A Level-Value Estimation Algorithm and Its Stochastic Implementation for Global Optimization

Zheng Peng * Donghua Wu † Quan Zheng ‡

Abstract: In this paper, we propose a new method for finding global optimum of continuous optimization problems, namely Level-Value Estimation algorithm (LVEM). First we define the variance function $v(c)$ and the mean deviation function $m(c)$ with respect to a single variable (the level value $c$), and both of these functions depend on the optimized function $f(x)$. We verify these functions have some good properties for solving the equation $v(c) = 0$ by Newton method. We prove that the largest root of this equation is equivalent to the global optimum of the corresponding optimization problem. Then we proposed LVEM algorithm based on using Newton method to solve the equation $v(c) = 0$, and prove convergence of LVEM algorithm. We also propose an implementable algorithm of LVEM algorithm, abbreviate to ILVEM algorithm. In ILVEM algorithm, we use importance sampling to calculate integral in the functions $v(c)$ and $m(c)$. And we use the main ideas of the cross-entropy method to update parameters of probability density function of sample distribution at each iteration. We verify that ILVEM algorithm satisfies the convergent conditions of (one-dimensional) inexact Newton method for solving nonlinear equation, and then we prove convergence of ILEVM algorithm. The numerical results suggest that ILVEM algorithm is applicable and efficient in solving global optimization problem.

Keywords: Global Optimization; Level-Value Estimation Algorithm; Variance Equation; Inexact Newton Method; Importance Sampling.

*Department of mathematics, NanJing University, PR. China 210093, and Department of mathematics, Shanghai University, PR. China 200444; Corresponding author, Email:pzheng008@yahoo.com
†Department of mathematics, Shanghai University, PR. China 200444
‡Department of mathematics, Shanghai University, PR. China 200444
1 Introduction

In this paper, we consider the following global optimization problem:

\[ c^* = \min_{x \in D} f(x) \]  

where \( D \subseteq \mathbb{R}^n \). When \( D = \mathbb{R}^n \), we call Problem (1.1) unconstrained global optimization. When \( D = \{ x \in \mathbb{R}^n : g_i(x) \leq 0, i = 1, 2, ..., r. \} \neq \emptyset \), we call Problem (1.1) constrained one. We always assume \( D \) is a compact subset in \( \mathbb{R}^n \).

The following notations will be used in this paper.

- \( \nu(\Omega) \): the Lebesgue measure of the set \( \Omega, \ \Omega \subset \mathbb{R}^n \).
- \( c^* \): the global optimal value, i.e., \( c^* = \min_{x \in D} f(x) \).
- \( c_p^* \): the global optimal value of penalty function \( F(x, a) \), i.e., \( c_p^* = \min_{x \in \mathbb{R}^n} F(x, a) \).
- \( H_c \): the global optimal solutions set, i.e., \( H_c = \{ x \in \mathbb{R}^n : f(x) \leq c \} \).
- \( H_c^p \): the global optimal solutions set of penalty function \( F(x, a) \), i.e., \( H_c^p = \{ x \in \mathbb{R}^n : F(x, a) \leq c_p^* \} \).
- \( \hat{c}^* \): the \( \varepsilon \)-optimal value.
- \( \hat{H}^*_{\varepsilon} \): the \( \varepsilon \)-optimal solutions set.

Throughout this paper, we assume that

1) \( D \) is a robust set (for this definition, one can refer to [3]), i.e., \( \text{cl} (\text{int} D) = \text{cl} (D) \), and moreover, when \( c > c^* \), \( 0 < \nu (H_c \cap D) \leq U < +\infty \).

2) \( f \) is lower Lipschitzian, and \( f \) and \( g_i \) (\( i = 1, 2, ..., r \)) are finite integrable on \( \mathbb{R}^n \).

Global optimization problem (1.1) arises in a wide range of applications, such as finance, allocation and location, operations research, structural optimization, engineering design and control, etc. This problem attracts a number of specialists in various fields, and they provided various methods to solve it. These methods can be classified into two classes: deterministic methods and stochastic methods. The deterministic method exploits analytical properties of Problem (1.1) to generate a deterministic sequence of points which converges to a global optimal solution. These properties generally include differentiability, convexity and monotonicity. Filled function method and tunnelling function method [1] are classical deterministic methods. The stochastic method generates a sequence of points which converges to a global optimal solution in probability [2, 24]. For examples, genetic algorithm, tabu search and simulated annealing algorithm, etc. Combining the deterministic method and the stochastic method, we are possible to get a method with better performance: We possibly overcomes some drawbacks of deterministic method (e.g., we have to use analytical properties of problem but we know them little, or we only get a local solution and we can not confirm whether is it a global solution.). And we possibly overcomes some drawbacks of some stochastic methods (e.g., some stochastic methods do not converge to the global optimization with probability one [12]).

A representative of this combinatorial methods is the integral level-set method.
(ILSM), which was first proposed by Chew and Quan [3]. In the integral level-set method, the authors construct the following two sequences:

\[ c_{k+1} = \frac{1}{\nu(H_{c_k})} \int_{H_{c_k}} f(x) d\nu \]  

\[ H_{c_k} = \{ x \in D : f(x) \leq c_k \} \]

(1.2)

(1.3)

where \( \nu \) is Lebesgue measure. The stopping criterion of ILSM method is:

\[ v(c_k) = \frac{1}{\nu(H_{c_k})} \int_{H_{c_k}} (c_k - f(x))^2 d\nu = 0. \]  

(1.4)

Under some suitable conditions, the sequences \( \{c_k\} \) and \( \{H_{c_k}\} \) converge to optimal value and optimal solutions set, respectively. However ILSM method has a drawback that the level-set \( \{H_{c_k}\} \) in (1.3) is difficult to be determined. So in the implementable algorithm of ILSM method, level-set \( H_{c_k} \) is approximately determined by using Monte-Carlo method. But to do in this way leads to another drawback: Convergence of the implementable algorithm can not be proved until now. Wu and Tian [4, 5] gave some modifications of the implementable ILSM method.

Phu and Hoffmann [17] gave a method for finding the essential supremum of summable function. They also proposed a accelerate algorithm in [20] and they implemented their conceptual algorithm by using Monte-Carlo method. Convergence of the implementable algorithm of Phu and Hoffmanns’ method has not been proved too. Wu and Yu [18] used the main ideas of Phu and Hoffmanns’ method to solve global optimization problem with box-constraints. Peng and Wu [19] extended the method of [18] to solve global optimization problem with general functional constrains. In [19], the authors used the uniform distribution in number theory to implement the conceptual algorithm and prove convergence of the implementable algorithm. But as well known, it is low efficient by using the uniform distribution in number theory in calculating integral, though it is polynomial-time with respect to the dimensions of the integrand function.

In this paper, we define the variance function \( v(c) \) [see (2.1)] and the mean deviation function \( m(c) \) [see (2.2)]. We verify these functions have some good properties for global optimization term, such as convexity, monotonicity and differentiability. Based on solving variance equation \( v(c) = 0 \) by Newton method, we propose the level-value estimation method (LVEM) for solving the global optimization problem (1.1).

To implement LVEM algorithm, we calculate integral of \( v(c) \) and \( m(c) \) by using Monte-Carlo method with importance sampling. We introduce importance sampling [6, 7], and the updating mechanism of sample distribution based on the main idea of the cross-entropy method [8, 9, 10]. Then we propose implementable ILVEM algorithm. By verifying ILVEM algorithm satisfies the convergent conditions of inexact Newton method [11] (We modify it to one-dimensional inexact Newton method), we get convergence of ILVEM algorithm.

3
We use discontinuous exact penalty function [4] to invert the problem with functional constraints into unconstrained one, and prove they are equivalent to each other.

This paper is organized as follows. We define the variance function and the mean deviation function, and study their properties in section 2. In section 3, based on solving variance equation \( v(c) = 0 \) by Newton method, we propose a conceptual LVEM algorithm. And we prove convergence of LVEM algorithm. In section 4, we introduce importance sampling, and the updating mechanism of sample distribution based on the main idea of the cross-entropy method. In section 5, we propose implementable ILVAM algorithm and prove its convergence. Some numerical results are given in section 6. Finally in section 7, we give our conclusions.

2 The variance function and the mean deviation function

We always assume \( H_c \cap D \neq \emptyset \) when \( c \geq c^* \). Moreover, when \( c > c^* \), we assume \( \{ x \in D : f(x) < c \} \neq \emptyset \) and \( \nu(H_c \cap D) > 0 \). By assumption R), we know that \( \nu(H_c \cap D) \leq U < +\infty \).

We define the variance function as follows:

\[
v(c) = \int_{H_c \cap D} (c - f(x))^2 d\nu, \tag{2.1}\]

and define the mean deviation function as follows:

\[
m(c) = \int_{H_c \cap D} (c - f(x)) d\nu, \tag{2.2}\]

where the integration is respect to \( x \) on the region \( H_c \cap D \).

