International Symposium on Nonlinear Theory and its Applications
Xi'an, PRC, October 7–11, 2002

An Algorithm for Globally Minimizing a Function with Many Local Minima whose Lower Envelope is Unimodal

Hideo KANEMITSU[1)], Hideaki KONNO[2)], Masaaki MIYAKOSHI[3)] and Masaru SHIMBO[4)]

[1) 2)] Hakodate Campus,
Hokkaido University of Education
Hakodate 040-8567, Japan
Phone: +81-138-44-{4351[1)], 4366[2)]}
Fax: +81-138-44-{4351[1)], 4366[2)]}
{hkanemt, konno}@cc.hokkyodai.ac.jp

[3)] Graduate School of Engineering,
Hokkaido University
Sapporo 060-8628, Japan
Phone: +81-11-706-6810
Fax: +81-11-706-7830
miyakosi@main.eng.hokudai.ac.jp

[4)] Information Science,
Hokkaido Information University
Ebetsu 069-8585, Japan
Phone: +81-11-385-4411
Fax: +81-11-384-0134
shimbo@do-johodai.ac.jp

Summary

We describe an algorithm for finding the global minimum of a function with many local minima whose lower envelope is unimodal. At first we focus on a univariate function that is expressed by the sum of a unimodal function and a periodic function, and we describe three conditions for a univariate function. We propose an algorithm uses a two-stage minimizer for this function, a *large-step minimizer* (main stage) and a *small-step local minimizer* (sub-stage), and we show using a numerical example that the algorithm effectively finds the global minimum with only a few function evaluations. Moreover, we assess the usefulness of an algorithm for globally minimizing multivariate functions, and we show using a numerical example that the algorithm finds the global minimum of multivariate functions with only a few function evaluations.

Keywords Global Minimization, Global Optimization, Optimization, Algorithm

1. Introduction

We deal with the following minimization problem (P):

$$\text{minimize} \quad f(\boldsymbol{x}) \equiv f(x^1, x^2, \ldots, x_n)\ , \qquad \boldsymbol{x} \in D^n \equiv \prod_{j=1,\ldots,n} [a_j, b_j]\ . \tag{P}$$

It is assumed that the function $f(\boldsymbol{x}) \in C^2$ has a finite number of isolated local minima $\boldsymbol{x}_k^* \in D^n$ $(k = 1, 2, \ldots, M)$.

Many methods for solving such a global minimization problem with multimodal functions have been proposed.

Clustering methods [3, 4] that enable the global minimum to be found with high accuracy and a only few function evaluations have usually been used. These method consists of two stages: (1) sampling points over the region and selecting the lower points, and (2) analyzing clusters of selected points and applying local minimizer from the best point in each cluster.

However, clustering methods do not work well for functions with many local minima because there are many clusters around the local minima.

To overcome such a problem, we propose an algorithm that minimizes a function with many local minima whose lower envelop is unimodal.

2. Algorithm for a Univariate Function

2.1. A Problem and An Example

We consider the following univariate function:

$$\text{minimize} \quad f(x) = u(x) + p(x), \quad x \in D^1 = [a_1, b_1] \tag{1}$$

where $u(x)$ is a unimodal function, and $p(x)$ is a periodic function with period T^p and a finite number of isolated local minima. Moreover, we assume the following conditions hold for $p(x)$ and $u(x)$.

Condition 1. The period T^p of $p(x)$ is much less than the width $b_1 - a_1$ of the interval $D^1 = [a_1, b_1]$, that is

$$T^p << b_1 - a_1. \tag{2}$$

Condition 2. The maximum absolute value of derivatives of $u(x)$ in $[a_1, b_1]$ is less than that of $p(x)$, that is

$$\max_{x \in [a_1,\, b_1]} \{ |\mathrm{d}u(x)/\mathrm{d}x| \} < \max_{x \in [a_1, b_1]} \{ |\mathrm{d}p(x)/\mathrm{d}x| \}. \tag{3}$$

Condition 3. The function $u(x)$ has only a unique minimum x_{**}, and the point x_{**} also becomes one of minima of $p(x)$.

