Research on the Importance Sampling Method for Evolutionary Algorithms Based on Probability Models

Takayuki Higo

Abstract

Evolutionary algorithms based on probability models (EAPM) are algorithms inspired by biological systems. The essential mechanism of EAPM consists of both statistical estimation and Monte Carlo integration. By using the two techniques, EAPM estimate the distribution of promising solutions and generate samples from it.

This thesis improves the basic framework of EAPM in three directions. First, this thesis proposes a technique for reusing the historical samples. The difficulty of employing the historical samples is that simply selecting good historical samples causes the bias of the statistical estimation. The proposed method weights historical samples in terms of importance sampling for possibly and theoretically removing the bias. Second, this thesis focuses on the convergence mechanism. In general EAPM, highly random sampling is employed in early stages, and the randomness is gradually decreased. Finally, the sampler distribution converges a point. This mechanism involves the problem of local optima. To overcome this problem, this thesis proposes to mix samples with different randomness. In other words, highly random sampling, slightly random sampling, and converged sampling are carried out simultaneously and iteratively. The point is that highly random samples can provide opportunities to escape from local optima. However, the difficulty is to retrieve information from mixed samples. In the proposed method, importance sampling with a mixture distribution plays an important role which provides theoretical validity for retrieving information from mixed samples. Third, this thesis proposes a novel convergence schedule. In EAPM, it is important not only to control the convergence speed but also to determine its speed. Actually, there is no convergence schedule with theoretical discussions. This thesis reveals the relationship between the randomness of the target distribution (i.e., the ideal sampler distribution which guides where samples should be generated) and the accuracy of the statistical estimation, that is, the entropy of the target distribution and the Fisher information. As a result, we obtain an approximately optimal convergence schedule, where the entropy of the target distribution is linearly reduced. This implies that the algorithm converges in linear time for a problem with an exponential size of the search space.

Consequently, this thesis theoretically and mathematically extends the basic framework of EAPM in new two directions: (1)mixing current and historical samples, and (2)mixing samples with different randomness. On the other hand, the proposed convergence schedule is an essential element of EAPM and the overall improvement can be expected. Through experiments with discrete problems and continuous problems, the effectiveness of each improvement is confirmed and the salient features are revealed: (1) employing historical samples overcomes the instability in statistical estimation, (2) mixing randomness overcomes the problem of local optima, and (3) the proposed convergence schedule is a practical method.

Acknowledgments

First, I would like to thank my supervisor, associate Prof. Keiki Takadama for a lot of his support. In 2003, I was inspired to start this research by his work, an optimization method using reinforcement learning. I also would like to thank Prof. Shigenobu Kobayashi, Prof. Katumi Nitta, Prof. Takao Terano, Prof. Yoshiyuki Kabashima, and assosicate Prof. Isao Ono for reviewing this thesis and providing comments to make it meaningful. This work was also supported by a Grant-in-Aid for JSPS Fellows (54103) and the 21st Century COE Program "Creation of Agent-Based Social System Sciences," and I would like to thank the leader, Prof. Hiroshi Deguchi.

I wish to thank the members of Takadama laboratory, Nitta laboratory, Watanabe laboratory, Kabashima laboratory, the work group for next generation algorithms, etc. They have encouraged and inspired me to proceed my research through discussions on science, engineering, human life and so on. Especially, Prof. Sumio Watanabe, who is my supervisor in my Bachelor course, associate Prof. Yukito Iba, Dr. Kazuho Watanabe, Takashi Sinzato and Masashi Sekino, have had a strong impact on my attitude towards this research and I appreciate them. Finally, I wish to thank my parents, Toshiaki and Toshiko, and my brother, Tomoyuki, for their support. Needless to say, my friends are also essential to enjoy proceeding this study.

Takayuki Higo

Contents

Chapter 1

Introduction

1.1 Background

Nowadays, it has become easy to obtain enormous computational resources, and one kind of the most interesting challenges is to apply computational techniques towards complex problems. The computational optimization is an effective and essential technique for dealing with complex problems. There are many optimization methods such as branch and bound methods, gradient methods, Newton methods, simulated annealing methods, and mean field annealing methods.

Evolutionary algorithms (EAs) such as genetic algorithms (GAs) [9, 44] are comparatively new optimization techniques. Since they have a relatively short history, EAs have not been widely employed and the fundamental principle have not been established yet. However, substantial studies on EAs have been carried out energetically and they have steadily contributed to reveal the essential mechanism of EAs. Consequently, we have one promising approach in sight now.

Actually, this approach have not possessed the official name and, in fact, the official definition have not been established yet. At least, it has been well confirmed that the essential key technique is statistical estimation. In general, estimation of the cost function is important task in computational optimization. For example, the Newton method predicts the cost function by using Taylor expansion. On the other hand, in the focused approach, statistical estimation predicts the structure of the cost function instead of Taylor expansion. This is clearly new approach.

This approach attracted attentions for the first time when the works of Mühlenbein [27, 29] appeared. It should be noted that there were similar studies such as [1] before them. Methods inspired by this approach are called estimation of distribution algorithms (EDAs). Actually, EDAs are intended to be a mathematical model of GAs and someone call methods based on this approach probabilistic model-building genetic algorithms (PMBGAs) [32]. Surprisingly, in a different field, that is, rare event simulation, a similar approach is proposed by Rubinstein as one of the Monte Carlo integration techniques and named the cross entropy method (CE) [40]. Currently, it has become well known that the three different names share the common and essential concept, that is, statistical estimation of the distribution of promising solutions, and the common name, evolutionary algorithms based on probability models (EAPM), is proposed in the congress on evolutionary computation (CEC) in 2007. This thesis uses this name.

One advantage of EAPM to other EAs is the mathematical definition of the algorithms. In EAPM, an objective is statistically estimation of promising solutions from the past samples. This problem setting can be defined mathematically and we can develop EAPM in theoretical manners. Recent studies on EAPM [20] mainly improve the statistical estimation, for example, employing complex probability models such as Bayesian networks.

1.2 Objectives and Contributions

The essential mechanism of EAPM consists of two techniques: statistical estimation and additionally Monte Carlo integration (MCI). Statistical estimation plays the central role in EAPM. On the other hand, MCI has an effect on the estimation accuracy. This thesis focuses not on the aspect of statistical estimation but on the aspect of Monte Carlo integration, whereas almost studies on EAPM focuses on statistical estimation methods.

This thesis improves the Monte Carlo integration of EAPM in three directions. First, this thesis proposes a novel technique, resampling population model (RPM), for reusing the historical samples. The difficulty of employing the historical samples is that simply selecting good historical samples causes the bias of the statistical estimation. RPM weights the historical samples in terms of importance sampling for possibly removing the bias.

Second, this thesis focuses on the convergence mechanism. In general EAPM, highly random sampling is employed in early stages, and the randomness is gradually decreased. Finally, the sampler distribution converges a point. This mechanism involves the problem of local optima because there is no opportunity to escape from local optima after convergence. To overcome this problem, this thesis proposes a novel method, hierarchical importance sampling (HIS) which mixes samples with different randomness. In other words, highly random sampling, slightly random sampling, and converged sampling are carried out simultaneously. Higher randomness contributes to escape from local optima and, in this method, high randomness is employed at all stages. The difficulty is to retrieve information from mixed samples. HIS overcomes this difficulty by using importance sampling with a mixture distribution. By using this technique, information can be retrieved from mixed samples in a theoretically valid manner. This method is also related to multi-start method.

Third, this thesis proposes a novel convergence schedule. In EAPM, it is important not only to control the convergence speed but also to determine its speed. Actually, there is no convergence schedule with theoretical discussions. This thesis reveals the relationship between the randomness of the target distribution (i.e., the ideal sampler distribution which guides where samples should be generated) and the accuracy of the statistical estimation, that is, the entropy of the target distribution and the Fisher information. As a result, we obtain an approximately optimal convergence schedule , entropy reduction schedule (ERS), where the entropy of the target distribution is linearly reduced. This implies that the algorithm converges in linear time for a problem with an exponential size of the search space.

Consequently, this thesis extends the basic framework of EAPM in new two directions: (1)mixing current and historical samples, and (2)mixing samples with different randomness. On the other hand, the convergence schedule is an essential element of EAPM and the overall improvement can be expected. The brief relationship is shown in Fig. 1.1. The aim of this thesis is to confirm the effectiveness of each improvement through experiments and additionally, with advanced benchmark problems such as Rosenbrock and Rastrigin function, to reveal comprehensive properties of RPM and HIS: (1) RPM has the robustness against the instability of the statistical estimation and (2) HIS has the robustness against local optima.

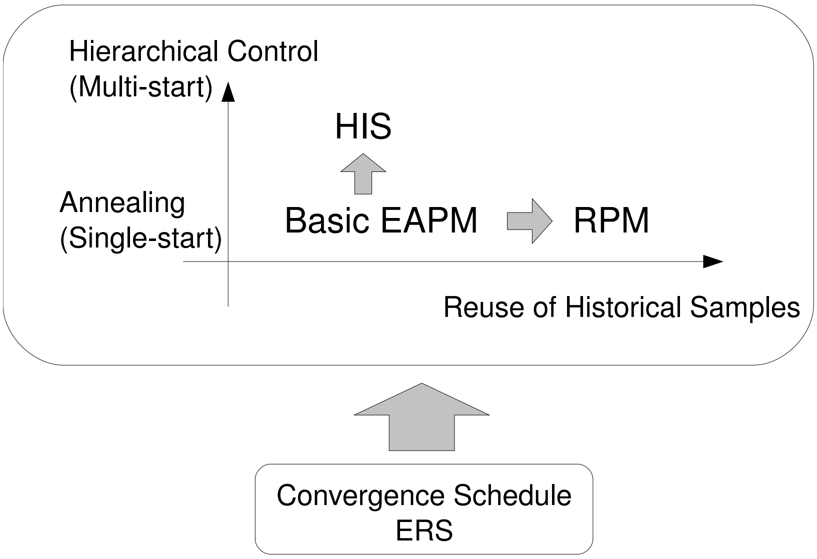


Figure 1.1: Relationship among the proposed methods.

1.3 Outline

This thesis is organized as follows: Chapter 2 introduces basic knowledge such as computational optimization, Monte Carlo integration and statistical estimation, and Chapter 3 explains the basics of EAPM. Chapter 4 describes a method for using the historical samples. Chapter 5 describes a hierarchically control of the randomness of generated samples. Chapter 6 describes a novel convergence schedule and its theoretical aspect. Chapter 7 conducts advanced experiments to investigate comprehensive properties of RPM and HIS through discrete and continuous problems. Chapter 8 summarizes the results and concludes this thesis.

Chapter 2

Preliminaries

This chapter introduces the essential knowledges for understanding EAPM.

2.1 Computational Optimization

The objective of the optimization problem is to find the solution x which minimizes or maximizes the cost function $f(x)$ (also called the objective function) under the constraint $x \in \mathrm{S}$. In this thesis, x is supposed to be a continuous or discrete vector. This thesis considers only minimization problems without loss of generality and does not consider the difficulty of the constraints.

As we know, in some simple problems, the optimum solution can be easily found by hand calculation. For example, it is clearly easy to minimize $f(x) = ax^2 + bx + c$, where $a, b, c, x \in \mathrm{R}$ and $0 < a$. This easiness comes from some basic properties of the cost function, for example, convexity, continuity, and differentiability.

In other cases, for example, where the derivative cannot be obtained or convexity is not guaranteed, optimization becomes difficult. For these types of optimization problems, computational methods are effective. The simplest method is the hill-climbing method (also called the local search), where a sequence of solutions, $x_1, x_2, \cdots, x_n$, is generated such that $f(x_1) > f(x_2) > \cdots > f(x_n)$. In the algorithm, first, the initial solution x_1 is generated randomly and is updated iteratively. In update steps, small difference Δx is somehow generated, for example, by using gradient, and if $f(x_t{+}\Delta x) < f(x_t)$ then the current solution is basically updated as $x_{t+1} = x_t + \Delta x$.

In hill-climbing methods, the presence of local optima is the serious problem. The local optima is defined as follows:

$$\{x^* | f(x^*) < f(x),\ \forall x \ s.t.\ |x^* - x| < |\epsilon|\ \}, \tag{2.1}$$

where $|\epsilon|$ is a small value. Clearly, hill-climbing methods can find a local optimum, but there is no guarantee that the obtained solution is the global optimum.

2.2 Monte Carlo Integration (MCI)

2.2.1 Basics of MCI

For an arbitrary function $g(x)$ and an arbitrary probability distribution $q(x)$, MCI can approximately calculate

$$I = \int q(x) g(x) \, dx \tag{2.2}$$

as follows:

$$\hat{I} \ = \ \frac{1}{M} \sum_{q(x)} g(x_i), \tag{2.3}$$

where $\sum_{q(x)}$ denotes summation over the samples generated from $q(x)$, and M is the number of the samples. Especially, if $M \to \infty$ then $\hat{I} \to I$. This is well known fact as the law of large numbers.

The speed to approach to the true value is important. This can be asymptotically discussed by the central limit theorem [22], which says that

$$\lim_{M \to \infty} \sqrt{M}(\hat{I} - I) \to \mathbf{N}(0, \sigma^2), \text{ in distribution,} \tag{2.4}$$

where $\mathbf{N}(u, \sigma^2)$ is a Gaussian distribution with the mean u and the variance σ^2; $\sigma^2 = \mathrm{Var}[g(x)]_{q(x)}$ and $\mathrm{Var}[\cdot]_{q(x)}$ denotes the variance of the random variable with respect to $q(x)$. In other words, the average squared error $E[(\hat{I} - I)^2]$ with M number of samples is given by

$$E[(\hat{I} - I)^2] = \frac{\sigma^2}{M}, \tag{2.5}$$

where E[$\cdot$] denotes the expectation.

2.2.2 Importance Sampling

In general, it can be difficult to directly generate samples from the probability distribution of interest and instead we have samples from another probability distribution. In this case, importance sampling is useful. In importance sampling, I of (2.2) is redefined as follows:

$$I = \int p(x)\frac{q(x)}{p(x)}g(x)\,dx, \tag{2.6}$$

where $p(x)$ is the sampler distribution. MCI is carried out as

$$\hat{I}_{\mathrm{IS}} = \frac{1}{M}\sum_{p(x)}\frac{q(x)}{p(x)}g(x_i). \tag{2.7}$$

Another application of importance sampling is shown as follows:

$$\begin{aligned}\int g(x)\,dx &= \int p(x)\frac{g(x)}{p(x)}\,dx &(2.8)\\ &\simeq \frac{1}{M}\sum_{p(x)}\frac{g(x_i)}{p(x)}. &(2.9)\end{aligned}$$

Also in these cases, the error can be assessed by the same way as (2.5) with $\sigma^2 = \mathrm{Var}[\frac{q(x)}{p(x)}f(x)]_{p(x)}$. The point is that the sampler distribution $p(x)$ has an effect on the error. In other words, the error can be controlled by changing $p(x)$.

In practice, normalized importance sampling is useful. Normalized importance sampling estimator is given by

$$\begin{aligned}\hat{I}_{\mathrm{NIS}} &= \frac{1}{\sum_{p(x)}\frac{q(x)}{p(x)}}\sum_{p(x)}\frac{q(x)}{p(x)}g(x_i) &(2.10)\\ &= \frac{1}{\sum_{p(x)}\frac{\tilde{q}(x)}{\tilde{p}(x)}}\sum_{p(x)}\frac{\tilde{q}(x)}{\tilde{p}(x)}g(x_i), &(2.11)\end{aligned}$$

where $\tilde{p}(x)$ and $\tilde{q}(x)$ are proportional to $p(x)$ and $q(x)$, respectively. The advantage of this method is that we can replace $p(x)$ and $q(x)$ with their proportional value $\tilde{p}(x)$ and $\tilde{q}(x)$, respectively. The validity of this calculation is confirmed by the following equations:

$$1 = \int \frac{q(x)}{p(x)}p(x)\,dx \tag{2.12}$$

$$\simeq \frac{1}{M}\sum_{p(x)}\frac{q(x)}{p(x)} \tag{2.13}$$

$$= \frac{1}{M}\frac{Z_p}{Z_q}\sum_{p(x)}\frac{\tilde{q}(x)}{\tilde{p}(x)}, \tag{2.14}$$

$$\frac{1}{\sum_{p(x)}\frac{\tilde{q}(x)}{\tilde{p}(x)}} \simeq \frac{1}{M}\frac{Z_p}{Z_q}, \tag{2.15}$$

where $Z_p = \int \tilde{p}(x)\,dx$ and $Z_q = \int \tilde{q}(x)\,dx$ are the normalizing constants of $\tilde{p}(x)$ and $\tilde{q}(x)$, respectively. Note that importance sampling estimator is unbiased, but normalizing importance sampling estimator is biased. This can be described as

$$E[\hat{I}_{\mathrm{IS}}] = I \tag{2.16}$$

and

$$E[\hat{I}_{\mathrm{NIS}}] \neq I. \tag{2.17}$$

2.3 Statistical Estimation

Statistical estimation is a task to estimate the probability distribution which underlies the given data. For example, observing the data $D = \{x_i\}$ which are generated according to a probability distribution $q(x)$, we make a probability distribution $p(x)$ which predicts $q(x)$. In this thesis, $q(x)$ represents the distribution of interest and is called the target distribution. On the other hand, $p(x)$ represents a probability model which is controlled to approximate the target distribution.

2.3.1 Kullback-Leibler Divergence

The most popular statistical estimation framework is one based on the Kullback-Leibler (KL) divergence, which is defined by

$$D_{\mathrm{KL}}(q \parallel p) = \int q(x)\log\frac{q(x)}{p(x)} \tag{2.18}$$

$$= \int q(x)\log q(x)\,dx - \int q(x)\log p(x)\,dx. \tag{2.19}$$

KL divergence is a measure of the distance between two probability distribution but not symmetric. It can be easily confirmed that if the two distributions are the same KL divergence becomes zero. In this framework, the objective of statistical estimation is to select $p(x)$ which minimizes KL divergence by using the given samples. Since $q(x)$ is unknown or $\int q(x) \log p(x)\, dx$ is difficult to calculate in practice, it is impossible to directly minimize KL divergence. Hence, some approximation methods are proposed, and the next section introduce one.

2.3.2 Maximum Likelihood Estimation

Maximum likelihood (ML) method is one of the practical methods for statistical estimation. In ML estimation, we find $p(x)$ which maximizes the empirical log-likelihood defined by

$$L(p(x)) = \sum_{q(x)} \log p(x). \tag{2.20}$$

In this thesis, the empirical log-likelihood is redefined by

$$\int q(x) \log p(x)\, dx \simeq \frac{1}{M} \sum_{q(x)} \log p(x), \tag{2.21}$$

where M is the number of the given samples. (2.20) and (2.21) are equivalent in finding $p(x)$ which maximizes them. It is clear that maximizing the empirical log-likelihood approximately minimizes the KL divergence. In practice, we use a parametrized probability distribution $p(x|w)$ for the probability model $p(x)$. Hence, in ML estimation, we find the parameter w which minimizes the empirical log-likelihood.

If the given data are generated from a different distribution $r(x)$ instead of $q(x)$, the empirical likelihood can be calculated through importance sampling as follows:

$$\int r(x) \frac{q(x)}{r(x)} \log p(x)\, dx \simeq \frac{1}{M} \sum_{r(x)} \frac{q(x)}{r(x)} \log p(x). \tag{2.22}$$

This type of problems is referred to as covariate shift [42].

2.3.3 Related Topics

Another example of KL divergence based statistical method is Bayes estimation. Readers who looks for better estimation methods like Bayes estimation, [2] becomes a good guide. Someone interested in minimizing $D_{\mathrm{KL}}(p, q)$ (q and p are exchanged) instead of $D_{\mathrm{KL}}(q, p)$, can find an approximation framework named *mean field approach* in [30]. Actually, EAPM has a relationship to mean field approach.

Chapter 3

Evolutionary Algorithms Based on Probability Models

3.1 Overview

In some optimization methods, estimation of the cost function is important. For example, the Newton method estimate the cost function via approximating it by a quadratic function through Taylor expansion. The mean field annealing approximates the cost function through the mean field technique. Actually, EAPM are optimization methods which estimate the cost function in statistical manners.

The basic concept of EAPM is estimation of the distribution of promising solutions. For this purpose, the cost function is transformed to a probability distribution, which is called the target distribution. It is assumed that samples generated from the target distribution tend to be promising solutions. EAPM make a probability model of the target distribution from the past samples and generate samples from it. If we have a probability model enough well approximating the target distribution, samples generated from the probability model will be promising solutions.

