Global and Stochastic Analysis
Vol. 5, No. 2, December (2018), 153-170

MUK PUBLICATIONS
Open Access Publisher

ADAPTIVE GAUSSIAN MIXTURE MODEL IN GLOBAL OPTIMIZATION

SOONHUI LEE

Abstract. We discuss stochastic optimization problems that involve unknown dependencies between control and random variables, where we lack information about both the distribution of uncertainty and the exact form of revenue function. We study adaptive algorithm that provides the decision which converges to the true optimal solution with learning uncertainty (for example, demand uncertainty) and optimizing the objective function (for example, maximizing revenue) iteratively.

1. Introduction

Optimization problems arises in various fields such as engineering, sciences, and business. Solution approaches are constantly developed and improved. It is often the case that the structure or relationship between the objective function and the decision variables are unknown, and hence, most algorithms designed to solve optimization problems in such situations treat the objective function as "blackbox" that returns the objective function value evaluated at a specific decision point. Finding global optimal solutions requires much computational effort especially when dealing with complex systems. There are various search methods on the decision space employed in a wide range of algorithms.

[2] classified search methods as being instance-based or model-based. The instance-based methods generate new decisions directly depending on the current solution set. Well-known instance-based methods include genetic algorithms [3] and simulated annealing algorithms [6]. Model-based search algorithms are relatively newer than instance-based algorithms. [2] provides a survey on well-established model-based search algorithms. In the model-based search methods, candidate solutions are sampled from a parameterized probabilistic model that is updated based on the previously generated solutions and the future search is designed to be concentrated on the high quality solutions area. The ant colony optimization [4], stochastic gradient ascent, cross-entropy [5], estimation of distributions [7], model reference adaptive search (MRAS) by [9], and model-based annealing random search methods (MARS) by [8] belong to this category.

In the MRAS and MARS methods, there is a reference distribution that guides a parametrized distribution whose parameter is updated in such a way that the

Date: Date of Submission August 6, 2018; Date of Acceptance September 8, 2018, Communicated by Yuri E. Gliklikh.

2000 *Mathematics Subject Classification.* Primary 49K99; Secondary 90-08.

Key words and phrases. global optimization, Gaussian mixture model, convergence.

minimized. These algorithms focus on the exponential family of distributions that makes the parameter optimization procedures analytically tractable.

In our study, we generate a candidate solution from a Gaussian mixture model with adaptively updated parameters. The adaptive Gaussian mixture (AGM) model used in this paper is structured with the same goal as the MRAS and MARS method; the probability model focuses more on the decision space containing high-quality solutions as iteration proceeds. The Gaussian mixture model is adaptively updated based on the current $\mathcal{M}$ best solutions. The best ranked decision continues to improve. Like many heuristics that are appealing because they are intuitively simple and work well in practice, our goal in developing AGM model is to maintain intuitively simple structure, and hence, be easy to implement in practice. The AGM algorithm does not require parameter optimization procedures but the algorithm has global convergence properties under mild conditions. We also note that there are not as many parameters in the AGM algorithm as in some other search algorithm; selecting the proper values of parameters that yield good performance of the algorithm is investigated.

The remainder of the paper begins with the problem description in Section 2. In Section 2, we also present the global convergence properties of the AGM algorithm. In Section 3, we present numerical results obtained by applying the AGM algorithm to solve benchmark problems often used in global optimization. In Section 4, we give concluding remarks.

2. Problem Description

We consider the following optimization problem:

$$q^* \in \arg\max_{q \in \mathcal{Q}} r(q), \;\; \mathcal{Q} \subseteq \mathbb{R}^n \tag{2.1}$$

where q is a decision variable in $\mathcal{Q}$, and $r : \mathcal{Q} \mapsto \mathbb{R}$ is a real-valued function. Throughout this paper, we assume that (2.1) has a unique global optimal decision q^*, $r(q)$ is bounded below, and evaluating $r(q)$ is expensive. We can think of solving (2.1) in the context of a sequential decision making process; a decision is made, data are observed, the estimate is updated, a decision is made, and so on. In this context, the objective function can be evaluated when a decision is made.

Our algorithms use a mixture of Gaussian kernel densities for a probabilistic model to generate a solution/decision each period. A mixture model is a statistical model where the probability density function is a convex sum of multiple density functions. Mixture models provide a flexible and powerful mathematical approach to modeling that is widely applied in many fields. A candidate decision q is generated from a multi-modal distribution $\hat{f}$ with Gaussian mixture model. To be concrete, $\hat{f}$ is a weighted sum of multiple Gaussian kernel density functions. At iteration j, the modes in $\hat{f}_j$ are determined by a collection of $\mathcal{M}$ (assuming $\mathcal{M} < j$) best decisions among $j-1$ previous decisions. Assuming that evaluating $r(q)$ is expensive, generating many candidate solutions in one iteration or increasing the number of candidate solutions as iteration proceeds are not feasible in our setting. A new decision q_{j+1} will be generated from $\hat{f}_j$ and if an objective value evaluated at a decision generated from $\hat{f}_j$ (i.e., $r(q_{j+1})$) is better than any decisions among the

$\mathcal{M}$ best decisions, then the $\mathcal{M}$ best decisions will also be updated, and hence, $\hat{f}_j$ will be also updated accordingly. As the algorithm may have to stop at some point, in which case the best decision should be employed for the remaining time periods, it is important to see the convergence results of the best decision. We will show that the best decision converges to the optimal decision with probability 1 under some conditions. We do not require $\hat{f}_j$ to converge to a degenerate distribution concentrating on the optimal solution unlike the model reference adaptive search (MRAS) method introduced in [9] and [10]. But concentration areas of $\hat{f}_j$ will move toward the promising areas, which will be accomplished by controlling the density at each mode or weights on each kernel density function. In the next section, we describe how the algorithm works in more detail.

2.1. Algorithm. In this section, we present the basic methodology of our algorithm when one candidate decision is generated per decision period. We define $\mathcal{M}$ to be the number of Gaussian kernel densities in $\hat{f}$ and $\mathcal{M}$ can be viewed as an input parameter indicating the level of exploration. At initialization, $\mathcal{M}$ decisions $q_1^1, \ldots, q_{\mathcal{M}}^1$ are sampled from the solution space $\mathcal{Q}$ and the corresponding objective values $r(q_1^1), \ldots, r(q_{\mathcal{M}}^1)$ are evaluated.