**Theorem 2.1** For all \( c \in (-\infty, +\infty) \), the mean deviation function \( m(c) \) is Lipschitz-continuous, non-negative, monotonous non-decreasing, convex and almost everywhere has the derivative

\[
m'(c) = \nu(H_c \cap D). \]

**Proof:** It is similar to Theorem 2.1 in [17] or [20], and we omit it here.

**Corollary 2.1** The mean deviation function \( m(c) \) is strictly increasing in \((c^*, +\infty)\).

**Proof:** By direct computing, we can get this assertion.

**Theorem 2.2** The variance function \( v(c) \) is Lipschitz-continuously differentiable on \((-\infty, +\infty)\), and

\[
v'(c) = 2m(c). \tag{2.3}\]

**Proof:** First we assume \( \Delta c > 0 \) in calculating \( v'(c) \) by using its definition. When \( \Delta c < 0 \), we can get the same assertion in the same way.
Note when $\Delta c > 0$, we have $H_c \subset H_{c+\Delta c}$. And $f$ is finite integral, and

\[ H_{c+\Delta c} \cap D = (H_{c+\Delta c} \setminus H_c \cup H_e) \cap D = (H_{c+\Delta c} \setminus H_c \cap D) \cup (H_c \cap D). \]

Hence we have

\[
v'(c) = \lim_{\Delta c \to 0} \frac{1}{\Delta c} [v(c + \Delta c) - v(c)]
= \lim_{\Delta c \to 0} \frac{1}{\Delta c} \left[ \int_{H_{c+\Delta c} \cap D} (c + \Delta c - f(x))^2 d\nu - \int_{H_c \cap D} (c - f(x))^2 d\nu \right]
= \lim_{\Delta c \to 0} \frac{1}{\Delta c} \left[ \int_{(H_{c+\Delta c} \setminus H_c) \cap D} (c + \Delta c - f(x))^2 d\nu 
+ \int_{H_c \cap D} (c + \Delta c - f(x))^2 d\nu - \int_{H_c \cap D} (c - f(x))^2 d\nu \right]
= \lim_{\Delta c \to 0} \frac{1}{\Delta c} \left[ \int_{(H_{c+\Delta c} \setminus H_c) \cap D} (c + \Delta c - f(x))^2 d\nu 
+ \lim_{\Delta c \to 0} \frac{1}{\Delta c} \int_{H_c \cap D} \Delta c(2c + \Delta c - 2f(x)) d\nu. \right]
\]

When $x \in (H_{c+\Delta c} \setminus H_c) \cap D$ we have $c \leq f(x) \leq c + \Delta c$, which implies $0 \leq (c + \Delta c - f(x))^2 \leq \Delta c^2$. Thus

\[
0 \leq \lim_{\Delta c \to 0} \frac{1}{\Delta c} \int_{(H_{c+\Delta c} \setminus H_c) \cap D} (c + \Delta c - f(x))^2 d\nu
\leq \lim_{\Delta c \to 0} \frac{1}{\Delta c} \int_{(H_{c+\Delta c} \setminus H_c) \cap D} (\Delta c)^2 d\nu
= \lim_{\Delta c \to 0} \Delta c \nu(H_{c+\Delta c} \setminus H_c) \cap D) = 0,
\]

and

\[
\lim_{\Delta c \to 0} \frac{1}{\Delta c} \int_{H_c \cap D} \Delta c(2c + \Delta c - 2f(x)) d\nu
= \lim_{\Delta c \to 0} \int_{H_c \cap D} (2c + \Delta c - 2f(x)) d\nu
= \int_{H_c \cap D} \left[ \lim_{\Delta c \to 0} (2c + \Delta c - 2f(x)) \right] d\nu
= \int_{H_c \cap D} (2c - 2f(x)) d\nu = 2m(c).
\]

In the third line, since $2c + \Delta c - 2f(x) > 0$ when $x \in H_c \cap D$, we can exchange integral and limit operators. Therefore we have $v'(c) = 2m(c)$.

On the other hand, by Theorem 2.1 we know that $m(c)$ is Lipschitz-continuous on $(-\infty, +\infty)$. Thus $v'(c) = 2m(c)$ is Lipschitz-continuous on $(-\infty, +\infty)$. This completes the proof of Theorem 2.2.
Theorem 2.3. The variance function $v(c)$ has the following properties:

1) When $c \in (-\infty, +\infty)$, it is monotonic non-decreasing; and when $c \in (c^*, +\infty)$, it is strictly increasing.

2) It is a convex function on $(-\infty, +\infty)$ and strictly convex in $(c^*, +\infty)$.

3) $v(c) = 0$ if and only if $c \leq c^*$.

Proof: The assertions 1), 2) follow from directly Theorem 2.2 and the monotonicity of $m(c)$. The proof of assertion 3) follows from Theorem 2.9 in [3]. Since $H_c \cap D = \emptyset$ and $\nu(\emptyset) = 0$ while $c < c^*$, the slight generalization for $c < c^*$ is trivial by the definition of $v(c)$.

Corollary 2.3 $m(c) = 0$ if and only if $c \leq c^*$.

We already assumed that $D = \{x \in R^n : g_i(x) \leq 0, i = 1, 2, ..., r\} \neq \emptyset$ is a compact subset of $R^n$. We then define discontinuous exact penalty function as follows:

$$F(x, a) = f(x) + ap(x), a > 1.0$$

where

$$p(x) = \begin{cases} 
0, & x \in D \\
\vartheta + d(x), & x \notin D 
\end{cases}, \vartheta > 0$$

$$d(x) = \sum_{i=1}^{r} \max(0, g_i(x)).$$

And we set $H_c^p = \{x | x \in R^n : F(x, a) \leq c\}$.

Clearly, by assumption 1), when $1 < a < B_1 < +\infty, 0 < \vartheta < B_2 < +\infty, F(x, a)$ is finite integral.

Furthermore, let

$$v_p(c) = \int_{H_c^p} (c - F(x, a))^2 d\nu,$$  \hspace{1cm} (2.6)

$$m_p(c) = \int_{H_c^p} (c - F(x, a)) d\nu.$$  \hspace{1cm} (2.7)

Then we have the following theorem:

**Theorem 2.4** When $a > 0$ is big enough, we have

$$v_p(c) = v(c).$$  \hspace{1cm} (2.8)

Proof: Note that $d(x) \geq 0$ and for any $a > 0$, $ap(x) \geq 0$. Hence $F(x, a) \leq c$ implies $f(x) \leq c$. Thus we get $H_c^p \supseteq H_c \cap D = \{x \in D : f(x) \leq c\}$. 
For $\forall \varepsilon > 0$, we have

\[
|v_p(c) - v(c)| = \left| \int_{H^p_c} (c - F(x, a))^2 d\nu - \int_{(H_c \cap D)} (c - f(x))^2 d\nu \right|
\]

\[
= \left| \int_{H^p_c \setminus (H_c \cap D)} (c - F(x, a))^2 d\nu + \int_{H_c \cap D} ((c - F(x, a))^2 - (c - f(x))^2) d\nu \right|
\]

\[
\leq \left| \int_{H^p_c \setminus (H_c \cap D)} (c - F(x, a))^2 d\nu \right| + \int_{H_c \cap D} (-ap(x))(2c - 2f(x) - ap(x)) d\nu
\]

\[
= A_1 + A_2.
\]

When $x \in H_c \cap D$, by using (2.5) we get $p(x) = 0$, which implies $A_2 = 0$.

On the other hand, note that if $\forall x \in H^p_c \setminus (H_c \cap D)$, then $x \in H^p_c$. It follows that if $x \in H_c$, then we know $x \notin D$. Thus we get $c - F(x, a) \geq 0$ and $p(x) = \vartheta + d(x) > 0$ for $d(x) \geq 0$, $\vartheta > 0$. For any fixed $c$, when $\frac{c-f(x)-\varepsilon}{p(x)} < a \leq \frac{c-f(x)}{p(x)}$, we have $|c - F(x, a)| = |c - f(x) - ap(x)| < \varepsilon$. Then we have

\[
A_1 = \left| \int_{H^p_c \setminus (H_c \cap D)} (c - F(x, a))^2 d\nu \right|
\]

\[
< \left| \int_{H^p_c \setminus (H_c \cap D)} \varepsilon^2 d\nu \right|
\]

\[
= \nu(H^p_c \setminus (H_c \cap D)) \varepsilon^2
\]

Therefore $A_1 \to 0$ when $\varepsilon \to 0$. But when $a > \frac{c-f(x)}{p(x)}$, $H^p_c$ is empty which implies that $\nu(H^p_c \setminus (H_c \cap D)) = 0$, and it follows $A_1 = 0$. Thus we have $A_1 = 0$.

Thus by summarizing, for a given $x$, when $a > 0$ is big enough we have $v_p(c) = v(c)$.