EAPM can be seen as sampling method which approximately generate samples from the target distribution. On the other hand, also Markov chain Monte Carlo methods (MCMC) are well known as sampling methods. When comparing EAPM with MCMC, the feature of EAPM is adaptively providing the sampler distribution, whereas in MCMC, the sampler distribution have to be designed by hand previously.

The optimization form of MCMC is well known as simulated annealing

(SA). There is the common concept to EAPM and SA. SA is started with the target distribution with high diversity (i.e., randomness or entropy), and then, the diversity is gradually decreased. This control method on the target distribution is called the annealing. In optimization, the annealing leads convergence. Actually, also EAPM employ the annealing.

To define the EAPM[1], the following sections, introduce the three essential concept: the target distributions, adaptive updating sampler distribution, and, the annealing. Additionally, one practical method which slightly differ from the EAPM is explained.

3.2 Target distribution

In the EAPM, the cost function is transformed to the target distribution which guides where the samples should be generated. To define the target distribution, there are two important features: (1) goodness and (2)randomness of the generated samples. The goodness means that the generated samples are preferred to have good cost function value. On the other hand, the randomness, which can be defined by the entropy, means that the generated samples are preferred to be distributed in the whole solution space. This is because there may be exists better solutions if a good solution have already been found. This is well known as the exploration/exploitation trade-off and the target distribution has a parameter which control this trade-off. In the following, two types of probability distributions are introduced.

3.2.1 Partially Uniform Distribution

The partially uniform distribution is the basic target distribution. It is defined as follows:

$$\begin{aligned} q(x) &= \frac{1}{Z}\tilde{q}(x) && (3.1) \\ \tilde{q}(x) &= I(f(x) < \tilde{f}) \\ &= \begin{cases} 1 & f(x) < \tilde{f} \\ 0 & else \end{cases} && (3.2) \\ Z &= \int \tilde{q}(x)\, dx, && (3.3) \end{aligned}$$

[1] Since there is no official definition of EAPM, this paper define "the EAPM". On the other hand, the word "EAPM" includes many similar algorithms. Note the difference between "the EAPM" and "EAPM".

where $I(\dot{})$ is the indicator function, $\tilde{f}$ is the threshold parameter which control the trade-off, and Z is the normalizing constant. Its entropy is given by

$$-\int q(x)\log q(x) = \log Z. \tag{3.4}$$

3.2.2 Boltzmann Distribution

The Boltzmann distribution (also called Gibbs distribution) is a well known probability distribution in statistical physics. The Boltzmann distribution is defined as follows:

$$\begin{aligned} q(x) &= \frac{1}{Z}\tilde{q}(x) && (3.5)\\ \tilde{q}(x) &= \exp(-f(x)\beta) && (3.6)\\ Z &= \int \tilde{q}(x)\,dx, && (3.7) \end{aligned}$$

where β is a parameter called the inverse temperature, which controls the trade-off.

The feature of the Boltzmann distribution is minimizing the free energy defined by

$$F(q(x)) = \int q(x)\beta f(x)\,dx + \int q(x)\log q(x)\,dx. \tag{3.8}$$

This can be understood that Boltzmann distribution maximizes the entropy and minimizes $f(x)$.

3.3 Adaptive Improvement of Sampler Distribution

The objective of the EAPM is to generate samples approximately according to the target distribution $q(x)$. The basic approach of the EAPM is to build a probability model of the target distribution by using ML estimation[2]. Let $p_{t-1}(x)$ and $p_t(x)$ denote the sampler distribution and the probability model, respectively. ML estimation is performed as follows:

$$p_t(x) = \underset{\hat{p}_t(x)}{\operatorname{argmax}} \frac{1}{M}\sum \frac{q(x)}{p_{t-1}(x)}\log \hat{p}_t(x). \tag{3.9}$$

[2] Obviously, we can use other methods such as Bayes estimation for building a probability model. However, currently, ML estimation is the best one in practice.

Naive EAPM

```
Initialization: Generate samples $X_{samp}^{(0)} = \{x_i\}_1^M$ from the uniform distribution $p_0(x)$. $t \Leftarrow 1$.
do{
  Calculate the empirical log-likelihood according to (3.9) from $X_{samp}^{(t-1)}$.
  Build a probability model $p_t(x)$.
  Generate samples $X_{samp}^{(t)}$ from $p_t(x)$.
  $t \Leftarrow t + 1$.
}until(stopping criterion reached)
```

Figure 3.1: The Pseudo-code of Naive EAPM.

Then, the samples are generated from the probability model $p_t(x)$. If $q(x)$ and $p_t(x)$ are enough similar, the obtained samples are approximately distributed according to $q(x)$.

The more similar $q(x)$ and $p_{t-1}(x)$ are, the better ML estimation works. Therefore, to generate samples from $p_t(x)$ and to built a new probability model by using ML estimation may provide a better new probability model. Hence, we can lead an algorithm which iteratively generates a probability model by using samples generated from the previous probability model. In this algorithm, it is expected that the built probability model is gradually improved. This thesis call this algorithm the naive EAPM and the pseudo code is shown in Fig. 3.1.

3.4 Annealing

The objective of the annealing is to reduce the variation of $\frac{q(x)}{p_{t-1}(x)}$ in (3.9), since if the variation is small, the importance sampling estimator becomes good. For this purpose, the target distribution is changed.

In general annealing, the target distribution is started from the uniform distribution and the randomness of the target distribution is gradually reduced. This is because it is difficult to build a probability model with less randomness, which means generating better solutions. The annealing can be easily employed by adding the procedure which changes the target distribution in each iteration of the naive EAPM. Hence, (3.9) is changed as

The EAPM

```
Initialization: Generate samples $X_{samp}^{(0)} = \{x_i\}_1^M$ from the uniform distribution $p_0(x)$. $t \Leftarrow 1$.
do{
  Determine the target distribution $q_t(x)$.
  Calculate the empirical log-likelihood according to (3.9) from $X_{samp}^{(t-1)}$.
  Build a probability model $p_t(x)$.
  Generate samples $X_{samp}^{(t)}$ from $p_t(x)$.
  $t \Leftarrow t + 1$.
}until(stopping criterion reached)
```

Figure 3.2: The Pseudo-code of the EAPM.

follows:

$$p_t(x) = \underset{\hat{p}_t(x)}{\operatorname{argmax}} \frac{1}{N} \sum \frac{q_t(x)}{p_{t-1}(x)} \log \hat{p}_t(x). \tag{3.10}$$

The pseudo-code and an illustration are shown in Figs 3.2 and 3.3, respectively. The control of the target distribution is discussed in Chapter 6. In this thesis, this algorithm is called the EAPM or cross entropy method (CE) [40].

3.5 A Practical Approximation

Although the EAPM seems to work well theoretically, not in practice. The reason can be that the variation of $\frac{q(x)}{p_{t-1}(x)}$ in (3.10) becomes too big. To overcome this problem, a slightly different method is often employed in practice. This algorithm is called the estimation of distribution algorithm (EDA) in this thesis.

The difference from the EAPM is that EDA does not use importance sampling. Instead of (3.10), EDA performs

$$p_t(x) = \underset{\hat{p}_t(x)}{\operatorname{argmax}} \frac{1}{M} \sum I(f(x) < \tilde{f}_t) \log \hat{p}_t(x). \tag{3.11}$$

This means that promising solutions are selected from the generated samples according to their cost function value and then the distribution of the selected solutions are estimated. This selection manner is called the truncation selection. (3.11) can be easily derived from (3.10) by supposing

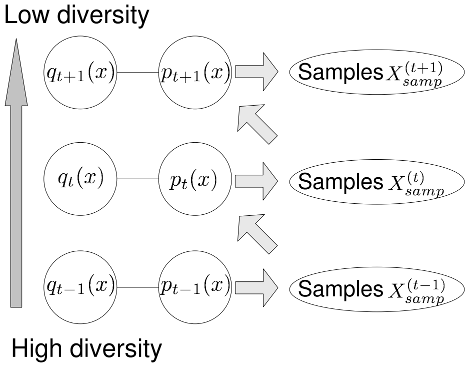


Figure 3.3: Illustration of Annealing.

that $p_{t-1}(x) = q_{t-1}(x)$, each $q_t(x)$ is a partially uniform distribution, and $f_{t-1} < f_t$. This implies that EDA assumes that $p_{t-1}(x)$ completely approximate $q_{t-1}(x)$. The pseudo-code is shown in Fig. 3.4. Experimental comparisons between the EAPM and EDA are shown in Appendix A. The experiments show that EDA outperforms the EAPM. However, this thesis do not recommend[3] this type of approximation and the proposed methods in the following sections are extensions of the EAPM.

[3] This thesis recommends another form of (3.10) as follows:

$$p_t(x) = \operatorname*{argmax}_{\hat{p}_t(x)} \frac{1}{N} \sum \left(\frac{q_t(x)}{p_{t-1}(x)} \right)^r \log \hat{p}_t(x), \tag{3.12}$$

where $0 \leq r \leq 1$. This form is friendly to importance sampling and we can represent the EDA by $r = 0$.

Estimation of Distribution Algorithm (EDA)

```
Initialization: Generate samples $X_{samp}^{(0)} = \{x_i\}_1^M$ from the uniform distribution $p_0(x)$. $t \Leftarrow 1$.
do{
  Determine the threshold parameter $f_t$.
  Calculate the empirical log-likelihood according to (3.11) from $X_{samp}^{(t-1)}$.
  Build a probability model $p_t(x)$ of $X_{pop}^{(t)}$.
  Generate samples $X_{samp}^{(t)} = \{x_i\}_1^M$ from $p_t(x)$.
  $t \Leftarrow t + 1$.
}until(stopping criterion reached)
```

Figure 3.4: The Pseudo-code of EDA.

Chapter 4

Maintaining Historical Samples

4.1 Introduction

The accuracy of the statistical estimation depends on not only the way of building probability models but also both the quality and the quantity of the samples. From this viewpoint, one drawback of the current EAPM is the absence of a population maintenance mechanism, where a part of the historical samples are reused, whereas Genetic Algorithms (GAs) [9] have used ones for a long time. In GAs, various population maintenance techniques [43] were proposed and they are intuitively combined with EAPM in some works, for example, Bayesian optimization algorithms (BOAs) [34], hierarchical Bayesian optimization algorithm (hBOA), [33, 36, 37], and iterated density estimation evolutionary algorithm (IDEA) [4, 5]. Indeed, they break the mathematical structure of EAPM; EAPM are regarded as methods building probability models approximating the target distributions, which explicitly represent the distributions of promising solutions. In the heuristic population mechanisms, the distribution of the population (i.e., the selected historical samples) is normally unknown and the built probability model is no longer related to the target distributions.

This chapter proposes a novel population maintenance method, resampling population model (RPM) that maintains a part of the historical samples such that they seem to follow the target distribution by weighting generated samples. This implies that the probability model of the population approximates the target distribution. To control the size of the population, resampling is employed. The aim of this chapter is to investigate the effectiveness of RPM through experimental comparisons with conventional methods.

4.2 Resampling Population Model (RPM)

The primary objective of RPM is to extend the calculation of (3.10) to using not only the currently generated samples but also the population, that is, a part of the historical samples. This implies that RPM is an extension of the EAPM. If all the historical samples in all previous iterations are completely maintained, according to importance sampling, the empirical log-likelihood is given by

$$L \simeq \frac{1}{|X|} \sum_{X} \frac{q_{t+1}(x)}{\sum_{i \leq t} \alpha_i p_i(x)} \log p_{t+1}(x), \tag{4.1}$$

where X is a set of all the historical samples, which are generated from all the built probability models and α_i is the ratio of the number of the samples generated from $p_i(x)$. [1] However, the secondary objective of RPM is to control the size of the population. In other words, important samples are selected and the remains are discarded.

In the following, first, the key calculation technique for weighting samples is described in Section 4.2.1. Section 4.2.2 describes how RPM maintains the population based on this technique. Finally, Section 4.2.3 describes the details for implementation.

4.2.1 Weighted Samples Approximating Distribution

If a sample set X whose samples are generated from $p(x)$ are given and their weights are defined by

$$w_i \propto \frac{q(x_i)}{p(x_i)} \text{ for } x_i \in X, \tag{4.3}$$

then we define a probability distribution $\hat{q}(x)$ as follows:

$$\hat{q}(x) = \frac{1}{\sum_X w_i} \sum_{x_j \in X} w_i \delta(x - x_i), \tag{4.4}$$

[1] If all the historical samples from the k previous iterations are completely maintained, according to importance sampling, the empirical log-likelihood is given by

$$L \simeq \frac{1}{|X|} \sum_{X} \frac{q_{t+1}(x)}{\sum_{t-k \leq i \leq t} \alpha_i p_i(x)} \log p_{t+1}(x), \tag{4.2}$$

where X is a set of all the historical samples, which are generated from all the built probability models and α_i is the ratio of the number of the samples generated from $p_i(x)$. This method may be practical, but we cannot control which samples are maintained in this way. Therefore, this is out of scope.

where $\delta(\cdot)$ is the delta function. The expectation of any function $f(x)$ with respect to $\hat{q}(x)$ is exactly given by

$$\int \hat{q}(x)f(x)dx = \frac{1}{\sum_X w_j}\sum_X w_i f(x_i). \tag{4.5}$$

This is equivalent to normalized importance sampling.

Let us assume $w_i = \frac{q(x_i)}{p(x_i)}$. Then, the expectation of $\sum_X w_i$ is the number of samples in X, that is, $E[\sum_X w_i] = |X|$. Especially, if $\sum_X w_i = |X|$, (4.5) reduces to normal importance sampling. Hence, we can regard $\hat{q}(x)$ as an approximation distribution of $q(x)$. This technique is well known in sequential Monte Carlo methods [7] [2].

In RPM, an additional extension is important. This can be extended to calculate the expectation with arbitrary $q^*(x)$ as follows:

$$\begin{aligned}
\int q^*(x)f(x)dx &= \int q(x)\frac{q^*(x)}{q(x)}f(x)dx && (4.6)\\
&\simeq \int \hat{q}(x)\frac{q^*(x)}{q(x)}dx && (4.7)\\
&= \frac{1}{\sum_X w_j}\sum_X w_i\frac{q^*(x)}{q(x)}f(x), && (4.8)
\end{aligned}$$

where w_i is defined by (4.3). If $q(x) = q^*(x)$, (4.8) reduces to (4.5). If $q(x) = p(x)$, (4.8) reduces to normal importance sampling. This gives an unified interpretation of normal importance sampling and normalized importance sampling. If the target distribution currently approximated by the weighted samples is changed the weights are redefined as follows:

$$\begin{aligned}
w_i^* &\propto w_i\frac{q^*(x_i)}{q(x_i)} && (4.9)\\
&\propto \frac{q(x)}{p(x)}\frac{q^*(x_i)}{q(x_i)} = \frac{q^*(x_i)}{p(x)}.
\end{aligned}$$

The advantage of this extension is that once the weights are determined we can forget how samples are generated, that is, $p(x)$.

[2] RPM is based on the technique of sequential Monte Carlo methods. Actually, the relationships between sequential Monte Carlo methods and Genetic Algorithms was pointed in [16].

4.2.2 Algorithm

If the population $X_{pop}^{(t)}$, which is supposed to be a set of unweighted samples here, is generated from the target distribution, the empirical log-likelihood can be calculated from the population and the currently generated samples as follows:

$$\begin{aligned} L &= \int r_t(x) \frac{q_{t+1}(x)}{r_t(x)} \log p_{t+1}(x) dx \qquad (4.10) \\ &\simeq \frac{1}{|X_m^{(t)}|} \sum_{X_m^{(t)}} \frac{q_{t+1}(x)}{r_t(x)} \log p_{t+1}(x), \qquad (4.11) \end{aligned}$$

$$r_t(x) = \alpha^{(t)} q_t(x) + (1-\alpha^{(t)}) p_t(x), \qquad (4.12)$$

$$\alpha^{(t)} = \frac{|X_{pop}^{(t)}|}{|X_{pop}^{(t)}| + |X_{samp}^{(t)}|}, \qquad (4.13)$$

$$X_m^{(t)} = X_{pop}^{(t)} \cup X_{samp}^{(t)}, \qquad (4.14)$$

where $X_{samp}^{(t)}$ is a set of the currently generated samples, which follow $p_t(x)$.

To maintain the population to follow the target distribution, RPM weights samples according to the method described in Section 4.2.1. It is assumed that the initial population follows a uniform distribution and the initial target distribution $q_1(x)$ is also a uniform distribution. Thus, the initial population is defined by

$$X_{pop}^{(1)} = \{(1, x_i^{(1)})\}_{i=1}^N,$$

where each x_i is generated from a uniform distribution and N is the number of weighted samples in the population. The population at time t is denoted by

$$X_{pop}^{(t)} = \{(w_i^{(t)}, x_i^{(t)})\}_{i=1}^N,$$

and their weights are calculated from the previous population as the following: Let us consider the following importance sampling:

$$\begin{aligned} L &\simeq \int \hat{r}_t(x) \frac{q_{t+1}(x)}{r_t(x)} \log p_{t+1}(x) dx \qquad (4.15) \\ &\simeq \frac{1}{|X_m^{(t)}|} \sum_{X_{samp}^{(t)}} \frac{q_{t+1}(x_i)}{r_t(x_i)} \log p_{t+1}(x_i) \\ &+ \frac{1}{|X_m^{(t)}|} \sum_{X_{pop}^{(t)}} w_i^{(t)} \frac{q_{t+1}(x_i)}{r_t(x_i)} \log p_{t+1}(x_i), \\ & \qquad (4.16) \end{aligned}$$

$$\hat{r}_t(x) = \alpha^{(t)}\hat{q}_t(x) + (1 - \alpha^{(t)})p_t(x). \tag{4.17}$$

This is equivalent to (4.11) whose $r(x)$ is replaced with $\hat{r}(x)$. In this thesis, $|X_{pop}^{(t)}| = \sum w_i^{(t)}$ is defined by the effective number which is explained later, and $\alpha^{(t)}$ can be determined. Let

$$X_{samp}^{(t)} = \{(w_i^{(t)}, x_i^{(t)})\}$$

where $w_i^{(t)} = 1$, then (4.16) can be rewritten as

$$L \simeq \frac{1}{\sum_{X_m} w_j} \sum_{X_m^{(t)}} w_i^t \frac{q_{t+1}(x_i)}{r_t(x_i)} \log p_{t+1}(x_i). \tag{4.18}$$

According to (4.9) and (4.18), the weight update rule is given by

$$w_i^{t+1} = w_i^t \frac{q_{t+1}(x_i)}{r_t(x_i)}. \tag{4.19}$$

The secondary objective is to select important samples for decreasing the size of the population. The point is to prevent the accuracy of the approximation of $\hat{q}(x)$ from becoming worse. For this purpose, RPM basically selects the samples with probabilities proportional to the weights and this is equivalent to sampling from $\hat{q}(x)$ defined by (4.4). Since it is preferable to avoid selecting overlapping samples, the population is represented by weighted samples. Obviously, one appropriate method is iteratively selecting samples by resampling with replacement and accumulating them but this is not practical in terms of computational cost. Another method is simply selecting the highest weight samples without changing the weights. $\hat{q}(x)$ can be a consistent estimator but this method breaks the consistency. There is an intermediate method between the previous two methods. that is to employs resampling (proportional to the weights) WithOut replacement (RWOR) without changing the weights. Note that RWOR can be quickly carried out [8]. In this chapter, RWOR is employed. Furthermore investigations on this topic are left as future works.

Effective Number of Samples

The population is defined by weighted samples as $\{(w_i^{(t)}, x_i^{(t)})\}_{i=1}^N$. The size of the population cannot be defined by $\sum w_i$ because each w_i is defined by a value proportional to $\frac{q(x)}{p(x)}$. Let us consider the normalized weight $\hat{q}(x_i) =$

$\frac{1}{\sum w_j} w_i$. This becomes a probability distribution and the average probability of $\hat{q}(x)$ is defined by $\bar{q} = \exp(\int \hat{q}(x) \log \hat{q}(x) dx)$. This can be regarded as an approximation, where w_i is given by $\bar{q}$ or 0. Under this approximation, the number of samples with nonzero weight is given by

$$\bar{q}^{-1} = \exp(-\int \hat{q}(x) \log \hat{q}(x) dx) \tag{4.20}$$

and this is called the effective number. The same idea have been introduced in [42].

