An Ellipsoid Algorithm for Equality-Constrained Nonlinear Programs

Sharmila Shah John E. Mitchell Michael Kupferschmid

18 Aug 99

Scope and Purpose – The purpose of this paper is to present a variant of the ellipsoid algorithm that can be used with equalities. This is a significant improvement over the classical algorithm, which yields accurate solutions to convex and many nonconvex nonlinear programming problems but requires the feasible set to be of full dimension and therefore cannot be used with equality constraints.

Abstract – This paper describes an ellipsoid algorithm that solves convex problems having linear equality constraints with or without inequality constraints. Experimental results show that the new method is also effective for some problems that have nonlinear equality constraints or are otherwise nonconvex.

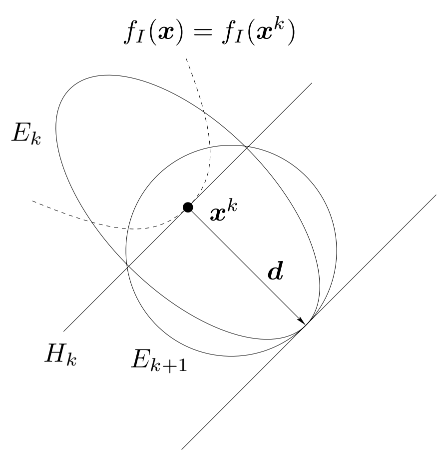
Keywords – nonlinear optimization, ellipsoid algorithm, equality constraints

1 The Ellipsoid Algorithm for Inequality Constraints

The classical ellipsoid algorithm [3] [11] [13] [8] solves nonlinear programming problems of the form

$$\begin{array}{llrcl} \text{INLP}: & \displaystyle\min_{\boldsymbol{x}\in\Re^n} & f_0(\boldsymbol{x}) & & \\ & \text{subject to} & f_i(\boldsymbol{x}) & \leq & 0, \quad i = 1 \ldots m_I \end{array}$$

where $f_i : \Re^n \rightarrow \Re$ for $i = 0 \ldots m_I$ are convex real-valued functions. Starting with bounds $\boldsymbol{U}$ and $\boldsymbol{L}$ containing the optimal point $\boldsymbol{x}^*$ for INLP the ellipsoid algorithm generates a sequence of ellipsoids $\boldsymbol{E}_k$, each guaranteed to contain $\boldsymbol{x}^*$, with the property that their volumes shrink to zero as the terms of a geometric progression. The starting ellipsoid $E_0 = \{\boldsymbol{x} \in \Re^n \mid (\boldsymbol{x} - \boldsymbol{x}^0)^T \boldsymbol{Q}_0^{-1} (\boldsymbol{x} - \boldsymbol{x}^0) \leq 1\}$ is the smallest ellipsoid containing $\boldsymbol{U}$ and $\boldsymbol{L}$, with $\boldsymbol{x}^0$ the midpoint of the bounds and $\boldsymbol{Q}_0$ positive definite and symmetric.



At each iteration, the algorithm finds the normalized gradient $\boldsymbol{g}$ of the objective function $f_0(\boldsymbol{x})$ if $\boldsymbol{x}^k$ is feasible or of a violated constraint $f_I(\boldsymbol{x})$ if $\boldsymbol{x}^k$ is infeasible, to calculate a direction

$$\boldsymbol{d} = -\frac{\boldsymbol{Q}_k \boldsymbol{g}}{\sqrt{\boldsymbol{g}^T \boldsymbol{Q}_k \boldsymbol{g}}}. \tag{1}$$

Using this $\boldsymbol{d}$ we find the next ellipsoid using the updates

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \frac{1}{n+1}\boldsymbol{d} \tag{2}$$

$$\boldsymbol{Q}_{k+1} = \frac{n^2}{n^2-1}\left(\boldsymbol{Q}_k - \frac{2}{n+1}\boldsymbol{d}\boldsymbol{d}^T\right). \tag{3}$$

Sharmila Shah is a Staff Analyst at United Airlines, 1200 E. Algonquin Rd., Elk Grove Village, IL 60007; her research interests are in mathematical programming and its applications. John Mitchell is an Associate Professor of Mathematics at Rensselaer Polytechnic Institute, 110 8th St., Troy, NY 12180-3590; his research interests are in interior point methods, integer programming, nonlinear programming, and their applications. Michael Kupferschmid (to whom correspondence should be addressed) is Scientific Programming Consultant in the Alan M. Voorhees Computing Center at Rensselaer; his research interests are in computational optimization and the experimental evaluation of algorithm performance.