If the above following conditions hold, then the function $f(x) = u(x) + p(x)$ has many local minima ($\approx \lfloor (b_1 - a_1)/T^p \rfloor$) in $[a_1, b_1]$, its lower envelope is unimodal, and the global minimum of $f(x)$ is x_{**}.

For example, let $u(x) = 0.1x^2 + 0.01$, $p(x) = -0.01\cos(40\pi x)$ and $D^1 = [-1, 1]$, we consider the following univariate problem:

$$\begin{aligned} \text{minimize} \quad f(x) \equiv& u(x) + p(x) \\ =& 0.1x^2 + 0.01 - 0.01\cos(40\pi x), \\ & x \in [-1, 1]. \end{aligned} \tag{P1}$$

In this problem, $T^p = 1/20$ ($<< b_1 - a_1 = 2$), and in equation (3),

$$\max_{x\in[a_1,b_1]}\{\, |\mathrm{d}(0.1x^2 + 0.01)/\mathrm{d}x| \,\} = \max_{x\in[-1,1]}\{\, |0.2x| \,\} = 0.2$$

is less than

$$\begin{aligned} &\max_{x\in[a_1,b_1]}\{\, |\mathrm{d}(-0.01\cos(40\pi x))/\mathrm{d}x| \,\} = \\ &\max_{x\in[-1,1]}\{\, |-0.4\pi\cos(40\pi x)| \,\} = 0.4\pi \approx 1.256. \end{aligned}$$

Therefore, the problem satisfies **Condition 1** and **Condition 2**, and the function has 41 local minima ($\approx (b_1 - a_1)/T^p = 2/0.05 = 40$) in the interval $[-1, 1]$. Since the unique minimum of $u(x)$ is 0 and 0 is also one minimum of $p(x)$, the minimum of $f(x)$ is 0.

The graph of the function $f(x)$ in this problem is shown in Figure 1.

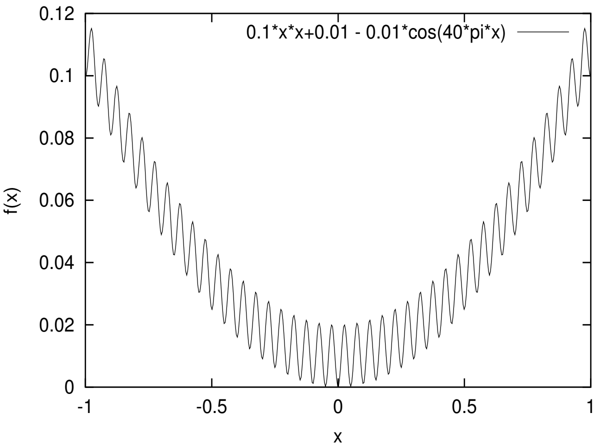

Figure 1: An example of a univariate function with many local minima

2.2. Minimizer

In order to describe our algorithm, we need the two general idea of *unimodal region* and *radius of unimodal region*. Thus, we show the definitions of the concepts at first.

Since the main idea of the proposed algorithm is to use a two-stage minimizer, i.e., a *large-step minimizer* (main stage) and a *small-step local minimizer* (sub-stage), we describe these two local minimizers.

2.2.1. Definitions

Definition 1 The *unimodal region* [2] $R_u(x_*^i)$ of a local minimum x_*^i of a function f be defined as the maximum region (interval) such that the function f is unimodal:

$$R_u(x_*^i) = \{\, [a_*^i, b_*^i] \mid \max_{[a^i,b^i]}\{f(x)\text{: unimodal in } [a^i, b^i]\,\}\,\}, \tag{4}$$

where $f(x)$: *unimodal* in $[a^i, b^i]$ implies that the following relationships hold.

$$\begin{cases} a^i \le \forall x^1 < \forall x^2 \le x_*^i \Longrightarrow f(x^1) > f(x^2) \\ x_*^i \le \forall x^1 < \forall x^2 \le b^i \Longrightarrow f(x^1) < f(x^2). \end{cases} \tag{5}$$

Definition 2 The radius $r(x_*^i)$ of the unimodal region $R_u(x_*^i) = [a_*^i, b_*^i]$ at a local minimum x_*^i is defined as the minimum of two distances from the local minimum x_*^i to the lower bound a_*^i and the upper bound b_*^i of the interval:

$$r(x_*^i) = \min\{\, x_*^i - a_*^i,\ b_*^i - x_*^i \,\}. \tag{6}$$

2.2.2. Large-step Minimizer

The steps of the minimizer are as follows:

- Bracketing global minimum by a interval, and
- Reducing the interval so that it includes the global minimum.