4.2.3 Numerical Calculation Details

The procedure of RPM is described as follows: In initialization, the initial population $X_{pop}^{(1)}$ and the initial set of generated samples $X_{samp}^{(1)}$ are generated from a uniform distribution. Thus, the target distribution $q_1(x)$ and the initial sampler $p_1(x)$ are uniform distributions. For generating the next population $X_{pop}^{(t+1)}$ and the next samples $X_{samp}^{(t+1)}$, first, the next target distribution $q_{t+1}(x)$ is defined. In this chapter, target distributions are defined by partially uniform distributions, which are defined by (3.1), and therefore the threshold parameter $\tilde{f}$ in (3.2) is determined to define the target distribution. $\tilde{f}$ is selected such that the number of samples which satisfies $q_{t+1}(x) \neq 0$, becomes $N'(< N)$, which is previously defined by the cutoff rate $c = \frac{N'}{N}$, where N is the number of samples in the current population. To investigate RPM with using other probability distribution families such as Boltzmann distribution for target distributions is carried out in Chapter 6.

The merged set $X_m^{(t)} = X_{pop}^{(t)} \cup X_{samp}^{(t)}$ is generated. The weights of the samples in $X_m^{(t)}$ are updated according to (4.19). (4.19) can be rewritten by

$$w_i^{t+1} = w_i^t \frac{1}{\alpha \frac{q_t(x_i)}{q_{t+1}(x_i)} + (1-\alpha) \frac{p_t(x_i)}{q_{t+1}(x_i)}}. \tag{4.21}$$

Since the normalizing constants of $q_t(x)$ and $q_{t+1}(x)$ are unknown and proportional values $\tilde{q}_t(x)$ and $\tilde{q}_{t+1}(x)$ are known, they are estimated from the samples with importance sampling. Let Z_t and Z_{t+1} be the normalizing constants of $q_t(x)$ and $q_{t+1}(x)$, respectively. To calculate $\frac{q_t(x)}{q_{t+1}(x)}$, $\frac{Z_{t+1}}{Z_t}$ is estimated by

$$\frac{Z_{t+1}}{Z_t} \simeq \frac{1}{\sum_{X_{pop}^{(t)}} w_j} \sum_{X_{pop}^{(t)}} w_i \frac{\tilde{q}_{t+1}(x)}{\tilde{q}_t(x)}. \tag{4.22}$$

Since (4.21) can be rewritten as

$$w_i^{t+1} = w_i^t \frac{1}{\alpha + (1-\alpha)\frac{p_t(x_i)}{q_t(x_i)}} \frac{q_{t+1}(x_i)}{q_t(x_i)}, \tag{4.23}$$

(4.22) can not be needed in practice. On the other hand, to calculate $\frac{p_t(x)}{q_{t+1}(x)}$, Z_{t+1} is estimated by

$$Z_{t+1} \simeq \frac{1}{M} \sum_{X_{samp}^{(t)}} \frac{\tilde{q}_{t+1}(x)}{p_t(x)}. \tag{4.24}$$

These are the simplest estimators and there may exist better estimators. This is left as a future work.

The next population $X_{pop}^{(t+1)}$ is generated by resampling from $X_m^{(t)}$ according to the new weights, and then the weights are normalized so that the sum becomes the effective number. In this chapter, samples whose weights are 0 are removed before the resampling operation even if the number of selected samples become less than N. This contributes to faster convergence but is not essential.

The next sampler $p_{t+1}(x)$ is generated not from $X_{pop}^{(t+1)}$ but from $X_m^{(t)}$ because $X_{pop}^{(t+1)}$ is a part of $X_m^{(t)}$ and thus $X_m^{(t)}$ contains more information. The empirical likelihood is given by (4.18) and the probability model maximizing the empirical log-likelihood is selected as the next probability model $p_{t+1}(x)$. Then, the next set of generated samples $X_{samp}^{(t+1)}$ are generated from $p_{t+1}(x)$. The pseudo-code of the algorithm is shown in Fig. 4.1.

4.3 Experiments

This section conducts experiments to compare RPM and conventional methods such as EDA, BOA, hBOA, and IDEA. The conventional methods are based on the common framework that iterates the following three steps: (1) selection, (2) sampling, and (3) replacement step. Especially, the replacement step plays the role of the population maintenance mechanism. In the present experiments, the truncation selection, which selects the best samples in the population, is employed. The percentage of the unselected samples is called cutoff rate, denoted by c, in this chapter. Note that this is not exactly the same as the cutoff parameter of RPM but they are similar. Therefore, they have the common name in this chapter. In the sampling step, a probability model of the selected samples is built and new samples are generated from it.

```
Resampling Population Model
Generate the initial population $X_{pop}^{(1)}$ and the initial samples $X_{samp}^{(1)}$
from the uniform distribution. $t \Leftarrow 1$.
do{
  Generate the target distribution $q_{t+1}(x)$.
  Reweight each sample $(w_i, x_i) \in X_m^{(t)} = X_{pop}^{(t)} \cup X_{samp}^{(t)}$ according to
  (4.19).
  Build a probability model $p_{t+1}(x)$ from $X_m^{(t)}$ according to (4.18).
  Generate samples $X_{samp}^{(t+1)} = \{(1, x_i)\}_{i=1}^{M}$ from $p_{t+1}(x)$.
  Generate the next population $X_{pop}^{(t+1)} = \{(1, x_i)\}_{i=1}^{N}$ by resampling
  from $X_m^{(t)}$.
  $t \Leftarrow t + 1$.
}until(stopping criterion reached)
```

Figure 4.1: The pseudo-code of RPM

In the replacement step, the next population is generated from the current population and the newly generated samples. In the present experiments, three types of replacement operators are considered: full replacement (FR), elitist replacement (ER) and restricted tournament replacement (RTR).

In FR, the current population is completely replaced with the newly generated samples. This case corresponds to EDA. In ER, the next population consists of the k best samples in the current population and the newly generated samples. For simplicity, $k = N \times (1 - c)$ where N is the size of the population and c is the cutoff parameter of the truncation selection. The feature is that the population is monotonically improved. This type of replacement operators have been used in the IDEA studies. In RTR, which is used in the hBOA studies, each generated sample is compared with one sample in the current population, and if the generated sample is better than the sample in the current population then they are exchanged. For selecting the sample to be compared from the population, a subset is generated by randomly selecting samples from the current population and the most similar sample to the generated one in the subset is selected. The size of the subset and the similarity is somehow provided previously. According to [21], the size is set at $\min\{d, \lceil N/20 \rceil\}$, where d is the problem size and N is the population size, and Manhattan distance is used in this chapter.

4.3.1 Benchmark Problems

In the benchmark problems, the domain for each variable is $x_i \in \{0, 1\}$ and the number of the dimension d is set at 400. Minimization problems are considered.

Onemax

This problem is defined as

$$f(x) = -\sum_{i=1}^{d} x_i. \tag{4.25}$$

The optimum cost function value is $-d$, and there is no correlation between any of the variables.

1D Ising model

This problem is defined as follows:

$$f(x) = -\sum_{i=1}^{d} J(x_i, x_{i+1}), \tag{4.26}$$

$$J(x_i, x_j) = \left\{ \begin{array}{ll} 1 & x_i = x_j \\ 0 & x_i \neq x_j \end{array} \right. . \tag{4.27}$$

Periodic boundary conditions, implying that x_{d+1} is treated as x_1, are employed. The optimum cost function value is $-d$. There are correlations between two variables, as illustrated in Fig. 4.2.

2D Ising model

We consider $d = r \times r = 20 \times 20$ grids, as illustrated in Fig. 4.3. If two connected variables attain the same value, the value of the cost function is improved. 2D Ising model can be defined as

$$f(x) = -\sum_{i=1}^{i=r} \sum_{j=1}^{j=r} \{J(x_{ij}, x_{i+1,j}) + J(x_{ij}, x_{i,j+1})\}. \tag{4.28}$$

Periodic boundary conditions are employed. The optimum cost function value is -2d. This problem is basically equivalent to a check-board problem [20].

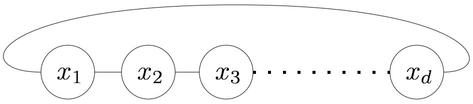


Figure 4.2: 1D Ising with Periodic Boundary Conditions.

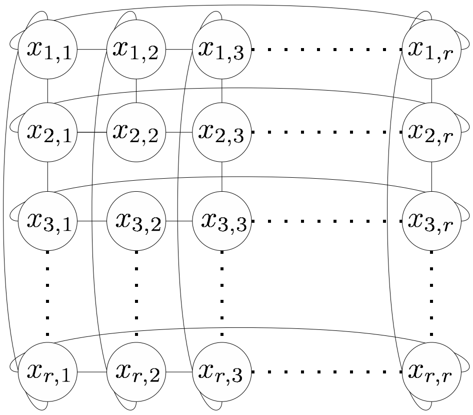


Figure 4.3: 2D Ising with Periodic Boundary Conditions.

Adding Noise

Since the threshold of the partially uniform distribution cannot function precisely when multiple solutions have the same cost function value, the original cost function $f(x)$ is slightly altered by adding small random values ϵ as follows:

$$f'(x) = f(x) + \epsilon. \tag{4.29}$$

In the experiments, ϵ is $u \times 10^{-10}$, where u is a random number uniformly distributed from 0 to 1. This is applied to all the three functions described above.

4.3.2 Experimental Setup

This thesis focuses on the simplest probability model, that is, a fully factorized one defined as follows:

$$p(x|w) = \prod_{i=0}^{i=d-1} p(x_i|w_i), \tag{4.30}$$

where each $p(x_i|w_i)$ is a Bernoulli distribution. This is because, for the first step in investigating the effects of HIS, the basic probability model is appropriate in terms of avoiding over-fitting. Instead of changing the complexity of the probability models, different kinds of problems are used for the experiments. As future work, investigations on the effects of model errors on HIS will be performed by changing the complexity of the probability models.

When using the factorized probability model, FR (EDA) corresponds to UMDA [29]. Probability models are built by using ML estimation. Only in the cases of FR, we employ online updating where the parameter is updated by

$$w_{new} = (1 - \alpha)w_{old} + \alpha w_{ML}, \tag{4.31}$$

where w_{new}, w_{old}, w_{ML} are the new parameter, previous parameter, and ML estimator, respectively, and α is the learning rate. This is because the population is quickly changed in FR and this causes instability. In the present experiments, $\alpha = 0.5$.

In all cases, there are basically three parameters: (1) the population size, denoted by N, (2) the number of generated samples in one sampling, denoted by M, and (3) cutoff rate, denoted by c. These values are experimentally determined as follows:

- FR (EDA)
  - N: 100, 500, 1000, 3000, or 6000.
  - $M = N$.
  - c: 0.1, 0.3, or 0.5.
- ER
  - N: 100, 500, 1000, or 3000.
  - $M = N \times c$.
  - c: 0.1, 0.3, 0.5, or 0.7.
- RTR
  - N: 10, 50, 100, 200, 300, 400, or 500.
  - $M(\leq N)$: 10, 50, 100, 200, 300, 400, or 500.
  - c: 0.1, 0.3, 0.5, 0.7, or 0.9.

Table 4.1: Results of ER for Onemax.

N	c	Best		Evaluations	
100	0.1	–397.4	(1.5)	3409	(83.72)
500	0.1	–400	(0)	15500	(196.21)
1000	0.1	–400	(0)	30300	(788.67)
3000	0.1	–400	(0)	90300	(1728.58)
100	0.3	–397.4	(0.92)	3103	(86.38)
500	0.3	–400	(0)	14045	(260.24)
1000	0.3	–400	(0)	28030	(283.02)
3000	0.3	–400	(0)	82200	(1505.99)
100	0.5	–397.1	(0.83)	2750	(100)
500	0.5	–400	(0)	12650	(165.83)
1000	0.5	–400	(0)	24750	(460.98)
3000	0.5	–400	(0)	74100	(994.99)
100	0.7	–394.5	(2.2)	2397.7	(75.9)
500	0.7	–400	(0)	11004.9	(104.7)
1000	0.7	–400	(0)	21410.8	(523.08)
3000	0.7	–400	(0)	62401.7	(1639.37)

- RPM
  - N: 10, 50, 100, 200, 300, 400, or 500.
  - $M(\leq N)$: 10, 50, 100, 200, 300, 400, or 500.
  - c: 0.01, 0.05, 0.1, or 0.2.

The stopping criteria are that the number of function evaluations is greater than 2.9×10^6, the variance of the cost function values of the generated samples is less than 10^{-20}, and the optimal solution is found.

4.3.3 Results

The results of FR (EDA) are shown in Appendix A. A part of the results for onemax are shown in Tables 4.1 , 4.2 and 4.3. The columns titled “Best” list the average cost function values, with the standard deviation in parenthesis, of the best obtained solutions over ten independent runs. The columns titled “Evaluations” list the average number, with the standard deviation in parenthesis, of function evaluations performed until stopping criteria are

Table 4.2: Results of RTR ($C = 0.7$) for Onemax.

N	M	Best		Evaluations	
10	10	–255.9	(7.44)	193	(16.16)
50	10	–331.7	(5.9)	2900010	(0)
50	50	–370.9	(4.91)	2900050	(0)
100	10	–372.6	(4.78)	2900010	(0)
100	50	–390.1	(2.12)	2900050	(0)
100	100	–397.1	(1.81)	2610730	(868110)
200	10	–398.1	(1.22)	2610823	(867561)
200	50	–399	(0.89)	2032990	(1324456.48)
200	100	–399.7	(0.46)	878810	(1323251.13)
200	200	–400	(0)	10820	(3552.41)
300	10	–400	(0)	18704	(3981.81)
300	50	–399.9	(0.3)	308135	(863982.34)
300	100	–400	(0)	18650	(2154.65)
300	200	–400	(0)	17320	(1695.76)
300	300	–400	(0)	18600	(2212.69)

met. The others show the parameters. Basically, FR, ER and RPM afford sufficient good results, that is, finding an optimal solution. In contrast, in some RTR cases, the population does not converge and the obtained solutions are bad. The results imply that RTR needs appropriate parameter setting especially on the cutoff rate, though it may be difficult to select an appropriate parameter in general. RTR with $c = 0.7$ is comparatively good for onemax.

The results for 1D and 2D Ising are shown in Figs. 4.4 and 4.5, respectively. In each figure, the horizontal axis represents the number of function evaluations, while the vertical axis represents the cost function value. Each point represents the average cost function value of the best obtained solutions and the average number of function evaluations performed over ten independent runs for FR, ER and RPM. Note that RPM with $C = 0.1$ and $C = 0.2$ are not shown because in these cases the population quickly converges and the results are not good. The results of these cases with $M = 500$ are shown in Fig. 4.6. In RTR, the population does not converge and therefore the average cost function value of the best case is represented by the horizontal line in each figure. The five best cases of RTR for 1D and 2D Ising are shown in Tables 4.4-(a) and 4.4-(b), respectively. For comparison, a part of results

Table 4.3: Results of RPM ($C = 0.01$) for Onemax.

N	M	Best		Evaluations	
10	10	–267.8	(8.82)	1455	(197.7)
50	10	–395.7	(1.73)	17935	(1268.47)
50	50	–399.9	(0.3)	118075	(4346.22)
100	10	–400	(0)	20291	(341.86)
100	50	–400	(0)	133730	(4321.41)
100	100	–400	(0)	313720	(9888.26)
200	10	–400	(0)	30225	(437.16)
200	50	–400	(0)	210360	(4185.08)
200	100	–400	(0)	476680	(8545.62)
200	200	–398.8	(3.28)	1040020	(122154.02)
300	10	–400	(0)	29187	(343.77)
300	50	–400	(0)	281865	(3792.63)
300	100	–400	(0)	609900	(11553.87)
300	200	–400	(0)	1319380	(23383.96)
300	300	–397.3	(6.87)	2019090	(315916.21)

of RPM are also shown in tables 4.5-(a) and 4.5-(b). The expression is the same as in the previous tables.

As shown in the results for 1D and 2D Ising, if sufficient number of function evaluations are performed until convergence RPM is clearly the best method. ER may be slightly better than FR. RTR can afford better solutions than ER, but RTR has the problem of convergence and may have the difficulty in parameter setting, It is pointed that RTR with tournament selection is better than RTR with truncation selection in [21].

4.4 Discussion

4.4.1 Diversity of Population

The quality of a population depends on both the average of the cost function value over the samples in the population and the diversity of the population. If we keep only samples with better cost function values, the population will quickly converge into local optima. To prevent this, it is important to search promising area where samples have not been generated yet, by keeping the diversity.

Table 4.4: The 5 best cases of RTR for 1D and 2D Ising. In all cases, the population does not converge and thus 2.9×10^6 function evaluations are performed.

(a) 1D Ising

N	M	c	Best
200	200	0.3	–370.2 (5.55)
500	10	0.7	–369.8 (2.89)
200	10	0.3	–369.6 (3.07)
300	10	0.5	–369.4 (3.35)
500	500	0.7	–369 (3.13)

(b) 2D Ising

N	M	c	Best
500	300	0.7	–733.4 (9.96)
200	10	0.3	–731.8 (18.41)
200	100	0.3	–729.8 (14.52)
200	50	0.3	–729 (7)
300	50	0.5	–728.2 (11.15)

In RPM, the diversity is kept not by resampling, but by weighting [3]. In calculating the empirical log-likelihood, small weight samples are basically ignored or removed. The weights are defined by (4.19). In (4.19) samples with higher value of $p(x)$ have smaller weight. This means that samples in an area where many samples are generated, are regarded as less important.

To generate samples from the target distribution $q(x)$ is supposed to be the optimal in the EAPM framework. However, in practice, the samples are generated from a probability model $p(x)$ approximating $q(x)$ and the samples have a bias because $p(x)$ is a simple function whereas $q(x)$ is a complex function in general. Then, the role of (4.19) is the correction of the sampling bias of $p(x)$.

CE actually has this bias removing mechanism but cannot afford good results. This is because generated samples are unstable due to the estimation error of $p(x)$ in approximating $q(x)$. In contrast, the population in RPM theoretically does not depend on $p(x)$ and consequently RPM affords good results. This implies that the population mechanism of RPM improves the

[3] It is clearly optimal that no samples are removed. Even in this case, the diversity can be controlled by weighting. The role of resampling is to keep the population size constant.

Table 4.5: The results of RPM ($N = 500, c = 0.01$) for 1D and 2D Ising.

(a) 1D Ising

M	Best		Evaluations	
10	–367.6	(2.5)	33113	(1197.29)
50	–376.4	(4.18)	212620	(6956.98)
100	–379.2	(2.99)	426020	(17388.26)
200	–379.6	(3.77)	870180	(23316.64)
300	–379.2	(3.92)	1336070	(48353.53)
400	–381.6	(4.54)	1867220	(57325.4)
500	–382.4	(2.94)	2365300	(93769.45)

(b) 2D Ising

M	Best		Evaluations	
10	–724.8	(14.29)	39582	(1398.18)
50	–738.8	(11.32)	230640	(9368.96)
100	–749.4	(8.81)	494280	(20965.72)
200	–766.4	(12.64)	1186340	(92668.69)
300	–749.2	(6.34)	1742000	(40892.32)
400	–761.6	(15.89)	2473260	(146213.15)
500	–761.4	(18.11)	2900500	(0)

quality of samples in calculating the empirical log-likelihood, where samples from $p(x)$ may be bad.

4.4.2 Control of Convergence Speed

In RPM, the convergence speed is controlled by the target distribution, that is, the threshold parameter of partially uniform distribution. It is guaranteed that the threshold is monotonically decreased in RPM and the population must converge eventually. ER possesses the similar property.

The convergence speed has an effect on the accuracy of importance sampling of (3.10). To generate samples from the current target distribution is the optimal case. In this case, the weight is defined by $\frac{q_{t+1}(x)}{q_t(x)}$. If $q_{t+1}(x)$ and $q_t(x)$ are similar, the accuracy of importance sampling becomes better. In other words, slow convergence improves the accuracy and helps to afford good results. Through the experiments, it is confirmed that RPM is actually the only method that can monotonically control its convergence speed by the cutoff rate as shown in Fig. 4.6.

4.5 Summary

This chapter proposed Resampling Population Model (RPM), where a part of the historical samples are stored as the population to follow the target distribution from the viewpoint of importance sampling. Experimental comparisons between RPM and the conventional population mechanisms have revealed the following two advantages of RPM: (1) RPM affords better solutions than the conventional methods and (2) RPM can monotonically control the convergence speed by using the cutoff rate.