A multi-modal distribution $\hat{f}(q|\bar{q}^1, \bar{r}^1)$ is constructed in such a way that the modes of the kernel densities in the mixture are at $q_1^1, \ldots, q_{\mathcal{M}}^1$ and the densities at the modes depend on $r(q_1^1), \ldots, r(q_{\mathcal{M}}^1)$ where $\bar{q}^1$ and $\bar{r}^1$ are vector representations of $q_1^1, \ldots, q_{\mathcal{M}}^1$ and $r(q_1^1), \ldots, r(q_{\mathcal{M}}^1)$, respectively. In the $(j+1)$st period, we generate a decision q_{j+1} according to a sampling distribution $\hat{f}(q|\bar{q}^j, \bar{r}^j)$, where $\bar{r}^j$ represents the set of $\mathcal{M}$ highest objective values observed up to the jth period and $\bar{q}^j$ represents the set of decisions corresponding to $\bar{r}^j$. That is,

$$\bar{r}^j = \left(r_{(1)}^j, r_{(2)}^j, \ldots, r_{(\mathcal{M})}^j\right), \ \bar{q}^j = \left(q_{[1]}^j, q_{[2]}^j, \ldots, q_{[\mathcal{M}]}^j\right)$$

where

$$r_{(1)}^j \leq r_{(2)}^j \leq \cdots \leq r_{(\mathcal{M})}^j$$

and $q_{[i]}^j$ gives $r_{(i)}^j$. After sampling the decision q_{j+1}, we score it according to the function, $r(\cdot)$ i.e., we compute $r(q_{j+1})$ and compare it with $r_{(1)}^j$. If $r(q_{j+1}) \geq r_{(1)}^j$, $r_{(1)}^j$ will be replaced by $r(q_{j+1})$ so that $\bar{r}^{j+1}$ consists of $\bar{r}^j \backslash r_{(1)}^j$ and $r(q_{j+1})$. Otherwise, $\bar{r}^{j+1} = \bar{r}^j$.

Algorithm 1
Initialization
Generate $q_1, \ldots, q_{\mathcal{M}}$ and evaluate $r(q_1), \ldots, r(q_{\mathcal{M}})$.
Set $\bar{q}^1 = (q^1_{[1]}, \ldots, q^1_{[\mathcal{M}]})$, $\bar{r}^1 = (r^1_{(1)}, \ldots, r^1_{(\mathcal{M})})$, and $\hat{f}(q|\bar{q}^1, \bar{r}^1)$

In j+1 st period (j=1,2,. . .),

Step 1: Generate q_{j+1} from a certain distribution $\hat{f}(q|\bar{q}^j, \bar{r}^j)$ that depends on $\bar{r}^j = \left(r^j_{(1)}, r^j_{(2)}, \ldots, r^j_{(\mathcal{M})}\right)$ and $\bar{q}^j = \left(q^j_{[1]}, q^j_{[2]}, \ldots, q^j_{[\mathcal{M}]}\right)$.

Step 2: Compute $r(q_{j+1})$.

Step 3: Set $\bar{r}^{j+1} = \left(r^{j+1}_{(1)}, r^{j+1}_{(2)}, \ldots, r^{j+1}_{(\mathcal{M})}\right)$ where $\bar{r}^{j+1}_{(i)}$ are the $\mathcal{M}$ highest values among $r^j_{(1)}, \ldots, r^j_{(\mathcal{M})}$ and $r(q_{j+1})$ and $\bar{q}^{j+1} = \left(q^{j+1}_{[1]}, q^{j+1}_{[2]}, \ldots, q^{j+1}_{[\mathcal{M}]}\right)$.

$j \leftarrow j + 1$**:**

Repeat Step 1 – Step 3.

At the initialization, $\bar{q}^1$ could be the decisions made in the first $\mathcal{M}$ periods, numbered $-(\mathcal{M}-2), \ldots, 0, 1$ using any rule that is chosen by the decision maker.

2.1.1. *Choice of sampling distribution.* We use the kernel density estimation to construct a sampling distribution $\hat{f}$. Suppose we have a random sample $x_1, \ldots, x_N$ drawn from a probability density $f_X(x)$, and we wish to estimate f_X at a point x_0. Then, the basic form of the kernel density estimate for the univariate case is as follows,

$$\hat{f}_X(x) = \sum_{i=1}^{N} \frac{1}{N\lambda} \mathcal{K}_\lambda(x_0, x_i) \tag{2.2}$$

where $\mathcal{K}_\lambda$ is a kernel function with the width λ. Our main purpose in using the kernel density estimation is nonparametric classification of the ranked decisions. Therefore, the random samples in (2.2) are replaced by the ranked (e.g., best $\mathcal{M}$) decisions. The weight $1/N$ and the width λ can be chosen differently depending on the rank of the decisions.

We use one of the most popular smoothing kernels, the Gaussian kernel. The basic form (2.2) of kernel density estimate defining the Gaussian mixture distribution $\hat{f}$ at iteration j is as follows,

$$\hat{f}(q|\bar{q}^j, \bar{r}^j) = \sum_{i=1}^{\mathcal{M}} \alpha_i^j \mathcal{K}\left(\frac{q - q_{[i]}^j}{\sigma_{[i]}^j}\right) \tag{2.3}$$

$$\text{where} \quad \sum_{i=1}^{\mathcal{M}} \alpha_i^j = 1 \text{ and}$$

$$\mathcal{K}\left(\frac{q - q_{[i]}^j}{\sigma_{[i]}^j}\right) = \frac{1}{\sqrt{2\pi}\sigma_{[i]}^j} \exp\left(-\frac{1}{2}\left(\frac{q - q_{[i]}^j}{\sigma_{[i]}^j}\right)^2\right) \tag{2.4}$$

where the $\mathcal{M}$ best solutions $(q_{[i]}^j, i = 1, \ldots, \mathcal{M})$ are the modes of each kernel. The weight α_i^j and width $\sigma_{[i]}^j$ differ in each kernel. Setting the $\mathcal{M}$ best solutions as the modes in the Gaussian mixture model can make a sampling area in (2.3) biased toward this elite group of solutions. As the spread of solutions should be changed each period and can be greater in one of the kernels than the others, we let the bandwidth $\sigma_{[i]}^j$ be updated depending on the period j and the objective function values corresponding to the current $\mathcal{M}$ best solutions.

A specific choice of α_i and $\sigma_{[i]}^j$ in (2.3) and (2.4) is

$$\text{Case 1: } \alpha_i = \frac{1}{\mathcal{M}}, \ \sigma_{[i]}^j = \frac{\sum_{p=1}^{\mathcal{M}} r_{(p)}^j}{r_{(i)}^j} \tag{2.5}$$

assuming that $r(q) > 0$ for all $q \in \mathcal{Q}$. Some analytical and numerical convergence results can be shown for Case 1 in (2.5). In Case 1, the weights on the kernel densities are equal and the standard deviation of ith kernel density is inversely proportional to the corresponding objective value. Therefore, $\hat{f}$ is more concentrated on the promising areas. The standard deviation in (2.5) will not be changed if the objective function is multiplied by some scaling factor. The $\sigma_{[i]}^j$ will lie within the range of 1 and $1 + (\mathcal{M} - 1) \sup_{q_t \neq q_s} \frac{r(q_t)}{r(q_s)}$. There is only one parameter $\mathcal{M}$ that needs to be determined. Selecting a good set of parameters is another problem to solve in other existing algorithms. The algorithm is very simple. It can work effectively for well-scaled, low dimensional problems; note that σ depends on the objective values directly—for well-scaled function, σ lies in the reasonable range of values—and σ is always larger than 1—for high dimensional problems, $\sigma > 1$ may not help the search of the algorithm at an appropriate level of exploitation.