Similarly, we have

**Corollary 2.5** When $a > 0$ is big enough, we get

\[
m_p(c) = m(c)
\]

\[
v'_p(c) = v'(c)
\]

The following corollary follows from Theorem 2.4 and Corollary 2.5 directly.

**Corollary 2.6** When $a > 0$ is large enough, the functions $m_p(c)$ and $v_p(c)$ are convex, monotonic and differentiable while $c \in (-\infty, +\infty)$, which are as same as the functions $m(c)$ and $v(c)$. 

7
3 Algorithm LVEM and convergence

Following from the discussed properties of $v(c)$ and $m(c)$ in the previous section, we know that $c^* = \min_{x \in D} f(x)$ is the largest root of variance equation $v_p(c) = 0$, and we can solve this equation by Newton method. Based on this, we propose the Level Value Estimation Method for solving global optimization problem (1.1) conceptually.

Algorithm 3.1: LVEM Algorithm

**step 1.** Let $\varepsilon > 0$ be a small number. Give a point $x_0 \in \mathbb{R}^n$, calculate $c_0 = F(x_0, a)$ and set $k := 0$.

**step 2.** Calculate $v_p(c_k)$ and $m_p(c_k)$ which are defined in the following:

$$v_p(c_k) = \int_{H^p_{c_k}} (c_k - F(x, a))^2 d\nu,$$

$$m_p(c_k) = \int_{H^p_{c_k}} (c_k - F(x, a)) d\nu,$$

where

$$H^p_{c_k} = \{x \in \mathbb{R}^n : F(x, a) \leq c_k\}.$$  

Let

$$\lambda_k = \frac{v_p(c_k)}{2m_p(c_k)}.$$  

**step 3.** If $\lambda_k < \varepsilon$, goto next step; else let

$$c_{k+1} = c_k - \lambda_k,$$

and let $k := k + 1$, return to step 2.

**step 4.** Let $c^*_p = c_k$ and $H^*_p = H^p_{c_k} = \{x \in \mathbb{R}^n : F(x, a) \leq c_k\}$, where $c^*_p$ is the approximal global optimal value, and $H^*_p$ is the set of approximal global optimal solutions.

LVEM algorithm is based on constrained problem, and it is clearly applicable to unconstrained problem.

On convergence of LVEM method, we have

**Theorem 3.1** The sequences $\{c_k\}$ generated by LVEM method is monotonically decreasing. Set $c^*_p = \min_{x \in \mathbb{R}^n} F(x, a)$ and $H^*_p = \{x \in \mathbb{R}^n : F(x, a) = c^*_p\}$. Then

$$\lim_{k \to \infty} c_k = c^*_p,$$

and

$$\lim_{k \to \infty} H^p_{c_k} = H^*_p.$$
Proof: By the properties of $v_p(c)$, we know that $c_p^*$ is the largest root of the variance equation $v_p(c) = 0$. To solve this equation by Newton method, we have

$$c_{k+1} = c_k - \frac{v_p(c_k)}{v'_p(c_k)}$$

(3.8)

substitute $v'_p(c_k) = 2m_p(c_k)$ into (3.8), we get the iterative formula (3.5). By (3.4) we know that $\lambda_k \geq 0$, and then the sequence $\{c_k\}$ generated by LVEM method is monotonically decreasing. By convergence of Newton method, when $k \to \infty$, $c_k$ converges to the root of $v_p(c) = 0$.

On the other hand, by Theorem 2.3 and Theorem 2.4, for all $c$ ($c \leq c_p^*$) is the root of equation $v_p(c) = 0$. Thus we only need to prove $c_k \geq c_p^*$, $k = 0, 1, 2, \ldots$. We can prove this assertion by mathematical induction.

It is clear that $c_0 = F(x_0, a) \geq c_p^*$. If $c_0 = c_p^*$, then $v_p(c_0) = 0$, and the process stops at step 3. Else if $c_0 > c_p^*$, then $v_p(c_0) > 0$ and the process goes on. Assume $c_k \geq c_p^*$, we have two cases: If $c_k = c_p^*$, then $v_p(c_k) = 0$, and the process stops; Else if $c_k > c_p^*$, since $v_p(c)$ is convex in $(-\infty, +\infty)$ and $c_p^*$, $c_k \in (-\infty, +\infty)$, then by Theorem 3.3.3 in page 89 in [16], we have

$$v_p(c_p^*) - v_p(c_k) \geq v'_p(c_k)(c_p^* - c_k).$$

(3.9)

since $v_p(c_p^*) = 0$ and $v'_p(c_k) = 2m_p(c_k) > 0$ while $c_k > c_p^*$, the inequality (3.9) is equivalent to

$$-v_p(c_k) \geq v'_p(c_k)(c_p^* - c_k).$$

That is to say

$$c_p^* - c_k \leq -\frac{v_p(c_k)}{v'_p(c_k)} = -\lambda_k.$$  

By this inequality and the iterative formula (3.5), we have

$$c_{k+1} = c_k - \lambda_k \geq c_p^*.$$  

By mathematical induction, we get $c_k \geq c_p^*$, $k = 0, 1, 2, \ldots$.

Thus we can conclude that:

$$\lim_{k \to \infty} c_k = c_p^*.$$  

To complete the proof, by the definition of $H^p_{c_k}$ and the Property 2 in [4], when $c_k \to c_p^*$ we have

$$\lim_{k \to \infty} H^p_{c_k} = H^p_{c_p^*}.$$  

Theorem 3.2 Set $c^* = \min_{x \in D} f(x)$, $H^* = \{x \in D : f(x) = c^*\}$ and $c_p^* = \min_{x \in R^n} F(x, a)$, $H^*_p = \{x \in R^n : F(x, a) = c_p^*\}$ respectively. When $a$ is big enough, we have

$$c_p^* = c^*, \quad H^*_p = H^*.$$  

(3.10)
Proof: By the definition of discontinuous exact penalty function (2.4), for $\vartheta > 0$ and $a > 0$ big enough, $F(x, a)$ attains its minimum over $\mathbb{R}^n$ only if $p(x) = 0$. Thus

$$
c^*_p = \min_{x \in \mathbb{R}^n} F(x, a)
= \min_{x \in \mathbb{R}^n} f(x) + ap(x)
= \min_{x \in D} f(x)
= c^*
$$

and the third equation holds since $p(x) = 0$ if and only if $x \in D$.

Next we prove $H^*_p = H^*$. For $\forall x \in H^*_p$, we have $F(x, a) = c^*_p = c^*$. Since $F(x, a)$ attains its minimum only if $p(x) = 0$, but $p(x) = 0$ if and only if $x \in D$, thus we have $x \in D$. Therefore, we have $F(x, a) = f(x) + ap(x) = f(x) = c^*$ and $x \in D$, i.e., $x \in H^*$. This follows $H^*_p \subseteq H^*$. Inversely, if $x \in H^*$, then we have $x \in D$ and $f(x) = c^* = c^*_p$. Thus $p(x) = 0$, and $F(x, a) = f(x) + ap(x) = f(x) = c^* = c^*_p$, which follows that $x \in H^*_p$. This implies $H^* \subseteq H^*_p$. By summarizing, we have $H^*_p = H^*$.

4 Importance sampling and updating mechanism of sampling distribution

To implement LVEM algorithm, we need to calculate the integrals of $v_p(c_k)$, $m_p(c_k)$ in (3.1) and (3.2). We do this by using Monte-Carlo method based on importance sampling [6]. And we use the main ideas of the cross-entropy method to update parameters of probability density function of sample distribution. This idea is different from the methods in literature [3, 4, 5]. So in this section we shall introduce importance sampling and updating mechanism of sampling distributions based on the main ideas of the cross-entropy method.

In this section, we assume that $\Psi(x)$ is an arbitrary integrable function with respect to measure $\nu$, where $\Psi$ is the integrand function in the formula (4.2). To calculate integral by using Monte Carlo method, we use importance sampling with probability density function (pdf) $g(x) = g(x; v)$, where $x = (x_1, ..., x_n)$ and $v$ is parameter (or parameter vector). For example, for given random samples $X_1, X_2, ..., X_N$ independent and identically distributed (iid) from distribution with density $g(x)$ on the region $\mathcal{X}$, we use

$$
\hat{I}_g = \frac{1}{N} \sum_{i=1}^{N} \frac{\Psi(X_i)}{g(X_i)}
$$

as an estimator of the integral:

$$
I = \int_{\mathcal{X}} \Psi(x) d\nu = \int_{\mathcal{X}} \frac{\Psi(x)}{g(x)} g(x) d\nu.
$$
It is easy to show this estimator is unbiased. Furthermore the variance of this estimator is:

$$\text{var}(\hat{I}_g) = \frac{1}{N} \int_X \left( \frac{\Psi(x)}{g(x)} - E_g \left[ \frac{\Psi(X)}{g(X)} \right] \right)^2 g(x) d\nu. \quad (4.3)$$

Moreover, if $\left[ \frac{\Psi(x)}{g(x)} \right]^2$ has a finite expectation with respect to $g(x)$, then the variance (4.3) can be also estimated from the sample $(X_1, X_2, ..., X_N)$ by the following formula:

$$\text{var}(\hat{I}_g) = \frac{1}{N^2} \sum_{j=1}^N \left( \frac{\Psi(X_j)}{g(X_j)} - \hat{I}_g \right)^2 = O\left( \frac{1}{N} \right), \text{ when } N \to +\infty. \quad (4.4)$$

See, for example, p83-84 and p92 Definition 3.9 in [26].

