On Solving a Certain Multi-Parametric Nonlinear Programming Problem

by

Hisashi MINE*, Masao FUKUSHIMA* and Yong-Joon RYANG*

(Received June 26, 1980)

Abstract

We propose a method of solving a nonlinear programming problem with two scalar parameters which satisfy a certain nonlinear equation. Given an optimal solution of the problem for an initial parameter value, the proposed method iteratively generates a sequence of optimal solutions of the parametric problem along the path determined by the equation in the two dimensional parameter space. The procedure is illustrated with an example.

1. Introduction

In the previous papers [3] and [4], the authors presented a method of finding optimal solutions for a nonlinear programming problem with a single scalar parameter. Extending the idea of the previous paper, this paper proposes a method for problems having multiple parameters.

We can generally write a parametric nonlinear programming problem of this type as follows:

$$\left.\begin{array}{lll}\text{minimize} & f(x, t) & \\ \text{subject to} & g(x, t)=0, & a\leq x\leq b,\end{array}\right\} \tag{1}$$

where f: $R^n\times R^p\to R$, g: $R^n\times R^p\to R^m$ are twice continuously differentiable functions, and $x\in R^n$ and $t\in R^p$ are a vector of decision variables and a vector of parameters, respectively, which satisfy the equation $\phi(t)=0$. Hereafter, we are mainly concerned with a two parameter case, namely, $t\in R^2$ and ϕ: $R^2\to R$, for simplicity of exposition.

An example of problem (1) is found in the following situation. Suppose we have a nonlinear programming problem:

$$\left.\begin{array}{lll}\text{minimize} & f(x, s, \tau) & \\ \text{subject to} & g(x, s, \tau)=0, & a\leq x\leq b,\end{array}\right\} \tag{2}$$

* Department of Applied Mathematics and Physics

where $x \in R^n$ is a vector of decision variables and $(s, \tau) \in R^2$ is a vector of parameters. Suppose moreover that s is given as a state of another system governed by a differential equation, say:

$$\frac{d^2s}{d\tau^2}+\alpha\frac{ds}{d\tau}+\beta s=\Phi(\tau), \tag{3}$$

where τ is a time parameter. Then the problem (2) becomes that of finding optimal solutions $x(s, \tau)$ for every pair (s, τ) of the state and time, which is determined from the differential equation (3). When initial conditions $s(0)=s_0$ and $s'(0)=s_1$ are given, the differential equation (3) might be uniquely solved with appropriate assumtions.

We denote an optimal solution of the problem (1) for the parameter value t by $x(t)$, and we shall assume that $x(t)$ is unique and a continuous function of t. Assume that it is known that the optimal solution $x(t^\circ)$ with respect to an initial parameter value $t^\circ=(t_1^\circ, t_2^\circ)$, thus satisfying $\phi(t^\circ)=0$. Then the optimal solution $x(t)$ of the nonlinear programming problem (1) can be obtained by changing t continuously so as to satisfy $\phi(t)=0$. It is noted that similar ideas may be found in [2] and [6] for unconstrained optimization and nonlinear equations.

In section 2, we describe a reduced formula for a parametric nonlinear programming problem. We also give expressions of its gradient and Hessian matrix of the reduced objective function, and the Kuhn-Tucker conditions for the reduced parametric formula. In section 3, using an iterative procedure for changing the value of two parameters, we present an algorithm to solve the parametric nonlinear programming problem. In section 4, numerical experiments are carried out to test the algorithms. In section 5, we briefly consider an extension to a three parameter case.

2. Reduced Problem

In this section, we sketch a fundamental idea of the parametric programming technique, which is regarded as a natural extension of our previous work [3] for a single parameter case.

For problem (1), we adopt the following nondegeneracy assumption: For any $t=(t_1, t_2)$ satisfying $\phi(t)=0$, and for any feasible solution x, i.e., $g(x, t)=0$ and $a \leq x \leq b$, there exists a partition of variables x into basic variables y and nonbasic variables z. In this conditions, y is m-dimensional and z is $(n-m)$-dimensional, such that the $m \times m$ matrix $\nabla_y g(y, z, t)$ is nonsingular and $a_y < y < b_y$ (a_y and b_y consist of the components of a and b corresponding to y, respectively). (See [1]).

