Murphy, Frederic H.

Working Paper

A Generalized LaGrange Multiplier Function Algorithm for Nonlinear Programming

Discussion Paper, No. 114

Provided in Cooperation with:
Kellogg School of Management - Center for Mathematical Studies in Economics and Management Science, Northwestern University

Suggested Citation: Murphy, Frederic H. (1974) : A Generalized LaGrange Multiplier Function Algorithm for Nonlinear Programming, Discussion Paper, No. 114, Northwestern University, Kellogg School of Management, Center for Mathematical Studies in Economics and Management Science, Evanston, IL

This Version is available at:
https://hdl.handle.net/10419/220474

DISCUSSION PAPER NO. 114

A GENERALIZED LAGRANGE MULTIPLIER FUNCTION ALGORITHM FOR NONLINEAR PROGRAMMING

by

Frederic H. Murphy

October 31, 1974

ABSTRACT

An exponential penalty function algorithm is developed. This algorithm is also shown to be similar to the method of multipliers of Hestines [5] and Powell [9]. The limiting convergence rate behavior is similar to that of the method of multipliers

A Generalized Lagrange Multiplier Function Algorithm for Nonlinear Programming

Consider the nonlinear programming problem (NLP)

$$\underset{x \in E^n}{\text{Minimize}} \quad f(x) \tag{1}$$

subject to

$$g_i(x) \leq 0 \text{ for } i = 1,\ldots,m \tag{2}$$

with an optimal solution x^*.

Penalty function methods [4] for solving NLP involve constructing and solving a sequence of easier unconstrained minimizations, where with each iteration the penalty for infeasibility increases. In a like manner, barrier function methods involve solving a sequence of unconstrained minimization problems where the penalty for being near the boundary while feasible decreases. As pointed out by Luenberger [6 , p. 289] a characteristic of penalty and barrier methods is that they are not really iterative, that is, the trial solution at iteration $k - 1$ is in no way related to the solution at iteration k. An alternative approach that is iterative, the method of multipliers [2 , 5 , 9], transforms NLP to a class of nonlinear programming problem with the same Kuhn-Tucker points:

$$\text{Minimize } f(x) + r \sum_{i=1}^{m} [\max(0, g_i(x))]^2 \tag{3}$$

subject to

$$g_i(x) \leq 0 \text{ for } i = 1,\ldots,m \ , \tag{4}$$

where $r \geq 0$.

Trial values of the Lagrange multiplier, $\lambda_1^k,\ldots,\lambda_m^k$ at iteration k, are then used to form a Lagrangian

$$f(x) + r \sum_{i=1}^{m} [\max(0, g_i(x))]^2 + \sum_{i=1}^{m} \lambda_i^k \, g_i(x) \ , \tag{5}$$

which is then minimized. Letting x_k minimize (5) we then update λ_i^k by the following procedure:

$$\lambda_i^{k+1} = \begin{cases} \lambda_i^k \text{ if } g_i(x)_k \leq 0 \\ 2r \, g_i(x) + \lambda_i^k \quad \text{otherwise} \end{cases} \tag{6}$$

Using this update procedure on the λ_i^k's with r sufficiently large, a Kuhn-Tucker point of NLP is achieved in the limit with very general conditions on $f(x)$, $g_1(x),\ldots,g_m(x)$ [2].

In penalty function algorithms the primary goal is to determine a good trial value for the primal problem with the trial dual variables a subsidiary interest. The method of multipliers has a different focus. The goal is to determine good estimates of the Lagrange multipliers which are used in the Lagrangian of a nonlinear program with the same Kuhn-Tucker points as NLP. The Lagrangian is then minimized and improved estimates of the multipliers are determined. The main importance of the trial x values within the algorithm is in estimating the new λ's. The estimates of the multipliers are improved by estimating the difference between the trial values of the multipliers and the actual values of the optimal Lagrange multipliers.