We get two useful assertions as follows. See, for example, Theorem 3.12 in [26], p94-95.

**Assertion 4.1:** If the expectation w.r.t $g(x)$

$$E_g \left[ \left( \frac{\Psi(X)}{g(X)} \right)^2 \right] = \int_X \frac{\Psi^2(x)}{g(x)} d\nu < +\infty,$$

then (4.1) converges almost surely to (4.2), that is

$$\Pr \left\{ \lim_{N \to +\infty} \frac{1}{N} \sum_{i=1}^N \frac{\Psi(X_i)}{g(X_i)} = \int_X \frac{\Psi(x)}{g(x)} g(x) d\nu \right\} = 1. \quad (4.5)$$

**Remark 4.1.** We have many approaches to get the estimation of sample size $N$ in Monte Carlo integration, see [29]. We relax (4.5) into the following form:

$$\Pr \left\{ \left| \frac{1}{N} \sum_{i=1}^N \frac{\Psi(X_i)}{g(X_i)} - \int_X \frac{\Psi(x)}{g(x)} g(x) d\nu \right| < \epsilon \right\} \geq 1 - \delta. \quad (4.6)$$

In the one-dimensional case (when $x \in \mathbb{R}$), by using the large deviation rate function, the simplest estimation of sample size $N$ is $N > -\frac{\ln \delta}{2\epsilon^2}$. In the worst case, since we always get a sample $X$ in $\mathbb{R}^n$ in component-independence style, a coarse estimator of sample size $N$ which satisfies (4.6) (when $x \in \mathbb{R}^n$) is

$$N > \left( -\frac{\ln \delta}{2\epsilon^2} \right)^n. \quad (4.7)$$

Take limit with respect to $\epsilon \to 0$ and $\delta \to 0$, we get (4.5) from (4.6) directly.

**Assertion 4.2:** The choice of $g$ minimizes the variance of the estimator (4.1) is as the following:

$$g^*(x) = \frac{|\Psi(x)|}{\int_X |\Psi(x)| d\nu}.$$
But in this case, $g^*(x)$ is respect to $\int_X |\Psi(x)|d\nu$. So in practice, we preassign a family pdf $g(x; v)$, and choose proper parameter (or parameter vector) $v$ to make $g(x; v)$ approximate to $g^*(x)$. The cross-entropy method gives an idea for choosing this parameter (or parameter vector) $v$.

The cross-entropy method was developed firstly by Rubinstein [10] to solve continuous multi-extremal and various discrete combinatorial optimization problems. The method is an iterative stochastic procedure making use of the importance sampling technique. The stochastic mechanism changes from iteration to iteration according to minimizing the Kullback-Leibler cross-entropy approach. Recently, under the assumption that the problem under consideration has only one global optimal point on its region, convergence of CE method is given in [30].

Let $g(x; v)$ be the n-dimensional coordinate-normal density which is denoted by $N(x; \mu, \sigma^2)$ and is defined as the following:

$$N(x; \mu, \sigma^2) = \frac{1}{(\sqrt{2\pi})^n \prod_{i=1}^{n} \sigma_i} \exp(-\frac{\sum_{i=1}^{n} (x_i - \mu_i)^2}{2\sigma_i^2}), \quad (4.8)$$

with component-independent mean vector $\mu = (\mu_1, \mu_2, ..., \mu_n)$ and variance vector $\sigma = (\sigma_1, \sigma_2, ..., \sigma_n)$.

Then based on the main ideas of the cross-entropy method, for estimating the integral (4.2) by using the estimator (4.1), the updating mechanism of sampling distribution can be described as follows:

**Updating algorithm of sampling distribution with pdf $N(x; \mu, \sigma^2)$**

(smoothing version [9] with a slight modification)

For given $T$ samples $\{X_1, X_2, ..., X_T\}$ generated iid from distribution with pdf $N(x; \mu, \sigma^2)$, we get the new pdf $N(x; \hat{\mu}, \hat{\sigma}^2)$ by using the following procedure:

**Step 1.** Compute $\Psi(X_i), i = 1, 2, ..., T$, select the $T_{\text{elite}}$ best samples (corresponding the $T_{\text{elite}}$ least value $\Psi(X_i)$), construct the elite samples set, denote it by $\{X^e_1, X^e_2, ..., X^e_{T_{\text{elite}}}\}$

**Step 2.** Update the parameters $(\hat{\mu}, \hat{\sigma}^2)$ by using the elite samples set (4.9), by the following formulas:

$$\hat{\mu}_j := \frac{1}{T_{\text{elite}}} \sum_{i=1}^{T_{\text{elite}}} X_{ij}, j = 1, 2, ..., n, \quad (4.10)$$

and

$$\hat{\sigma}^2_j := \frac{1}{T_{\text{elite}}} \sum_{i=1}^{T_{\text{elite}}} (X_{ij} - \hat{\mu}_j)^2, j = 1, 2, ..., n. \quad (4.11)$$

**Step 3.** Smooth:

$$\hat{\mu} = \alpha \hat{\mu} + (1 - \alpha)\mu, \quad \hat{\sigma}^2_j = \beta \hat{\sigma}^2_j + (1 - \beta)\sigma_j^2, \quad j = 1, 2, ..., n. \quad (4.12)$$

where $0.5 < \alpha < 0.9, \quad 0.8 < \beta < 0.99, \quad 5 \leq q \leq 10.$
5 Implementable algorithm ILVEM and convergence

We first introduce the following function:

\[
\psi(x, c) = \begin{cases} 
  c - F(x, a), & x \in H^c_p \\
  0, & x \notin H^c_p,
\end{cases}
\]  

where \( H^c_p = \{ x \in \mathbb{R}^n : F(x, a) \leq c \} \). Then it is clear that

\[
\psi^2(x, c) = \begin{cases} 
  (c - F(x, a))^2, & x \in H^c_p, \\
  0, & x \notin H^c_p.
\end{cases}
\]  

Then we rewrite \( v_p(c) \) and \( m_p(c) \) by the following formulas:

\[
v_p(c) = \int_{H^c_p} \psi^2(x, c) d\nu = \int_{\mathbb{R}^n} \frac{\psi^2(x, c)}{g(x)} dG(x) = E_g \left[ \frac{\psi^2(X, c)}{g(X)} \right],
\]

\[
m_p(c) = \int_{H^c_p} \psi(x, c) d\nu = \int_{\mathbb{R}^n} \frac{\psi(x, c)}{g(x)} dG(x) = E_g \left[ \frac{\psi(X, c)}{g(X)} \right],
\]

where \( X \) is a stochastic vector in \( \mathbb{R}^n \) iid from distribution with density function \( g(x) \), \( dG(x) = g(x) d\nu \).

We use

\[
\hat{v}_p(c) = \frac{1}{N} \sum_{i=1}^{N} \frac{\psi^2(X_i, c)}{g(X_i)},
\]

and

\[
\hat{m}_p(c) = \frac{1}{N} \sum_{i=1}^{N} \frac{\psi(X_i, c)}{g(X_i)},
\]

as the unbiased estimators of \( v_p(c) \) and \( m_p(c) \), respectively.

We choose the density function family \( g(x) = N(x, \mu, \sigma^2) \), which is the n-dimensional coordinate-normal density defined by (4.8). And we update its parameter vectors \( \mu \) and \( \sigma \) (from \( \mu_k, \sigma_k \) to \( \mu_{k+1}, \sigma_{k+1} \)) at the \( k \)-th iteration by using the updating mechanism which is introduced in the previous section, such that the density function \( g(x) \) is as similar as possible to the integrand function \( \psi^2(x, c) \) (and \( \psi(x, c) \)) in its shape.

We propose the implementable algorithm of LVEM in follows:

**Algorithm 5.1. ILVEM algorithm**

Step 1. Let \( \varepsilon > 0 \) be a small number. Let \( \varepsilon_0 > 0 \) and \( \delta > 0 \) be also small numbers. Given a point \( x_0 \in \mathbb{R}^n \) and calculate \( c_0 = F(x_0, a) \), choose the initiative parameter vectors \( \hat{\mu}_0 = (\hat{\mu}_{01}, ..., \hat{\mu}_{0n}) \) and \( \hat{\sigma}_{0}^2 = (\hat{\sigma}_{01}^2, ..., \hat{\sigma}_{0n}^2) \), and let \( 0 < \rho < 0.1 \). Let \( k := 0 \).
Step 2. Generate \( N = \left( \left[ \frac{-\ln \delta}{2\epsilon_k^2} \right] + 1 \right) \) (where \( \epsilon_k > 0 \), we shall discuss this parameter in Remark 5.3 in the later) samples \( X^k = \{X^k_1, X^k_2, ..., X^k_N\} \) in \( \mathbb{R}^n \) iid from distribution with density \( g(x) = N(x; \hat{\mu}_k, \hat{\sigma}^2_k) \). Calculate the estimators of integral by using (5.5) and (5.6), and let

\[
\hat{\lambda}_k = \frac{\hat{v}(c_k)}{2m(c_k)}. \tag{5.7}
\]

Step 3. If \( \hat{\lambda}_k < \epsilon \), goto step 7; else goto next.