For a given starting point $x^{(0)}$, the algorithm repeatedly generates a new point $x^{(k+1)}$ from the previous point $x^{(k)}$ and $\Delta^{(k)}$ that is determined by a certain rule as follows:

$$x^{(k+1)} \leftarrow x^{(k)} + \Delta^{(k)}, \qquad (k = 0, 1, 2, \ldots). \tag{a}$$

Moreover, for each generated point $x^{(k)}$ the *small-step local minimizer* is applied, and a local minimum is obtained. In order to avoid converging in duplicate to the same local minimum in the *small-step local minimizer*, the large-step algorithm must keep to the step length $|\Delta^{(k)}| > T^p$. Thus, step (a) is replaced by the following step:

$$\begin{cases} \underline{\Delta}^{(k)} \leftarrow \mathrm{sign}\{\, \max\{\, |\Delta^{(k)}|,\ T^p \,\},\ \Delta^{(k)} \,\}\,; \\ x^{(k+1)} \leftarrow x^{(k)} + \underline{\Delta}^{(k)}, \qquad (k = 0, 1, 2, \ldots). \end{cases} \tag{b}$$

2.2.3. Small-step Local Minimizer

For a given starting point $x^{(0)}$, it is desirable that the algorithm of a local minimizer converges to minimum x_* such that the sequence $\{x^{(k)}\}$ of points is always included in *the unimodal region* $R_u(x_*)$ [2]. That is, if the step of the algorithm that generates a new point $x^{(k+1)}$ is

$$x^{(k+1)} \leftarrow x^{(k)} + \delta^{(k)}, \qquad (k = 0, 1, 2, \ldots), \tag{c}$$

then the following condition holds:

$$\begin{cases} \{x^{(k)}\} \in R_u(x_*), \qquad (k = 0, 1, 2, \ldots), \\ \lim_{k\to\infty} \{x^{(k)}\} = x_*. \end{cases} \tag{7}$$

In step (c), if the step length $\delta^{(k)}$ is too large, then a new point, $x^{(k+1)}$, probably falls in a unimodal region different from $R_u(x_*)$. In order to avoid this difficulty and to guarantee bracketing of a local minimum by three points, $\delta^{(k)}$ must be less than or equal to half of the radius of the unimodal region [2]:

$$\delta^{(k)} \le \frac{1}{2} r(x_*^i), \tag{8}$$

where $r(x_*^i)$ is the radius of the unimodal region at x_*^i.

In general, the user sets an upper limit of step $\overline{\delta}$ such that $\overline{\delta} \le \frac{1}{2} r_{**}$, where $\frac{1}{2} r_{**}$ is the minimum of all radii of unimodal regions. By using this parameter $\overline{\delta}$, we substitute the following step for step (c):

$$\begin{cases} \overline{\delta}^{(k)} \leftarrow \min\{\, \delta^{(k)},\ \overline{\delta}\,\}\,; \\ x^{(k+1)} \leftarrow x^{(k)} + \overline{\delta}^{(k)}, \quad (k = 0, 1, 2, \ldots). \end{cases} \tag{d}$$

Thus, the specification of a small-step minimizer algorithm $MLuf$ that finds the local minimum x_*^i and its function value f_*^i of a function $f(x)$ in a domain $D^1 = [a_1, b_1]$ from a starting point x^i with an upper limit of step $\overline{\delta}$ and a tolerance ε is

$$(f_*^i,\ x_*^i) \leftarrow MLuf(\, f,\ D^1,\ x^i,\ \overline{\delta},\ \varepsilon\,).$$

The the algorithm for locally minimizing a univariate function is described in detail in the book by Brent [1].