We present here an algorithm that estimates the percentage of error in the trial multipliers and corrects for this error. The advantage of this approach is that the algorithm becomes self-scaling. That is, the constraints need not be multiplied by constants to transform NLP so that the multipliers are relatively close to each other. Therefore, as an alternative to (3) and (4) we may consider substituting the following set of constraints for (2) in NLP:

$$\exp[rg_i(x)]g_i(x) \leq 0 \text{ for } i = 1,\ldots,m, \tag{7}$$

where $r > 0$. However, even if $g_i(x)$ is convex, $\exp[rg_i(x)]g_i(x)$ is not necessarily even pseudo-convex. To remedy this we define

$$g_i(x,r) = \begin{cases} \exp[rg_i(x)]g_i(x) \text{ if } g_i(x) \geq -\frac{1}{r} \\ -\frac{1}{r}e^{-1} \quad \text{otherwise.} \end{cases} \tag{8}$$

Substituting (8) for (2) we have the following nonlinear program (NLP1):

$$\text{Minimize } f(x) \tag{9}$$

subject to

$$g_i(x,r) \leq 0 \text{ for } i = 1,\ldots,m. \tag{10}$$

We now state two preliminary results.

Theorem 1.

a) $(\bar{x},\bar{\lambda})$ is a Kuhn-Tucker point of NLP1 if and only if it is a Kuhn-Tucker point of NLP.

b) If $g_i(x)$ is convex and differentiable for $i = 1,\ldots,m$, then $g_i(x,r)$ is convex and differentiable for $i = 1,\ldots,m$.

Proof:

Part a: If $(\bar{x},\bar{\lambda})$ is a K-T point for NLP then

$$\nabla f(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \nabla g_i(\bar{x}) = 0 \tag{11}$$

$$\bar{\lambda}_i g_i(\bar{x}) = 0 \text{ for } i = 1,\ldots,m \tag{12}$$

$$\bar{\lambda}_i \geq 0,\ g_i(\bar{x}) \leq 0 \text{ for } i = 1,\ldots,m. \tag{13}$$

Now

$$\begin{aligned}
&\nabla f(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \nabla g_i(\bar{x},r) \\
&= \nabla f(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \begin{cases} (rg_i(\bar{x}) + 1)\exp[rg_i(\bar{x})]\nabla g_i(\bar{x}) & \text{if } g_i(\bar{x}) \geq -\frac{1}{r} \\ 0 & \text{otherwise} \end{cases} \\
&= \nabla f(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i (rg_i(\bar{x}) + 1)\exp[rg_i(\bar{x})]\nabla g_i(\bar{x}) \quad \text{(by (12))} \qquad (14) \\
&= \nabla f(\bar{x}) + \sum_{i=1}^{m} \bar{\lambda}_i \nabla g_i(\bar{x}) \\
&= 0
\end{aligned}$$

with the second to the last equality holding by substituting 0 for $g_i(x)$ when $\lambda_i > 0$, by (12) again. Clearly $g_i(x,r) = 0$ if and only if $g_i(x) = 0$ and $g_i(x,r) \leq 0$ if and only if $g_i(x) \leq 0$. Therefore, $(\bar{x},\bar{\lambda})$ is a K-T point of NLP1. For the converse we need only reverse the process in (14).

Part b: To prove b we need only show that for $y \in E^1$

$$h(y) = \begin{cases} ye^y & x \geq -1 \\ -e^{-1} & x \leq -1 \end{cases} \tag{15}$$

is convex and monotonic, since the composition of a convex monotonic function with a convex function is convex [7].

$$D_y(ye^y) = (y + 1)e^y. \tag{16}$$

Now (16) is equal to zero if and only if $y = -1$ making $h(y)$ differentiable, and (16) is positive if $y > -1$, that is, $h(y)$ is monotonic. Finally,

$$D_y^2(ye^y) = (y + 2)e^y \tag{17}$$

which is greater than zero for $y \geq -1$, with the result that $h(y)$ is convex.

Note that Theorem 1 holds for all $r > 0$.

The Algorithm

We now state an algorithm based on determining trial solutions to NLP1 and prove convergence to an optimal solution to NLP.

Step 1: At iteration k given $\lambda_1^k,\ldots,\lambda_m^k$ with $\lambda_i^k > 0$ for $i = 1,\ldots,m$ and given $r = r_k$ where $r_k \to \infty$ as $k \to \infty$ we minimize

$$f(x) + \sum_{i=1}^{m} \lambda_i^k g_i(x,r_k) \tag{18}$$

letting x_k be the x that minimizes (18).