Step 4. Compute \( \psi(X^k_i), i = 1, 2, ..., N \), select the \( N_{\text{elite}} = \lfloor \rho N \rfloor \) best samples (corresponding the \( N_{\text{elite}} \) least value \( \psi(X^k_i) \)), construct the elite samples set, denote it by \( \hat{H}^p_{c_k} = \{X^k_1, X^k_2, ..., X^k_{N_{\text{elite}}}\} \) \( \tag{5.8} \)

Step 5. Update parameter vectors \((\hat{\mu}_k, \hat{\sigma}^2_k)\) by using the following formulas:

\[
\hat{\mu}_{k+1,j} := \frac{1}{N_{\text{elite}}} \sum_{i=1}^{N_{\text{elite}}} X^k_{ij}, j = 1, 2, ..., n, \tag{5.9}
\]

and

\[
\hat{\sigma}^2_{k+1,j} := \frac{1}{N_{\text{elite}}} \sum_{i=1}^{N_{\text{elite}}} (X^k_{ij} - \hat{\mu}_{k+1,j})^2, j = 1, 2, ..., n. \tag{5.10}
\]

where \( X^k_i \in \hat{H}^p_{c_k} \).

Smooth these parameters by

\[
\hat{\mu}_{k+1} = \alpha \hat{\mu}_{k+1} + (1 - \alpha) \hat{\mu}_k, \quad \hat{\sigma}^2_{k+1} = \beta_k \hat{\sigma}^2_{k+1} + (1 - \beta_k) \hat{\sigma}^2_k. \tag{5.11}
\]

where

\[
\beta_k = \beta - \beta(1 - \frac{1}{k})^q \quad k = 1, 2, ...
\]

\(0.5 < \alpha < 0.9, \quad 0.8 < \beta < 0.99, \quad 5 \leq q \leq 10.\)

Step 6. Let

\[
c_{k+1} = c_k - \hat{\lambda}_k. \tag{5.13}
\]

Let \( k := k + 1 \), goto step 2.

Step 7. Let \( \hat{H}^*_c = \hat{H}^p_{c_k} = \{X^k_1, X^k_2, ..., X^k_{N_{\text{elite}}}\} \) be the approximation of global optimal solutions set, and \( \hat{\epsilon}^*_c = c_{k+1} \) be the approximation of global optimal value.

**Remark 5.1.** According to Remark 4.1, given a small number \( \delta > 0 \) (for example \( \delta = 0.05\% \)), and given the precision \( \epsilon_k \), we can get the sample size \( N \), which is used in step 2 of the \( k^{th} \) iteration, such that the estimators of integral satisfies the \( \epsilon_k \)–precision condition with high probability (for example, not less than 99.95%).
The parameter $\epsilon_k$ in the formula of the sample size $N$ will be discussed in the later, see Remark 5.3.

**Remark 5.2.** According to the analysis of Theorem 2.4, we set the penalty parameter $a = a_k > \frac{c_0}{\vartheta > c_k} (\forall k = 1, 2, ...)$ in computing $F(x, a)$ function value in step 2 of algorithm ILVEM, where $\vartheta > 0$ is refer as to jump. This setting ensures $F(x, a)$ matching the characters of discontinuous exact penalty function which are discussed in [3]. The initiative estimator of $a = a_0$ is a large positive real number, for example, $a_0 = 1.0e + 6$. Indeed, numerical experiments show that it is not sensitive in the parameter $a$ in ILVEM algorithm.

We are now to prove convergence of ILVEM algorithm. We will do this by showing ILVEM algorithm satisfies the convergent conditions of the one-dimensional inexact Newton method.

Without loss of general, we assume that $f(x)$ and $g_i(x) (i = 1, 2, ..., r)$ in Problem (1.1) satisfy some suitable conditions, such that when functions $\psi(x)$ defined by (5.1) and $\psi^2(x)$ defined by (5.2), both functions $\psi^2(x)/g(x)$ and $\psi^4(x)/g(x)$ have finite expectation with respect to density function $g(x)$.

**Lemma 5.1** For a given $\epsilon > 0$ and a given $c > c^*$, if function $\psi^2(x)/g(x)$ has finite integral in $\mathbb{R}^n$, then we have

$$P_r \{ \lim_{N \to +\infty} |\hat{m}_p(c) - m_p(c)| < \epsilon \} = 1.$$  

(5.14)

**Proof:** It is direct consequence of Assertion 4.1 in the previous section.

**Lemma 5.2**

$$\frac{\partial \psi^2(x, c)}{\partial c} = 2\psi(x, c).$$  

(5.15)

**Proof:** Set $\Delta c > 0$, then we know $H^p_{c+\Delta c} \supset H^p_c$, and $H^p_{c+\Delta c} = H^p_c \cup (H^p_{c+\Delta c} \setminus H^p_c)$, $H^p_c \cap (H^p_{c+\Delta c} \setminus H^p_c) = \emptyset$. Thus we have

$$\psi^2(x, c) = \begin{cases} (c - F(x, a))^2, & x \in H^p_c \\ 0, & x \in H^p_{c+\Delta c} \setminus H^p_c \\ 0, & x \notin H^p_{c+\Delta c} \end{cases}$$

and

$$\psi^2(x, c + \Delta c) = \begin{cases} (c + \Delta c - F(x, a))^2, & x \in H^p_c \\ (c + \Delta c - F(x, a))^2, & x \in H^p_{c+\Delta c} \setminus H^p_c \\ 0, & x \notin H^p_{c+\Delta c} \end{cases}$$
Therefore,

\[
\psi^2(x, c + \Delta c) - \psi^2(x, c) = \begin{cases} 
(c + \Delta c - F(x, a))^2 - (c - F(x, a))^2, & x \in H^p_c \\
(c + \Delta c - F(x, a))^2, & x \in H^p_{c+\Delta c} \not\subset H^p_c \\
0, & x \not\in H^p_{c+\Delta c} 
\end{cases}
\]

It is obvious that

\[
\lim_{\Delta c \to 0^+} H^p_{c+\Delta c} = H^p_c \quad \text{and} \quad \lim_{\Delta c \to 0^+} H^p_{c+\Delta c} \setminus H^p_c = \emptyset.
\]

Thus we have

\[
\lim_{\Delta c \to 0^+} \frac{\psi^2(x, c + \Delta c) - \psi^2(x, c)}{\Delta c} = \begin{cases} 
\lim_{\Delta c \to 0^+} \frac{\Delta c(2c + \Delta c - 2F(x, a))}{\Delta c}, & x \in H^p_c \\
0, & x \not\in H^p_c 
\end{cases}
\]

When \(\Delta c < 0\), by \(H^p_{c+\Delta c} \subset H^p_c\), we can prove in the same way that

\[
\lim_{\Delta c \to 0^-} \frac{\psi^2(x, c + \Delta c) - \psi^2(x, c)}{\Delta c} = 2\psi(x, c).
\]

By summarizing, we have

\[
\frac{\partial \psi^2(x, c)}{\partial c} = \lim_{\Delta c \to 0} \frac{\psi^2(x, c + \Delta c) - \psi^2(x, c)}{\Delta c} = 2\psi(x, c).
\]

**Lemma 5.3** For all \(c > c^*\) and a given \(\epsilon > 0\), if \(|\hat{m}_p(c) - m_p(c)| < \epsilon\), then we have

\[
|\hat{v}_p(c) - v_p(c)| < 2\epsilon(c - c^*). \tag{5.16}
\]

**Proof:** Firstly we assume that \(\hat{m}_p(c) \geq m_p(c)\) which leads to \(\hat{v}_p(c) \geq v_p(c)\) obviously. (The case \(\hat{m}_p(c) \leq m_p(c)\) can be proved in the same way.) Notice that
\( \hat{m}_p(c^*) = m_p(c^*) = 0 \) and \( |\hat{m}_p(c) - m_p(c)| < \epsilon \), by Lemma 5.2 we have

\[
|\hat{v}_p(c) - v_p(c)| = \frac{1}{N} \sum_{i=1}^{N} \frac{\psi^2(X_i, c)}{g(X_i)} - \int_{H_{c}} \psi^2(x, c)dv
\]

\[
= \frac{1}{N} \sum_{i=1}^{N} \int_c^c \frac{2\psi(X_i, c)}{g(X_i)} dc - \int_{H_{c}} \left[ \int_c^c 2\psi(x, c)dv \right] dc
\]

\[
= \int_c^c \left[ \frac{1}{N} \sum_{i=1}^{N} \frac{2\psi(X_i, c)}{g(X_i)} \right] dc - \int_{H_{c}} 2\psi(x, c)dv dc
\]

\[
= \int_c^c 2[\hat{m}_p(c) - m_p(c)] dc
\]

\[
< 2\epsilon(c - c^*)
\]

This completes the proof.