Similarly, the matrices $\nabla_x f(x, t)$ and $\nabla_x g(x, t)$ are partitioned as $[\nabla_y f(x, y), \nabla_z f(x, t)]$ and $[\nabla_y g(x, t), \nabla_z g(x, t)]$, respectively. Let $\bar{x}$ and $\bar{t}$ be any points in R^n and R^2. Then by the above nondegeneracy assumption and the implicit function theorem, there exist a partition $x=(y, z)$ and a function $h: \Omega\times\Sigma\to R^m$, where Ω and Σ are certain neighborhoods of $\bar{y}$ and $\bar{t}$, such that

$$g(h(z, t), z, t)=0 \tag{4}$$

for all $z\in\Omega$, $t\in\Sigma$.

Consequently, we may define, at least locally, a reduced formula of problem (1) as follows: For $t\in R^2$

$$\left.\begin{array}{ll}\text{minimize} & F(z, t)\equiv f(h(z, t), z, t)\\ \text{subject to} & a_z\leq z\leq b_z,\end{array}\right\} \tag{5}$$

where components of a_z and b_z correspond to the lower and upper bounds of z. Notice that the objective function F is twice continuously differentiable in x and t, because both f and h have similar properties.

We now state the Kuhn-Tucker conditions for an isolated local minimum of problem (5). A statement of the conditions for more general nonlinear programs can be found elsewhere, for example, in [5, p. 235], and hence, no proof is given here. For $t\in R^2$, if $z(t)$ is optimal to problem (5), then the following conditions hold:

$$\begin{array}{ll}\nabla_{z_I}F(z(t), t)=0 & a_I\leq z_I\leq b_I\\ z_{J_1}=a_{J_1} & \nabla z_{J_1}F(z(t), t)z(t)\geq 0\end{array} \tag{6}$$

and

$$z_{J_2}=b_{J_2} \qquad \nabla z_{J_2}F(z(t), t)z(t)\leq 0,$$

where $I\cup J_1\cup J_2=\{1, 2, \cdots, n-m\}$.

Next, we given the gradient and Hessian matrix of the function F. Let

$$\lambda(x, t)=\nabla_y f(x, t)[\nabla_y g(x, t)]^{-1} \tag{7}$$

Then from (4) and (5) we obtain by direct calculation

$$\nabla_z F(z, t)=\nabla_z f(h(z, t), z, t)-\lambda(h(z, t), z, t)\nabla_z g(h(z, t), z, t). \tag{8}$$

Furthermore, differentiating (8) yields

$$\nabla_{zz}{}^2F(z, t)=[\nabla_z h^T, I_{n-m}]\begin{bmatrix}\nabla_{yy}{}^2f-\lambda\nabla_{yy}{}^2g & \nabla_{yz}{}^2f-\lambda\nabla_{yz}{}^2g\\ \nabla_{zy}{}^2f-\lambda\nabla_{zy}{}^2g & \nabla_{zz}{}^2f-\lambda\nabla_{zz}{}^2g\end{bmatrix}\begin{bmatrix}\nabla_z h\\ I_{n-m}\end{bmatrix} \tag{9}$$

$$=\Gamma^T[\nabla_{xx}{}^2f-\lambda\nabla_{xx}{}^2g]\Gamma, \tag{10}$$

where

$$\Gamma=\begin{bmatrix}-\nabla_y g^{-1}\nabla_z g\\ I_{n-m}\end{bmatrix}.$$

3. Algorithm

To describe the approach to solving nonlinear programming problems with a two dimensional parameter, we suppose that a parameter vector $t^k=(t_1^k, t_2^k)$ satisfying $\phi(t^k)=0$ and the optimal solution $x(t^k)$ are known.

First, an iterative procedure is introduced to obtain the parameter value $\tilde{t}^{k+1}=(\tilde{t}_1^{k+1}, \tilde{t}_2^{k+1})$, which differs slightly from t^k and satisfies the equation $\phi(t^{k+1})=0$. The procedure starts with the moving of t from t^k by a fixed amount along the line tangent to the curve defined by $\phi(t)=0$. Thus, the direction $d^k=(d_1^k, d_2^k)\in R^2$ of the movement is orthogonal to $\nabla\phi(t^k)$, namely

$$\nabla\phi(t^k)d^k=0 \tag{11}$$

and

$$\|d^k\|=\Delta \tag{12}$$

for some small fixed real number $\Delta>0$. (See Fig. 1).