Step 2: For $i = 1,\ldots,m$, let

$$\gamma_i^k = \lambda_i^k [r_k g_i(x_k) + 1] \exp [r_k g_i(x_k)]. \tag{19}$$

then set

$$\lambda_i^{k+1} = \min \left[U_k, \max(\gamma_i^k, L_k) \right] \text{ for } i = 1,\ldots,m \qquad (20)$$

where $0 < L_k < U_k$, $L_k \to 0$, $U_k \to \infty$, $r_k L_k \to \infty$, and $U_k/r_k \to 0$ as $k \to \infty$

Return to Step 1.

We introduce L_k and U_k as lower and upper bounds on the trial Lagrange multipliers to ensure convergence of the algorithm. They do not, however, represent much of a practical limitation. Admittedly, not allowing $\lambda_i^k = 0$ means we always have to carry along constraints that are not binding. The burden is not excessive, however, since for these constraints $\nabla g_i(x,r_k) = 0$ when $g_i(x) \leq -\frac{1}{r_k}$, eliminating a major part of the computation with respect to a nonbinding constraint in any unconstrained algorithm. This algorithm is a member of the class of expotential penalty functions as presented in Evans and Gould [3] and a variant of the penalty function in Murphy [8].

We now prove convergence of the algorithm.

Theorem 2.

Assume (a) The feasible region is compact and nonempty

(b) $x_k \in X$ a compact set

(c) $r_k \to \infty$ as $k \to \infty$.

(d) $f(x), g_1(x),\ldots,g_m(x)$ are continuous

then any convergent subsequence of x_k converges to an optimal solution of NLP.

Proof: Let x_{k_u} be a convergent subsequence with limit $\bar{x}$. We first show that $\bar{x}$ is feasible. Assume that $\bar{x}$ is not feasible, that is, $g_h(\bar{x}) \geq \epsilon > 0$ for some h with $1 \leq h \leq m$. By (d) we can say that there exists a K such that for $k_u \geq K$ $g_h(x_{k_u}) \geq \frac{\epsilon}{2}$. This means that as $k_u \to \infty$ $g_h(x_{k_u}, r_{k_u}) \to \infty$. Since $\lambda_i^{k_u} \geq L_{k_u} > 0$, by the definition of L_k the limit of the minimum of (18) is then $+\infty$ which contradicts (a), because (18) evaluated at a feasible point is uniformly bounded for all r. Consequently, $\bar{x}$ is feasible. We now show that $\bar{x}$ is optimal in NLP. Because x* is feasible,

$$
\begin{aligned}
f(x^*) &\geq f(x^*) + \sum_{i=1}^{m} \lambda_i^k g_i(x^*, r_k) \\
&\geq f(x_k) + \sum_{i=1}^{m} \lambda_i^k g_i(x_k, r_k) \\
&\geq f(x_k) + U_k m\left(\frac{-1}{r_k}\right) \\
&\geq f(x^*) - U_k m / r_k,
\end{aligned}
\tag{21}
$$

for all x_k since $-1/r_k$ is a lower bound for $g_i(x,r)$ for all x whether feasible or not for $i = 1,\ldots,m$, and U_k is an upper bound for λ_i^k for $i = 1,\ldots,m$. Taking limits on the subsequence k_u as $k \to \infty$, $f(\bar{x}) = f(x^*)$ or $\bar{x}$ is an optimal solution to NLP.

The proof of convergence presented here is weaker than the proof of convergence of the method of multipliers as presented in [2], in that we require $r_k \to \infty$. However, none of the local duality conditions that are necessary for guaranteeing convergence with r_k bounded in the method of multipliers are needed here. Also, in computational experiments Bertsekas [2] found it advantageous to update r_k as is done with the standard exterior penalty function, for example, setting $r_{k+1} = a \cdot r$ with $a > 1$.

From now on we assume that $f(x)$, $g_1(x),\ldots,g_m(x)$ are convex and satisfy Slater's constraint qualification. Using standard procedures [4] the following theorem can be proved:

Theorem 3:

Let x^* be an optimal solution to NLP. Let $\bar{\gamma}_1,\ldots,\bar{\gamma}_m$ be the limit of any convergent subsequence of $\gamma_1^k,\ldots,\gamma_m^k$. Then $x^*,\bar{\gamma}_1,\ldots,\bar{\gamma}_m$ exist and constitute a Kuhn-Tucker point for NLP and NLP1.