An iteration can be visualized geometrically as shown. A hyperplane H_k is constructed supporting the contour $f_I(\boldsymbol{x}) = f_I(\boldsymbol{x}^k)$ and dividing the ellipsoid E_k in half. The next ellipsoid E_{k+1} is the smallest ellipsoid enclosing the half of E_k that contains $\boldsymbol{x}^*$. In this construction, the direction vector $\boldsymbol{d}$ can be found by translating H_k parallel to itself until it is tangent to E_k at $\boldsymbol{x}^k + \boldsymbol{d}$. Analytically this amounts to minimizing $\boldsymbol{g}^T\boldsymbol{x}$ over E_k, or solving

$$\begin{array}{ll} \min\limits_{\boldsymbol{d}\in\Re^n} & \boldsymbol{g}^T\boldsymbol{d} \\ \text{subject to} & \boldsymbol{d}^T\boldsymbol{Q}_k^{-1}\boldsymbol{d} \leq 1. \end{array}$$

For this problem the Karush-Kuhn-Tucker conditions yield the formula (1) given above for $\boldsymbol{d}$. The volumes $V[E_k]$ of the ellipsoids decrease according to $V[E_{k+1}] = c_n V[E_k]$ or $V[E_k] = (c_n)^k V[E_0]$, where [4]

$$c_n = \frac{n}{n+1}\left(\frac{n^2}{n^2-1}\right)^{(n-1)/2} \quad < \quad 1. \tag{4}$$

This volume reduction ratio depends only upon the dimension n of the space.

The convexity of the $f_i(\boldsymbol{x})$ ensures that each ellipsoid E_k will contain $\boldsymbol{x}^*$ and $c_n < 1$ ensures that the volumes $V[E_k]$ decrease. For the algorithm to converge it is also necessary [9] [5] for the feasible set $S = \{\boldsymbol{x} \in \Re^n \big| f_i(\boldsymbol{x}) \leq 0, \quad i = 1 \ldots m_I\}$ to have dimension n. This prevents the algorithm from being used to solve equality-constrained problems.

2 The New Algorithm

We want to use the ellipsoid algorithm for problems of the form

$$\text{NLP}: \qquad \begin{array}{rlcl} \min\limits_{\boldsymbol{x}\in\Re^n} & f_0(\boldsymbol{x}) & & \\ \text{subject to} & f_i(\boldsymbol{x}) & \leq & 0, \quad i = 1\ldots m_I \\ & \boldsymbol{A}\boldsymbol{x} & = & \boldsymbol{b} \end{array}$$

where the $f_i(\boldsymbol{x})$ are convex as before and $\boldsymbol{A}$ has full rank $m_E < n$. To solve these problems we constrain $\boldsymbol{d}$ to lie in the flat $F = \{\boldsymbol{x} \in \Re^n \big| \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$ by requiring $\boldsymbol{A}(\boldsymbol{x}^k + \boldsymbol{d}) = \boldsymbol{b}$ or $\boldsymbol{A}\boldsymbol{d} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^k = \boldsymbol{0}$. Then $\boldsymbol{d}$ solves

$$\begin{array}{ll} \min\limits_{\boldsymbol{d}\in\Re^n} & \boldsymbol{g}^T\boldsymbol{d} \\ \text{subject to} & \boldsymbol{d}^T\boldsymbol{Q}_k^{-1}\boldsymbol{d} \leq 1 \\ & \boldsymbol{A}\boldsymbol{d} = \boldsymbol{0}. \end{array}$$

Using the Karush-Kuhn-Tucker conditions we get the direction $\boldsymbol{d}$ that minimizes $\boldsymbol{g}^T\boldsymbol{d}$ over the ellipsoid E_k and is in the flat F,

$$\boldsymbol{d} = -\frac{(\boldsymbol{Q}_k - \boldsymbol{Q}_k\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{Q}_k\boldsymbol{A}^T)^{-1}\boldsymbol{A}\boldsymbol{Q}_k)\boldsymbol{g}}{\sqrt{\boldsymbol{g}^T(\boldsymbol{Q}_k - \boldsymbol{Q}_k\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{Q}_k\boldsymbol{A}^T)^{-1}\boldsymbol{A}\boldsymbol{Q}_k)\boldsymbol{g}}}.$$

If $\boldsymbol{x}^0 \in F$, we can use this $\boldsymbol{d}$ in the updates (2-3) given earlier to generate a different sequence of ellipsoids E_k, each containing $\boldsymbol{x}^*$ and having centers $\boldsymbol{x}^k$ in the flat F.