It is easy to see that (11) and (12) are satisfied by

$$\left.\begin{aligned} d_1^k &= \frac{\partial\phi(t^k)}{\partial t_2}\Delta \,/\, \|\nabla\phi(t^k)\| \\ d_2^k &= -\frac{\partial\phi(t^k)}{\partial t_1}\Delta \,/\, \|\nabla\phi(t^k)\| \end{aligned}\right\} \tag{13}$$

provided that $\nabla\phi(t^k)\neq 0$. Then, we set

$$\tilde{t}_i^{k+1}=t_i^k+d_i^k, \qquad i=1, 2. \tag{14}$$

However, the point $\tilde{t}^{k+1}=(\tilde{t}_1^{k+1}, \tilde{t}_2^{k+1})$ determined by (14) may not satisfy $\phi(\tilde{t}_1^{k+1}, \tilde{t}_2^{k+1})=0$ in general, so we compute $t^{k+1}=(t_1^{k+1}, t_2^{k+1})$ such that $\phi(t^{k+1})=0$ in the following

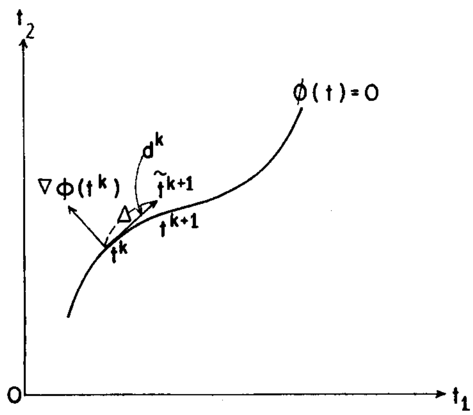

Fig. 1.

way. Fixing $t_1^{k+1}=\tilde{t}_1^{k+1}$, t_2^{k+1} is obtained as a limit of the sequence $\{\tau^i\}$ generated by Newton's method.

$$\left.\begin{aligned} &\tau^\circ = t_2^{k+1} \\ &\tau^{i+1}=\tau^i-\frac{\phi(t_1^{k+1},\tau^i)}{\dfrac{\partial\phi}{\partial t_2}(t_1^{k+1},\tau^i)}, \qquad i=0,1,2,\cdots \end{aligned}\right\} \tag{15}$$

where we assume that $\dfrac{\partial\phi}{\partial t_2}(t^{k+1},\tau)\neq 0$ for all i.

In practice, whenever $|\phi(t_1^{k+1},\tau^i)|<\varepsilon$ for some u, where ε is a sufficiently small positive number, the iteration terminates, and the value of t_2^{k+1} is set equal to τ^{i+1}. It is expected that the point $t^{k+1}=(t_1^{k+1}, t_2^{k+1})$ satisfying $|\phi(t_1^{k+1}, t_2^{k+1})|<\varepsilon$ can be found within a few iterations of (15), provided ε is small enough.

Once t^{k+1} is determined, we have to calculate the optimal solution $x(t)$ of problem (1) for $t=t^{k+1}$. This can be done first by solving the system (6) to obtain $z(t^{k+1})$, and then by solving the equation $g(y, z(t^{k+1}), t^{k+1})=0$ to obtain $y(t^{k+1})$. Since this procedure is essentially the same as in a single parameter case, in the present case, we also employ the method given in [3].

Now, we are ready to state an algorithm to solve problem (5) parametrically.

Algorithm

Step 1: Determine an initial point $x(t^\circ)$ with $t^\circ=(t^\circ_1, t^\circ_2)$. Fix the upper bound of the parameter value t_1^* and t_2^*, and choose sufficiently small numbers $\alpha>0$, $\delta>0$ and $\rho>0$. Set $k=0$ and proceed to step 2.1.

Step 2.1: At $t=t^k$, partition x into basic variables y and nonbasic variables z, with $a_i+\alpha<y_i(t^k)<b_i-\alpha$ for each i. Proceed to step 2.2.

Step 2.2: The nonbasic variables z are divided into (z_I, z_{J_1}, z_{J_2}) in such a way that, $a_I<z_I(t^k)<b_I$, $z_{J_1}(t^k)=a_{J_1}$, and $z_{J_2}(t^k)=b_{J_2}$. Proceed to step 3.1.

Step 3.1: Determine the point $t^{k+1}=(t_1^{k+1}, t_2^{k+1})$ as described previously. Proceed to step 3.2.

Step 3.2: Set $z^\circ=z(t^k)$ and $j=0$.

Step 3.3: Compute z^{j+1} by

$$z_I^{j+1}=z_I^j-\nabla_{z_I z_I}^2 F(z^j, t^{k+1})^{-1}\nabla_{z_I}F(z^j, t^{k+1})$$
$$z_{J_1}^{j+1}=a_{J_1}, \qquad z_{J_2}^{j+1}=b_{J_2}$$

Step 3.4: If $\|z^{j+1}-z^j\|<\rho$, set $z(t^{k+1})=z^{j+1}$ and go to step 4.1. Otherwise, set $j=j+1$ and return to step 3.3.