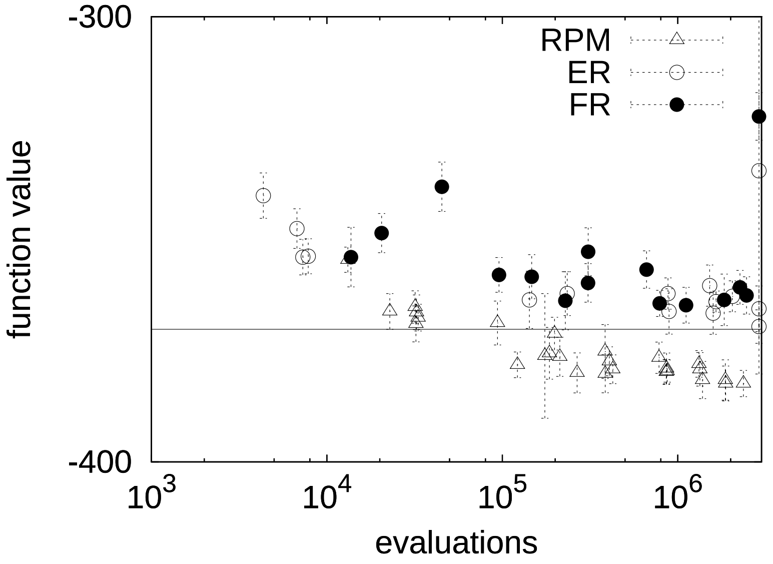


Figure 4.4: Results for 1D Ising.

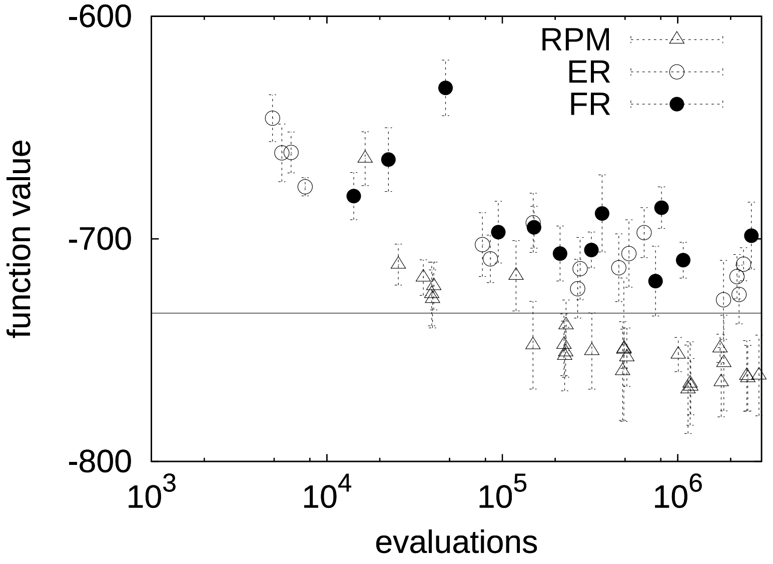


Figure 4.5: Results for 2D Ising.

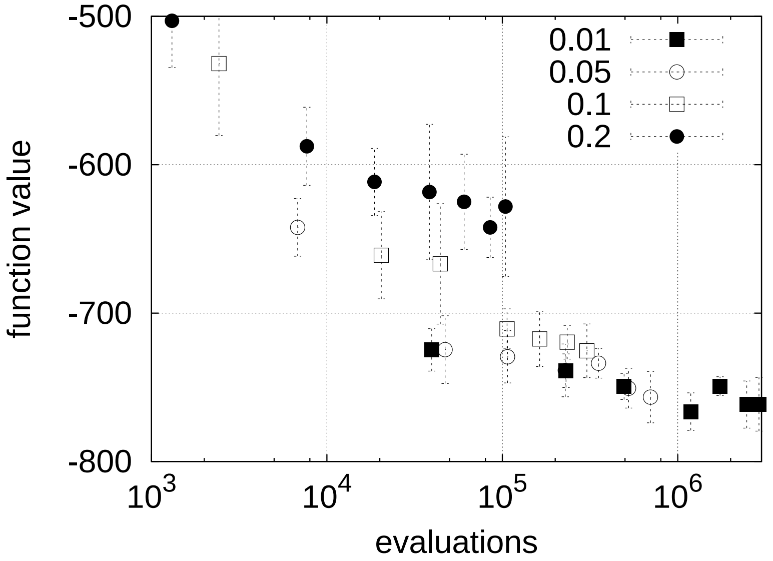


Figure 4.6: Results of RPM ($N = 500$) for 2D Ising.

Chapter 5

Hierarchical Control of The Divergence

5.1 Introduction

The annealing is the essential concept of the EAPM. However, the annealing is an unstable method because the obtained solutions cannot be further improved once the EAPM converges. This is the problem of local optima.

To overcome this problem, this paper proposes a novel method, Hierarchical Importance Sampling (HIS) that can be used instead of the annealing. The basic principle is to generate multiple sample sets with different diversities [1]. For example, one sample set may be almost random and another, almost converged. HIS employs multiple target distributions, builds a probability model of each target distribution, respectively, and generates samples from all the built probability models simultaneously. Therefore, the obtained samples consist of a number of sample sets, each of which is generated from a different probability distribution. The salient feature is that mixed samples are used for building probability models of the target distributions according to importance sampling [2,39], which guarantees mathematical validity. The aim of this paper is to investigate the effectiveness of the proposed method through experimental comparisons

[1] The exchange Monte Carlo method (EMC) [17] uses the same concept of sampling from multiple target distributions with different diversities. EMC is one of the Markov chain Monte Carlo methods (MCMC) [2]. MCMC and EMC are essentially different from EAPM and HIS. The relationships among EAPM, MCMC, HIS, and EMC are summarized in Section 5.4.6.

5.2 Hierarchical Importance Sampling (HIS)

5.2.1 Theoretical Overview

HIS maintains L number of layers, each of which consists of a sample set X_l, a probability model $p_l(x)$, and a target distribution $q_l(x)$. Each X_l is a set of samples generated from the corresponding probability model $p_l(x)$. Each $p_l(x)$ is built with ML estimation to approximate the corresponding target distribution $q_l(x)$, which is assumed to be previously provided here. Thus, X_l is approximately distributed according to $q_l(x)$. It is supposed that $q_l(x)$ has less diversity (i.e.,e ntropy) than $q_{l-1}(x)$. Therefore, it is also expected that $p_l(x)$ has less diversity than $p_{l-1}(x)$, and X_l contains better solutions than X_{l-1}. Normally, $q_0(x)$ is the uniform distribution, and $q_{L-1}(x)$ is the converged distribution, which generates only the best obtained solution.

Basically, HIS iterates the following two steps: (1) sampling and (2) estimation. In the sampling step, each X_l is updated by sampling from $p_l(x)$ and replacing the current sample set with the newly generated samples; the sampling step is illustrated in Fig. 5.1–(a). In the estimation step, each $p_l(x)$ is updated to approximate $q_l(x)$ more accurately than the previous one. The important feature is that all the sample sets $X_m = X_0 \cup \cdots \cup X_{L-1}$ are used for updating each $p_l(x)$. The probability distribution of X_m is given by a mixture distribution, which is defined as follows:

$$p_m(x) = \sum_l \alpha_l p_l(x), \tag{5.1}$$

$$\alpha_l = \frac{M_l}{\sum_i M_i}, \tag{5.2}$$

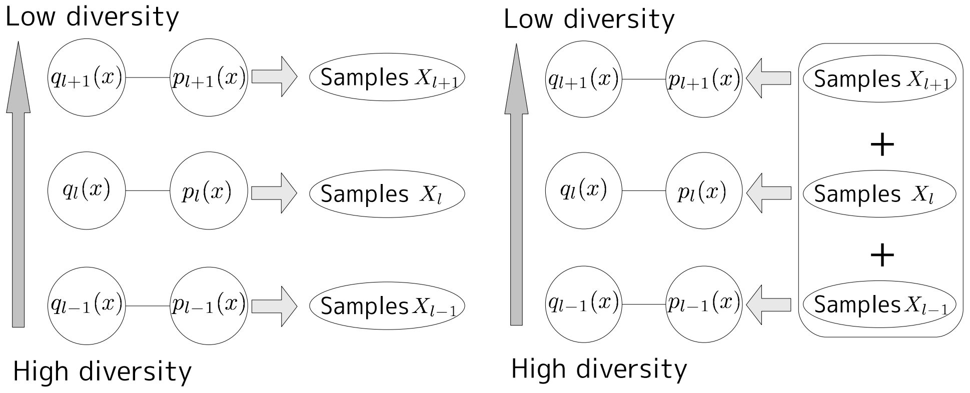
where M_l is the number of samples in X_l; thereby, the empirical log-likelihood with respect to $q_l(x)$ can be calculated via importance sampling as follows:

$$L \simeq \frac{1}{\sum_i M_i} \sum_{X_m} \frac{q_l(x)}{p_m(x)} \log p_l(x). \tag{5.3}$$

This corresponds to (3.10). The estimation step is illustrated in Fig. 5.1–(b).

5.2.2 Comparison between HIS and the EAPM

Suppose that the target distributions are previously provided in the EAPM and HIS. Let L be the number of the layers of HIS. At time t, the EAPM



(a) Sampling (b) Estimation

Figure 5.1: Illustration of Hierarchical Importance Sampling.

generates a probability model $p_t(x)$ approximating the corresponding target distribution $q_t(x)$, whereas HIS generates L number of probability models $p_0^{(t)}(x) \cdots p_{L-1}^{(t)}(x)$ approximating the corresponding target distributions $q_0(x) \cdots q_{L-1}(x)$, respectively. To generate $p_t(x)$, the EAPM uses only one sample set X_{t-1}, which is generated in the previous step. On the other hand, to generate $p_l^{(t)}(x)$, HIS uses all the sample sets $X_0^{(t-1)} \cdots X_{L-1}^{(t-1)}$, generated in the previous step. In other words, the difference is that the EAPM sequentially generates probability models and sample sets, whereas HIS generates probability models and sample sets both simultaneously and iteratively.

If only the $l-1$th sample set X_{l-1} is used for updating the lth probability model $p_l(x)$ in the estimation step of HIS, HIS, indeed, corresponds to iterative execution of the EAPM, which means that the EAPM is restarted from the initialization if the EAPM converges. This implies that HIS is a mathematical extension of the EAPM.

5.2.3 Target Distribution Control

HIS can theoretically operate if the target distributions are previously defined in any manner. However, in practice, HIS requires appropriate target distributions to produce good results. This section explains a manner in which the target distributions of HIS are provided. Note that the proposed target distribution control method in this section cannot be directly applied with any probability distribution other than the partially uniform distribution defined by (3.1) for target distributions. Further discussion is left in Chapter 6.

It is supposed that $q_0(x)$ and $q_{L-1}(x)$ are given[2]; then the objective of the control method is to determine $q_l(x)$ for $l = 1 \cdots L-2$. Each $q_l(x)$ is represented by the partially uniform distribution and denoted by $q_l(x|\tilde{f}_l)$ with the threshold parameter $\tilde{f}$. In terms of importance sampling, $q_{l-1}(x)$ and the next target distribution $q_l(x)$ should be similar because the accuracy of the empirical log-likelihood given by the importance sampling depends on this similarity. Thus, the objective is to select $\tilde{f}_l$ such that $q_{l-1}(x|\tilde{f}_{l-1})$, $q_l(x|\tilde{f}_l)$, and $q_{l+1}(x|\tilde{f}_{l+1})$ are similar.

The present concept is based on the size of the search space. In the case of the partially uniform distribution, a set of drawable samples is defined by $C_l = \{x|\tilde{q}(x|\tilde{f}_l) = 1\}$, where $\tilde{q}(x|\tilde{f})$ is defined by (3.2), and the number of drawable samples is given by $\int_C dx = \int \tilde{q}(x)dx = Z$. Thus, the size of the search space can be provided by the normalizing constant defined by (3.3). Note that the normalizing constant is normally unknown, but its estimator can be calculated through importance sampling as follows:

$$\begin{aligned} Z_l(\tilde{f}) &= \int \tilde{q}(x|\tilde{f})dx \\ &\simeq \frac{1}{M}\sum_{p(x)} \frac{\tilde{q}(x|\tilde{f})}{p(x)} \\ &= \hat{Z}_l(\tilde{f}), \end{aligned} \tag{5.4}$$

where $\sum_{p(x)}$ denotes summation over the samples generated from $q(x)$ and M is the number of the samples. In an importance sampling calculation,

$$\frac{1}{M}\sum_{q_{l-1}(x)} \frac{q_l(x)}{q_{l-1}(x)} f(x), \tag{5.5}$$

the probability of generating an acceptable sample, whose weight $\frac{q_l(x)}{q_{l-1}(x)}$ is not zero, is given by

$$\int_{C_{l-1}} q_{l-1}(x)\frac{\tilde{q}_l(x)}{\tilde{q}_{l-1}(x)}dx = \frac{Z_l}{Z_{l-1}}, \tag{5.6}$$

where it is assumed that $C_l \subseteq C_{l-1}$. It is clear that the rejected samples do not contribute to the importance sampling. In the EAPM under an assumption that samples are generated not from the probability models but from

[2] In the experiments, a probability distribution that generates only the best obtained sample is used for $q_{L-1}(x)$.

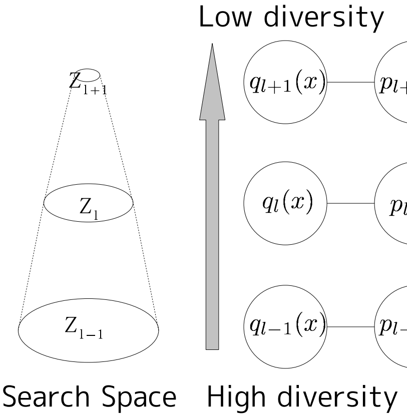


Figure 5.2: Search Space Reduction.

the target distributions, the sum of the number of the accepted samples throughout the optimization process is given by

$$\sum_{l=1}^{L-1} M_{l-1} \frac{Z_l}{Z_{l-1}}, \tag{5.7}$$

where L is the number of iterations. The maximization condition of Z_l with respect to (5.7) is given by

$$M_{l-1} \frac{Z_l^*}{Z_{l-1}} = M_l \frac{Z_{l+1}}{Z_l^*}, \tag{5.8}$$

where Z_l^* is the optimal value.

If Z_{l-1} and Z_{l+1} are given[3], the target normalizing constant Z_l^* is obtained from (5.8). Then, the threshold parameter $\tilde{f}_l$ is updated to satisfy

$$Z_l^* = \hat{Z}_l(\tilde{f}_l), \tag{5.9}$$

where $\hat{Z}_l(\tilde{f}_l)$ is the estimator of the normalizing constant given by (5.4). A method for solving (5.9) is described in Section 6.2.4. Figure 5.2 shows an illustration of the search space reduction.

[3] Note that Z_0 and Z_{L-1} are normally previously provided and thus, all Z_l can be previously determined according to (5.8). However, this paper uses the estimators of Z_{l-1} and Z_{l+1} to determine Z_l because, in some cases, it can be difficult to build a probability model approximating a target distribution with a certain normalizing constant.

```
Hierarchical Importance Sampling (HIS)
Initialize the probability models $p_0(x) \cdots p_{L-1}(x)$ and the sample sets $X_0 \cdots X_{L-1}$. $l \Leftarrow 0$.
do{
  Adjust the target distribution $q_l(x)$ according to (5.8).
    Calculate the empirical log-likelihood with respect to $q_l(x)$ from $X_{l-1}, X_l, X_{l+1}$ through importance sampling according to (5.3).
    Update the probability model $p_l(x)$ according to the empirical log-likelihood.
    Generate samples from $p_l(x)$ and replace the sample set $X_l$ with the generated samples.
    $l \Leftarrow (l+1)\%L$.
}until(stopping criterion reached)
```

Figure 5.3: The Pseudo-code of Hierarchical Importance Sampling.

5.2.4 Practical Procedure

In the practical procedure of HIS, first of all, each $p_l(x)$ is initialized to a uniform distribution and each X_l is generated from $p_l(x)$. For each l, the lth layer (i.e., $q_l(x)$, $p_l(x)$, and X_l) is sequentially and iteratively updated. To update the lth layer, first, $q_l(x)$ is updated according to (5.8), and then $p_l(x)$ is updated. To calculate the empirical log-likelihood with respect to $q_l(x)$, we use only three sample sets, which are the upper one X_{l-1}, the current one X_l, and the lower one X_{l+1}[4] for two reasons: calculating the marginal probability (5.1) consumes a considerable amount of time, and the samples in X_i, generated from $p_i(x)$, tend not to contribute to the importance sampling (5.3) if $p_i(x)$ and $q_l(x)$ are not similar. Finally, the sample set X_l is replaced with samples newly generated from $p_l(x)$. The pseudo-code of HIS is shown in Fig. 5.3.

[4] X_{-1} and X_L are supposed to be null sets.

Table 5.1: Results of HIS for Onemax.

Samples	Cutoff	Best	Evaluations
10	10	–400 (0)	29155 (12095.51)
10	20	–400 (0)	32743 (11000.43)
50	10	–400 (0)	48435 (20868.97)
10	30	–400 (0)	56170 (16627.74)
10	40	–400 (0)	67595 (20897.39)
50	20	–400 (0)	82215 (15277.81)
50	30	–400 (0)	113680 (23360.03)
50	40	–400 (0)	157715 (35272.9)

5.3 Experiments

This section describes the experiments conducted to investigate the advantages of HIS through comparison with EDA.

5.3.1 Experimental Setup

The present experiments are set up according to the experiments of Section 4.3.

HIS Setting

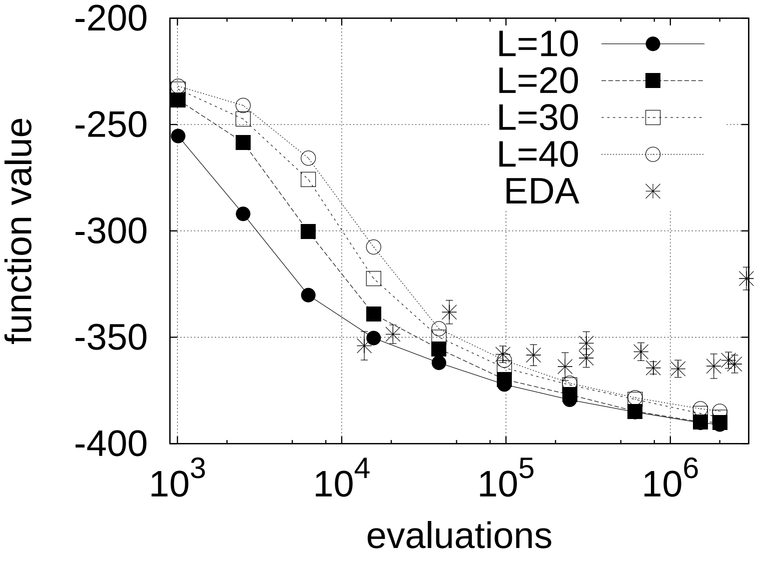
HIS employs the online updating defined in Section 4.3.2. All the parameter settings are described as follows:

- The number of generated samples in one sampling M : 10 or 50.
- The number of the layers L: 10, 20, 30, or 40.
- Learning rate α: 0.5.

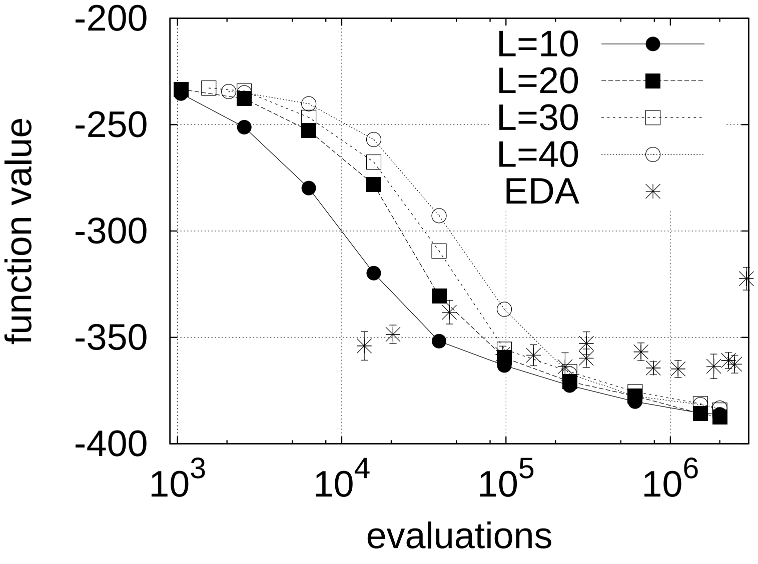
These values are experimentally determined. Note that the number of samples contained in X_i is denoted by M_i and $M_i = M_j = M$.