There are many ways of constructing $\hat{f}$ with different choices of $\mathcal{K}(\cdot)$, α_i, and $\sigma_{[i]}^j$. However, to perform effectively, some properties are desired: For example, (a) the standard deviation of each kernel density is inversely proportional to the correponding objective values, (b) the density spreads the candidate decisions around the modes so that it prevents local convergence, and (c) weights are updated in the way that gives more weight on the promising kernel functions. In Figure 1, we show a graphical representation of a multi-modal distribution given a set of

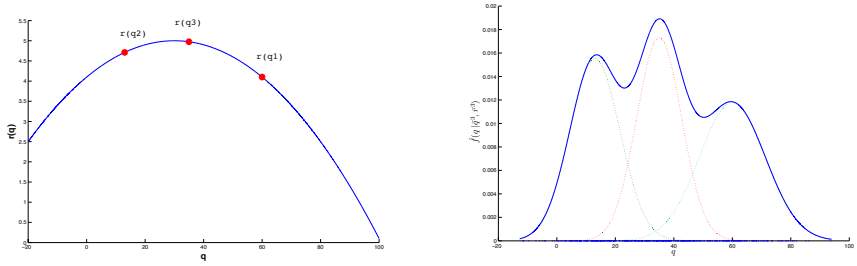

FIGURE 1. Observations $(q, r(q))$ (left) and kernel density estimates q vs. $\hat{f}(q|\bar{q}^3, \bar{r}^3)$ (right)

decision points and objective values. Consider the one dimensional optimization problem, $r(q) = -\frac{1}{1000}(q-30)^2 + 5$ where the maximum is at $q^* = 30$ and $\mathcal{Q} = \mathbb{R}$. Suppose that $\bar{q} = \left(q_{[1]}, q_{[2]}, q_{[3]}\right) = (13, 35, 60)$ is a set of current best decisions and $\bar{r} = \left(r_{(1)}, r_{(2)}, r_{(3)}\right) = (4.711, 4.975, 4.1)$ are the corresponding objective values for $\mathcal{M} = 3$. Figure 1 describes objective function values in the left and a sampling distribution $\hat{f}(q|\bar{q}, \bar{r})$ determined according to (2.3), (2.4), and (2.5). It can be seen from the figure that the sampling area for generating the next decision is more focused on the promising area around the current best decision.

For badly scaled and high-dimensional problems, we can consider a case where σ decays without depending on the objective values like Case 1. For high-dimensional problems, the decay of the variance in each Gaussian kernel will help the algorithm exploit the promising areas efficiently. One such case is as follows,

$$\text{Case 2: } \alpha_i = \frac{1}{\mathcal{M}}, \ \sigma^j_{[i]} = \frac{c}{\sqrt{\mathcal{M}}(\log j)^g} \tag{2.6}$$

where c and g are positive constants.

In Case 2, the weights on the kernel densities are equal and the standard deviation of ith kernel density decays as iteration proceeds. Larger values of $\mathcal{M}$ and c will help the search of the algorithm more scattered while larger values of g will help exploitative search concentrated on around the $\mathcal{M}$ best decisions.

We now extend a kernel density function to the multi-dimensional case by defining a multivariate Gaussian density function using the product kernel.

Definition 2.1. When $\mathcal{Q} = \mathbb{R}^n$, let $q = (q_1, q_2, \ldots, q_n)$, $q^j_{[i]} = (q^j_{[i]1}, q^j_{[i]2}, \ldots, q^j_{[i]n})$, and

$$\mathcal{K}\left(\frac{q - q_{[i]}}{\sigma^j_{[i]}}\right) = \prod_{l=1}^{n} \frac{1}{\sqrt{2\pi}\sigma^j_{[i]}} \exp\left(-\frac{1}{2}\left(\frac{q_l - q^j_{[i]l}}{\sigma^j_{[i]}}\right)^2\right).$$

For the multi-dimensional case, we use Definition 2.1 throughout the paper.

Our interest is to show global convergence of the algorithm, which will depend on the properties of $\hat{f}$. In the next section, we provide some results on convergence

for Algorithm 1 under a different set of assumptions where the global convergence properties can be easily established.

2.2. Results on convergence when $\mathcal{Q}$ is compact. To obtain analytical convergence results for Algorithm 1, we begin with a case where $\mathcal{Q}$ is compact in $\mathbb{R}^n$. We also make the following assumptions and definitions.

Definition 2.2. For $\delta > 0$, $B_\delta(q^*) = \{q \in \mathcal{Q} | \ \|q - q^*\| < \delta\}$ where $\|q - q^*\|$ denotes the Euclidean distance between q and q^*.

Assumption 2.3. *For any $\delta > 0$, $\mu(B_\delta(q^*)) > 0$ where $\mu(\cdot)$ is Lebesque measure.*

Assumption 2.4. $r : \mathcal{Q} \mapsto \mathbb{R}^+$, *i.e.,* $r(q) > 0$ *for* $\forall q \in \mathcal{Q}$.

Assumption 2.5. $r : \mathcal{Q} \mapsto \mathbb{R}$ *is a continuous or a bounded function.*

Assumption 2.6. *For all $\tilde{r} < r(q^*)$, there always exists $\epsilon > 0$ such that $B_\epsilon(q^*) \subseteq \{q : r(q) > \tilde{r}\}$.*

We make Assumption 2.4 without loss of generality since we can always introduce a strictly increasing function $T : \mathbb{R} \mapsto \mathbb{R}^+$, which can be used when the values of $r(\cdot)$ are negative. Assumption 2.6 quarantees q^* is in the interior of $\mathcal{Q}$. We re-define a truncated version of the multi-modal distribution on the compact set $\mathcal{Q}$.

Definition 2.7.

$$\hat{f}(q|\bar{q}^j, \bar{r}^j) = \sum_{i=1}^{\mathcal{M}} \alpha_i \ \frac{\mathcal{K}\left(\frac{q - q^j_{[i]}}{\sigma^j_{[i]}}\right)}{\int_{\mathcal{Q}} \mathcal{K}\left(\frac{q - q^j_{[i]}}{\sigma^j_{[i]}}\right) dq} \tag{2.7}$$

where $\sum_{i=1}^{\mathcal{M}} \alpha_i = 1$ and $\mathcal{K}(\cdot)$ is as in Definition 2.1.

In step 1 of Algorithm 1, q_{j+1} can be either sampled from $\hat{f}(\cdot|\bar{q}^j, \bar{r}^j)$ in (2.7) directly or a mixture distribution combined with the uniform distribution. Both cases give similar convergence results, which will be shown in Propositions 2.8 and 2.9, respectively.