To control round-off errors, and to allow starting points $\boldsymbol{x}^0$ that are not feasible for the equality constraints, we project each $\boldsymbol{x}^k$ onto the flat F. When the equality constraints are nonlinear, we linearize them at each iteration and project $\boldsymbol{x}^k$ onto that flat. It is because the linearization can be different at each iteration for a nonlinear problem that we use the method outlined above rather than simply using the equalities to eliminate variables and solving the reduced problem. Confining $\boldsymbol{d}$ to the flat F can result in the ellipsoid E_k becoming highly aspheric more quickly than in the classical algorithm, leading to imprecise numerical results. To refine the estimate of $\boldsymbol{x}^*$, we sometimes restart the algorithm with a new smaller E_0 centered on the best iterate $\boldsymbol{x}^k$ found so far.

3 The Volume Reduction Ratio

For the classical ellipsoid algorithm the ratio of volumes of successive ellipsoids is c_n given above (4), and $c_n < 1$ so the volumes $V(E_k)$ decrease monotonically to zero. The ellipsoid $\tilde{E_k} = E_k \cap F$ is the intersection of the ellipsoid $E_k = \{x \in \Re^n \big| (\boldsymbol{x} - \boldsymbol{x}^k)^T \boldsymbol{Q}_k^{-1} (\boldsymbol{x} - \boldsymbol{x}^k) \leq 1\}$ with the flat $F = \{\boldsymbol{x} \in \Re^n \big| \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\}$. The volumes $V(\tilde{E_k})$ of the ellipsoids of intersection need to decrease monotonically to zero for the new algorithm to converge. To show that this happens, we let the columns of $\boldsymbol{B}$ denote a basis for the null space of $\boldsymbol{A}$, so $\boldsymbol{B}$ is $n \times (n - m_E)$. Then any solution $\boldsymbol{x}$ to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ satisfies $\boldsymbol{x} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{b} + \boldsymbol{B}\boldsymbol{y}$ for some $\boldsymbol{y} \in \Re^{n-m_E}$ [12]. The ellipsoid $\hat{E_k}$ with center $\boldsymbol{y}^k$ in $y-$space can be obtained by substituting for $\boldsymbol{x}$ and $\boldsymbol{x}^k$ in the above formula for E_k and simplifying, so $\boldsymbol{x} \in E_k \cap F$ if and only if

$$\begin{aligned}
((\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{b} + \boldsymbol{B}\boldsymbol{y}) - (\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{b} + \boldsymbol{B}\boldsymbol{y}^k))^T \boldsymbol{Q}_k^{-1} & \\
((\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{b} + \boldsymbol{B}\boldsymbol{y}) - (\boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{b} + \boldsymbol{B}\boldsymbol{y}^k)) &\leq 1 \\
\Longleftrightarrow (\boldsymbol{B}\boldsymbol{y} - \boldsymbol{B}\boldsymbol{y}^k)^T \boldsymbol{Q}_k^{-1} (\boldsymbol{B}\boldsymbol{y} - \boldsymbol{B}\boldsymbol{y}^k) &\leq 1 \\
\Longleftrightarrow (\boldsymbol{y} - \boldsymbol{y}^k)^T \boldsymbol{B}^T \boldsymbol{Q}_k^{-1} \boldsymbol{B} (\boldsymbol{y} - \boldsymbol{y}^k) &\leq 1.
\end{aligned}$$