5.3.2 Results

Table 5.1 shows the results of HIS for Onemax. The first and the second columns list the number of generated samples per sampling and the number of layers, respectively. The third column lists the average cost function value,

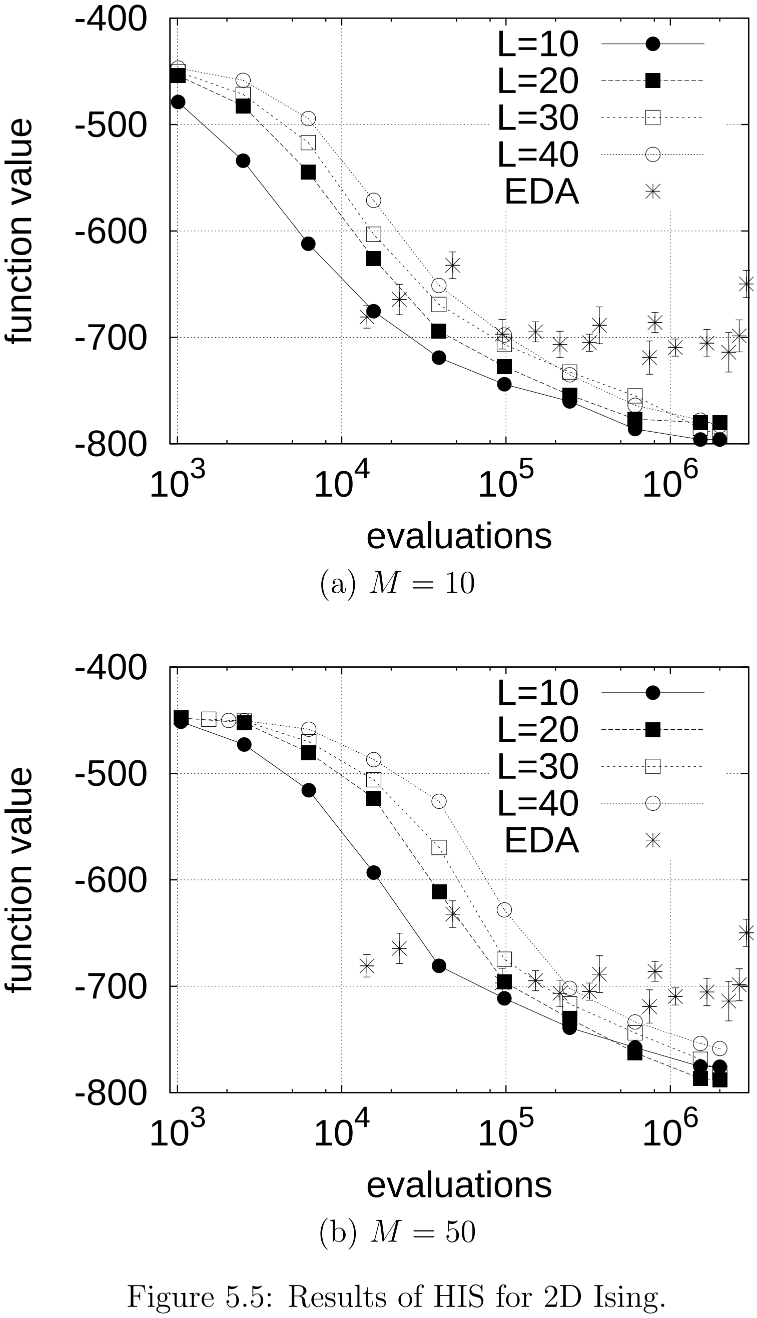


(a) $M = 10$



(b) $M = 50$

Figure 5.4: Results of HIS for 1D Ising.



(a) $M = 10$

(b) $M = 50$

Figure 5.5: Results of HIS for 2D Ising.

with the standard deviation in parenthesis, of the best obtained solutions over ten independent runs. The forth column lists the number of function evaluations.

Figures 5.4 and 5.5 show the results of HIS for the 1D and 2D Ising models, respectively. In each figure, the horizontal axis represents the number of function evaluations, while the vertical axis represents the average cost function value. Each point represents the average cost function value of the best obtained solutions over ten independent runs for the corresponding number of function evaluations performed. The standard deviations are negligibly small and may be ignored. Additionally, the results of EDA are appended for comparison. The points correspond to the results in Tables A.2 or A.3 in Appendix A.

The results for Onemax show that HIS performs as well as EDA. For EDA, M should be set at more than 100; otherwise, EDA can not find the optima. Figures 5.4 and 5.5 show that HIS can find better solutions than EDA. EDA may exhibit faster convergence than HIS; however, given sufficient time (i.e., a sufficient number of function evaluations), HIS can find better solutions than EDA.

5.4 Discussion

5.4.1 Escaping Local Optima

As shown in Figs. 5.4 and 5.5, it is clear that HIS can afford better solutions than EDA. The number of samples employed by HIS for building a probability model is given by

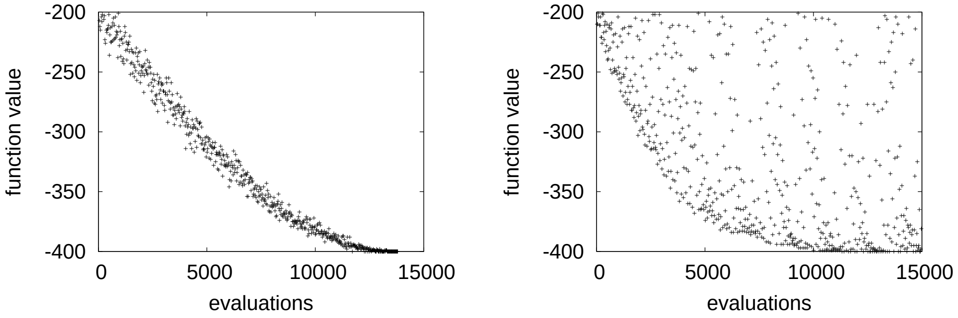
$$3 \times M. \tag{5.10}$$

The number of samples that EDA uses for building a probability model is given by

$$(1 - c) \times M. \tag{5.11}$$

When $M = 10$, HIS uses 30 samples; on the other hand, when $M = 100$ and $c = 0.3$, EDA uses 70 samples. This implies that HIS can escape from local optima by using fewer samples.

In EDA and the EAPM, the entropy of the target distribution is decreased in a stepwise fashion and the target distribution is tracked by a probability model. For tracking the target distribution, the expected log-likelihood must



(a) EDA

(b) HIS

Figure 5.6: Optimization Process of EDA ($M = 100, c = 0.3$) and HIS ($L = 10, M = 10$) for 400-dimensional Onemax.

be estimated. The accuracy of an estimator of the expected log-likelihood is dependent on the accuracy of the approximation of the probability model. Thus, once an inferior probability model is built, the accuracy of the estimator of the log-likelihood with respect to the next target distribution is also compromised. Subsequently, acceptable probability models cannot be generated. This phenomenon can be understood as dropping into local optima.

On the other hand, HIS overcomes this problem by maintaining multiple probability models. In HIS, the larger is the entropy of a target distribution, the easier it is to approximate it. More specifically, low layers tend to have good probability models and high layers tend to have bad probability models. HIS iteratively improves the probability models in the higher layers with samples generated from the lower layers. Thus, if the lower layers have good probability models, the expected log-likelihood can be estimated well at the layers above them. Once a good probability model is built, it tends not to make a change for the worse. Consequently, HIS sequentially improves all the probability models from the lowest layer.

5.4.2 Iterative EDA

The more the number of function evaluations performed, the better the solutions afforded by HIS. This is because the samples generated by HIS always have certain diversity. Figure 5.6 shows the cost function values of the samples generated by HIS and EDA. The horizontal axis represents the number of function evaluations, while the vertical axis represents the cost function value. HIS has no convergence, and therefore, can find the optimum solution eventually. However, this is not an unique advantage of HIS because no convergence can also be realized by iterative EDA.

As the results of EDA for 1D Ising and 2D Ising show, iterative EDAs do not perform as well as HIS because the standard deviations of the best values are insufficiently small. For example, the 10 best obtained solutions in 100 trials of EDA with $M = 3000$ and $c = 0.5$ for the 2D Ising are -746, -736, -732, -732, -730, -730, -730, -728, -726, and -726. If the target distributions are assumed to be previously provided, HIS is an extension of the EAPM. The advantage of HIS is the use of the samples and probability models of other trials, whereas each trial in iterative EDA is independently executed.

5.4.3 Parameters

In sampling-based optimization, there exists a trade-off between the number of function evaluations and the quality of the obtained solutions. In other words, the greater is the number of function evaluations, the better are the solutions afforded. In EDA, the number of function evaluations perfomed until convergence depends on the parameters: the number of generated samples in one sampling and the cutoff rate. If a solution with a certain quality is needed, it becomes necessary to provide good parameters.

On the other hand, HIS does not converge, and the best obtained value is gradually improved. Thus, it can be said that the setting of the parameters in HIS is easier than in EDA. However, both the number of function evaluations necessary and the efficiency of HIS depend on the number of layers. A greater number of layers in HIS affords greater similarity between adjacent target distributions (i.e., $q_l(x)$ and $q_{l-1}(x)$), implying that it is easier for HIS with a greater number of layers to escape from local optima. On the other hand, HIS with a greater number of layers requires more function evaluations because the samples generated from bad probability models are useless, and the probability models in the higher layers tend to be bad at the early stages. The number of layers may be expected to be determined adaptively according to the accuracy of the probability models: this will be the subject of future work.

5.4.4 Computational Cost

HIS can provide better results, but at greater computational cost than EDA. First, HIS requires L times the memory space required by EDA: L number of probability models and L number of sample sets maintained in HIS. Second, HIS consumes greater computational time than EDA: the calculation of the probability of the mixture distribution given by (5.1) requires considerable time.

5.4.5 Mixture Model-based EDAs

In terms of using a mixture distribution, some mixture model-based EDAs such as [31] can be considered similar works. However, they are classified as methods with annealing because they simply split samples generated from one probability model into a number of groups and gradually converge each group, whereas HIS organizes the diversity of all the probability models. Thus, the optimization process of them is almost equivalent to one illustrated in Fig. 5.6-(a).

Note that HIS can simply employ a mixture distribution as the probability model of each target distribution. In terms of statistical estimation, the model error can be reduced by using a mixture model. On the other hand, HIS improves the accuracy of the empirical log-likelihood in terms of importance sampling.

5.4.6 Comparison with Markov Chain Monte Carlo

Calculating the expectation with respect to the distribution of interest is common to Markov chain Monte Carlo methods (MCMC) [2] and EAPM. Table 5.2 briefly shows the relationship between MCMC and EAPM.

The key concepts behind MCMC are local transition, which realizes effective sampling, and designing it as a Markov chain by satisfying *detailed balance*, which guarantees mathematical validity. On the other hand, the principle feature of EAPM is estimating an effective probability distribution and sampling from it. The mathematical validity is guaranteed by importance sampling.

In the practical methods, there exist correspondence relations. For example, simulated annealing (SA) [19] corresponds to general EDA and the

Table 5.2: MCMC and EDA.

	MCMC	EAPM
Mathematical Validity	Detailed Balance	Importance Sampling
Effective Sampling	Local Transition	Estimated Probability Model
Sequential	SA	the EAPM
Parallel	EMC	HIS

EAPM in terms of sequentially tracking a target distribution, and the exchange Monte Carlo method (EMC) [17] corresponds to HIS in terms of sampling from multiple target distributions.

5.5 Summary

This chapter proposed Hierarchical Importance Sampling (HIS), a method that can be used instead of the annealing for the EAPM. Experimental comparisons between HIS and EDA revealed that HIS outperforms EDA. The advantages of HIS can be summarized as follows: (1)it affords better solutions than EDA by escaping from local optima, and (2)it allows the parameters to be set easily.

Chapter 6

Convergence Schedule

6.1 Introduction

One difficulty of the EAPM is to determine the target distributions. This is called the problem of the convergence schedule or also called the annealing schedule. In related works, one promising method is standard deviation schedule (SDS) [25, 26]. However, SDS is based on an empirical rule.

This chapter proposes a novel convergence schedule method and highlights the theoretical aspects, that is, a relationship between the entropy of the target distribution and the Fisher information, which can assess the accuracy of the statistical estimation. The proposed method is designed to improves the accuracy of the statistical estimation. The aim of this chapter is to investigate the efficiency of the proposed convergence schedule.

6.2 Entropy Reduction Schedule (ERS)

6.2.1 Search Space Reduction

The target distributions have an effect for the variance of the importance sampling of (3.10). The objective of the convergence schedule is to reduce the variance of the importance sampling by controling the target distributions.

For simplicity, let us introduce two assumptions. One is that each probability model $p_t(x)$ completely approximates the corresponding target distribution $q_t(x)$, that is, $p_t(x) = q_t(x)$. The other is that each target distribution is a partially uniform distribution. For partially uniform distributions, the search space can be defined by $\Omega = \{x | q(x) \neq 0\}$. It holds that $q(x) = \frac{1}{|\Omega|}$ for

$x \in \Omega$ and $\Omega = Z$, where Z is the normalizing constant. Normally $\Omega_{t+1} \subseteq \Omega_t$ is satisfied in the EAPM, and then $x \in \Omega_{t+1}$ satisfies the following:

$$\frac{q_{t+1}(x)}{p_t(x)} = \frac{q_{t+1}(x)}{q_t(x)} = \frac{|\Omega_t|}{|\Omega_{t+1}|}. \tag{6.1}$$

$(\frac{|\Omega_t|}{|\Omega_{t+1}|})^{-1} = \frac{|\Omega_{t+1}|}{|\Omega_t|}$ can be understood as the accept probability, which means the probability of a sample generated from Ω_t to enter the region Ω_{t+1}. In importance sampling, rejected samples that is $\frac{q_{t+1}(x)}{q_t(x)} = 0$ (i.e., $x \notin \Omega_{t+1}$) have no effect, and therefore a probability distribution that maximizes the accept probability should be selected.

In the EAPM, multiple importance sampling are carried out. The number of the accepted samples in a whole optimization process is written by

$$\sum_{t=1}^{T} M_t \frac{|\Omega_{t+1}|}{|\Omega_t|}, \tag{6.2}$$

where M_t is the number of generated samples in the t-th iteration and T is the number of the iterations. By maximizing (6.2), the following equations are obtained:

$$M_{t-1}\frac{|\Omega_t|}{|\Omega_{t-1}|} = M_t \frac{|\Omega_{t+1}|}{|\Omega_t|}. \tag{6.3}$$

In general, $M_{t-1} = M_t$ and the equations are rewritten by

$$\frac{|\Omega_{t+1}|}{|\Omega_t|} = c, \tag{6.4}$$

where $0 \leq c \leq 1$ is a constant value, called the cutoff ratio. (6.4) is understood that the size of search space is reduced with a common ratio c.

6.2.2 Theoretical Justification

The property of (6.4) is intuitively good, but there remains a question why (6.4) is good from theoretical viewpoints. For answering this question, the asymptotic error of MCI is highlighted.

In the EAPM, an expected log-likelihood,

$$L(\theta) = \int q(x) \log p(x|\theta), \tag{6.5}$$

where θ is the parameter of the probability model $p(x|\theta)$, is calculated via importance sampling as follows:

$$\hat{L}(\theta) = \frac{1}{M} \sum_{p(x)} \frac{q(x)}{p(x)} \log p(x|\theta). \tag{6.6}$$

The true parameter θ^*, which maximizes $L(\theta)$, is satisfies

$$\frac{\partial L}{\partial \theta} = E\left[\frac{\partial}{\partial \theta} \log p(x|\theta^*)\right]_{q(x)} = 0. \tag{6.7}$$

On the other hand, the variance

$$\begin{aligned} \sigma^2 &= \mathrm{Var}\left[\frac{\partial}{\partial \theta} \log p(x|\theta^*)\right]_{q(x)} & (6.8) \\ &= \mathrm{E}\left[\left(\frac{\partial}{\partial \theta} \log p(x|\theta^*)\right)^2\right]_{q(x)} & (6.9) \end{aligned}$$

is called the Fisher information and becomes an assessment of the ML estimator. In importance sampling, the Fisher information is calculated as follows:

$$\begin{aligned} \sigma_{IS}^2 &= \mathrm{Var}\left[\frac{q(x)}{p(x)} \frac{\partial}{\partial \theta} \log p(x|\theta^*)\right]_{p(x)} & (6.10) \\ &= \mathrm{E}\left[\left(\frac{q(x)}{p(x)} \frac{\partial}{\partial \theta} \log p(x|\theta^*)\right)^2\right]_{p(x)}. & (6.11) \end{aligned}$$

By using the assumption of (6.1), the Fisher information in the EAPM is given by

$$\sigma_{IS}^2 = \frac{\Omega_t}{\Omega_{t+1}} \sigma_t^2, \tag{6.12}$$

where $\sigma_t^2 = \mathrm{Var}\left[\frac{\partial}{\partial \theta} \log p(x|\theta^*)\right]_{q_t(x)}$. This shows that the original Fisher information is multiplied by the inverse of the accept probability.

The squared error,

$$\left|\frac{\partial \hat{L}(\theta^*)}{\partial \theta} - \frac{\partial L(\theta^*)}{\partial \theta}\right|^2 = \left|\frac{\partial \hat{L}(\theta^*)}{\partial \theta}\right|^2 \tag{6.13}$$

represents the empirical Fisher information. According to (2.5), the average squared error is written with the accept probability as follows:

$$E\left[\left|\frac{\partial \hat{L}(\theta^*)}{\partial \theta}\right|^2\right] = \frac{\sigma_t^2}{\frac{\Omega_{t+1}}{\Omega_t} M_t}. \tag{6.14}$$

Here, it becomes clear that our maximization condition (6.3) minimizes the sum of the inverse[1] of Fisher information, that is,

$$\sum_t \frac{\Omega_{t+1}}{\Omega_t} M_t \frac{1}{\sigma_t^2}, \tag{6.16}$$

however, under the assumption that each σ_t^2 for the t-th target distribution is equivalent to each other.

6.2.3 Search Space Size and Entropy

In the previous section, we consider only the partially uniform distribution, and the concept of the search space Ω plays the central role. In the EAPM, we control other types of probability distributions such as Boltzmann distributions, and however it seems to be difficult to define the search space for the Boltzmann distribution.

Actually, the size of the search space can be approximately measured by entropy, which is defined as follows:

$$S = -\int q(x) \log q(x) dx. \tag{6.17}$$

This integration can be understood as the geometric average. The geometric average of $q(x)$ with respect to $q(x)$ is given by

$$\bar{q} = \exp \int q(x) \log q(x) dx. \tag{6.18}$$

This is understood as an approximation of a general probability distribution by a partially uniform distribution. Then, the search space size is given by

$$|\Omega| = \log S. \tag{6.19}$$

[1] The inverse of Fisher information has important aspect.

$$\mathrm{Var}[\hat{\theta}] \geq \frac{1}{\sigma^2} \tag{6.15}$$

is well known bound on any unbiased estimator as Cramér-Rao bound.

Finally, from (6.4), the convergence schedule is given by

$$S_{t+1} - S_t = \log c. \tag{6.20}$$

Note that $0 \leq c \leq 1$ and $-\infty \leq \log c \leq 0$. This shows that the entropy is linearly reduced. Hence, the proposed schedule is called the entropy reduction schedule (ERS).

6.2.4 Implementation Issues

In practice, the entropy cannot be calculated exactly, except for uniform distributions. In ERS, entropy is estimated by using MCI. The following sections show the methods to find a parameter value such that the probability distribution has the given entropy for the partially uniform distribution and the Boltzmann distribution.

Partially Uniform Distribution

For partially uniform distributions, the entropy is given by

$$S = \log Z. \tag{6.21}$$

Then, the objective is to solve the following equation:

$$\begin{aligned} Z^* &= \hat{Z}(\tilde{f}), & (6.22)\\ &= \frac{1}{M}\sum_{p(x)} \frac{\tilde{q}(x|\tilde{f})}{p(x)} & (6.23) \end{aligned}$$

where Z^* is the given search space size, $\hat{Z}$ is the estimated normalizing constant, and $\tilde{q}(x|\tilde{f})$ is defined by (3.2).

The estimator of the normalizing constant is a monotonically decreasing step function with respect to $\tilde{f}$, and its change-points are given by $f(x_1)\cdots f(x_M)$, where $x_1 \cdots x_M$ are the given samples. Thus, the solution is selected from $f(x_1)\cdots f(x_M)$. Assuming $f(x_1) < \cdots < f(x_M)$ without loss of generality, we have the following:

$$\hat{Z}(\tilde{f}(x_{i+1})) = \hat{Z}(\tilde{f}(x_i)) + \frac{1}{M \times p_m(x_{i+1})}. \tag{6.24}$$

A linear search on $f(x_1)\cdots f(x_M)$ can afford an approximate solution. In the experiments, for $\tilde{f}$, we select $f(x_k)$ such that $\hat{Z}(f(x_k)) - Z^*$ is minimized under $Z^* < \hat{Z}(f(x_k))$.