2.3. Main Algorithm

We introduce the main algorithm that finds the global minimum x_{**} and its function value f_{**} of a function $f(x)$ in a searching region $D^1 = [a_1, b_1]$ for a given initial point x^1, an initial step size $\Delta^{(0)}$ ($> T^p$), an upper limit of step size $\overline{\delta}$, a period T^p and a tolerance ε.

The algorithm consists of the following steps.

$(f_{**},\ x_{**}) \leftarrow MGuf(\, f,\ D^1,\ x^1,\ \Delta^{(0)}, \overline{\delta}, T^p\ \varepsilon\,)$

MG1. (Initial Setting)

$x^2 \leftarrow x^1 + \Delta^{(0)}\,;\ \ k \leftarrow 0\,;$
$(f_*^1,\ x_*^1) \leftarrow MLuf(\, f,\ D^1,\ x^1,\ \overline{\delta},\ \varepsilon\,)\,;$
$(f_*^2,\ x_*^2) \leftarrow MLuf(\, f,\ D^1,\ x^2,\ \overline{\delta},\ \varepsilon\,)\,;$
if $f_*^1 > f_*^2$ **then** $\Delta^{(0)} \leftarrow \gamma\Delta^{(0)}\,;$
else $x_*^1 \leftrightarrow x_*^2\,;\ \ f_*^1 \leftrightarrow f_*^2\,;\ \ \Delta^{(0)} \leftarrow -\gamma\Delta^{(0)}\,;$ **fi** ;
$x^3 \leftarrow x^2 + \gamma\Delta^{(0)}\,; \qquad (\gamma > 1)$
$(f_*^3,\ x_*^3) \leftarrow MLuf(\, f,\ D^1,\ x^3,\ \overline{\delta},\ \varepsilon\,)\,;$

MG2. (Bracketing step in large-step minimizer)

repeat
$k \leftarrow k + 1\,;$
if $f_*^1 > f_*^2$
then $x_*^1 \leftarrow x_*^2\,;\ \ x_*^2 \leftarrow x_*^3\,;$
Choose step length $\Delta^{(k)} > T^p\,;$
$x^3 \leftarrow x_*^2 + \Delta^{(k)}\,;$
$(f_*^3,\ x_*^3) \leftarrow MLuf(\, f,\ D^1,\ x^3,\ \overline{\delta},\ \varepsilon\,)\,;$
else $x_*^3 \leftarrow x_*^2\,;\ \ x_*^2 \leftarrow x_*^1\,;$
Choose step length $\Delta^{(k)} > T^p\,;$
$x^1 \leftarrow x_*^1 - \Delta^{(k)}\,;$ **fi** ;
$(f_*^1,\ x_*^1) \leftarrow MLuf(\, f,\ D^1,\ x^1,\ \overline{\delta},\ \varepsilon\,)\,;$
until $f_*^1 > f_*^2 < f_*^3\,;$

MG3. (Reducing step in large-step minimizer)

Reduce the interval $[x_*^1,\ x_*^3]$ until the following conditions hold:

- $x_*^1 < x_*^2 < x_*^3$ and $f(x_*^1) > f(x_*^2) < f(x_*^3)$,
- the unimodal regions $R_u(x_*^1)$ and $R_u(x_*^2)$ are adjoined, and
- the unimodal regions $R_u(x_*^2)$ and $R_u(x_*^3)$ are adjoined.

MG4. (Last setting)

$x_{**} \leftarrow x_*^2\,;\ \ f_{**} \leftarrow f_*^2\,;$

2.4. A Numerical Experiment

In order to confirm the efficacy of the algorithm, we used the following problem (P1):

$$f(x) = 0.1x^2 + 0.01 - 0.01\cos(40\pi x), \qquad x \in [-1, 1].$$

The conditions of the experiment were:

- $T^p = 0.05,\ \overline{\delta} = 0.2T^p$.
- The *small-step local minimizer* was constructed by the method of Brent [1].
- The tolerance ε with respect to x was 10^{-5}.
- The initial point and step length were $x^{(0)} = -0.91$ and $\Delta^{(0)} = 4T^p$, respectively.