Thus, $\hat{E_k} = \{\boldsymbol{y} \in \Re^{n-m_E} \big| (\boldsymbol{y} - \boldsymbol{y}^k)^T \boldsymbol{B}^T \boldsymbol{Q}_k^{-1} \boldsymbol{B} (\boldsymbol{y} - \boldsymbol{y}^k) \leq 1\}$ is the ellipsoid in F described by the intersection of E_k with F and $\tilde{E_k} = \{\boldsymbol{x} \in \Re^n \big| \boldsymbol{x} = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{A}^T)^{-1}\boldsymbol{b} + \boldsymbol{B}\boldsymbol{y}, \boldsymbol{y} \in \hat{E_k}\}$. Now recall that the volume of an ellipsoid $E = \{x \in \Re^n \big| (\boldsymbol{x} - \boldsymbol{c})^T \boldsymbol{W}^T \boldsymbol{W} (\boldsymbol{x} - \boldsymbol{c}) \leq 1\}$ is $Vol(E) = det(\boldsymbol{W}^{-1}) \times Vol(S(0,1))$ where $S(0,1)$ is the unit ball in n dimensions (centered at the origin) [4]. An ellipsoid is the image of the unit ball $S(0,1)$ under some affine transformation, so we can assume that E_k is itself a unit ball centered at the origin (then $\boldsymbol{Q}_k$ is the $n \times n$ identity). The flat F of equalities passes through the center of the ellipsoid, so if $E_k = S(0,1)$ then F goes through the origin and $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{0}$. If F goes through the origin, another affine transformation could be used to rotate it into the position of a coordinate hyperplane, so without loss of generality we can assume that $\boldsymbol{d} = [1, 0, \ldots 0]^T$. These assumptions correspond to letting

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{I} \\ \boldsymbol{Z} \end{bmatrix}$$

where $\boldsymbol{I}$ is the $(n - m_E) \times (n - m_E)$ identity and $\boldsymbol{Z}$ is an $m_E \times (n - m_E)$ zero matrix. Then $\boldsymbol{B}^T \boldsymbol{Q}_k^{-1} \boldsymbol{B} = \boldsymbol{I}$ and $Vol(\tilde{E_k}) = Vol(\tilde{S}(0,1))$. Applying the update formula (3),

$$\begin{aligned}
\boldsymbol{Q}_{k+1} &= \frac{n^2}{n^2-1}(\boldsymbol{Q}_k - \frac{2}{n+1}\boldsymbol{d}\boldsymbol{d}^T) \\
&= \begin{pmatrix} \frac{n^2}{(n+1)^2} & 0 & \ldots & 0 \\ 0 & \frac{n^2}{n^2-1} & \ldots & 0 \\ 0 & & \ddots & 0 \\ 0 & 0 & \ldots & \frac{n^2}{n^2-1} \end{pmatrix}.
\end{aligned}$$

Then

$$\boldsymbol{B}^T \boldsymbol{Q}_{k+1}^{-1} \boldsymbol{B} = \begin{pmatrix} \frac{(n+1)^2}{n^2} & 0 & \ldots & 0 \\ 0 & \frac{n^2-1}{n^2} & \ldots & 0 \\ 0 & & \ddots & 0 \\ 0 & 0 & \ldots & \frac{n^2-1}{n^2} \end{pmatrix},$$

where $\boldsymbol{B}^T \boldsymbol{Q}_k^{-1} \boldsymbol{B}$ is an $(n - m_E) \times (n - m_E)$ matrix. Thus the ratio of the volumes of successive ellipsoids of intersection is

$$\frac{Vol(\tilde{E}_{k+1})}{Vol(\tilde{E_k})} = det(\boldsymbol{W}^{-1}) = \frac{n}{n+1} \left(\frac{n^2}{n^2-1} \right)^{(n-m_E-1)/2}.$$

For problems where $\operatorname{rank}(\boldsymbol{A}) = m_E = n-1$, the above ratio is $\dfrac{n}{n+1}$ which is strictly less than 1. For problems where $\operatorname{rank}(\boldsymbol{A}) = 1$, the above ratio takes on its largest value,

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

Thus the ratio of successive volumes, which depends on rank of $\boldsymbol{A}$ and the dimension of the problem, is strictly less than 1 for all $n \geq 2$ (we only consider problems with equality constraints for $n \geq 2$). Hence the volumes of the ellipsoids of intersection decrease monotonically to zero as k increases and this algorithm converges in polynomial time like the classical ellipsoid algorithm but faster.

4 Preliminary Computational Experience

We implemented the algorithm using double precision arithmetic and tested it on the test problems listed below. We used a feasibility tolerance of 10^{-6} for the equality constraints, and started from published starting points for all problems except HS109 (whose published starting point yields a matrix $\boldsymbol{A}$ that does not have full row rank). The following table contains a summary of the results obtained by the new algorithm.