Boltzmann Distribution

For Boltzmann distributions, the entropy is given by

$$S = \bar{f}\beta + \log Z, \tag{6.25}$$

where $\bar{f} = \int q(x)f(x)dx$. Its derivative is given by

$$\frac{\partial S}{\partial \beta} = -\sigma^2 \beta, \tag{6.26}$$

where $\sigma^2 = \mathrm{Var}[f(x)]_{q(x)}$ and this can be calculated via importance sampling. Our approach is based on the gradient. By using the assumption that σ^2 is a constant value, the difference of the entropy is approximately given by

$$\Delta S \quad \simeq \quad -\sigma^2 \int \beta \, d\beta. \tag{6.27}$$

Then, we obtain the update rule

$$\beta_{t+1} = \sqrt{\beta_t^2 - \frac{\Delta S}{\sigma_t^2}}. \tag{6.28}$$

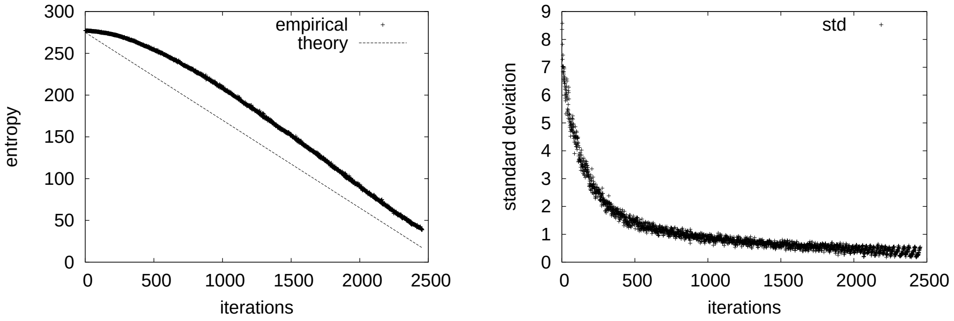
This is the single step updating. Of course we can employ a multi-step updating way.

6.3 Experiments

For the basic framework, the resampling population model (RPM) [14] is employed because it is experimentally confirmed that RPM is more efficient than the EAPM. the EAPM has one parameter that is the number of generated samples in one sampling, denoted by M. RPM has one additional parameter to the EAPM. That is the number of the stored samples, denoted by N. ERS has the parameter of the cutoff ratio defined in (6.4), and denoted by c. The difference of entropy is obtained by $\Delta S = \log c$.

6.3.1 Basic Property

This section describes experiments conducted to investigate the basic property of ERS. The parameters are setup as $N = 200$, $M = 200$, $c = 0.1$ ($\Delta S = \log 0.1 \simeq 0.105$). The parameters are experimentally determined so that the Monte Carlo error is sufficiently ignored. Two cases which employ the partially uniform distribution and the Boltzmann distribution, respectively, are experimented.



(a) Entropy

(b) Standard Deviation

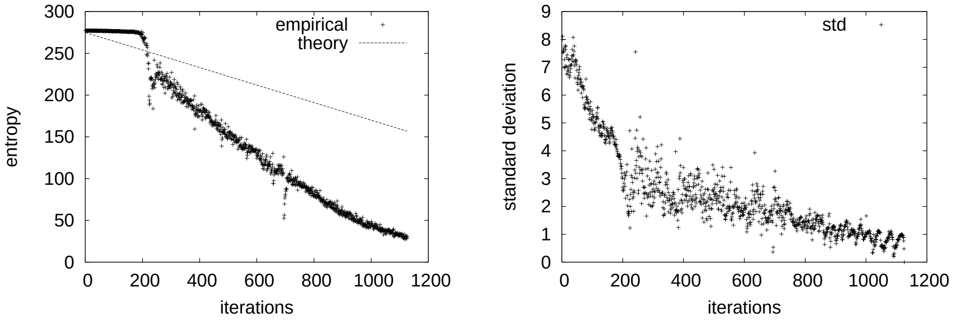
Figure 6.1: A typical evolution for onemax in using partially uniform distributions.

Result

The Left-hand side figures of Figs. 6.1, $\cdots$, 6.6 show the evolution of the entropy. In each figure, the vertical axis represents the empirical entropy. The horizontal axis represents the number of iterations. In theory, the entropy is linearly reduced. The difference is $\log 0.1 \simeq 0.105$ and the line in each figure represents $y = 0.105x + b$, where b is fitted to the data.

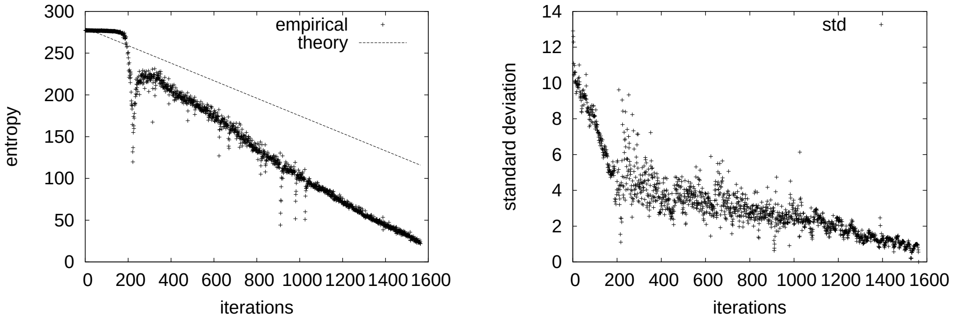
For onemax, the theoretical entropy transition is almost realized in both cases, that is, the partially uniform distribution and the Boltzmann distribution cases. However, for 1D and 2D Ising model, the realized entropy transition is different from the theoretical one in both cases.

We show the evolution of the standard deviation of the cost function value of the generated samples in the right-hand side figure of each figure. ERS with the partially uniform distribution, the difficulty of adjusting the threshold parameter described in Section 6.2.4 depends on the variance of the cost function value of the generated samples. ERS with the Boltzmann distribution assumes that the variance of the cost function value with respect to the current target distribution is not changed if the β is slightly moved. The rapid change can be seen for 1D and 2D Ising model and, at that time, the entropy is quickly decreased in both cases.



(a) Entropy (b) Standard Deviation

Figure 6.2: A typical evolution for 1D Ising in using partially uniform distributions.



(a) Entropy (b) Standard Deviation

Figure 6.3: A typical evolution for 2D Ising in using partially distributions.

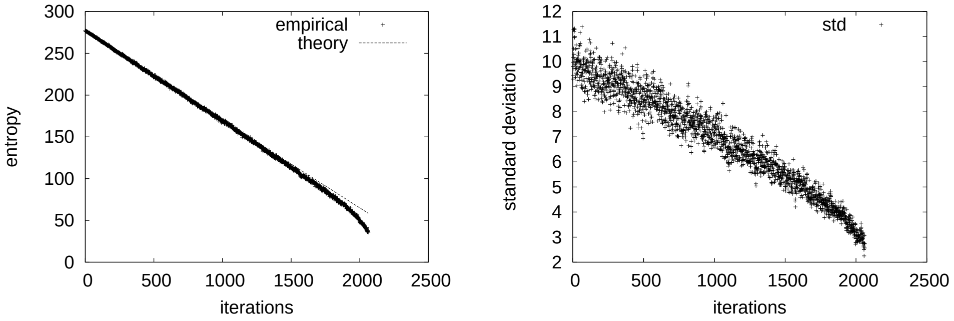
6.3.2 Comparison with Standard Deviation Schedule

In this section, ERS with the Boltzmann distribution is compared with SDS, which is a convergence schedule derived from experimental rule of genetic algorithms. SDS is defined by

$$\beta_{t+1} = \beta_t + \sqrt{\frac{d}{\sigma_t^2}} \tag{6.29}$$

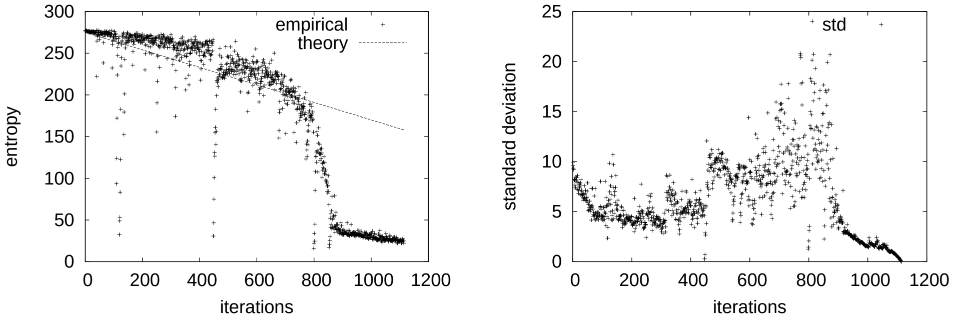
where σ_t^2 is the variance of the cost function value and d is a parameter. If $\beta_t = 0$, ERS is equivalent to SDS. The comparison is also shown in Table 6.1

For each schedule, the parameters are set as $N = 200$ and $M = 200$. For ERS, $c = 0.4,\ 0.2,\ 0.1,\ 0.05$. Note that $\Delta S = \log c$ and the parameter d of



(a) Entropy (b) Standard Deviation

Figure 6.4: A typical evolution for onemax in using Boltzmann distributions.



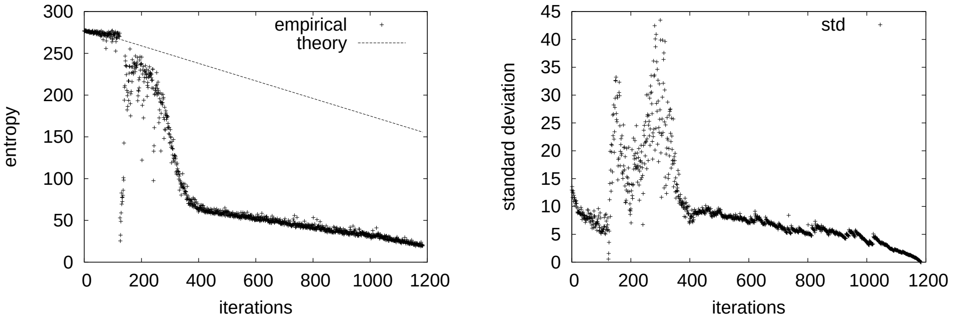
(a) Entropy (b) Standard Deviation

Figure 6.5: A typical evolution for 1D Ising in using Boltzmann distributions.

SDS corresponds to $-\Delta S$ if $\beta = 0$. For SDS, $d = -0.001 \times \log c$. This is experimentally determined so that the number of function evaluations taken until the optimization converge becomes almost the same number.

Table 6.1: Comparison between ERS and SDS

ERS	$(\beta_{t+1} - \beta_t)^2 = \frac{d}{\sigma_t^2}$
SDS	$\beta_{t+1}^2 - \beta_t^2 = -\frac{\Delta S}{\sigma_t^2}$



(a) Entropy (b) Standard Deviation

Figure 6.6: A typical evolution for 2D Ising in using Boltzmann distributions.

Result

Figure 6.7 shows the result. The vertical axis represents the cost function value of the best obtained solution. The horizontal axis represents the number of function evaluations. The figure shows that both schedule are almost equivalent in terms of the cost function value of the best obtained solution and, however, ERS consumes slightly smaller number of function evaluations than SDS.

6.4 Discussion

6.4.1 Numerical Calculation Error

In some ideal situations such as the onemax cases, the entropy of the target distribution can be controlled efficiently. However, in some cases such as the 1D and 2D Ising cases, the control of the entropy becomes difficult. If we have an enough number of samples and enough computational time, the numerical method for adjusting the threshold or the inverse temperature can realize the given entropy. The results show that ERS outperforms SDS. Hence the numerical method is enough practical and this problem of the accuracy of the control may not be important.

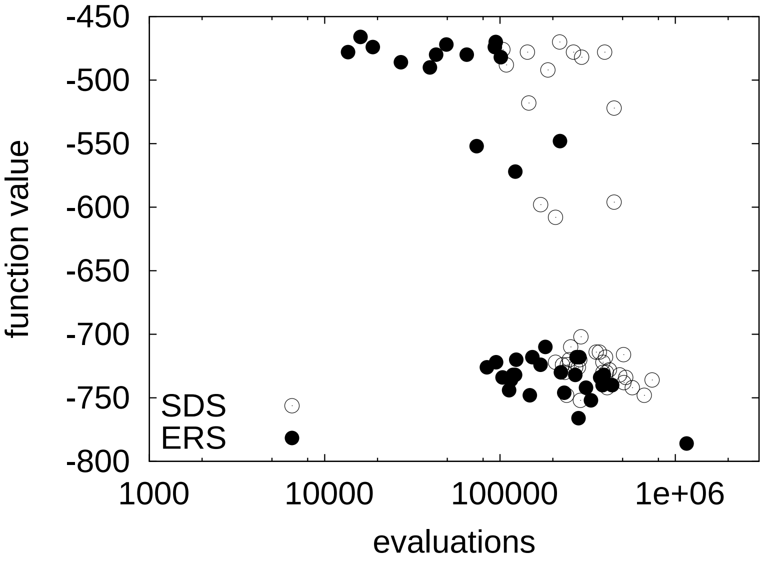


Figure 6.7: The result of standard deviation schedule and entropy reduction schedule.

6.4.2 Linear Time Convergence

If the entropy is linearly decreased, the algorithm converge in linear time for an exponential size of the search space. Hence, the EAPM with ERS is designed as linear time algorithm. Unfortunately, there is no guarantee to converge in linear time because the entropy cannot be controlled precisely in practice. At least, in our experiments, the convergence is faster than the theoretical evolution or the almost same. This property is useful for predicting the convergence time.

6.4.3 Application

Actually, ERS is equivalent to the truncation selection of EDA. In the EAPM, RPM, and HIS, ERS provides a novel method for controling the target distribution. ERS is a general convergence framework.

6.5 Summary

The entropy reduction schedule (ERS) is based on the following two assumptions: (1)the target distributions are partially uniform distributions and (2)the probability models perfectly approximates the target distributions. Under these assumptions, this chapter have revealed the relationship between entropy and Fisher information. As a result, we have obtained ERS.

In ERS, the entropy is decreased linearly and this means linear time convergence for an exponential size of the search space in theory. In practice, it is difficult to exactly realize the target distribution with a given entropy and the linear time convergence is not guaranteed. Through experiments, the proposed numerical method seems to work well but not perfectly and it has been revealed that ERS outperforms standard deviation schedule.

Chapter 7

Advanced Experimental Analysis

7.1 Introduction

In the previous chapters, onemax function, 1D and 2D Ising models are employed for the benchmark problems. The difference among the three problems is the number of correlations, which has an effect on the complexity of the cost function and the target distributions. However, there are some different types of the difficulty of optimization problems. Hence, the objective of this chapter is to conduct experiments to reveal the comprehensive effectiveness and property of RPM and HIS by using different types of problems.

In the following, first, this chapter approaches towards the problems of 2D Ising with frustration. The frustration of 2D Ising has an effect of instability, which means increasing the number of solutions which have the same cost function value.

Second, this chapter approaches towards continuous problems. Continuous problems seems to be more difficult than discrete problems. In Ising models, undetected correlations are the difficulty. In continuous space, additionally, the complexity of each dimension can have an effect on the difficulty.

7.2 Advanced Benchmark Problems

7.2.1 Frustration

2D Ising model with Frustration

In 1D or 2D Ising models, the connections, which are defined by $J(x_i, x_j)$ in (4.27), can be seen as constraints and the cost function value represents the number of satisfied connections. The frustration means existing unsatisfied constraints in any solutions. This situation can be easily realized by changing a part of connections $J(x_i, x_j)$ to

$$J^{-}(x_i, x_j) = \begin{cases} 0 & x_i = x_j \\ 1 & x_i \neq x_j \end{cases} . \tag{7.1}$$

In 2D Ising models without frustration, the optimum cost function value is given by $-2d$, where d is the number of the dimensions. If just one connection is changed to be J^{-} the cost function value of the optimum solution becomes $-2d + 1$. Hence, if just $k\%\,(k < 50)$ connections are changed to be J^{-} and they are independent, the cost function value of the optimum solution can become $-2d(100 - k)/100$. However, if there exists some regularities, for example, horizontal connections are J and vertical connections are J^{-}, the cost function value of the optimum solution becomes $\{-2d(100-k)/100\}-\alpha$, where α is an improvement term and its value is difficult to calculate in general.

7.2.2 Continuous Problems

This section introduces two continuous problems.

Rosenbrock Function

Rosenbrock Function [41] is defined as follows:

$$f(x) = \sum_{i=1}^{d-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2). \tag{7.2}$$

If the second term is ignored, the condition of $f'(x) = 0$ is $x_i^2 = x_{i+1}$. This feature is shown in Fig. 7.1 The property of this problem is ill-scaled, which means each dimension is correlated to each other. This function can be seen almost as unimodal, which means that there is no local optima.

Rastrigin Function

Rastrigin Function is defined as follows:

$$f(x) = 10d + \sum_{i=1}^{d} \{x_i^2 - 10\cos(2\pi x_i)\}. \tag{7.3}$$

The feature is that there are many local optima as shown in Fig. 7.2 and, therefore, this is quite hard problem for gradient-based methods. There is no correlation.

7.3 Experiments on Frustration

7.3.1 Experimental Setup

Three problems are employed. One is a normal 2D Ising model. For the others, 5% are 10% of the conncetions are changed to J^-, respectively. The number of the dimension is 400.

EDA, RPM and HIS are compared. Their parameters are determined as follows:

- EDA
  - N: 100, 500, 1000, 3000, or 6000.
  - c: 0.1, 0.3, or 0.5.
- RPM
  - N: 10, 50, 100, 200, 300, 400, or 500.
  - $M(\leq N)$: 10, 50, 100, 200, 300, 400, or 500.
  - c: 0.01 or 0.05
- HIS
  - M: 10
  - L: 30

The setting of the probability models is the same as the previous chapters. For each setting, we perform ten independent runs.

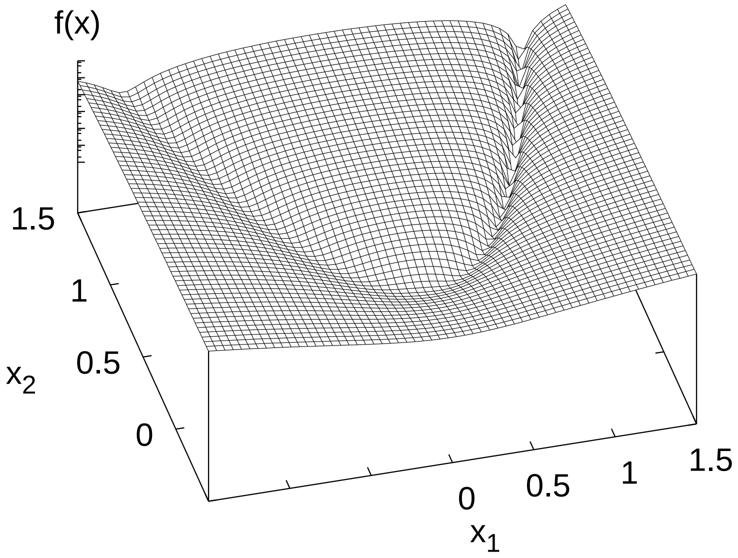


Figure 7.1: 2D Rosenbrock Function

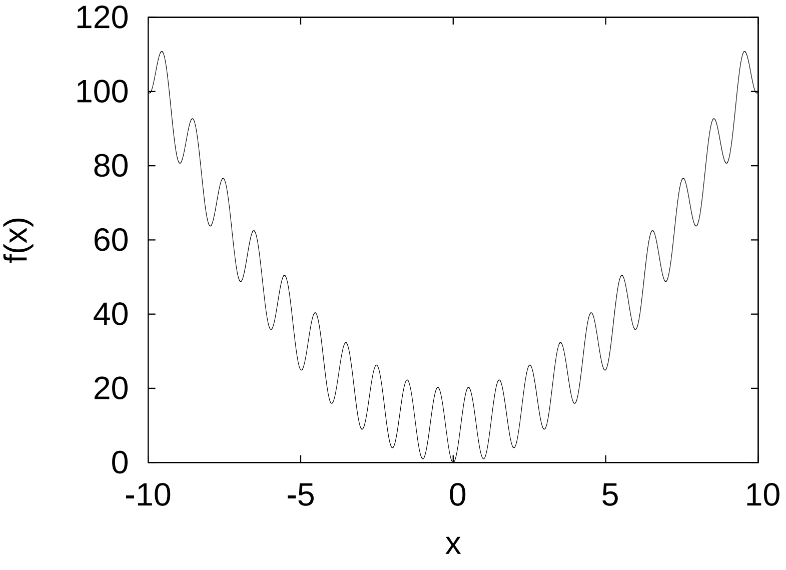


Figure 7.2: 1D Rastrigin Function

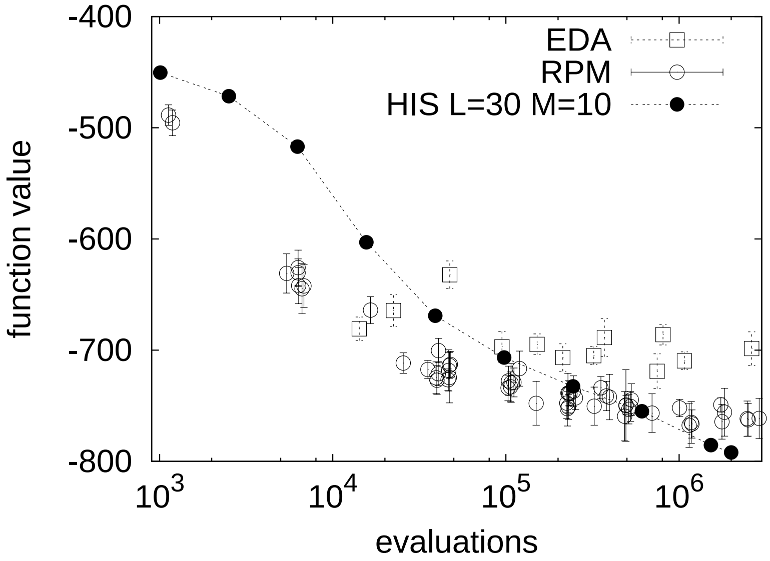


Figure 7.3: 2D Ising with $p(J^-) = 0$.