It is obvious that functions \( c \mapsto \Psi(x, c) \) and \( c \mapsto \Psi^2(x, c) \) are increasing convex functions. When \( c > F(x, a) \), the function \( c \mapsto \Psi^2(x, c) \) is strictly increasing and strictly convex. Since \( \hat{v}_p(c) \) and \( \hat{m}_p(c) \) are finite positive combinations of \( \Psi(x, c) \) and \( \Psi^2(x, c) \), they have the same monotonicity and convexity properties for all \( c \in (-\infty, +\infty) \).

**Lemma 5.4** For a given \( c > c^* \) and \( 0 < \eta < 1 \), there exists \( 0 < \frac{\eta}{2 + \eta} < \theta < \frac{\eta}{2 - \eta} < 1 \), if

\[
\epsilon \leq \min \left\{ \tau_1 m_p(c), \tau_2 \frac{v_p(c)}{2(c - c^*)} \right\} |0 < \tau_1, \tau_2 < \theta \}, \quad (5.17)
\]

and \( |\hat{m}_p(c) - m_p(c)| < \epsilon \), then we have

\[
|1 - \frac{m_p(c)\hat{v}_p(c)}{\hat{m}_p(c)v_p(c)}| < \eta < 1.
\]

**Proof:** If \( |\hat{m}_p(c) - m_p(c)| < \epsilon \leq \tau_1 m_p(c) \), then we have

\[
\left| \frac{\hat{m}_p(c) - m_p(c)}{m_p(c)} \right| < \tau_1,
\]

which implies

\[
1 - \tau_1 < \frac{\hat{m}_p(c)}{m_p(c)} < 1 + \tau_1.
\]

The last inequality is equal to

\[
\frac{1}{1 + \tau_1} < \frac{m_p(c)}{\hat{m}_p(c)} < \frac{1}{1 - \tau_1}. \quad (5.19)
\]
On the other hand, when $|\hat{m}_p(c) - m_p(c)| < \epsilon \leq \tau_2 \frac{v_p(c)}{2(c-c^*)}$, by Lemma 5.3 we have

$$\left| \frac{\hat{v}_p(c) - v_p(c)}{v_p(c)} \right| < \frac{2c(c-c^*)}{v_p(c)} < \tau_2,$$

which implies

$$1 - \tau_2 < \frac{\hat{v}_p(c)}{v_p(c)} < 1 + \tau_2. \quad (5.20)$$

From (5.19) and (5.20), we get

$$\frac{1 - \theta}{1 + \theta} < \frac{1 - \tau_2}{1 + \tau_1} < \frac{m_p(c)\hat{v}_p(c)}{m_p(c)v_p(c)} < \frac{1 + \tau_2}{1 - \tau_1} < \frac{1 + \theta}{1 - \theta}. \quad (5.21)$$

Let

$$\frac{1 - \theta}{1 + \theta} > 1 - \eta, \quad \frac{1 + \theta}{1 - \theta} < 1 + \eta.$$

By $0 < \eta < 1$ we have

$$0 < \frac{\eta}{2 + \eta} < \theta < \frac{\eta}{2 - \eta} < 1. \quad (5.22)$$

Since $\theta$ satisfies (5.22) and $\epsilon$ is defined by (5.17), when $|\hat{m}_p(c) - m_p(c)| < \epsilon$ we have

$$1 - \eta < \frac{m_p(c)\hat{v}_p(c)}{m_p(c)v_p(c)} < 1 + \eta,$$

which implies (5.18) and this completes the proof.

On convergence of the one-dimensional inexact Newton method for solving single variable equation $h(x) = 0$, we have the following lemma:

**Lemma 5.5** Let $h : R \rightarrow R$ be a continuously differentiable function, $\{x_k\}$ is a sequence generated by the one-dimensional inexact Newton method for solving the equation $h(x) = 0$. This is:

$$x_{k+1} = x_k + s_k, \quad \text{where} \quad s_k \text{ is satisfied that } h'(x_k)s_k = -h(x_k) + r_k.$$

Suppose the following conditions are satisfied:

a). There exists $x^*$ such that $h(x^*) = 0$ and $h'(x^*) = 0$;

b). The function $h(x)$ is convex, and it has continuous derivative in a neighbor $(x^* - \delta, x^* + \delta)$, $\forall \delta > 0$;

c). When $x_t$ is in a neighborhood of $x^*$ and $x_t \neq x^*$, it has $h'(x_t) \neq 0$. And

$$\lim_{x_t \rightarrow x^*} \frac{h(x_t)}{h'(x_t)} = 0. \quad (5.23)$$

d). There exists $0 < \eta < 1$ and a sequence $\{\eta_k : 0 < \eta_k < \eta\}$, such that $|r_k| < \eta_k h(x_k), \ k = 0, 1, 2, ...$. 

18
Then the sequence \( \{x_k\} \) converges to \( x^* \).

**Proof:** We get from condition d),

\[
|s_k| = \left| \frac{-h(x_k) + r_k}{h'(x_k)} \right| \\
\leq \left| \frac{h(x_k)}{h'(x_k)} \right| + \left| \frac{r_k}{h'(x_k)} \right| \\
\leq \left| \frac{h(x_k)}{h'(x_k)} \right| + \left| \frac{\eta h(x_k)}{h'(x_k)} \right| \\
\leq (1 + \eta) \left| \frac{h(x_k)}{h'(x_k)} \right| .
\]  
(5.24)

By using Taylor’s theorem, we get

\[
h(x_{k+1}) = h(x_k) + h'(x_k)s_k + O(|s_k|^2) \\
= h(x_k) - (h(x_k) - r_k) + O\left( \left| \frac{h(x_k)}{h'(x_k)} \right|^2 \right) \\
= r_k + O\left( \left| \frac{h(x_k)}{h'(x_k)} \right|^2 \right).
\]

By taking norm on both sides of the above equation and recalling condition d), we have

\[
|h(x_{k+1})| \leq |r_k| + O\left( \left| \frac{h(x_k)}{h'(x_k)} \right|^2 \right) \\
\leq \eta_k |h(x_k)| + O\left( \left| \frac{h(x_k)}{h'(x_k)} \right|^2 \right) \\
\leq \eta |h(x_k)| + O\left( \left| \frac{h(x_k)}{h'(x_k)} \right|^2 \right).
\]

Take limit with respect to \( x_k \to x^* \) on both sides of the previous inequality, using condition a) c) and d), we get

\[
\lim_{x_k \to x^*} |h(x_{k+1})| = 0.
\]

Thus by convexity of \( h(x) \) (condition b)), we know \( |x_{k+1} - x^*| < \gamma |x_k - x^*| \), where \( 0 < \gamma < 1 \). That is to say, sequence \( \{x_k\} \) converge to \( x^* \).

**Lemma 5.6**

\[
\lim_{c \to c^* + m_p(c)} \frac{v_p(c)}{c} = 0.
\]

**Proof:** By using the mean value theorem, we get

\[
v_p(c) - v_p(c^*) = v'_p(\xi)(c - c^*), \quad \xi \in (c^*, c).
\]  
(5.25)
Note \( v_p(c^*) = 0 \) and \( v'_p(\xi) = 2m_p(\xi) \), we have

\[
\lim_{c \to c^*+} \frac{v_p(c)}{m_p(\xi)} \cdot 2(c - c^*) = 0.
\]

(5.26)

Note \( \xi \in (c^*, c) \), when \( c \to c^* \), \( \xi \to c^* \). Thus the assertion of Lemma 5.6 follows from (5.26) directly.

**Theorem 5.7** Set the sequences \( \{c_k\} \) and \( \{\hat{H}_p c_k\} \) are generated by ILVEM algorithm. Then the sequence \( \{c_k\} \) converges to \( c^* \) and the sequence \( \{\hat{H}_p c_k\} \) converges to \( H^*_p = \{x \in \mathbb{R}^n : F(x, a) = c^*_p\} \), respectively.

**Proof:** We first prove \( \{c_k\} \) converges to the root of the variance equation \( v_p(c) = 0 \). We do this by verifying ILVEM algorithm satisfies the convergent conditions of the one-dimensional inexact Newton method (Lemma 5.5).