The algorithm found the global minimum of this function $f(x)$ with 68 function evaluations. A trace of the searching process of the algorithm is shown Figure 2.

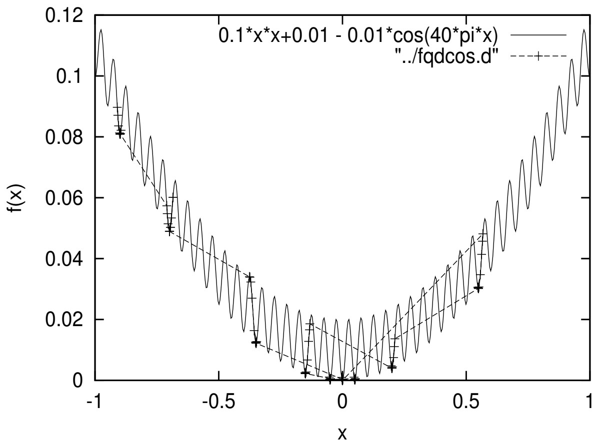

Figure 2: Trace of the searching process of the algorithm.

3. Algorithm for a Multivariate Function

The minimization problem (P) of a multivariate function $f(\boldsymbol{x})$ that is expressed by the sum of a unimodal function $u(\boldsymbol{x})$ and a periodic function $p(\boldsymbol{x})$ is as follows:

$$\begin{aligned} &\text{minimize } \ f(\boldsymbol{x}) \equiv u(\boldsymbol{x}) + p(\boldsymbol{x})\ , \\ &\qquad \boldsymbol{x} \in D^n \equiv \prod_{j=1,\ldots,n} [a_j, b_j]\ . \end{aligned} \tag{Q}$$

In order to solve this problem, a new point $\boldsymbol{x}^{(k+1)}$ is repeatedly generated using step length $\alpha^{(k)}$ that is determined by minimizing from the previous point $\boldsymbol{x}^{(k)}$ in the direction $\boldsymbol{d}^{(k)}$:

$$\begin{cases} \alpha^{(k)} = \underset{\alpha}{\operatorname{argmin}}\{\ \phi(\alpha) \equiv f(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)})\ \}, \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha^{(k)}, \quad (k = 0, 1, 2, \ldots), \end{cases} \tag{9}$$

where this minimization step is called *line search*.

Generally, line search procedures can only find a local minimum of the function $\phi(\alpha)$. Thus, even though the line search is repeatedly applied, the possibility of finding the global minimum of function f is low.

In order to overcome this problem, we modify the specification of algorithm $MGuf$ into the following specification so that the algorithm can find the global minimum $\boldsymbol{x}^{(k+1)}$ and its function value $f^{(k+1)}$ from a starting point $\boldsymbol{x}^{(k)} \in D^n$ in the direction $\boldsymbol{d}^{(k)}$:

$$(f^{(k+1)}, \boldsymbol{x}^{(k+1)}) \leftarrow MGufn\,(\,f, D^n, \boldsymbol{x}^{(k)}, \boldsymbol{d}^{(k)}, \Delta^{(0)}, \overline{\delta}, T^p, \varepsilon\,).$$

In the problem (Q), if $u(\boldsymbol{x})$ is a *strongly quasi-convex function*, then the function $\phi(\alpha) = u(\boldsymbol{x}^i + \alpha \boldsymbol{d}^i)$ becomes unimodal for all $\boldsymbol{x}^i \in D^n$ and $\boldsymbol{d}^i$. If conditions similar to **Conditions 1–3** hold, then the possibility of finding the global minimum of the function f will become high under certain conditions by iteratively applying the line search algorithm.