Proposition 2.8. *Consider Algorithm 1 with Case 1 and a sampling distribution defined by Definition 2.7. Let $A_{j+1}|\mathcal{F}_j = \{||q_{j+1} - q^*|| < \epsilon|\mathcal{F}_j\}$. If Assumptions 2.3-2.4 are satisfied, $P(A_j \ i.o.) = 1$.*

Proof. Let $\mathcal{F}_j$ be a filtration, i.e., $\mathcal{F}_j = \sigma(q_1, \ldots, q_j)$ for $j \geq 1$. By the second Borel-Cantelli lemma II (in [11]), it suffices to prove $\sum_j P(A_j|\mathcal{F}_j) = \infty$.

$$\begin{aligned} P(A_{j+1}|\mathcal{F}_j) &= \int_{q \in B_\epsilon(q^*)} \hat{f}(q|\bar{q}^j, \bar{r}^j) dq \\ &\geqslant \sum_{i=1}^{\mathcal{M}} \frac{1}{\mathcal{M}} \int_{q \in B_\epsilon(q^*)} \mathcal{K}\left(\frac{q - q^j_{[i]}}{\sigma^j_{[i]}}\right) dq \end{aligned} \tag{2.8}$$

The inequality in (2.8) holds since $\int \mathcal{K}\left(\frac{q-q^j_{[i]}}{\sigma^j_{[i]}}\right)dq < 1$ for each i and j. There exists $r^\circ = \min_{q\in\mathcal{Q}} r(q)$ since $r(\cdot)$ is continuous or bounded on a compact set $\mathcal{Q}$. It is clear that $\inf_{i,j}\sigma^j_{[i]} \geqslant \frac{Mr^\circ}{r(q^*)}$ and $\sup_{i,j}\sigma^j_{[i]} \leqslant \frac{Mr(q^*)}{r^\circ}$. Let

$$\sigma^\circ = \frac{Mr^\circ}{r(q^*)},\ \sigma^\bullet = \frac{Mr(q^*)}{r^\circ} \tag{2.9}$$

and

$$\delta^*_\epsilon = \max_{\substack{i=1,\ldots,n\\ q_b\in B_\epsilon(q^*)\\ q\in\mathcal{Q}}} |e_i^T q_b - e_i^T q|$$

where $e_i = (0,0,\ldots,1,\ldots,0)^T$ $(\in \mathbb{R}^n)$. Since $\mathcal{Q}$ is compact, we know $\delta^*_\epsilon (< \infty)$ exists. Then, for any i, j, and l,

$$|q_l - q^j_{[i]l}| \quleq \delta^*_\epsilon \tag{2.10}$$

for $(q_1, q_2, \ldots, q_n) \in B_\epsilon(q^*)$ and $(q^j_{[i]1}, q^j_{[i]2}, \ldots, q^j_{[i]n}) \in \mathcal{Q}$. It follows from (2.9) and (2.10) that

$$\begin{aligned}\int_{q\in B_\epsilon(q^*)} \mathcal{K}\left(\frac{q-q^j_{[i]}}{\sigma^j_{[i]}}\right)dq &\geqslant \left(\frac{1}{\sqrt{2\pi}\sigma^\bullet}\right)^n \int_{q\in B_\epsilon(q^*)} e^{-(\delta^*_\epsilon/\sigma^\circ)^2 n}dq \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma^\bullet}\right)^n \mu(B_\epsilon(q^*))e^{-(\delta^*_\epsilon/\sigma^\circ)^2 n}. \end{aligned} \tag{2.11}$$

From (2.8) and (2.11),

$$P(A_{j+1}|\mathcal{F}_j) \geqslant \left(\frac{1}{\sqrt{2\pi}\sigma^\bullet}\right)^n \mu(B_\epsilon(q^*))e^{-(\delta^*_\epsilon/\sigma^\circ)^2 n} = C_1 > 0.$$

Because $\sum_{j=0}^{n-1} P(A_{j+1}|\mathcal{F}_j) \geqslant nC_1$, $\sum_{j=0}^{\infty} P(A_{j+1}|\mathcal{F}_j) \to \infty$.

□

The next proposition shows a similar result for the case of a mixture distribution combined with the uniform distribution and its proof is more straightforward.

Proposition 2.9. *Consider Algorithm 1 with a mixture distribution combined with the uniform distribution, $\bar{f}(q|\bar{q}^j, \bar{r}^j)$ defined as $\lambda\hat{f}(q|\bar{q}^j, \bar{r}^j) + (1-\lambda)f_0(q)$ where*

$$f_0(q) = \frac{1}{\mu(\mathcal{Q})}\mathbf{1}_{\{q\in\mathcal{Q}\}}$$

and $\mu(\cdot)$ is Lebesque measure. Let $q_{j+1} \sim \bar{f}(q|\bar{q}^j, \bar{r}^j)$ and $A_{j+1}|\mathcal{F}_j = \{||q_{j+1} - q^*|| < \epsilon|\mathcal{F}_j\}$. If Assumptions 2.3 is satisfied, for any $\epsilon > 0$, $P(A_j \text{ i.o.}) = 1$.

Proof. By the second Borel-Cantelli lemma II (in [11]), it suffices to prove $\sum_j P(A_j|\mathcal{F}_j) = \infty$. It follows from $P(A_{j+1}|\mathcal{F}_j) = \int_{q\in\mathcal{B}_\epsilon(q^*)} \bar{f}(q|\bar{q}^j, \bar{r}^j)dq$ that

$$(1-\lambda)\frac{1}{\mu(\mathcal{Q})}\mu(\mathcal{B}_\epsilon(q^*)) \leq P(A_{j+1}|\mathcal{F}_j).$$

Thus, we have $\sum_j P(A_j|\mathcal{F}_j) = \infty$. □

Theorem 2.10 shows that the objective function value corresponding to the best decision converges almost surely regardless of the choices of the distribution.

Theorem 2.10. *Consider Algorithm 1. $r(q^j_{[\mathcal{M}]})$ converges almost surely as $j \to \infty$.*

Proof. Let $q_{j+1} \sim \hat{f}(\cdot|\bar{q}^j, \bar{r}^j))$. It follows in Step 3 for all j that

$$E\left[r(q^{j+1}_{[\mathcal{M}]})|\mathcal{F}_j\right] = E\left[r(q_{j+1})\mathbf{1}_{\{r(q_{j+1})-r(q^j_{[\mathcal{M}]})>0\}} + r(q^j_{[\mathcal{M}]})\mathbf{1}_{\{r(q_{j+1})-r(q^j_{[\mathcal{M}]})\leq 0\}}|\mathcal{F}_j\right].$$

Therefore, $E[r(q^{j+1}_{[\mathcal{M}]})|\mathcal{F}_j] \geq r(q^j_{[\mathcal{M}]})$ for all j. Since $E[r(q^j_{[\mathcal{M}]})] \leq E[r(q^*)] < \infty$ for all j, $Y_j = r(q^j_{[\mathcal{M}]})$ is a submartingale with respect to $\mathcal{F}_j$. By the martingale convergence theorem [12], we know that Y_j converges almost surely. □

We use Theorem 2.10 for the following theorem.