At the \( k \)th iteration, the step size of LVEM algorithm is \( \lambda_k = \frac{v_p(c_k)}{2m_p(c_k)} \) (which is Newton method for solving the equation \( v_p(c) = 0 \) essentially), and the step size of ILVEM algorithm is \( \hat{\lambda}_k = \frac{\hat{v}_p(c_k)}{2\hat{m}_p(c_k)} \) (which is one-dimensional inexact Newton method for solving the equation \( v_p(c) = 0 \) essentially). By comparing these two step sizes, we know the residual of ILVEM algorithm is that:

\[
|r_k| = 2m_p(c_k)|\lambda_k - \hat{\lambda}_k|.
\]

(5.27)

Thus

\[
\left| \frac{r_k}{v_p(c_k)} \right| = \left| \lambda_k - \hat{\lambda}_k \right| \cdot \frac{2m_p(c_k)}{v_p(c_k)} = \left| \frac{v_p(c_k)}{2m_p(c_k)} - \frac{\hat{v}_p(c_k)}{2\hat{m}_p(c_k)} \right| \cdot \frac{2m_p(c_k)}{v_p(c_k)} = \left| 1 - \frac{m_p(c_k)}{\hat{m}_p(c_k)} \frac{\hat{v}_p(c_k)}{v_p(c_k)} \right|
\]

On the other hand, the density function of sample distribution \( g(x) = N(\hat{\mu}_k, \hat{\sigma}^2_k) \) and the level value \( c_k \) are fixed at the \( k \)th iteration (which are given by the previous iteration), thus Assertion 4.1 and Remark 4.1 are applicable. Therefore, for given \( 0 < \eta < 1 \) and \( \frac{n}{\tau_1 \eta} < \theta < \frac{n}{\tau_2 \eta} \), if

\[
\epsilon_k \leq \min \left\{ \tau_1 m_p(c_k), \tau_2 \frac{v_p(c_k)}{2(c_k - c^*)} \mid 0 < \tau_1, \tau_2 < \theta \right\},
\]

(5.28)

and the sample size is \( N = \left\lceil \left( \frac{-\ln \delta}{2\epsilon_k^2} \right)^n \right\rceil + 1 \), by Remark 4.1 we have

\[
P_r \{ |\hat{m}_p(c_k) - m_p(c_k)| < \epsilon_k \} = 1 - \delta.
\]
Take limit with respect to $\delta \to 0$, we have

$$P \left\{ \lim_{N \to \infty} |\hat{m}_p(c_k) - m_p(c_k)| < \epsilon_k \right\} = 1. \quad (5.29)$$

Thus by Lemma 5.4, the following inequality holds with probability one,

$$\frac{r_k}{v_p(c_k)} = \left| 1 - \frac{m_p(c_k)}{\hat{m}_p(c_k)} \frac{\hat{v}_p(c_k)}{v_p(c_k)} \right| < \eta < 1. \quad (5.30)$$

The inequality (5.30) shows that ILVEM algorithm satisfies the convergent condition d) of the one-dimensional inexact Newton method (Lemma 5.5). Thus, by Lemma 5.6 and using Lemma 5.5 again, we have that the sequence $\{c_k\}$ generated by ILVEM algorithm converges to the root of the variance equation $v_p(c) = 0$.

In addition, we know that $c_p^* = \min_{x \in \mathbb{R}^n} F(x, a)$ is the largest root of the variance equation $v_p(c) = 0$. Therefore we only need to prove $c_k \geq c_p^* (k = 0, 1, 2, \ldots)$ in the next. We do this by using mathematical induction.

It is obvious that $c_0 = F(x_0, a) \geq c_p^*$. We assume $c_k \geq c_p^*$ then $\lambda_k = 0$, and the process stops at step 3. If $c_k > c_p^*$, by the convexity of $\hat{v}_p(c)$ (see the proof of Lemma 5.3) and $\hat{v}_p(c_p^*) = 0$, we have

$$-\hat{v}_p(c_k) = \hat{v}_p(c_p^*) - \hat{v}_p(c_k) \geq 2\hat{m}_p(c_k)(c_p^* - c_k),$$

which follows that

$$c_{k+1} = c_k - \lambda_k = c_k - \frac{\hat{v}_p(c_k)}{2\hat{m}_p(c_k)} \geq c_p^*.$$

By using mathematical induction, we have

$$c_k \geq c_p^*, k = 0, 1, 2, \ldots.$$

By summarizing, we have that $\{c_k\}$ converges to $c_p^*$. When $\{c_k\}$ converges to $c_p^*$, by using the Property 2 in [4] we immediately have that $\{H_{c_k}\}$ converges to $H_{c_p^*}$.

Theorem 3.2 shows that, if $a > 0$ is big enough, then $c_p^* = c^*$ and $H_{c_p^*} = H^*$. Thus by Theorem 5.7, we indeed have that the sequences $\{c_k\}$ and $\{H_{c_k}\}$ generated by ILVEM algorithm converge to $c^*$ and $H^*$ respectively.

**Remark 5.3.** By Lemma 5.4, if and only if the estimator $\hat{m}(c_k)$ satisfies the precision condition (5.17), the control convergent condition d) of Lemma 5.5 holds when it is used in the proof of Theorem 5.7. See (5.28), (5.29) and (5.30). On the other hand, in the step 2 of ILVEM algorithm, the sample size $N$ depends on $\epsilon_k$. And the parameter $c_k$ depends on $c_k$, see (5.28). But in (5.28), the values of functions $m(c_k)$, $v(c_k)$ and $c^*$ are all unknown. At the $k$–th iteration, $c_k$ is unknown. Thus in practice, a suitable alternative of (5.28) is as the following:

$$\epsilon_k \leq \min \left\{ \tau_1 \hat{m}_p(c_k), \tau_2 \frac{\hat{v}_p(c_k-1)}{2(c_k-1 - c)} \right\}, \quad (5.31)$$
where
\[ \bar{c} = \min \{ \psi(X_i^{k-1}), i = 1, 2, ..., N, X_i^{k-1} \in X^{k-1} \} \quad (5.32) \]

By substituting (5.31–5.32) into \( N = \left\lceil \left( -\ln \delta \right)^{\frac{1}{2k\epsilon_{\delta}}} \right\rceil + 1 \) in step 2 of ILVEM algorithm, we can get the sample size \( N \) at the \( k \)-th iteration.

6 Numerical results

We choose some classical global optimization problems to test the performance of ILVEM algorithm. They are commonly multiple-extreme, multidimensional and difficult to optimize by the other methods (especially local search methods). These benchmarks are chosen from [25, 27, 28] respectively. We classify them into three classes:

**Class A:** Two-dimensional, multimodal or difficult to optimize functions.

**Class B:** Multidimensional, multimodal with huge number of local extremes.

**Class C:** Constrained problems.

We use ILVEM algorithm to solve these problems on a notebook personal computer with conditions: CPU 1.8 GHZ, 256 MB EMS memory; Matlab 6.5. The parameters in practice are set in the following: \( \alpha = 0.8, \beta = 0.80, q = 6. \) We compare the numerical results to the ones of computing by integral level-set method (abbreviating ILSM) and by the cross-entropy method (abbreviating CE). We list the problems in the following, and state the numerical results in table 6.1 to table 6.3, respectively.

**Remark 6.1.** In our ILVEM algorithm, the accelerated method in Newton-step introduced in [20] is used.

**Exam 1.** [A, 29] Three-humps camel back function: In the search domain \( x_1, x_2 \in [-5, 5] \) this function is defined as follows and has \( f_{\min}(0, 0) = 0. \)

\[ f(x) = 2x_1^2 - 1.06x_1^4 + x_1^6/6 + x_1x_2 + x_2^2. \]

**Exam 2.** [A, 29] Six-hump camel back function is a global optimization test function. Within the bounded region it has six local minima, two of them are global ones. Test area is usually restricted to the rectangle \( x_1 \in [-3, 3], x_2 \in [-2, 2]. \) Two global minima equal \( f(x^*) = -1.0316, x^* = (0.0898, -0.7126), (-0.0898, 0.7126) \)

\[ f(x) = \left( 4 - 2.1x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1x_2 + (-4 + 4x_2^2) x_2^2. \]
Exam 3. [A, 30] Modified Schaffer function 4: In the search domain \( x_i \in [-100, 100], i = 1, 2, \) \( f_{\text{min}}(0, 1.253132) = 0.292579. \)

\[
f(x) = 0.5 + \frac{\cos^2[\sin(\sqrt{|x_1^2 - x_2^2|})] - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}.
\]

Exam 4. [A, 30] Goldstein-Price function is a global optimization test function. Test area is usually restricted to the square \( x_i \in [-2, 2], i = 1, 2. \) Its global minimum equal \( f(x^*) = 3, x^* = (0, -1). \)

\[
f(x) = [1 + (x_1 + x_2 + 1)^2 \ast (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]
\times [30 + (2x_1 - 3x_2)^2 \ast (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]
\]

<table>
<thead>
<tr>
<th>prob.</th>
<th>ILVEM</th>
<th>ILSM</th>
<th>CE</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>iter.</td>
<td>eval.</td>
<td>error</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>2020</td>
<td>1.08e-6</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>2424</td>
<td>1.02e-6</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>2222</td>
<td>6.22e-5</td>
</tr>
<tr>
<td>4</td>
<td>18</td>
<td>3636</td>
<td>2.16e-5</td>
</tr>
</tbody>
</table>

Remark 6.2. 1. In all tables, the notation \textit{iter.} denotes the number of iterations, \textit{eval.} denotes the number of function evaluations, and \textit{error} denotes the absolute error which is measured by \(|\hat{c}^* - c^*|\), where \(c^*\) is the known global minimal value.
2. In all of the test processes, the stop conditions are same which are stated in the following: In II\text{V}EM, we choose $\varepsilon = 1.0e - 6$, when $\lambda < \varepsilon$, the process stops. In ILSM, they choose $\varepsilon = 1.0e - 10$, when $\hat{V} < \varepsilon$, the process stops. And in CE, they choose $\varepsilon = 1.0e - 3$, when $\max\{\sigma_i\} < \varepsilon$, the process stops.