In particular, if functions $u(\boldsymbol{x})$ and $p(\boldsymbol{x})$ are separable with respect to each variable x_i, that is,

$$\begin{aligned} f(\boldsymbol{x}) \equiv f(x_1, x_2, \ldots, x_n) &= u(\boldsymbol{x}) + p(\boldsymbol{x}) \\ &= \sum_{i=1}^{n} \{u_i(x_i) + p_i(x_i)\}\ , \end{aligned} \tag{10}$$

then the global minimum of this function can be found by applying the line search procedure $MGufn$ along the coordinate vector $\boldsymbol{e}^i$ only n-times.

We show an algorithm for finding the global minimum $\boldsymbol{x}_{**}$ and its function value f_{**} from an initial point $\boldsymbol{x}^{(0)}$ for the above function f with periods $\boldsymbol{T}^p = (T_1^p, T_2^p, \ldots, T_n^p)$. The algorithm consists of the following steps.

$(f_{**}, \boldsymbol{x}_{**}) \leftarrow MGnf(\,f,\ D^n,\ \boldsymbol{x}^{(0)},\ \Delta^{(0)},\ \overline{\boldsymbol{\delta}}, \boldsymbol{T}^p,\ \varepsilon\,)$
for $i = 1$ **to** n
$\quad (f^{(k+1)}, \boldsymbol{x}^{(k+1)}) \leftarrow MGufn\,(\,f, D^n, \boldsymbol{x}^{(k)}, \boldsymbol{e}^i, \Delta^{(0)}, \overline{\delta}_i, T_i^p, \varepsilon\,)\,;$
$f_{**} \leftarrow f^{(k+1)}\,;\ \ \boldsymbol{x}_{**} \leftarrow \boldsymbol{x}^{(k+1)}\,;$

3.1. A Numerical Experiment

We used the following Griewank test function as a numerical experiment:

$$f(\boldsymbol{x}) = \sum_{i=1}^{2} \frac{x_i^2}{200} - \prod_{i=1}^{2} \cos(x_i/\sqrt{i}) + 1, \quad x \in [-100, 100].$$

This function has the global minimum at $\boldsymbol{x}_{**} = (0, 0)$ and has 500 further local minima.

The conditions of the experiment were as follows:

- The period of the i-th coordinate was $T_i^p = \sqrt{i}\pi,\ \overline{\delta}_i = 0.2T_i^p\ (i = 1, 2)$.
- The initial point was $\boldsymbol{x}^{(0)} = (-80, -80)$.

The algorithm found a minimum $\boldsymbol{x}^{(1)} = (0, 0)$ with 142 function evaluations by applying line search from $\boldsymbol{x}^{(0)}$ in the first coordinate direction $\boldsymbol{e}^1 = (1, 0)$ and the second coordinate direction $\boldsymbol{e}^2 = (0, 1)$. This result (142 function evaluations) is less than one tenth of the number of function evaluations required in the experiment by Snyman and Fatti (1496 function evaluations)[4].

4. Conclusions

We have mainly studied an algorithm for finding the global minimum of a univariate function $f(x)$ with many local minima whose lower envelope is unimodal. This algorithm is characterized by the use of a large-step minimizer and a small-step local minimizer. Numerical results showed that the algorithm efficiently finds the global minimum with only a few function evaluations.

Moreover, we considered a multivariate function $f(\boldsymbol{x})$ similar to the above unimodal function and showed an algorithm for minimizing a certain class of the function. The results of a numerical example showed that the algorithm effectively finds the global minimum.

References

[1] Brent, R. P. : *Algorithm for Minimization without Derivatives,* Prentice-Hall:New York 1973.

[2] Kanemitsu, H. and Shimbo, M.: "Two Methods for Finding Local and Global Maxima of a Multimodal Function of One Variable Based on the Concept of a Unimodal Region," *IPSJ*, 37: 1646–1656, 1996 (In Japanese).

[3] Rinnooy Kan, A.H.G. and Timmer, G.T.: "Stochastic Global Optimization Methods. Part I: Clustering Methods," *Mathematical Programming*, 39: 27–56, 1987.

[4] Törn, A.A. and Žilinskas, A.A.: "*Global Optimization*," Springer-Verlag, Lecture Notes in Computer Science 350, Berlin, 1989.