Problem	N	MI	ME	INLP convex?	Largest = Error	Our $f_0(\boldsymbol{x}^*)$	Source $f_0(\boldsymbol{x}^*)$	CPU sec to $\geq$ 6 digits	
HS6 [7]	2	0	(1)	Yes	-1.11022302D-15	0.00000000D+00	0.00000000D+00	0.01	
HS7 [7]	2	0	(1)	No	5.49743101D-07	-1.73205085D+00	-1.73205081D+00	0.01	
HS8 [7]	2	0	(2)	Yes	-3.55271368D-15	-1.00000000D+00	-1.00000000D+00	0.03	
HS26 [7]	3	0	(1)	Yes	0.00000000D+00	0.00000000D+00	0.00000000D+00	0.04	
HS28 [7]	3	0	1	Yes	2.22044605D-16	0.00000000D+00	0.00000000D+00	0.01	
HS39 [7]	4	0	(2)	Yes	-3.78440229D-24	-1.00000000D+00	-1.00000000D+00	0.05	
HS40 [7]	4	0	(3)	No	8.84675277D-08	-2.50000013D-01	-2.50000000D-01	0.02	
HS46 [7]	5	0	(2)	Yes	0.00000000D+00	0.00000000D+00	0.00000000D+00	0.08	
HS48 [7]	5	0	2	Yes	0.00000000D+00	0.00000000D+00	0.00000000D+00	0.10	
HS49 [7]	5	0	2	Yes	0.00000000D+00	0.00000000D+00	0.00000000D+00	0.08	
HS50 [7]	5	0	3	Yes	0.00000000D+00	0.00000000D+00	0.00000000D+00	0.09	
HS51 [7]	5	0	3	Yes	0.00000000D+00	0.00000000D+00	0.00000000D+00	0.10	
HS52 [7]	5	0	3	Yes	0.00000000D+00	5.32664756D+00	5.32664756D+00	0.06	
HS107 [7]	9	8	(6)	No	6.59650744D-16	5.05501145D+03	*5.05501074D+03*	0.95	
HS109 [7]	9	20	(6)	No	2.08545138D-11	5.32685133D+03	*5.36206769D+03*	0.93	$\star$
HS119 [7]	16	32	8	Yes	-8.88178420D-16	1.31363614D+02	*1.32850557D+02*	206.05	$\star$
LINEAR [10]	2	0	1	Yes	0.00000000D+00	7.50000000D-01	7.50000000D-01	0.01	
BS400 [1]	2	0	1	Yes	0.00000000D+00	3.35528345D+01	3.35528345D+01	0.02	
BS401 [1]	3	0	(2)	Yes	-8.24308712D-07	-6.82207508D+00	-6.82207465D+00	0.01	
BS403 [1]	2	1	(1)	No	5.45599933D-07	-6.71459233D-01	-6.71459186D-01	0.02	
BS403C [1]	2	1	(1)	No	-4.44089210D-16	7.22128130D-01	7.22128130D-01	0.04	
BS467 [1]	4	4	2	Yes	8.88178420D-16	-7.16129032D+00	-7.16129032D+00	0.17	
BS475 [1]	3	4	1	Yes	0.00000000D+00	-2.40000000D+01	-2.40000000D+01	0.01	
BS475A [1]	2	1	1	Yes	0.00000000D+00	1.00000000D+00	1.00000000D+00	0.01	
BS476 [1]	3	1	(1)	No	3.23176964D-07	-2.01416790D+01	-2.01416753D+01	0.01	
BS486 [1]	2	2	(1)	Yes	0.00000000D+00	1.20000000D+01	1.20000000D+01	0.02	
JM [10]	3	1	1	Yes	0.00000000D+00	-1.50000000D+00	-1.50000000D+00	0.01	
HIM4 [6]	10	10	3	No	0.00000000D+00	-4.77610909D+01	*-4.77699981D+01*	0.96	$\star$
HIM4A [6]	10	0	(3)	No	6.81422498D-07	-4.77610950D+01	*-4.77699981D+01*	1.29	$\star$
HIM5 [6]	3	3	(2)	No	2.87427202D-07	9.61715172D+02	9.61715172D+02	0.01	
HIM15 [6]	6	12	(4)	Yes	3.27834981D-13	8.82759774D+03	*8.82758000D+03*	0.15	$\star$
HIM20 [6]	24	30	(14)	No	-2.22044605D-16	5.17277178D-02	*5.70109712D-02*	504.63	$\star$
BRM4 [2]	10	20	(7)	No	-9.51536104D-13	-1.16133689D+03	*-1.76798529D+03*	1.22	$\star$