Theorem 2.11. *Consider Algorithm 1 with a multi-modal distributon defined in Definition 2.7. Let $A_{j+1}|\mathcal{F}_j = \{||q_{j+1} - q^*|| < \epsilon|\mathcal{F}_j\}$. If Assumptions 2.3-2.6 are satisfied, then $P(|r(q^j_{[\mathcal{M}]}) - r(q^*)| > \epsilon \text{ i.o.}) = 0$ for each $\epsilon > 0$.*

Proof. By Theorem 2.10, there exists q° such that $P(|r(q^j_{[\mathcal{M}]}) - r(q^\circ)| > \epsilon \text{ i.o.}) = 0$ for each $\epsilon > 0$. Suppose that $q^\circ \neq q^*$. This implies that $r(q^\circ) < r(q^*)$ and $r(q^j_{[\mathcal{M}]}) \leqslant r(q^\circ)$ for $\forall j$ since q^* is unique and $r(q^j_{[\mathcal{M}]})$ is monotonically nondecreasing. Because

$$\begin{aligned}
&P(|r(q^j_{[\mathcal{M}]}) - r(q^\circ)| > \epsilon) \\
&\quad = P(|r(q^{j-1}_{[\mathcal{M}]}) - r(q^\circ)| > \epsilon | r(q_j) - r(q^{j-1}_{[\mathcal{M}]}) < 0) P(r(q_j) - r(q^{j-1}_{[\mathcal{M}]}) < 0) \\
&\qquad + P(|r(q_j) - r(q^\circ)| > \epsilon | r(q_j) - r(q^{j-1}_{[\mathcal{M}]}) \geq 0) P(r(q_j) - r(q^{j-1}_{[\mathcal{M}]}) \geq 0) \\
&\qquad\quad \geqslant \quad P(\{|r(q_j) - r(q^\circ)| > \epsilon\} \cap \{r(q_j) \geqslant r(q^{j-1}_{[\mathcal{M}]})\}) \\
&\qquad\quad \geqslant \quad P(\{r(q_j) - r(q^\circ) > \epsilon\} \cap \{r(q_j) \geqslant r(q^{j-1}_{[\mathcal{M}]})\}) \\
&\qquad\quad = \quad P(r(q_j) > r(q^\circ) + \epsilon),
\end{aligned}$$

we have

$$P(|r(q^j_{[\mathcal{M}]}) - r(q^\circ)| > \epsilon \text{ i.o.}) \geqslant P(r(q_j) > r(q^\circ) + \epsilon \text{ i.o.}). \tag{2.12}$$

It follows from Assumption 2.6 that there always exists $\tilde{\epsilon}$ such that

$$\{r(q_j) > r(q^\circ) + \epsilon\} \supseteq B_{\tilde{\epsilon}}(q^*) \tag{2.13}$$

for any ϵ satisfying $r(q^\circ) + \epsilon < r(q^*)$. From (2.12) and (2.13), we have

$$P(|r(q^j_{[\mathcal{M}]}) - r(q^\circ)| > \epsilon \text{ i.o.}) \geqslant P(r(q_j) > r(q^\circ) + \epsilon \text{ i.o.}) \geqslant P(q_j \in B_{\tilde{\epsilon}}(q^*) \text{ i.o.}) \tag{2.14}$$

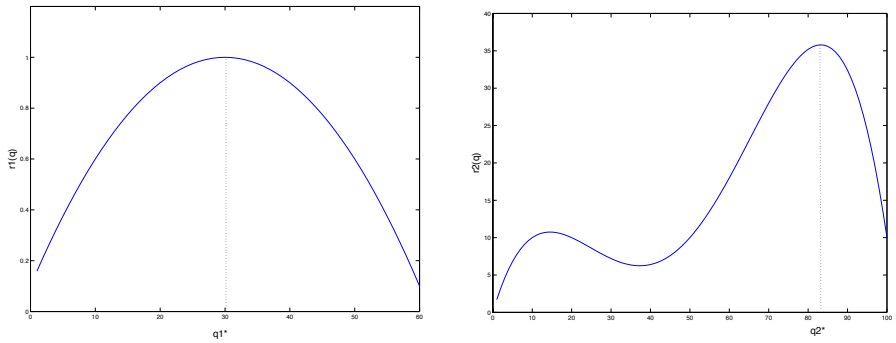

FIGURE 2. $r_1(q)$ (left) and $r_2(q)$ (right)

for any ϵ satisfying $r(q^\circ) + \epsilon < r(q^*)$. We know the right hand side in (2.14) =1 by Proposition 2.8, which contradicts the supposition. Thus, q° is equal to q^*. □

Remark 2.12. We can extend the solution space $\mathcal{Q}$ to the unbounded space under some assumption: There exists a compact set $\mathcal{C}$ and a positive constant d such that $q_0 \in \mathcal{C} \cap \mathcal{Q}$ and $r(q) \leq r(q_0) + d$ for any $q \in \mathcal{C}^c \cap \mathcal{Q}$. It is reasonable to assume that all solutions beyond some (unknown) distance from the initial setting are inferior. Since the current best decision $q^j_{[\mathcal{M}]}$ will never visit the inferior region under this assumption, we have the same convergence results.

3. Numerical experiments

In this section, we demonstrate the performance of the algorithm for continuous optimization problems. We first show the numerical results on two one-dimensional problems that are obtained from the algorithm with Case 1. For Case 2, we use 10 benchmark problems that have often been used for testing in global optimization.

3.1. Algorithm with Case 1. We use two one-dimensional functions, $r_1(q) = -\frac{1}{1000}(q-30)^2 + 1$ and $r_2(q) = -\frac{1}{100000}(q-10)(q-20)(q-50)(q-100) - 1$ where $q_1^* = 30$ and $q_2^* = 83.2165$ are optimal solutions, respectively (Figure 2). Figure 3 shows the performance of the algorithm on $r_1(q)$ for $\mathcal{M} = 1$ and $\mathcal{M} = 3$. Figure 4 shows the performance of the algorithm on $r_2(q)$ for $\mathcal{M} = 1$ and $\mathcal{M} = 5$. The algorithm converges faster with higher values of $\mathcal{M}$ as the search becomes more explorative with larger values of $\mathcal{M}$. It is possible that the algorithm stays around $q^\circ = 14.5065$ (local maxima) for a long time for smaller values of $\mathcal{M}$ on $r_2(q)$ depending on the initial decision point. The algorithm with Case 1 can be applied to well-scaled low-dimensional problems.

3.2. Algorithm with Case 2. The following benchmark problems with various dimensions (n) are used to test the algorithm with Case 2. Functions r_1–r_3 are low-dimensional problems and functions r_4–r_{10} are high-dimensional problems varying from $n = 20$ to $n = 100$.