Step 4.1: Calculate $y(t^{k+1})$ by solving $g(y, z(t^{k+1}), t^{k+1})=0$. If there exists some i such that $a_i+\alpha>y_i(t^{k+1})$ or $y_i(t^{k+1})<b_i-\alpha$, set $y_i(t^{k+1})=a_i$ for each i such that

$a_i+a<y_i(t^{k+1})$ is not satisfied. Also set, $y_i(t^{k+1})=b_i$ for all i, such that $y_i(t^{k+1})<b_i-a$ does not hold, and return to step 2.1. Otherwise, proceed to step 4.2.

Step 4.2: For each $i\in I$, if $a_i\leq z_i$ is not satisfied, set $z_i=a_i$. Similary, if $z_i\leq b_i$ does not hold, set $z_i=b_i$. Set $z_{J_1}=a_{J_1}$ for all $j\in J_1$ such that $\nabla_{z_j}F(z(t^{k+1}))<\delta$ is not satisfied. Likewise set $z_{J_2}=b_{J_2}$ for all $j\in J_2$ such that $\nabla_{z_j}F(z(t^{k+1}))>\delta$ does not hold, and return to step 2.2. Otherwise, proceed to step 5.

Step 5: If $t_1\geq t_1^*$ or $t_2\geq t_2^*$, then terminate. Otherwise, return to step 2.1.

4. Numerical Experiments

In this section, computer experiments for nonlinear programming problem (1) with a two dimensional parameter are carried out to test the algorithm. Specifically, the parametric programming problem solved here is expressed as follows:

$$\left.\begin{array}{ll}\text{minimize} & t_1f_1(x)+(1-t_1)f_0(x)\\ \text{subject to} & t_2g_1(x)+(1-t_2)g_0(x)=0, \quad a\leq x\leq b,\end{array}\right\} \tag{16}$$

where $g_0(x)$ have the same dimension as $g_1(x)$. Also, assume that the optimal solution of the problem for $t^0=(0,0)$ is known. For t_1, $t_2\in[0, 1]$ satisfying a certain nonlinear equation $\phi(t)=0$, we can obtain the optimal solution of problem (1) parametrically.

In our experiment, the functions f_1 and f_0 are respectively

$$f_1(x)=x_1^2+x_2^2+2x_3^2+x_4^2-5x_1-5x_2-21x_3+7x_4,$$

$$f_0(x)=\sum_{i=1}^{7}(x_i-x_i^\circ)^2,$$

where $x^\circ=(x_1^\circ, x_2^\circ, \cdots, x_7^\circ)$ is a given vector, and the functions g_1 and g_0 are respectively

$$g_{11}(x)=x_1^2+x_2^2+x_3^2+x_4^2+x_1-x_2+x_3-x_4+x_5-8$$

$$g_{12}(x)=x_1^2+2x_2^2+x_3^2+2x_4^2-x_1-x_4+x_6-10$$

$$g_{13}(x)=2x_1^2+x_2^2+x_3^2+2x_1-x_2-x_4+x_7-5$$

and

$$g_{01}(x)=g_{11}(x)-g_{11}(x^\circ)$$

$$g_{02}(x)=g_{12}(x)-g_{12}(x^\circ)$$

$$g_{03}(x)=g_{13}(x)-g_{13}(x^\circ).$$

The bounds for the variables x are $0\leq x_i$, $i=5$, 6, 7. Moreover, the parameter $t=(t_1, t_2)\in[0, 1]\times[0, 1]$ is supposed to satisfy the equation

$$\phi(t)=t_1^2-t_2=0.$$

Table 1.