Convergence Rate Results

The convergence rate results are quite similar to those for the method of multipliers. However, they have to be developed in a different manner. We need to establish the following lemmas to determine our theoretical rate of convergence. The proofs of these lemmas use the fact that the algorithm converges which in turn uses $r_k \to \infty$ as $k \to \infty$.

Lemma 1. Let $I = \{ i | g_i(x^*) = 0 \}$. Let $\lambda_1^*, \ldots, \lambda_m^*$ be the corresponding dual variables. Assume $\lambda_1^*, \ldots, \lambda_m^*$ are unique and for all $i \in I$ $\lambda_i^* > 0$. Then for $i \in I$, $\lim_{k \to \infty} \lambda_i^k = \bar{\gamma}_i = \lambda_i^*$.

Proof: By Theorem 3 and the uniqueness of $\lambda_1^*, \ldots, \lambda_m^*$, $\gamma_i^k \to \lambda_i^*$ for $i = 1, \ldots, m$, and since $L_k \leq \lambda_i^* \leq U_k$ for $k \geq K$ for some K by the strict complementarity assumption

$$\lim_{k \to \infty} \min[U_k, \max(L_k, \gamma_i^k)] = \lambda_i^* \text{ for } i \in I, \tag{22}$$

or

$$\lim_{k \to \infty} \lambda_i^k = \lambda_i^* \text{ for } i \in I.$$

We now present our first convergence rate result.

Lemma 2. If the conditions of Lemma 1 hold, then

$$\lim_{k \to \infty} r_k g_i(x_k) = 0 \text{ for } i \in I . \tag{23}$$

Proof. For $i \in I$

$$\gamma_i^k = \lambda_i^k r_k g_i(x_k) \exp[r_k g_i(x_k)] + \lambda_i^k (\exp[r_k g_i(x_k)] - 1) + \lambda_i^k. \tag{24}$$

Since γ_i^k and λ_i^k have the same limit by Lemma 1, $\gamma_i^k - \lambda_i^k \to 0$ as $k \to \infty$ and

$$\lambda_i^k r_k g_i(x_k) \exp[r_k g_i(x_k)] + \lambda_i^k (\exp[r_k g_i(x_k)] - 1) \to 0. \tag{25}$$

Since the sign of each term in (25) is the same as the sign of $g_i(x_k)$, each term must converge to zero as $k \to \infty$. From either term we see that (23) holds.

This result illustrates the superiority of the convergence rate of this algorithm over penalty and barrier methods. Examining a barrier method first, we have for the barrier function

$$f(x) - \sum_{i=1}^{m} [r_k g_i(x)]^{-1} \tag{26}$$

evaluated at its minimum x_k

$$\nabla f(x_k) + \sum_{i=1}^{m} [r_k g_i^2(x_k)]^{-1} \nabla g_i(x_k) = 0 \tag{27}$$

Now $[r_k\ g_i^2(x_k)]^{-1}$ is uniformly bounded [4p.99] and for $i \in I, [g_i(x_1)]^{-1} \to \infty$ which means $[r_k\ g_i(x_k)]^{-1} \to 0$, or $r_k\ g_i(x_k) \to \infty$. In like manner with the penalty function

$$f(x) + \sum_{i=1}^{m} r_k\ [\max(0, g_i(x))]^2 \tag{28}$$

any convergent subsequence of $2\ r_k\ g_i(x_k)$ for $i = 1,\ldots,m$ converges to λ_i^* the optimal Lagrange multiplier for constraint i, which we assume to be greater than zero for $i \in I$. The conclusion we may draw is that the algorithm presented here gives us an x_k with $g_i(x_k)$ converging to zero for $i \in I$ more quickly than the penalty or barrier function algorithms.

Also this lemma is important from a computational point of view. The point $-r_k^{-1}$ is where there is a discontinuity in the Hessian of $g_i(x, r_k)$ that could interfere with estimating the Hessian of the penalty function. Since $r_k g_i(x_k) \to 0$, we know that for $i \in I$, $g_i(x_k) > -r_k^{-1}$ for k sufficiently large. With the trial solution sufficiently far from the discontinuity it would seem that numerical problems in dealing with this discontinuity should occur only as an accident.