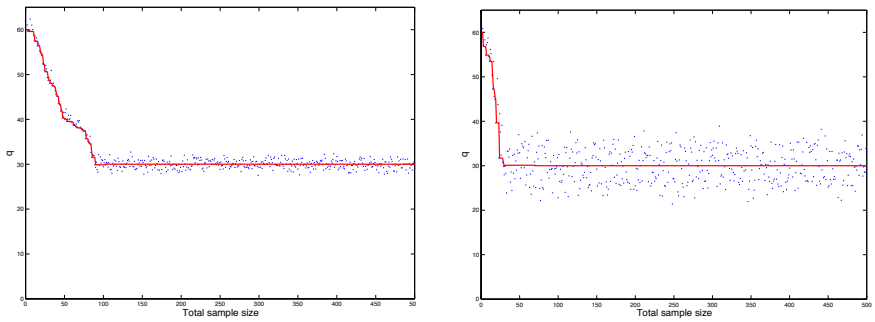

FIGURE 3. Iterative decisions q_j (blue) and the best rank decisions $q^j_{[\mathcal{M}]}$ (red) when $\mathcal{M} = 1$ (left) vs. $\mathcal{M} = 3$ (right), on r_1

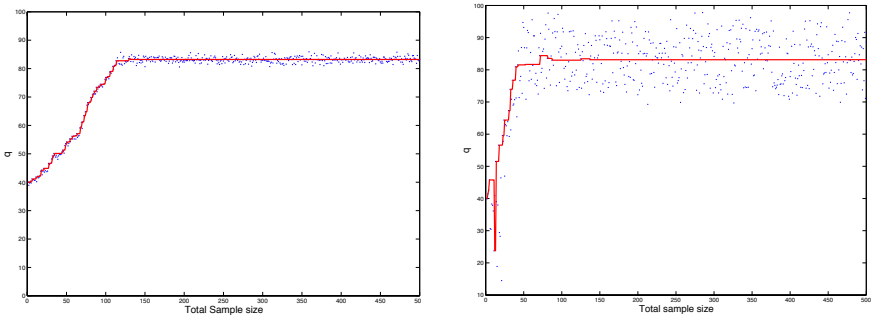

FIGURE 4. Iterative decisions q_j (blue) and the best rank decisions $q^j_{[\mathcal{M}]}$ (red) when $\mathcal{M} = 1$ (left) vs. $\mathcal{M} = 5$ (right), on r_2

(1) Dejong's 5th function ($n = 2$)

$$r_1(q) = \left[0.002 + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (q_i - a_{j,i})^6}\right]^{-1}$$

where

$$a_{j,1} = \{-32, -16, 0, 16, 32, -32, -16, 0, 16, 32, -32, -16, 0, \\ 16, 32, -32, 16, 0, 16, 32, -32, -16, 0, 16, 32, \},$$

$$a_{j,2} = \{-32, -32, -32, -32, -32, -16, -16, -16, -16, -16, 0, 0, 0, 0, 0, \\ 16, 16, 16, 16, 16, 32, 32, 32, 32, 32\},$$

$q^* = (-32, -32)^T$, $r_1(q^*) \approx 0.998$, $\mathcal{Q} = [-50, 50]^n$.

(2) Shekel's function ($n = 4$)

$$r_2(q) = -\sum_{i=1}^{5}((q-a_i)^T(q-a_i)+c_i)^{-1}$$

where $a_1 = (4,4,4,4)^T, a_2 = (1,1,1,1)^T, a_3 = (8,8,8,8)^T, a_4 = (6,6,6,6)^T$, $a_5 = (3,7,3,7)^T$, $c = (0.1, 0.2, 0.2, 0.4, 0.4)$, $q^* = (4,\ldots,4)^T$, $r_2(q^*) = -10.153$, $\mathcal{Q} = [0,10]^n$.

(3) Hartmann function ($n = 6$)

$$r_3(q) = -\sum_{i=1}^{4} a_i \exp\left(-\sum_{j=1}^{6} B_{i,j}(q_j - C_{i,j})^2\right)$$

where $a_i = (1, 1.2, 3, 3.2)^T$,

$$B = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix},$$

$$C = \begin{pmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{pmatrix},$$

$q^* = (0.20169, 0.150011, 0.476874, 0.275332, 0.311652, 0.6573)^T$, $r_3(q^*) = -3.32237$, $\mathcal{Q} = [0,1]^n$.

(4) Rosenbrock function ($n = 20$)

$$r_4(q) = \sum_{i=1}^{n-1} 100(q_{i+1} - q_i^2)^2 + (q_i - 1)^2$$

where $q^* = (1,\ldots,1)^T$, $r_4(q^*) = 0$, $\mathcal{Q} = [-5,5]^n$.

(5) Pintér's function ($n = 20$)

$$\begin{aligned} r_5(q) \quad = \quad & \sum_{i=1}^{n} i q_i^2 + \sum_{i=1}^{n} 20i \sin^2(q_{i-1}\sin q_i - q_i + \sin q_{i+1}) \\ & + \sum_{i=1}^{n} i \log_{10}(1 + i(q_{i-1}^2 - 2q_i + 3q_{i+1} - \cos q_i + 1)^2) \end{aligned}$$

where $q_0 = q_n$, $q_{n+1} = q_1$, $q^* = (0,\ldots,0)^T$, $r_5(q^*) = 0$, $\mathcal{Q} = [-5,5]^n$.

(6) Trigonometric function ($n = 20$)

$$r_6(q) = 1 + \sum_{i=1}^{n} 8\sin^2(7(q_i-9)^2) + 6\sin^2(14(q_i-9)^2) + (q_i-9)^2$$

where $q^* = (0.9,\ldots,0.9)^T$, $r_6(q^*) = 1$, $\mathcal{Q} = [-3,3]^n$.

(7) Griewank function ($n = 20, 100$)

$$r_7(q) = \frac{1}{4000}\sum_{i=1}^{n} q_i^2 - \prod_{i=1}^{n}\cos\left(\frac{q_i}{\sqrt{i}}\right) + 1$$

where $q^* = (0, \ldots, 0)^T$, $r_7(q^*) = 0$, $\mathcal{Q} = [-10, 10]^n$.

(8) Rastrigin function ($n = 20$)

$$r_8(q) = An + \sum_{i=1}^{n}[q_i^2 - A\cos(2\pi q_i)]$$

where $q^* = (0, \ldots, 0)^T$, $r_8(q^*) = 0$, $\mathcal{Q} = [-5.12, 5.12]^n$.

(9) Sinusoidal function ($n = 30$)

$$r_9(q) = \sum_{i=1}^{n} 2.5\prod_{i=1}^{n}\sin\left(\frac{\pi q_i}{180}\right) + \sum_{i=1}^{n}\sin\left(\frac{\pi q_i}{36}\right) - 3.5$$

where $q^* = (90, \ldots, 90)^T$, $r_9(q^*) = 0$, $\mathcal{Q} = [0, 180]^n$.

(10) Zakharov function ($n = 20$)

$$r_{10}(q) = \sum_{i=1}^{n} x_i^2 + \left(\sum_{i=1}^{n} 0.5iq_i\right)^2 + \left(\sum_{i=1}^{n} 0.5iq_i\right)^4$$

where $q^* = (0, \ldots, 0)^T$, $r_{10}(q^*) = 0$, $\mathcal{Q} = [-5, 10]^n$.

