL)
      I=1
      KI=I+1
1009  IG=I
      IF(KI. GT. LL) GO TO 1012
      IF(SUM(I). LE. SUM(KI)) I=KI
      KI=KI+1
      GO TO 1009
1012  IF(SUM(IG).LT.O.) GO TO 1015
      IF(K.GE.NL) GO TO 1018
      K=K+1
      DEN=0.
      DO 157 I=1, KK
      PRO=0.
      DO 154 J=1, KK
154   PRO=PRO+A(IG, J)*B(J, I)
157   DEN=DEN+PRO*A(IG, I)
      WRITE(IW, 5001) K, DEN, (A(IG, J), J=
       1, KK)
5001  FORMAT(//, I5, 5E20.10)
      AX=0.
      DO 158 J=1, KK
158   AX=AX+A(IG, J)*X(J)
      U=(A(IG, JL)-AX)/DSQRT(DEN)
      IF(U.LE.-1.) GO TO 1018
      DO 160 I=1, KK
      BA(I)=0.
      DO 160 J=1, KK
160   BA(I)=BA(I)+B(I, J)*A(IG, J)
      DO 163 J=1, KK
      X(J)=X(J)-(1.-U*KK)*BA(J)/(JL*
       DSQRT(DEN))
      DO 163 I=1, KK
163   B(I, J)=(1.-U*U)*KK*KK/(KK*KK-
       1.)*(B(I, J)-2.*(1.-KK*U)*BA(I)*
       BA(J)/(JL*(1.-U)*DEN))
      WRITE(IW, 5002) ((B(I, J), J=1, KK), I
       =1, KK)
```

```
5002  FORMAT(/, 4(5X, 4E20.10,/))
      WRITE(IW, 5005) (X(J), J=1, KK)
5005  FORMAT(5X, 4E20.10)
      GO TO 1006
1015  WRITE (IW, 9) (X(J) ,J=1, KK)
   9  FORMAT(///, 5X, 4F10.2,/)
      GO TO 1021
1018  WRITE(IW, 12)
  12  FORMAT(///, 5X, 'THIS PROBLEM HAS
       NO SOLUTION')
1021  STOP
      END
```

REFERENCES

1. Aspvall, Bengt & R.E. Stone, "Khachiyan's Linear Programming Algorithm", J. of Algorithms, 1, pp. 1—13, 1980.
2. Chung Sung J. & Katta G. Murty, "A Polynomially Bounded Algorithm for Positive Definite Symmetric LCPs", Technical Report No. 79—10, Department of Industrial and Operations Engineering, The University of Michigan, 1979.
3. Chung Sung J. & Katta G. Murty, "Polynomially Bounded Elliposoid Algorithms for Convex Quadratic Programming", Department of Industrial and Operations Engineering, The University of Michigan, June 1980.
4. Gacs, P. & L. Lovasz, "Khachiyan's Algorithm for Linear Programming", Technical Report STANCS-79-750, Computer Science Department, Stanford University, 1979.
5. Khachiyan, L.G., "A Polynomial Algorithm in Linear Programming, Soviet Math. Dokl., Vol. 20, No. 1, pp. 191—194, 1979.
6. Kozlov, M.K., S.P. Tarasov & L.G. Khachiyan, "Polynomial Solvability of Convex Quadratic Programming", Soviet Math. Dokl., Vol. 20, No. 5, pp. 1108—1111, 1979.
7. Mangasarian, O.L., "Simplified Characterization of Linear Complementarity Problems Solvable as Linear Programs", Math. of Operations Research, Vol. 4, No. 3, pp. 268—273, 1979.
8. Murty, Katta G., *Complementarity Problems, Linear and Combinatorial Programming*, John Wiley & Sons. Inc., 1976.
9. Murty, Katta G., *Linear, Network and Conbinatorial-Programming*, Department of Industrial and Operations Engineering, The University of Michigan, December 1979.