Let us assume that the constraints $1,\ldots,p$ with $p \leq m$ are the binding constraints. Since the gradients of the nonbinding constraints in NLP are zero near the optimal solutions for r_k sufficiently large, we need only consider the problem (NLP2).

$$\text{minimize } f(x) \tag{29}$$

subject to

$$g_i(x,r) = 0 \text{ for } i = 1,\ldots,p \tag{30}$$

for determining limiting results on the convergence rate. Also assume $\lambda_i > 0$ for $i = 1,\ldots,p$. Therefore, after a finite number of iterations $\gamma_i^k = \lambda_i^k$ for $i = 1,\ldots,p$. Also by assumption $f(x)$ and $g_i(x)$ are convex. Finally, we also assume that either $f(x)$ or a $g_i(x)$ for $i = 1,\ldots,p$ is strictly convex and $\nabla g_1(x^*),\ldots,\nabla g_p(x^*)$ are linearly independent.

Using the notions of local duality in Luenberger [6, p. 321] we define the dual function φ of NLP2

$$\varphi(\lambda) = \underset{x \in E^n}{\text{minimum}} \left[f(x) + \sum_{i=1}^{p} \lambda_i g_i(x,r) \right] \tag{31}$$

where $\lambda = (\lambda_1,\ldots,\lambda_p)$

The function φ is convex, since we have assumed $f(x), g_1(x),\ldots,g_p(x)$ to be convex. With r_k sufficiently large so that $g_i(x_k) > -r_k^{-1}$ for $1 \leq i \leq p$, φ has gradient [6, p. 321],

$$\nabla\varphi(\lambda^k) = \begin{bmatrix} e^{r_k g_1(x_k)} g_1(x_k) \\ \vdots \\ e^{r_k g_p(x_k)} g_p(x_k) \end{bmatrix} \tag{32}$$

The gradient (32) is the direction of steepest descent for the dual function. The direction of change for the λ_i's provided by the algorithm, however, is

$$\lambda_i^{k+1} - \lambda_i^{k} = \lambda_i^{k} r_k g_i(x_k)\exp[r_k g_i(x_k)] + \lambda_i^{k}(\exp[r_k g_i(k)] - 1). \qquad (33)$$

In the method of multipliers the procedure for estimating λ^* at iteration k is to move in the direction of steepest descent from λ^k. We see that (33) is not the direction of steepest descent. The relationship between (32) and (33) is now to be developed.

Let

$$\nabla g(x) = \begin{bmatrix} \nabla g_1(x) \\ \vdots \\ \nabla g_p(x) \end{bmatrix}, \qquad (34)$$

let

$$H(r_k,x) = \begin{bmatrix} [r_k g_1(x) + 1]\exp[r_k g_1(x)] & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & [r_k g_p(x)+1]\exp[r_k g_p(x)] \end{bmatrix}, \qquad (35)$$

let

$$\Lambda(\lambda) = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & \lambda_p \end{bmatrix}. \qquad (36)$$

and let $F(x)$ be the Hessian of $f(x)$ and let $G_i(x)$ be the Hessian of $g_i(x)$ for $i = 1,\ldots,p$.

From this we can construct the Hessian of $\varphi(\lambda^k)$, [6].

$$H(r_k,x_k)\nabla g(x_k)[F(x_k) + \sum_{i=1}^{p} \lambda_i^k[r_k g_i(x_k) + 1]\exp[r_k g_i(x_k)]G_i(x_k)$$

$$+ r_k \nabla g(x_k)'\Lambda(\lambda^k)H(r_k x_k)\nabla g(x_k)]^{-1} \nabla g(x_k)'H(r_k,x_k)' \tag{37}$$

The inverse exists because we have assumed strict convexity for one of the functions $f(x)$, $g,(x),\ldots,g_p(x)$. Multiplying and dividing by r_k we have

$$\frac{1}{r_k} H(r_k,x_k)\nabla g(x_k)[F(x_k)/r_k + \sum_{i=1}^{p} \lambda_i^k/r_k[r_k g_i(x_k)+1]\exp[r_k g_i(x_k)]G_i(x_k)$$

$$+ \nabla g(x_k)'\Lambda(\lambda_k)H(r_k,x_k)\nabla g(x_k)]^{-1} \nabla g(x_k)'H(r_k,x_k)'. \tag{38}$$