When the test function is a minimization problem, we convert the problem to $\max -r_i(q)$. In the numerical experiments, we used scaled versions of test functions when needed. For example, $r_i(c\tilde{q})$, $-1 \leqslant \tilde{q} \leqslant 1$ is solved instead of $r_i(q)$, $-c \leqslant q \leqslant c$. We have tried different sets of parameters and found that the performance of the algorithm depends on the values of $\mathcal{M}$, c, and g especially for high-dimensional problems.

Table 1 presents the averaged performance of the algorithm with a different set of parameters c, g, and $\mathcal{M}$ based on 10 independent replications for each problem. In Table 1, $\bar{r}_{\mathcal{M}}$ is the averaged objective function value at the best solution and *se* is its standard error. The numerical simulation results corresponding to $\bar{r}^*_{\mathcal{M}}$ in the table are presented graphically in Figures 5 and 6.

For low-dimensional problems, a relatively small number of function evaluations are needed to find the global optimum. r_1 is a two-dimensional function with 25 local minima. Functions r_2 and r_3 have 5 and 6 local minima, respectively. For r_1, the algorithm converges after around 6000 function evaluations for all runs. For r_2, the algorithm find values bigger than 9 in 9 replications out of 10 but the results are not as good as that of r_1 and r_3. For r_3, the algorithm gives robust results for all values of parameters and Figure 5 shows the algorithm converges within around 2000 function evaluations.

For r_4, the global optimum is inside a long, narrow, parabolic shaped flat valley. It is known that finding the valley is trivial, however convergence to the global optimum is difficult. It is also known that most gradient methods fail to minimize r_4 since the two successive gradients are opposite to each other. Due to this unique structure of the function, unlike other multimodal test functions, the larger values of $\mathcal{M}$ does not help convergence on the high-dimensional setting of r_4.

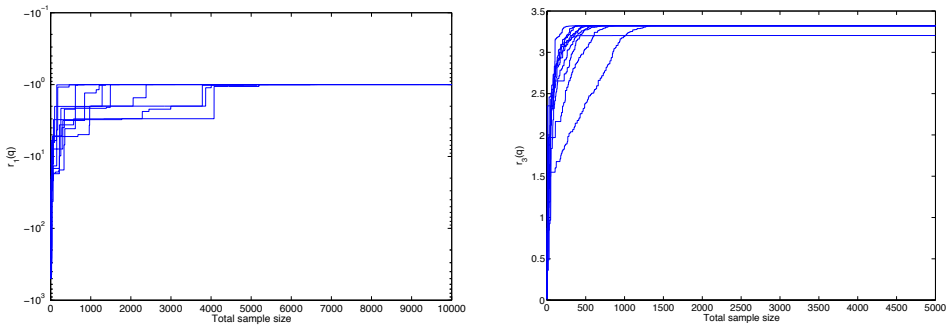

FIGURE 5. 10 independent runs on r_1 (left) and r_3 (right)

Function r_5 is a badly scaled and highly multimodal function. In our preliminary experiments, we found that a small values of $\mathcal{M}$ does not maintain the search explorative for high-dimensional setting. For $n = 20$, $\mathcal{M} = 300$, the algorithm found near-optimal solutions for some parameter settings. The performance seems to be more sensitive to the change of parameters than other multimodal function because the function is badly scaled. The algorithm with $\mathcal{M} = 300, c = 10, g = 3$ found near-optimal solution in all replications (out of 10) within 300,000 function evaluations. The algorithm with $\mathcal{M} = 300, c = 0.1, g = 1$ found near-optimal solutions in 9 replications (out of 10) within 300,000 function evaluations; $c = 10, g = 3$ and $c = 0.1, g = 1$ seem to work well with other highly multimodal functions with $\mathcal{M}$ smaller than or around 300.

Function r_6 is a highly multimodal function. Similarly with r_5, the algorithm with $\mathcal{M} = 300, c = 10, g = 3$ found near-optimal solution in 9 replications (out of 10) within 300,000 function evaluations. The algorithm with $\mathcal{M} = 300, c = 0.1, g = 1$ found near-optimal solutions in 8 replications (out of 10) within 300,000 function evaluations.

Functions r_7, r_8, and r_9 are also highly multimodal functions. They have many widespread local minima/maxima and the location of the minima/maxima are regularly distributed. For r_7, with various settings of parameters, the algorithm found near optimal solutions for all runs. The algorithm performs well even when $n = 100$ for larger values of $\mathcal{M}$. For r_8, the parameter setting $c = 10, g = 3$ work better than other settings that we tried. When $\mathcal{M} = 300$ is changed to $\mathcal{M} = 700$, the solution is slightly improved from $\bar{r}_{\mathcal{M}} = -0.797$ to $\bar{r}_{\mathcal{M}} = -1.127$ ($r(q^*) = 0$). For r_9, the parameter setting $c = 1, g = 1$ work well for $\mathcal{M} = 100$, $\mathcal{M} = 10$, and $\mathcal{M} = 1$. Near optimal solutions were found for all runs. For r_{10}, there are no local minima except the global one. Smaller values of $\mathcal{M}$ do not help convergence to the optimal point and gives robust results for all parameters.

From our numerical experiments, we have found various parameter settings that the algorithm performs reasonably well. Small values of $\mathcal{M}$ will often be preferred for low-dimensional problems. Larger values of $\mathcal{M}$ are appropriate for high-dimensional multimodal problems as many candidate solutions will be kept,

and hence, the search becomes more explorative. But $\mathcal{M}$ (i.e., the number of kernel functions in the mixture model) will not affect the computational complexity in the implementation of the algorithm since a candidate solution will be generated from a randomly selected Gaussian kernel density among $\mathcal{M}$ Gaussian kernel densities.

Func. $(r(q^*))$	n	N	$\mathcal{M}$	c	g	$\bar{r}_{(\mathcal{M})}$	se
r_1 (-0.998)	2	10K	10	3	1	-0.998^*	4.424e-6
	2	10K	10	1	1	-1.396	0.162
r_2 (10.153)	4	40K	10	0.1	1	8.900	0.854
	4	100K	10	0.1	1	9.648	0.505
r_3 (3.32237)	6	5K	10	0.1	1	3.310^*	$1.192e-2$
	6	5K	100	0.1	1	3.322	$1.296e-4$
r_4 (0)	20	400K	1	0.01	2.1	$-6.876e-2^*$	$4.105e-2$
	20	300K	1	0.01	1	-1.389	0.597
r_5 (0)	20	300K	300	10	3	$-9.972e-2^*$	$1.446e-2$
	20	300K	300	0.1	1	-0.178	$3.043e-2$
r_6 (0)	20	300K	300	0.1	1	-1.000	$4.761e-7$
	20	300K	300	10	3	-1.045^*	$4.486e-2$
r_7 (0)	20	300K	10	0.1	1	$-2.802e-2^*$	$1.825e-2$
	100	300K	300	0.1	1	$-6.509e-5$	$1.562e-6$
r_8 (0)	20	300K	700	10	3	-0.797^*	0.289
	20	300K	300	10	3	-1.127	0.343
r_9 (0)	30	100K	10	1	1	$-3.909e-5$	$1.515e-5$
	30	100K	100	1	1	-0.140	$2.679e-3$
r_{10} (0)	20	300K	1	1	3	$-1.838e-6$	$6.960e-8$
	20	300K	1	1	1	$-4.275e-2^*$	$1.223e-3$