3. In most cases, algorithm ILSM preserves advantage in the number of iterations and of evaluations, but it spends a lot of cpu-time in finding approximating level set, so it has disadvantage in cpu-time.

**Exam 5.** [B, 27] The Trigonometric function: $\eta = 7$, $\mu = 1$, $x_i^* = 0.9$.

$$f(x) = \sum_{i=1}^{n} 8 \sin^2(\eta(x_i - x_i^*))^2 + 6 \sin^2(2\eta(x_i - x_i^*))^2 + \nu(x_i - x_i^*)^2$$

**Exam 6.** [B, 29] Ackley’s function is a widely used multimodal test function. It is recommended to set $a = 20$, $b = 0.2$ $c = 2\pi$, test area is usually restricted to hypercubes $x_i \in (-32.78, 32.78)$, $i = 1, 2, ..., n$. $f^*(x^*) = 0$.

$$f(x) = -a \exp \left( -b \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^{n} \cos(cx_i) \right) + a + \exp(1).$$

**Exam 7.** [B, 30] The Shekel-5 function: Its search domain: $-32.195 \leq x_i \leq 32.195$, $i = 1, 2, ..., n$.

$$f(x) = 20 + e - 20e^{(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2})} - e^{(\frac{1}{n} \sum_{i=1}^{n} \cos(\pi x_i))}$$

**Exam 8.** [B, 30] Rastrigin’s function is based on the function of De Jong with the addition of cosine modulation in order to produce frequent local minima. Thus, the test function is highly multimodal. However, the location of the minima are regularly distributed. Test area is usually restricted to hypercubes $x_i \in [-5.12, 5.12]$, $i = 1, 2, ..., n$. $f^*(x^*) = 0$.

$$f(x) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)].$$
Table 6.2. Numerical Results on Class B

<table>
<thead>
<tr>
<th>prob. id</th>
<th>n</th>
<th>ILVEM iter.</th>
<th>ILVEM eval.</th>
<th>ILVEM error</th>
<th>ILSM iter.</th>
<th>ILSM eval.</th>
<th>ILSM error</th>
<th>CE iter.</th>
<th>CE eval.</th>
<th>CE error</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td>21</td>
<td>21084</td>
<td>9.00e-3</td>
<td>26</td>
<td>119253</td>
<td>6.02e-5</td>
<td>178</td>
<td>89400</td>
<td>9.23e-4</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>31</td>
<td>7874</td>
<td>3.16e-3</td>
<td>70</td>
<td>3877</td>
<td>9.92e-5</td>
<td>283</td>
<td>28300</td>
<td>1.87e-3</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>37</td>
<td>12360</td>
<td>1.05e-4</td>
<td>6</td>
<td>4802</td>
<td>1.87e-4</td>
<td>260</td>
<td>26000</td>
<td>1.70e-3</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>43</td>
<td>4429</td>
<td>2.0e-10</td>
<td>7</td>
<td>974</td>
<td>3.59e-7</td>
<td>77</td>
<td>7700</td>
<td>1.8e-10</td>
</tr>
</tbody>
</table>

Remark 6.3. In this class, when the stop rule is $\hat{\lambda}_k < 1.0e - 6$, almost all of samples in ILVEM algorithm are in the current level set until the process stops.

Exam 9. [C, 27] Constrained problem (1): The bounds: $U = (10, 10), L = (0, 0)$.

$$\begin{align*}
\max_x f(x) &= \frac{\sin(2\pi x_1) \sin(2\pi x_2)}{x_1^2(x_1 + x_2)} \\
\text{s.t.} & \quad g_1(x) = x_1^2 - x_2 + 1 \leq 0, \\
& \quad g_2(x) = 1 - x_1 + (x_2 - 4)^2 \leq 0.
\end{align*}$$

Exam 10. [C, 27] Constrained problem (2): The theoretical solution is $x^* = (1, 1), f(x^*) = 1$.

$$\begin{align*}
\min_x f(x) &= (x_1 - 2)^2 + (x_2 - 1)^2 \\
\text{s.t.} & \quad x_1 + x_2 - 2 \leq 0 \\
& \quad x_1^2 - x_2 \leq 0.
\end{align*}$$

Exam 11. [C, 27] Constrained problem (3): The bounds: $(10, 10, \ldots, 10), L = (-10, -10, \ldots, -10)$.

$$\begin{align*}
\min_x f(x) &= (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\
& \quad + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7 \\
\text{s.t.} & \quad g_1(x) = 2x_1^2 + 3x_2^2 + x_3 + 4x_4^2 + 5x_5 - 127 \leq 0, \\
& \quad g_2(x) = 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282 \leq 0, \\
& \quad g_3(x) = 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196 \leq 0, \\
& \quad g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0.
\end{align*}$$
Exam 12. [C, 27] Constrained problem (4): Its known solution is $x^* = (4.75, 5.0, -5.0)$, $f(x^*) = -14.75$

$$\begin{align*}
\min & \quad f(x) = -x_1 - x_2 + x_3; \\
\text{s.t.} & \quad \sin(4\pi x_1) - 2\sin^2(2\pi x_2) - 2\sin^2(2\pi x_3) \geq 0; \\
& \quad -5 \leq x_j \leq 5, \ j = 1, 2, 3.
\end{align*}$$

Table 6.3. Numerical Results on Class C

<table>
<thead>
<tr>
<th>prob.</th>
<th>n</th>
<th>ILVEM</th>
<th>ILSM</th>
<th>CE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>iter.</td>
<td>eval.</td>
<td>error</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>51</td>
<td>2193</td>
<td>4.21e-7</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>36</td>
<td>1548</td>
<td>4.39e-3</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>40</td>
<td>28120</td>
<td>9.25e-2</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>26</td>
<td>7878</td>
<td>1.32e-4</td>
</tr>
</tbody>
</table>

Remark 6.4: 1. CE for solving exam 11 can not stop regularly, it is break off forcibly after 200 iterations.

2. In class C, ILSM algorithm almost has no more advantage with comparison to ILVEM algorithm.

7 Further discussions and conclusions

In this paper, we propose a new method for solving continuous global optimization problem, namely the level-value estimation method (LVEM). Since we use the updating mechanism of sample distribution based on the main idea of the cross-entropy method in the implementable algorithm, ILVEM algorithm automatically update the “importance region” which the “better” solutions are in. This method overcomes the drawback that the level-set is difficult to be determined, which is the main difficult in the integral level-set method [3]. Numerical experiments show the availability, efficiency and better accuracy of ILVEM algorithm.

However, ILVEM algorithm may be inefficient in dealing with some particular problems. For example, the fifth function of De Jong,

$$\begin{align*}
\min & \quad f(x) = \{0.002 + \sum_{i=-2}^{2} \sum_{j=-2}^{2} [5(i + 2) + j + 3 + (x_1 - 16j)^6 + (x_2 - 16i)^6]^{-1}\}^{-1}, \\
\text{s.t.} & \quad x_1, x_2 \in [-65.536, 65.536].
\end{align*}$$

Its figure is as follows:
This function is hard to minimize. ILVEM algorithm, ILSM algorithm and CE algorithm all fail in dealing with this problem.

When the number of global optima is more than one, ILVEM algorithm can generally get one of them, see the result on exam 2 (six humps camel function). However, it fails in the fifth function of De Jong. This difficult case may be improved by using the density function with multi-kernel in sample distribution. For example, Rubinstein [9] proposed a coordinate-normal density with two kernels. This conjecture needs verification by using more numerical experiments in our further work.

It is possible to use a full covariance matrix for better approximation. But in high dimensional case, to get a random sample by using a full covariance matrix is expensive until now.

Acknowledgement: We would like to give our thanks to Professor A. Hoffmann, for his help to improve the quality of the final version of this paper. And we are also grateful to the anonymous reviewers, for their detailed comments. This work is supported by Natural Sciences Foundation of China (No. 10671117).

References