t_1	t_2	x_1	x_2	x_3	x_4	x_5	x_6	x_7	f	cumulative CPU time (msec)
0.0	0.0	0 0	2.0	1.0	2.0	3.0	2.0	4.0	0.0	14
0.0997	0.0099	0.0653	1.9132	1.0197	1.9063	2.9900	1.9812	4.0158	−0.063	122
0.1967	0.0386	0.0844	1.8072	1.1548	1.7065	2.9691	1.9686	3.9988	−0.3493	245
0.2881	0.0830	0.0435	1.7078	1.3269	1.4579	2.9308	1.9681	3.9388	−1.0265	369
0.3731	0.1392	−0.0313	1.6259	1.4755	1.2182	2.8708	1.9866	3.8218	−2.1729	491
0.4518	0.2041	−0.1037	1.5619	1.5858	1.0143	2.7928	2.0280	3.6381	−3.7532	611
0.5246	0.2752	−0.1559	1.5121	1.6683	0.8351	2.7038	2.0907	3.3882	−5.7027	730
0.5924	0.3510	−0.1876	1.4730	1.7335	0.6533	2.6050	2.1722	3.0734	−7.9889	853
0.6862	0.4709	−0.2026	1.4219	1.8042	0.3069	2.4077	2.3208	2.4821	−12.1062	1040
0.7443	0.5540	−0.1865	1.3727	1.8311	0.0070	2.1775	2.4078	2.0472	−15.4391	1199
0.8259	0.6821	−0.1090	1.2460	1.8860	−0.3909	1.5517	2.3991	1.3848	−21.5955	1439
0.8771	0.7693	−0.0210	1.1431	1.9729	−0.5737	0.8806	2.1983	0.7668	−26.6614	1583
0.9261	0.8577	0.1121	1.0482	2.1386	−0.7249	0.0	1.5558	0.0	−33.6434	1738
0.9731	0.9470	0.3698	0.9949	2.0360	−0.9162	0 0	1.3111	0.0	−39.8501	1866
1.0	1.0	0.0	1.0	2.0	−1.0	0.0	1.0	0.0	−44.0	1973

Then the vector x° is obviously the optimal solution to problem (16) for $t=(0,0)$. It is noted that for $t=(1,1)$, problem (16) is equivalent to the well-known Rosen-Suzuki problem, whose optimal solution is $(0,1,2,-1,0,1,0)$ and the minimal value of the objective is -44. These computations were performed using double precision on a FACOM M 200 computer of the Kyoto University Computation Center. The results for $x^\circ=(0,2,1,2,3,2,4)$ of the calculations are summarized in Table 1.

5. Conclusion and Extensions

In this paper, we have presented a nonlinear programming problem with two parameter $t=(t_1, t_2)$, and tested the numerical experiment by a computer. The technique proposed so far iteratively generates a sequence of optimal solutions $x(t^k)$ for the successive values of the parameter t, such that $\phi(t^k)=0$.

Moreover, it is not difficult to establish similar results for the case of a three dimensional parameter, satisfying $\phi(t_1, t_2, t_3)=0$. In this case, equations (11) and (12) become

$$\nabla\phi_1(t)d=0,$$
$$\nabla\phi_2(t)d=0, \quad \text{and}$$
$$\|d\|=\sqrt{d_1^2+d_2^2+d_3^2}=\Delta$$

from which each component of the vector d is calculated in a manner analogous to (13). The expressions for the direction vector are given as follows:

$$d_1=D_{23}/D$$
$$d_2=D_{31}/D$$
$$d_3=D_{12}/D$$

where

$$D_{31}=\frac{\partial\phi_1}{\partial t_3}\frac{\partial\phi_2}{\partial t_1}-\frac{\partial\phi_2}{\partial t_3}\frac{\partial\phi_1}{\partial t_1}$$
$$D_{12}=\frac{\partial\phi_1}{\partial t_1}\frac{\partial\phi_2}{\partial t_2}-\frac{\partial\phi_2}{\partial t_1}\frac{\partial\phi_1}{\partial t_2}$$
$$D_{23}=\frac{\partial\phi_1}{\partial t_2}\frac{\partial\phi_2}{\partial t_3}-\frac{\partial\phi_2}{\partial t_2}\frac{\partial\phi_1}{\partial t_3}$$

and

$$D=\sqrt{D_{12}^2+D_{23}^2+D_{31}^2}\,.$$

References

1) J. Abadie and J. Carpentier, "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints," in Fletcher, R. (ed), *Optimization*, Academic Press, New York, 1969, pp. 34–47.
2) G. H. Meyer, "On Solving Nonlinear Equations with a One-parameter Operator Imbedding," *SIAM J. Numer. Anal*, Vol. 5, No. 4, 1968, pp. 739–752.
3) H. Mine, M. Fukushima, and Y. J. Ryang, "Parametric Nonlinear Programming for General Cases and Its Application to Some Problems," *Memoirs of the Faculty of Engineering*, Kyoto University, Vol. XL, Part 3, July 1978, pp. 198–211.
4) H. Mine and Y. J. Ryang, "Deformation Method Using Parametric Approach for Solving Nonlinear Programming Problems," *Proceedings of the Pacific Conference on Operations Research*, Seoul, Vol. 2, April 1979, pp. 831–847.
5) D. G. Luenberger, "Introduction to Linear and Nonlinear Programming," Addison Wesley, Reading, Mass., 1973.
6) J. M. Ortega and W. C. Rheinboldt, "Iterative Solution of Nonlinear Equations in Several Variables," Academic Press, New York, 1970.