TABLE 1. Average performance on r_1-r_{10}

4. Conclusion

In this paper, we proposed a search algorithm to solve deterministic optimization problems. We presented the convergence properties of AGM algorithms. In our numerical experiments, the performance of the algorithm is presented on multidimensional optimization problems. In our future work, we intend to investigate the performance of AGM algorithm on discrete optimization problems. The structure of Gaussian mixture model needs to be investigated; for example, its relationship with the topology of the problems and an initial set of bandwidths or the number of kernel functions which give robust results. It is also worthwhile to investigate a way that the number of kernel functions and the number of samples generated each period can be adaptively changed.

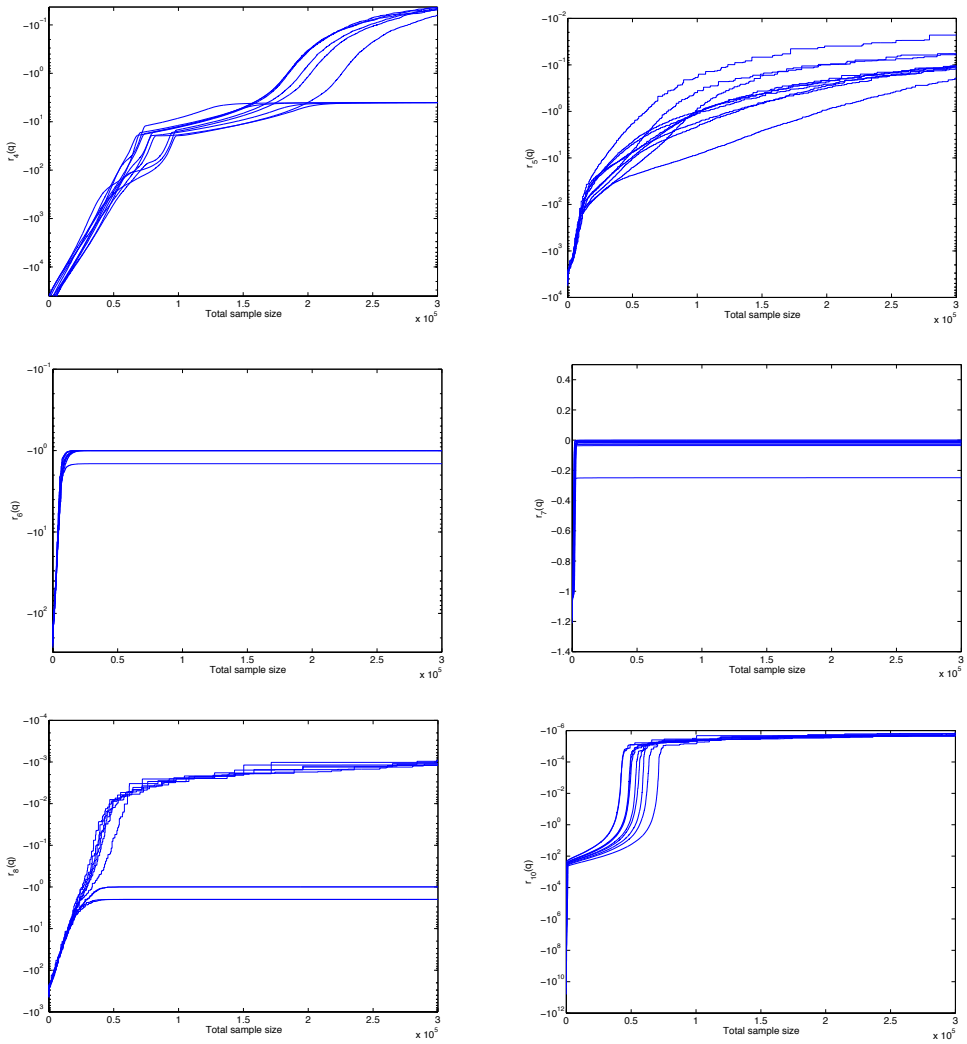

FIGURE 6. 10 independent runs of AGM on r_4–r_8, r_{10}

Acknowledgment. This research was supported by Hankuk University of Foreign Studies Research Fund.

References

1. Gross, L.: Abstract Wiener spaces, in: *Proc. 5th Berkeley Symp. Math. Stat. and Probab.* **2**, part 1 (1965) 31–42, University of California Press, Berkeley.
2. Zlochin, Mark, Mauro Birattari, Nicolas Meuleau, and Marco Dorigo: Model-Based Search for Combinatorial Optimization: A Critical Survey, *Annals of Operations Research* **131(1)** (2004) 373–395.

3. Holland, J.H.: Adaptation in Natural and Artificial Systems, The Michigan University Press, 1975.
4. Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni: The Ant System: Optimization by a colony of cooperating agents, *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART B* **26(1)** (1996) 29–41.
5. Reuven Y. Rubinstein and Dirk P. Kroese: The Cross Entropy Method: A Unified Approach To Combinatorial Optimization, Monte-carlo Simulation (Information Science and Statistics),Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2004.
6. Kirkpatrick, S., C.D. Gelatt, and M.P. Vecchi: Optimization by simulated annealing, *Science* **220** (1983) 671–680.
7. Mühlenbein, H. and G. Paaß,G: From recombination of genes to the estimation of distributions I. Binary parameters, *In Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature-PPSN IV* (1996) 178–187.
8. Hu, Jiaqiao and Ping Hu: Annealing adaptive search, cross-entropy, and stochastic approximation in global optimization, *Naval Research Logistics (NRL)* **58(5)** (2011) 457–477.
9. Hu, Jiaqiao, Michael C.Fu, and Steven I. Marcus: A model reference adaptive search method for global optimization, *Operations Research* **55(3)** (2007) 549–568.
10. Hu, Jiaqiao, Michael C. Fu, and Steven I. Marcus: Stochastic optimization using model reference adaptive search, *Proceedings of the 2005 Winter Simulation Conference* (2005) 811-818.
11. Durrett, Richard: Probability : Theory and Examples, Duxbury Press, 1995.
12. Billingsley, Patrick: Probability and Measure, Wiley-Interscience, New York, NY, 1995.

Soonhui Lee: College of Business, Hankuk University of Foreign Studies, 107, Imunro, Dongdaemun-gu, Seoul, 02450, Republic of Korea

E-mail address: shlee2016@hufs.ac.kr, soonhui.lee@gmail.com