Let

$$\Gamma_k = \begin{bmatrix} \sqrt{\lambda_1^k} & & \\ & \ddots & \\ & & \sqrt{\lambda_p^k} \end{bmatrix}, \tag{39}$$

that is $\Lambda(\lambda^k) = \Gamma_k \Gamma_k$. Now

$$\Gamma_k M_k \Gamma_k = \Gamma_k M_k \Gamma_k' \tag{40}$$

$$= \frac{1}{r_k} \Gamma_k H(r_k,x_k) \nabla g(x_k)[F(x_k)/r_k + \sum_{i=1}^{p} \lambda_i^k/r_k[r_k g_i(x_k)+1]\exp[r_k g_i(x_k)]G_i(x_k)$$

$$+ [\Gamma_k \nabla g(x_k)]' \Gamma_k H(r_k,x_k)\nabla g(x_k)]^{-1} \nabla g(x_k)' H(r_k,x_k)' \Gamma_k' .$$

Noting that $H(r_k,x_k) \to I_p$ as $k \to \infty$ by [9,p.294] or [6,p.321] we can say

$$\lim_{r\to\infty} r_k \Gamma_k M_k \Gamma_k = I. \tag{41}$$

We may now conclude.

Theorem 4.

Let $\lambda_1^*,\ldots,\lambda_p^*$ be the corresponding dual variables to NLP2. Assume $f(x)$, $g_1(x),\ldots,g_m(x)$ are convex with one of these functions strictly convex. If $\lambda_1^*,\ldots,\lambda_p^*$ are unique for all $i = 1,\ldots,p$ and $\lambda_i^* > 0$ then the limit of $r_k M_k$ as $k \to \infty$ is $\Lambda(\lambda^*)^{-1}$ where

where

$$\Lambda(\lambda^*)^{-1} = \begin{bmatrix} 1/\lambda_1^* & \cdots & 0 \\ & \ddots & \\ 0 & \cdots & 1/\lambda_p^* \end{bmatrix} .$$

Proof:

$$\begin{aligned} r_k M_k &= \Gamma_k^{-1} \Gamma_k r_k M_k \Gamma_k \Gamma_k^{-1} \\ &= \Gamma_k^{-1} \left[r_k \Gamma_k M_k \Gamma_k \right] \Gamma_k^{-1} . \end{aligned} \tag{42}$$

Since the limit of the bracketed matrix exists and is equal to the identity matrix, the limit of the product is the product of the limits and

$$\lim_{k\to\infty} r_k M_k = \lim_{k\to\infty} \Gamma_k^{-1} \; I \; \lim_{k\to\infty} \Gamma_k^{-1} \tag{43}$$

which is just $\Lambda(\lambda^*)^{-1}$.

As a consequence, for large k the Hessian of $\varphi(\lambda)$ is approximately $\frac{1}{r_k} \Lambda (\lambda^k)^{-1}$. This is different from the method of multipliers where for large r_k the Hessian of $\varphi(x)$ is approximately $\frac{1}{2r_k} I$. The key to the success of the method of multipliers is that the update of λ^k to λ^{k+1} is steepest ascent on $\varphi(\lambda)$ with the Hessian of $\varphi(\lambda)$ approximately the identity matrix divided by r_k. That is we have approximately a Newton type iteration on maximizing the dual function.

Let us see how the difference in the Hessians is accounted for in this algorithm.

Theorem 5.

With the same assumptions as in Theorem 4:

$$(\lambda^{k+1}-\lambda^{k}) = 2r_k \Lambda(\lambda^{k}) [g_i(x_k,r_k),\ldots,g_m(x_k,r_k)]' + o[r_k g_1(x_k),\ldots,r_k g_p(x_k)] \quad (44)$$

where

$$h(x) = o(x) \text{ means } \frac{h(x)}{x} \to 0 \text{ as } x \to 0.$$

Proof:

Let us re-examine (33) for a single constraint i. Defining $y = r_k g_i(x_k)$ we have

$$\lambda_i^{k+1} - \lambda_i^{k} = \lambda_i^{k}(ye^{y} + e^{y} - 1). \quad (45)$$