The first column of the table gives the name of the problem and a reference where it can be found; for the starting points we used, see [10]. The second column lists the number of variables N, the number of inequality constraints MI and the number of equality constraints ME. Parentheses around the value of ME indicate that some or all of the equalities are nonlinear. The third column tells whether the problem without the equality constraints, earlier called INLP, is convex. The fourth and fifth columns give the largest equality constraint function value and the objective value at the best point found using the new algorithm. The sixth column gives the true minimum value if that is known exactly, or else (in *slanting* type) the objective value calculated at the optimal point reported in the source from which we took the problem. To find the results given for the new algorithm we ran each problem until the calculations could go no further (typically because $\boldsymbol{d}$ became the zero vector), restarted with a new smaller E_0 centered on the best iterate x^k found so far, ran until the calculations could go no further, and repeated this process until the x^k stopped changing. In each case the solution we found is strictly feasible for all inequality constraints and satisfies the equality constraints within 10^{-6}. For some of the problems our answers are much closer to the set of equalities than the answers published in the source. In a few cases the best point we found has an objective value lower than the known optimal value, because of the nonzero tolerance we used for the equality constraints. In other cases, marked with a $\star$, the exact solution is unknown to us and either the objective value at the best point we found is lower than that reported in the source or the solution given in the source violates an equality constraint by more than 10^{-6}.

This algorithm, like the classical ellipsoid algorithm, usually finds feasible points that are not far from optimal early in the solution process, so in practical applications it is unnecessary to continue restarting until the answer is as precise as those we found. To measure the CPU time reported in the table for our experimental code, we reran each problem under the UNIX™ `time` command, on an IBM RS6000 model 250 workstation, for long enough to get the first 6 (or more) digits of the objective value correct. From the measured times it is clear that some of the test problems are much more difficult than others, but that even the most difficult of them can be solved in no more than a matter of minutes using a computer of only modest speed.

5 Summary

The new algorithm solves convex problems with linear equality constraints, with or without inequality constraints, starting from points that are feasible or infeasible for the equalities. In addition, it solves some otherwise convex problems having nonlinear equality constraints, and it solved most of the problems we tried that are nonconvex even ignoring the equalities.

For the numerical stability of the algorithm, it is necessary to re-project the center found at each iteration onto the flat of equalities. The accuracy of this algorithm is improved by recentering.

References

[1] M. S. Bazaraa, H. D. Sherali and C. M. Shetti, *Nonlinear Programming Theory and Algorithms.* John Wiley & Sons, New York (1993).

[2] J. Bracken and G. P. McCormick, *Selected Applications of Nonlinear Programming.* John Wiley & Sons, New York (1968).

[3] J. G. Ecker and M. Kupferschmid, *Introduction to Operations Research.* Krieger, Malabar, Florida (1988).

[4] Shu-Cherng Fang and S. Puthenpura, *Linear Optimization and Extensions: Theory and Algorithms.* Prentice Hall, New Jersey (1993).

[5] Jean-Louis Goffin, Convergence Rates of the Ellipsoid Method on General Convex Functions. *Mathematics of Operations Research* **8,** 135-150 (1983).

[6] D. M. Himmelblau, *Applied Nonlinear Programming.* McGraw-Hill, New York (1972).

[7] W. Hock and K. Schittkowski, *Test Examples For Nonlinear Programming.* Lecture Notes in Economics and Mathematical Systems Volume 187, Springer-Verlag, New York (1981).

[8] L. G. Khachian, A Polynomial Algorithm in Linear Programming. *Soviet Mathematics Doklady* **20,** 191-194 (1979).

[9] Hans-Jakob Lüthi, On the Solution of Variational Inequalities by the Ellipsoid Method. *Mathematics of Operations Research* **10,** 515-522 (1985).

[10] Sharmila Shah, *An Ellipsoid Algorithm for Equality-Constrained Nonlinear Programs*. Ph. D. Dissertation, Rensselaer Polytechnic Institute, Troy, NY (1998).

[11] N. Z. Shor, Cut-off Method With Space Extension in Convex Programming Problems. *Cybernetics* **12,** 94-96 (1977).

[12] Gilbert Strang, *Linear Algebra and Its Applications*. Academic Press, New York (1976).

[13] D. B. Yudin and A. S. Nemirovskii, Informational Complexity and Effective Methods for Solving Convex Extremal Problems. *Matekon* **13,** 24-45 (1977).