7.3.2 Results

The results are shown in Figs. 7.3, 7.4, and 7.5. The horizontal axis represents the number of function evaluations performed and the vertical axis represents the cost function value of the best obtained solution. Additionally, the horizontal lines in Figs. 7.4 and 7.5 show the estimated optimal cost function values given by $-2 \times d \times (1 - p(J^-))$.

The results show that the frustration cases have no serious problem since the estimated optimal values are almost obtained. However, in 2D Ising model with changing 10% connections, the difference between the cost function value of the obtained solutions of RPM and HIS becomes smaller than the others.

7.4 Experiments on Continuous Problems

7.4.1 Experimental Setup

EDA, RPM and HIS are compared. Their parameters are determined as follows:

- EDA

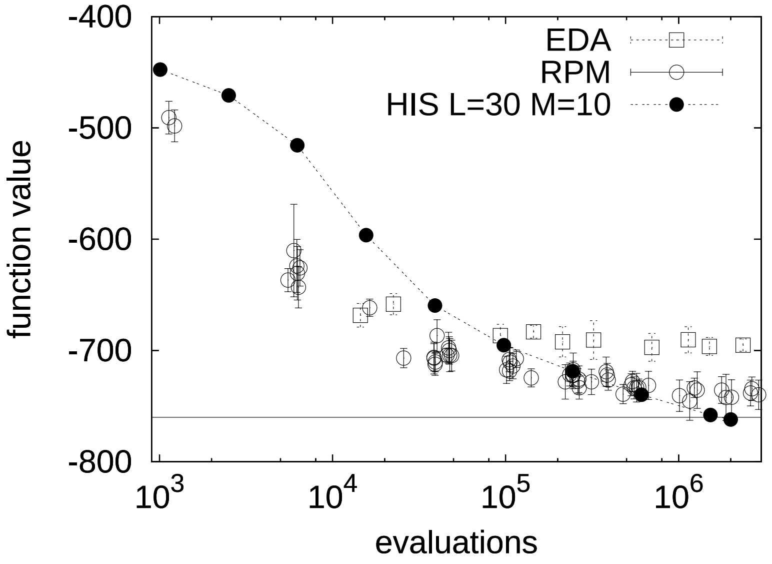


Figure 7.4: 2D Ising with $p(J^-) = 0.05$. The horizontal line is $y = 800 \times 0.95$ as a lower bound of the optimal solution.

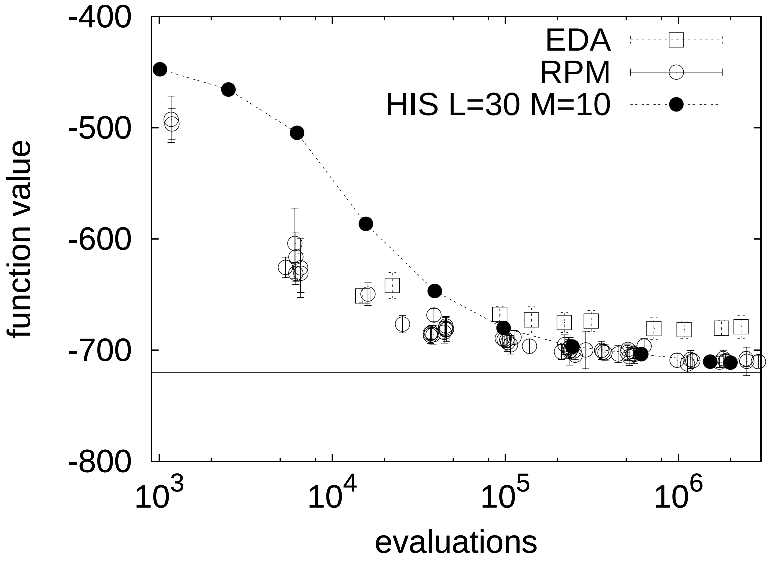


Figure 7.5: 2D Ising with $p(J^-) = 0.1$. The horizontal line is $y = 800 \times 0.9$ as a lower bound of the optimal solution.

  - M: 500, 1000 or 3000.
  - c: 0.3.

- RPM
  - $M = N$: 50, 100 or 200.
  - c: 0.01.

- HIS
  - M: 10 (in UG cases) or 100 (in MG cases).
  - L: 20, 40 or 60.

For probability models, two types of Gaussian distributions are employed. One is univariate Gaussian (UG), which means we assume that each dimension is independent and, hence, estimate only the diagonal elements of the covariance matrix. Another is the normal multivariate Gaussian (MG).

The initialized region is $[-10, 10]^d$. Additionally, we experiment the case where the initialized region is $[0, 10]^d$ for Rastrigin function. This setting is named shifted-Rastrigin (S-Rastrigin). The number of dimensions is 10. For each setting, we perform ten independent runs.

7.4.2 Results

The number of successful trials in ten independent runs are shown in Figs. 7.1, 7.2, and 7.3. The success means to find a solution with cost function value less than 10^{-4}. The average and the standard deviation of the number of function evaluations until convergence of EDA and RPM are shown in Figs. 7.4 and 7.7. On the other hand, Fig. 7.9 shows the number of function evaluations until finding global optima of HIS. Note that the HIS cases where the global optima have not been found are ignored. The optimization task of HIS is stopped if the number of function evaluations become more than 10^7. Figures 7.4, 7.7, and 7.9 show the average cost function value of the obtained best solutions.

The results show that EDA cannot find the optima in Rosenbrock function where as can find the optima in Rastrigin function. In S-Rastrigin, EDA using MG cannot find the optimal solution where as EDA using UG can find. When using UG, only RPM can find the optima in Rosenbrock function.

Table 7.1: The Number of Successful Trials Finding Global Optima of EDA.

	M	Rosenbrock	Rastrigin	S-Rastrigin
UG	500	0	10	10
	1000	0	10	10
	3000	0	10	10
MG	500	0	10	0
	1000	0	10	0
	3000	0	10	0

Table 7.2: The Number of Successful Trials Finding Global Optima of RPM.

	$N = M$	Rosenbrock	Rastrigin
UG	50	9	0
	100	10	0
	200	10	1
MG	50	10	0
	100	10	0
	200	10	0

Table 7.3: The Number of Successful Trials Finding Global Optima of HIS.

	L	Rosenbrock	Rastrigin	S-Rastrigin
UG	20	0	10	10
	40	0	10	10
	60	0	10	10
MG	20	10	5	2
	40	10	6	7
	60	10	10	10

Table 7.4: The avege number of function evaluations of EDA.

	M	Rosenbrock	Rastrigin
UG	500	1.36E+05 (1.38E+03)	1.52E+05 (7.56E+03)
	1000	2.73E+05 (1.70E+03)	2.98E+05 (4.84E+03)
	3000	8.21E+05 (3.60E+03)	9.02E+05 (2.32E+04)
MG	500	1.21E+05 (5.39E+03)	1.44E+05 (1.75E+03)
	1000	2.38E+05 (1.76E+03)	2.90E+05 (7.50E+03)
	3000	7.09E+05 (4.07E+03)	8.99E+05 (8.57E+03)

Table 7.5: The avege number of function evaluations of EDA.

	M	Rosenbrock	Rastrigin
UG	500	8.30E+00 (2.41E-02)	7.64E-05 (1.38E-05)
	1000	8.28E+00 (1.50E-02)	7.38E-05 (9.15E-06)
	3000	8.25E+00 (4.27E-02)	8.22E-05 (1.07E-05)
MG	500	7.89E+00 (1.59E-01)	8.30E-05 (1.95E-05)
	1000	7.98E+00 (9.23E-02)	7.88E-05 (1.20E-05)
	3000	7.91E+00 (6.77E-02)	7.36E-05 (1.22E-05)

Table 7.6: The average cost function value of the obtained solutions and the average number of function evaluations of EDA for S-Rastrigin.

	M	Value	Num. of Evaluations
UG	500	8.36E-05 (1.44E-05)	1.79E+05 (9.67E+03)
	1000	7.38E-05 (1.51E-05)	3.39E+05 (1.51E+04)
	3000	8.48E-05 (1.01E-05)	9.98E+05 (1.76E+04)
MG	500	5.16E+01 (6.93E+00)	4.85E+06 (4.72E+06)
	1000	4.63E+01 (4.98E+00)	3.28E+06 (4.40E+06)
	3000	3.93E+01 (8.03E-01)	1.09E+06 (6.35E+04)

Table 7.7: The avege number of function evaluations of RPM.

	M=N	Rosenbrock		Rastrigin	
UG	50	7.93E+05	(5.31E+05)	4.72E+05	(2.53E+05)
	100	1.71E+06	(6.61E+04)	2.92E+06	(2.07E+06)
	200	3.88E+06	(1.56E+05)	1.60E+07	(2.56E+05)
MG	50	3.06E+05	(7.69E+03)	3.49E+05	(9.27E+04)
	100	1.46E+06	(3.13E+04)	2.71E+06	(5.32E+04)
	200	4.00E+06	(1.04E+05)	1.10E+07	(2.50E+05)

Table 7.8: The average cost function value of the obtained solutions of RPM.

	M=N	Rosenbrock		Rastrigin	
UG	50	2.95E+02	(8.84E+02)	1.48E+01	(1.39E+01)
	100	2.31E-04	(2.70E-05)	1.67E+01	(1.63E+01)
	200	1.37E-04	(1.68E-05)	2.59E+00	(1.95E+00)
MG	50	7.94E-05	(1.48E-05)	1.03E+01	(9.19E+00)
	100	8.45E-05	(1.06E-05)	1.09E+01	(4.36E+00)
	200	7.84E-05	(1.57E-05)	6.57E+00	(3.12E+00)

On the other hand, Rastrigin function is too difficult for RPM HIS can find the optima both Rastrigin and S-Rastrigin if the number of the layers is sufficiently large. In general results depend on the number of function evaluations, that is, the convergence speed. The difference of the number of function evaluations among the settings is not significantly large in log-scale.

7.5 Discussion

7.5.1 Robustness against Frustration

The frustration has an effect on the instability. Through experiments, it seems that the frustration does not cause serious problems because the cost function value of the obtained solution is near the estimated optimal cost function value. The instability can be removed through the annealing process.

7.5.2 Difficulty in 2D Ising Model

The difference among the thee method is reduced by adding the frustration. This can be explained by the size of the clusters. Basically, the optimization

Table 7.9: The avege number of function evaluations of HIS.

	L	Rosenbrock		Rastrigin	
UG	20	1.00E+07	(0.00E+00)	2.01E+05	(5.88E+04)
	40	1.00E+07	(0.00E+00)	4.23E+05	(1.84E+05)
	60	1.00E+07	(0.00E+00)	5.94E+05	(1.75E+05)
MG	20	2.74E+05	(2.64E+04)	2.40E+06	(3.08E+06)
	40	5.09E+05	(3.56E+04)	2.65E+06	(1.77E+06)
	60	7.91E+05	(6.64E+04)	2.23E+06	(4.28E+05)

Table 7.10: The average cost function value of the obtained solutions of HIS.

	L	Rosenbrock		Rastrigin	
UG	20	4.13E+00	(8.12E-02)	8.80E-05	(1.47E-05)
	40	4.07E+00	(5.05E-01)	9.05E-05	(8.98E-06)
	60	7.02E+00	(6.42E-01)	8.47E-05	(1.53E-05)
MG	20	6.64E-05	(1.61E-05)	5.97E-01	(6.60E-01)
	40	7.21E-05	(2.24E-05)	6.97E-01	(8.95E-01)
	60	7.89E-05	(1.37E-05)	8.33E-05	(1.83E-05)

Table 7.11: The average cost function value of the obtained solutions and the average number of function evaluations of HIS for S-Rastrigin.

	L	Value		Num. of Evaluations	
UG	20	7.35E-05	(2.66E-05)	1.93E+05	(9.52E+04)
	40	8.87E-05	(1.17E-05)	4.15E+05	(1.45E+05)
	60	8.96E-05	(1.12E-05)	5.40E+05	(1.47E+05)
MG	20	1.49E+00	(1.20E+00)	2.60E+05	(3.90E+06)
	40	3.98E-01	(6.60E-01)	2.53E+06	(4.16E+06)
	60	8.76E-05	(1.10E-05)	1.96E+06	(9.47E+05)

process of EAPM for 2D Ising consists of three phases. The first phase is the transition from random to some clusters. A cluster means a set of connected variables which have the same value. In the second phase, the clusters are merged. In the third phase, two clusters remains and consequently one has to be eliminated in order to obtain the optimal solution. Actually, to eliminate a big cluster is quite difficult for EAPM. However, the frustration breaks the large clusters.

7.5.3 RPM: Robustness against Model Error

For Rosenbrock function, RPM can find the global optima in spite of using the inappropriate probability model, that is, UG, whereas HIS cannot in the same condition. This shows the robustness against the model error, which means the difference between the target distribution and the probability model. RPM can store a part of the historical samples and this mechanism do not depend on the probability model in theory.

7.5.4 HIS: Robustness against Local Optima

HIS works well for Rastrigin, whereas RPM do not. The reason would be the presence of the local optima. The number of local optima around the global optima is at least $3^d - 1$ and RPM drops into one of them. In practice $d = 5$, RPM can find the global optima.

The reason of HIS finding the global optima is preserving the best probability model. This can be understood as preserving the best obtained samples, whereas RPM may discard the best samples in order to remove the bias. Note that HIS removes the bias according to importance sampling with different manner from RPM.

7.5.5 Estimation Bias in EDA

In cases using MG, EDA seems to be effective for Rastrigin function. However, we have to note the bias of EDA. In importance sampling, EDA assumes the following:

$$\frac{q_{t+1}(x)}{p_t(x)} \log p(x|w) \simeq \frac{q_{t+1}(x)}{q_t(x)} \log p(x|w), \tag{7.4}$$

	Model Error	Local Optima	Removing Bias
EDA	?	?	Bad
RPM	Good	Bad	Good
HIS	Bad	Good	Good

where target distributions are supposed to be partially uniform distributions. The simple aspect is the assumption of

$$q_t(x) = p_t(x). \tag{7.5}$$

This is the assumption that the statistical estimation is completely successful. This has an effect on the variance reduction but we estimate the following distribution:

$$q_{t+1}^{bias}(x) = \frac{p_t(x)}{q_t(x)} q_{t+1}(x). \tag{7.6}$$

This can be the reason of the failure in Rosenbrock function and S-Rastrigin function with MG.

7.6 Summary

This chapter provides experimental results to investigate additional properties of EDA, RPM and HIS. The experiments have revealed that RPM has the robustness against the model error and HIS has the robustness against local optima and.

Chapter 8

Conclusions and Future Directions

8.1 Summary of This Thesis

This thesis has theoretically approached to establish fundamental technique of EAPM. This thesis has focused on the importance sampling manner, which calculates the empirical log-likelihoods with respect to the target distribution for statistical estimation, in EAPM and has improved the importance sampling estimator in the three manners. In the following, each section summarizes each improvement.

8.1.1 RPM for Model Error

The EAPM generates many samples and they are used once. Since Monte Carlo estimator becomes improved as the number of the samples increases, it is important to reuse the historical samples. In evolutionary algorithms, a method to reuse the historical samples is called a population mechanism, and, this is a fundamental problem not only in EAPM but also in evolutionary algorithms (EAs). This thesis has provided a theoretical method, resampling population model (RPM), for this purpose.

In employing the historical samples, the bias of the reused samples is the problem. In the EAPM, the bias is removed by using importance sampling, which requires the probability distribution of the samples. However, the probability distribution of the selected historical samples is unknown.

RPM consists of the weighting and resampling procedures. The weighting is a kind of importance sampling calculation and provides a manner to mix the current samples and a part of the historical samples. On the other hand, the resampling can change the size of the maintained historical samples without changing the distribution of them. RPM is an extension of the EAPM (CE).

Through experiments, it is confirmed that PRM outperforms conventional methods to maintain the historical samples such as the full replacement, the elitist replacement, and the restricted tournament replacement. Additionally, through experiments using Rosenbrock function, it has revealed that especially RPM has the robustness against the model error.

8.1.2 HIS for Local Optima

The difference between the target distribution and the estimated distribution is important because the large difference means that the next generated samples will be strongly biased and the next estimated distribution is also different from the next target distribution. This phenomenon can be understood as dropping into local optima. In the general annealing process of the EAPM, to correct the difference is quite difficult and the only method is to restart after convergence, that is, the multi-starting. To overcome the problem of the local optima, this thesis has proposed another convergence method, hierarchical importance sampling, instead of annealing.

It can be intuitively understood that high entropy samples are effective for escaping from local optima. Hence, this thesis has proposed hierarchical importance samling (HIS) that generates samples with different entropies simultaneously and calculates the empirical log-likelihood from the mixed samples. The difficulty is to calculate the empirical log-likelihood from mixed samples. This can be calculated via importance sampling with a mixture distribution. This is derived simply according to the definition of importance sampling and this implies there there can be no risk at this point. HIS is a mathematical extension of the EAPM (CE).

Through experiments, it is confirmed that the proposed method outperforms EDA and surely the EAPM (CE). The point is iteratively estimating the probability distribution towards the same target distribution with mixing high entropy samples. This method is understood as repeating the EAPM ,that is, multi-starting, with using the information of the previous trials. Additionally, through experiments using Rastrigin function, it has revealed that especially HIS has the robustness against the local optima.

8.1.3 ERS and Linear Time Convergence

To determine and control the annealing speed is a basic problem of the EAPM. In fact, it has not been well known what the factor is. This thesis has revealed the fundamental relationship between the entropy of the target distribution and the Fisher information, and, consequently, has proposed a general annealing schedule, entropy reduction schedule (ERS).

Basically, the entropy of the target distribution represents the size of the region where samples are generated. In other words, the entropy represents the size of the search space. In terms of the statistical estimation, the approximately optimal convergence speed is realized by lineally reducing the entropy, and this is called ERS. If linearly reduction of the entropy is realized, the algorithm converges in linear time for the number of the dimensions. However, in practice, the numerical calculation of the entropy is not easy. This thesis has proposed numerical calculation methods for this purpose.

Through comparison with a conventional method, standard deviation schedule, the effectiveness of the proposed convergence schedule is confirmed. ERS will converge in linear time in theory, but, in practice, the entropy cannot be controlled exactly. However, experiments show that the convergence time can be approximately estimated as linear.

8.2 Future Directions

8.2.1 HIS with RPM

The advantages of RPM and HIS are different from each other. Hence, it is expected that a new method is obtained by combining RPM and HIS. Actually, we can simply combine HIS with RPM. However, there exists some problems. One is which samples should be emplyed for determining the target distribtuions. Another is whether the simple probability model is appropriate. In RPM, we have a part of the histrical samples and they do not depend on the probability model. Hence, the distribution of the maintained historical samples may be too complex. In RPM, the size of the maintained historical samples is naturally decreased, whereas not in HIS.