Using the Taylor's formula for e^{y}, the term in the parenthesis in (45) can be expressed as

$$\begin{aligned} y\,e^{y}+e^{y}-1 &= y+y^2+\tfrac{1}{2}\exp(z_1)y^3+1+y+\frac{y^2}{2}+\frac{1}{6}\exp(z_2)y^3-1 \\ &= 2y+\frac{3}{2}y^2+(\tfrac{1}{2}\exp(z_1)+\frac{1}{6}\exp(z_2))y^3 \\ &= 2y(1+y)-\tfrac{1}{2}y^2+(\tfrac{1}{2}\exp(z_1)+\frac{1}{6}\exp(z_2))y^3 \\ &= 2y(1+y+\tfrac{1}{2}\exp(z_3)y^2)+\tfrac{1}{2}y^2+(\tfrac{1}{2}\exp(z_1)+\frac{1}{6}\exp(z_2)-\tfrac{1}{2}\exp(z_3))y^3 \\ &= 2y\,e^{y}+o(y). \end{aligned} \quad (46)$$

where $0 \leq z_1, z_2, z_3 \leq y$. Consequently

$$\lambda_i^{k+1} - \lambda_i^k = 2r_k \lambda_i^k g_i(x_k) \exp [r_k g_i(x_k)] + o[r_k g_i(x_k)] \tag{47}$$

Or

$$\lambda^{k+1} - \lambda^k = 2r_k \Lambda(\lambda^k)[g_1(x_k,r_k),\ldots,g_p(x_k,r_k)]' + [o[r_k g_1(x_k)],\ldots,o[r_k g_p(x_k)]]' \tag{48}$$

and the theorem holds.

Let us examine the implications of Theorem 5. In searching for the unconstrained minimum of a function h(x) a Newton iteration from a point x_k is

$$x_{k+1} = x_k + 2[H(x_k)]^{-1} \nabla h(x_k) \tag{49}$$

where H(x) is the Hessian of h(x).

Since the Hession of $\varphi(x_k)$ for k sufficiently large is approximately $\Lambda(\lambda^k)^{-1}$, the inverse of the Hessian is approximately $\Lambda(\lambda^k)$. From Theorem 5 we have

$$\lambda^{k+1} \approx \lambda^k + 2r_k \Lambda(\lambda^k) \nabla \varphi(\lambda^k) \tag{50}$$

where the right hand side of (50) is a Newton iteration as in (49)

Because $\Lambda(\lambda)$ is a diagonal matrix, the updating procedure for λ^k is approximately steepest ascent with a transformation of scale in the vector space of λ's. Within the transformed space the Hessian of $\varphi(\lambda)$ is approximately the identity matrix times $\frac{1}{r_k}$ for large r_k. The convergence of steepest ascent is linear and dependent on the ratio of the largest and smallest eiginvalues of the Hessian. Since the ratio of the eigenvalues of $\varphi(\lambda)$ within the transformed space approach 1, we have arbitrarily fast linear convergence or superlinear convergence.

Conclusion:

Now that we have examined $\varphi(\lambda)$, we can heuristically show the difference in philosophy between this algorithm and the method of multipliers. The method of multipliers tries to estimate the difference between λ^k and λ^*. Here, however, we are estimating the percentage of error between λ^k and λ^* and correcting λ^k with this estimate. At this point in time it is difficult to say which algorithm is superior if either is. However, on a few small test problems the algorithm presented here was clearly superior to SUMT.

1. Bertsekas, Dimitri, "On Penalty and Multiplier Methods for Constrained Minimization," Dept. Electrical Engineering, working paper, University of Illinois at Urbana - Champaign, April 1974.

2. Bertsekas, Dimitri, "Combined Primal - Dual and Penalty Function Methods for Constrained Minimization," SIAM J. Control, Vol. 13, No. 3 1975.

3. Evans, J. P. and F. J. Gould, "An Existence Theorm for Penalty Function Theory," Center for Mathematical Studies in Business and Economics, Report 7248, Univ. Chicago, October 1972.

4. Fiacco, A. V. and G. P. McCormick, Nonlinear Programming: Sequential Unconstrained Minimization Techniques, Wiley, New York, 1968.

5. Hestines, M. R., "Multiplier and Gradient Methods," J. Optimization Theory and Appl. 4 (1969), pp. 303 - 320.

6. Luenberger, D. G., Introduction to Linear and Nonlinear Programming, Addison-Wesley, Reading, Mass., 1973.

7. Mangasarian, O. L., Nonlinear Programming, McGraw-Hill New York, 1969.

8. Murphy, F. H., "A Class of Exponential Penalty Functions," SIAM J. Control, 1974.

9. Powell, M.J.D., "A Method for Nonlinear Constraints in Minimization Problems," in Optimization, R. Fletcher (ed.) Academic Press, New York, 1969.