8.2.2 Constraints

In this thesis, constraints are basically out of scope. However, some problems such as traveling salesman problems (TSP)

have constraints. Unfortunatel, the EAPM and our extensions cannot be directly applied to TSP and also other constrained problems. Some works [40,45] are proposed but we have not obtained sufficiently theoretical method yet. To deal with constraints is one of the most important current problems of EAPM.

8.2.3 Statistical Estimation

This thesis focuses not on the statistical estimation but on the Monte Carlo framework, that is, importance sampling. However, the quality of the samples strongly depends on the accuracy of the statistical estimation. Some methods for EAPM are proposed [20]. However, there can remain problems such as the control of the model complexity, that is, model selection, and learning mixture distribution with effective computational method.

8.2.4 Evolution into Reinforcement Learning

[40] has pointed that the EAPM and the reinforcement learning, which solves Bellman Equation with a sampling method, share the same concept, that is, importance sampling and the annealing. Hence, the reinforcement learning may be extended by the similar manner as our proposed extensions.

8.2.5 Killer Applications

The EAPM has been practically and theoretically developed so far. However, there remains the most important question: “ Are EAPM really better than other methods? ” Trying to answer this question, we can understand advantages and disadvantages of the EAPM, and subsequently we can realize further improvements.

Appendix A

Comparison between EDA and the EAPM (CE)

This appendix provides experimental studies on comparisons between EDA and the EAPM. In this appendix, the EAPM is denoted by CE. In spite of the theoretical aspects of CE, CE does not work well in practice.

A.1 Experimental Setup

A.1.1 EDA Setting

We employ UMDA [29] as the EDA. Thus, the probability model is defined as

$$p(x|w) = \prod_{i=0}^{i=d-1} p(x_i|w_i) \tag{A.1}$$

and ML estimation is employed for building the probability models. Here, the learning rate α is introduced. The parameter w is updated by the following equation:

$$w_{new} = (1-\alpha)w_{old} + \alpha w_{ML}, \tag{A.2}$$

where w_{new}, w_{old}, w_{ML} are the new parameter, previous parameter, and ML estimator, respectively. This mechanism affords stable estimation.

The selection operator employed is the truncation selection operator. The truncation selection operator includes the cutoff rate parameter c, which represents the percentage of samples that are removed. For example, if $c =$ 0.3 and the number of generated samples is 100, then the best $70 = 100 \times$

$(1-0.3)$ samples are selected and the rest are discarded. All the parameter settings are described as follows:

- The number of generated samples in one sampling M: 100, 500, 1000, 3000, or 6000.
- Cutoff rate c: 0.1, 0.3, or 0.5.
- Learning rate α: 0.5.

These values are experimentally determined.

A.1.2 CE Setting

CE uses the same probability model and estimation method as EDA. However, instead of truncation selection, CE employs the $(1-\delta)$-quantile method [40], which selects the best $k = M \times \delta$ samples, where M is the number of generated samples, and removes the rest. Truncation selection and the $(1-\delta)$-quantile method are basically the same: the parameter $(1-\delta)$ corresponds to the cutoff rate in truncation selection. Thus, $(1-\delta)$ is referred to as the cutoff parameter in this paper. All the parameter settings are described as follows:

- The number of generated samples in one sampling M: 100, 500, 1000, 3000, or 6000.
- Cutoff rate $c = 1-\delta$: 0.3, 0.5, or 0.7.
- Learning rate α: 0.5.

These values are experimentally determined.

A.2 Results

Tables A.1, A.2, and A.3 show the results of EDA. Tables A.4, A.5, and A.6 show the results of CE. The values in the first and second columns are the number of generated samples per sampling and the cutoff rate value, respectively. The third column lists the average cost function value, with the standard deviation in parenthesis, of the best obtained solutions over ten independent runs. The forth column lists the number of function evaluations

Table A.1: Results of EDA for Onemax.

Samples	Cutoff	Best		Evaluations	
100	0.5	–400	(0)	8570	(148.66)
500	0.5	–400	(0)	41950	(610.33)
500	0.3	–400	(0)	64750	(512.35)
1000	0.5	–400	(0)	85200	(1326.65)
1000	0.3	–400	(0)	130200	(1400)
500	0.1	–400	(0)	159600	(2406.24)
3000	0.5	–400	(0)	260400	(3231.1)
1000	0.1	–400	(0)	314900	(4109.74)
3000	0.3	–400	(0)	391800	(4069.4)
6000	0.5	–400	(0)	523800	(7613.15)
6000	0.3	–400	(0)	792000	(8485.28)
3000	0.1	–400	(0)	945600	(5969.92)
6000	0.1	–400	(0)	1891200	(6462.2)
100	0.3	–399.9	(0.3)	13400	(275.68)
100	0.1	–395.1	(1.7)	36030	(1500.03)

Table A.2: Results of EDA for 1D Ising.

Samples	Cutoff	Best		Evaluations	
3000	0.3	–364.8	(4.02)	1114800	(62183.29)
3000	0.5	–364.4	(2.94)	788400	(54212.91)
1000	0.5	–363.8	(6.54)	228600	(22037.24)
6000	0.5	–363.6	(5.78)	1839600	(120365.44)
6000	0.3	–362.6	(4.2)	2463000	(118922.66)
3000	0.1	–360.8	(3.82)	2260200	(49060.78)
1000	0.3	–359.8	(4.33)	307800	(12064.82)
500	0.3	–358.4	(4.96)	146900	(13931.62)
500	0.5	–358	(3.9)	95700	(4648.66)
1000	0.1	–356.8	(4.21)	663200	(35312.32)
100	0.5	–354	(6.69)	13720	(570.61)
500	0.1	–352.8	(5.38)	308550	(22005.06)
100	0.3	–348.6	(4.39)	20510	(1328.5)
100	0.1	–338.2	(5.55)	45170	(6008.5)
6000	0.1	–322.4	(5.35)	2904000	(0)

Table A.3: Results of EDA for 2D Ising.

Samples	Cutoff	Best		Evaluations	
3000	0.5	–719	(15.68)	746700	(63617.69)
6000	0.3	–714	(18.57)	2269800	(277094.14)
3000	0.3	–709.6	(8.04)	1073400	(130579.63)
1000	0.5	–706.6	(12.33)	213100	(22997.61)
6000	0.5	–705.4	(12.84)	1671000	(165043.63)
1000	0.3	–705	(8.06)	321600	(38257.55)
3000	0.1	–698.6	(15.07)	2625900	(261206.99)
500	0.5	–697	(13.89)	94800	(6021.63)
500	0.3	–694.8	(9.39)	151400	(14902.68)
500	0.1	–688.6	(17.32)	370050	(56346.45)
1000	0.1	–686	(9.34)	807500	(118196.66)
100	0.5	–680.8	(10.59)	14230	(445.08)
100	0.3	–664.4	(14.31)	22410	(1602.78)
6000	0.1	–649.8	(12.79)	2904000	(0)
100	0.1	–632.2	(12.47)	47430	(2609.23)

Table A.4: Results of CE for Onemax.

Samples	Cutoff	Best		Evaluations	
6000	0.7	–399.9	(0.3)	538800	(31269.15)
6000	0.5	–394.6	(3.56)	2835000	(207000)
3000	0.7	–380.1	(8.83)	272400	(7800)
3000	0.5	–359.2	(7.15)	2901000	(0)
1000	0.7	–339.3	(9.18)	72200	(3124.1)
500	0.7	–319.5	(4.92)	32300	(1661.32)
1000	0.5	–317.3	(8.96)	279600	(27122.68)
500	0.5	–298	(5.59)	74550	(4660.74)
100	0.7	–286.3	(4.24)	4870	(272.21)
100	0.5	–273.9	(4.87)	8120	(622.58)
500	0.3	–269.1	(7.33)	2900500	(0)
1000	0.3	–265.7	(3.13)	2901000	(0)
3000	0.3	–264.8	(1.83)	2901000	(0)
6000	0.3	–264	(1.55)	2904000	(0)
100	0.3	–254.9	(8.35)	2900100	(0)

Table A.5: Results of CE for 1D Ising.

Samples	Cutoff	Best		Evaluations	
1000	0.7	–288.4	(7.94)	2901000	(0)
500	0.7	–287.6	(5.99)	2613250	(861750)
500	0.5	–273.2	(5.31)	2900500	(0)
100	0.7	–269.8	(6.72)	183420	(206344.9)
500	0.3	–259.2	(6.21)	2900500	(0)
100	0.5	–258.2	(8.12)	1931100	(1221777.89)
1000	0.3	–251.6	(2.5)	2901000	(0)
3000	0.5	–250.8	(2.56)	2901000	(0)
6000	0.5	–250.2	(2.27)	2904000	(0)
1000	0.5	–250	(2)	2901000	(0)
3000	0.7	–249.6	(2.65)	2901000	(0)
6000	0.3	–249.6	(1.2)	2904000	(0)
3000	0.3	–249.4	(1.8)	2901000	(0)
6000	0.7	–249.2	(1.6)	2904000	(0)
100	0.3	–246.6	(8.67)	2900100	(0)

Table A.6: Results of CE for 2D Ising.

Samples	Cutoff	Best		Evaluations	
1000	0.7	–533	(12.53)	2901000	(0)
500	0.7	–525	(14.81)	2613900	(859800)
500	0.5	–506.6	(13.97)	2900500	(0)
100	0.7	–501.2	(9)	168440	(176983.23)
500	0.3	–484.2	(16.04)	2900500	(0)
100	0.5	–480.4	(10.07)	2191050	(937454.9)
100	0.3	–473.4	(9.3)	2900100	(0)
3000	0.5	–472.8	(2.56)	2901000	(0)
3000	0.7	–472.6	(3.23)	2901000	(0)
6000	0.7	–472.4	(3.56)	2904000	(0)
6000	0.3	–472.4	(3.67)	2904000	(0)
3000	0.3	–472	(3.22)	2901000	(0)
1000	0.5	–471.4	(3.9)	2901000	(0)
6000	0.5	–471.2	(2.99)	2904000	(0)
1000	0.3	–470.8	(2.56)	2901000	(0)

until the population converges. The convergence criterion is that the number of function evaluations is greater than $2.9e6$ or the variance of the cost function values of the generated samples is less than $1e - 20$.

Clearly, the performance of CE is inferior to that of EDA, despite the fact that CE is basically equivalent to or plausibly better than EDA. The results show that the populations of CE do not converge well. The performance of CE is dramatically improved in 4 by adding a population mechanism.

Bibliography

[1] Shumeet Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning,. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, 1994.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[3] Tobias Blickle and Lothar Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, Vol. 4, No. 4, pp. 361–394, 1996.

[4] Peter A.N. Bosman and Dirk Thierens. Continuous iterated density estimation evolutionary algorithms within the IDEA framework. In *Proceedings of the Optimization by Building and Using Probabilistic Models OBUPM Workshop at GECCO2000*, pp. 197–200, 2000.

[5] Peter A.N. Bosman and Dirk Thierens. Expanding from discrete to continuous estimation of distribution algorithms: The idea. In *Parallel Problem Solving From Nature - PPSN VI,*, pp. 767–776. Springer-Verlag, 2000.

[6] Mark Denny. Introduction to importance sampling in rare-event simulations. *European Journal of Physics*, Vol. 22, pp. 403–411, 2001.

[7] Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. *Sequential Monte Carlo Methods in Practice.* Springer-Verlag, New York, 2001.

[8] Pavlos S. Efraimidis and Paul G. Spirakis. Weighted random sampling with a reservoir. *Information Processing Letters*, Vol. 97, No. 5, pp. 181–185, 2006.

[9] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley Publishing Company, 1989.

[10] Nikolaus Hansen and Andreas Ostermeier. Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *In Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pp. 312–317, 1996.

[11] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning.* Springer-Verlag, New York, 2001.

[12] Mark Hauschild, Martin Pelikan, Claudio F. Lima, and Kumara Sastry. Analyzing probabilistic models in hierarchical BOA on traps and spin glasses. MEDAL Report No. 2007001, Missouri Estimation of Distribution Algorithms Laboratory, University of Missouri, St. Louis, MO, 2007.

[13] Masayuki Henmi, Ryo Yoshida, and Shinto Eguchi. Importance sampling via the estimated sampler. *Biometrika*, Vol. 94, No. 4, pp. 985–991, 2007.

[14] Takayuki Higo and Keiki Takadama. Resampling-based population mechanism for evolutionary algorithms based on probability models. In *11th Asia-Pacific Workshop on Intelligent and Evolutionary Systems*, 2007.

[15] Takayuki Higo and Keiki Takadama. Maintaining multiple populations with different diversities for evolutionary optimization based on probabiity models. *IPSJ Transactions on Mathematical Modeling and Its Applications*, 2008. to appear.

[16] Tomoyuki Higuchi. Monte Carlo filter using the genetic algorithm operators. *Journal of Statistical Computation and Simulation*, Vol. 59, No. 1, pp. 1–23, 1997.

[17] Koji Hukushima and Koji Nemoto. Exchange monte carlo method and application to spin glass simulations. *Journal of the Physical Society of Japan*, Vol. 65, No. 6, pp. 1604–1608, 1996.

[18] Stefan Kern, Sibylle D. Müller, Nikolaus Hansen, Dirk Büche, Jiri Ocenasek, and Petros Koumoutsakos. Learning probability distributions in continuous evolutionary algorithms - a comparative review. *Natural Computing: an international journal*, 2004.

[19] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, Vol. 220, pp. 671–680, 1983.

[20] Pedro Larranaga and Jose A. Lozano, editors. *Estimation of Distribution Algorithm*. Kluwer Academic Publishers, 2002.

[21] Claudio F. Lima, Martin Pelikan, David E. Goldberg, Fernando G. Lobo, Kumara Sastry, and Mark Hauschild. Influence of selection and replacement strategies on linkage learning in BOA. Technical report, IlliGAL Technical Report No. 2007013, University of Illinois at Urbana-Champaign, 2007.

[22] Jun S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[23] Jose A. Lozano, Pedro Larranaga, Inaki Inza, and Endika Bengoetxea, editors. *Towards an New Evolutionary Computation*. Springer, 2006.

[24] M. Lundy and A. Mees. Convergence of an annealing algorithm. *Mathematical Programming*, Vol. 34, No. 1, pp. 111–124, 1986.

[25] Thilo Mahnig and Heinz Mühlenbein. Comparing the adaptive Boltzmann selection schedule SDS to truncation selection. In *Proceedings of the Third Internatinal Symposium on Adaptive Systems*, pp. 121–128, 2001.

[26] Thilo Mahnig and Heinz Mühlenbein. A new adaptive Boltzmann selection schedule SDS. In *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001.

[27] Heinz Mühlenbein, J. Bendisch, and H.-M Voigt. From recombination of genes to the estimation of distributions II. continuous parameters. In *Parallel Problem Solving from Nature IV*, pp. 188–197, 1996.

[28] Heinz Mühlenbein and Robin Hons. The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, Vol. 13, No. 1, pp. 1–27, 2005.

[29] Heinz Mühlenbein and Gerd Paaß. From recombination of genes to the estimation of distributions I. binary parameters. In *Parallel Problem Solving from Nature IV*, pp. 178–1187, 1996.

[30] Manfred Opper and David Saad, editors. *Advanced Mean Field Methods: Theory and Practice (Neural Information Processing Systems)*. MIT Press, 2001.

[31] Martin Pelikan and David E. Goldberg. Genetic algorithms, clustering, and the breaking of symmetry. In *Parallel Problem Solving from Nature - PPSN VI 6th International Conference*, pp. 840–846, 2000.

[32] Martin Pelikan and David E. Goldberg. Research on the Bayesian optimization algorithm. In *Optimization By Building and Using Probabilistic*, pp. 216–219, Las Vegas, Nevada, USA, 8 2000.

[33] Martin Pelikan and David E. Goldberg. Escaping hierarchical traps with competent genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pp. 511–518, 7-11 2001.

[34] Martin Pelikan, David E. Goldberg, and Erick Cantú-Paz. BOA: The Bayesian optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, Vol. I, pp. 525–532, 1999.

[35] Martin Pelikan, David E. Goldberg, and Fernando G. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, Vol. 21, pp. 5–20, 2002.

[36] Martin Pelikan, David E. Goldberg, Jiri Ocenasek, and Simon Trebst. Robust and scalable black-box optimization, hierarchy, and ising spin glasses. IlliGAL Report No. 2003019, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, 2003.

[37] Martin Pelikan, Shigeyoshi Tsutsui, and Rajiv Kalapala. Dependency trees, permutations, and quadratic assignment problem. MEDAL Report No. 2007003, Missouri Estimation of Distribution Algorithms Laboratory, University of Missouri, St. Louis, MO, 2007.

[38] Christian P. Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, 2004.

[39] Reuven Y. Rubinstein. *Simulation and the Monte Carlo Method*. Wiley-Interscience, 1981.

[40] Reuven Y. Rubinstein and Dirk P. Kroese. *The Cross-Entropy Method*. Springer, 2004.

[41] Yun-Wei Shang and Yu-Huang Qiu. A note on the extended rosenbrock function. *Evolutionary Computation*, Vol. 14, No. 1, pp. 119–126, 2006.

[42] Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, Vol. 90, pp. 227–244, 2000.

[43] Hisashi Shimodaira. An empirical performance comparison of niching methods for genetic algorithms. *IEICE transactions on information and systems*, Vol. E85-D, No. 11, pp. 1872–1880, 11 2002.

[44] Thomas Weise. Global Optimization Algorithms Theory and Application. `http://www.it-weise.de/`.

[45] Shigeyoshi Tsutsui. Node histogram vs. edge histogram: A comparison of probabilistic model-building genetic algorithms in permutation domains. In *The 2006 IEEE Congress on Evolutionary Computation (CEC-2006)*, 2006.

[46] 深尾毅. 分散システム論 -熱力学的システム論-. 昭晃堂, 1987.

[47] Walter Greiner, Ludwig Neise, and Horst Stöcker. 熱力学・統計力学. シュプリンガー・フェアラーク東京, 1999.

[48] 樋口知之. 遺伝的アルゴリズムとモンテカルロフィルタ. 統計数理, Vol. 44, No. 1, 1996.

[49] 伊庭幸人, 種村正美, 大森裕浩, 和合肇, 佐藤整尚, 高橋明彦. 計算統計 II マルコフ連鎖モンテカルロ法とその周辺. 岩波書店, 2005.

[50] 長尾智晴. 最適化アルゴリズム. 昭晃堂, 2000.

[51] 下平英寿, 伊藤秀一, 久保川達也, 竹内啓. モデル選択 –予測・検定・推定の交差点. 岩波書店, 2004.

[52] 渡邊澄夫. データ学習アルゴリズム. 共立出版, 2001.

[53] 渡邊澄夫, 村田昇. 確率と統計 -情報学への架橋-. コロナ社, 2005.

[54] 汪 金芳, 田栗正章, 手塚集, 樺島祥介, 上田修功. 計算統計 –確率計算の新しい手法. 岩波書店, 2003.

List of Publications

Journal Articles

Takayuki Higo and Keiki Takadama. Maintaining Multiple Populations with Different Diversities for Evolutionary Optimization Based on Probability Models. *IPSJ Transaction on Mathematical Modeling and Its Applications*, 2008, to appear.

Conference Proceedings

Takayuki Higo and Keiki Takadama. Resampling-based Population Mechanism for Evolutionary Algorithms based on Probability Models. *11th Asia-Pacific Workshop on Intelligent and Evolutionary Systems*, 2007.

Takayuki Higo and Keiki Takadama. Hierarchical Importance Sampling Instead of Annealing. *IEEE Congress on Evolutionary Computation*, 2007.

Takayuki Higo and Keiki Takadama. Neighbor based Parents Selection for Real-coded Genetic Algorithms *Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on advanced Intelligent Systems*, 2006.

Takayuki Higo and Keiki Takadama. Comparison between Self-organization with Sampling and Genetic Algorithms in multi-modal function *International Symposium on Artificial Life and Robotics*, 2006.

Technical Reports and Conference Papers in Japanese

比護貴之, 高玉圭樹. 多点探索型の最適化計算における収束問題のための階層型インポータンスサンプリング. 情報論的学習理論ワークショップ, 2006.

比護貴之, 高玉圭樹. 自己組織化サンプリングの提案とその最適化手法 -多峰性関数への適用-. 人工知能学会全国大会（第１９回）, 2005.

比護貴之, 高玉圭樹. 高次元多峰性関数最適化手法の提案. 計測自動制御学会 システム・情報部門 システム・情報部門学術講演会, 2005.

比護貴之, 高玉圭樹. 適応的探索領域分割に基づく最適化手法の提案. 計測自動制御学会　システム・情報部門 システム・情報部門学術講演会, 2004.