Title	An evolutionary Monte Carlo algorithm for identifying short adjacent repeats in multiple sequences
Author(s)	Xu, J; Li, Q; Fan, X; Li, VOK; Li, SYR
Citation	The 2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Hong Kong, China, 18-21 December 2010. In Proceedings of BIBM, 2010, p. 643-648
Issued Date	2010
URL	http://hdl.handle.net/10722/142819
Rights	Creative Commons: Attribution 3.0 Hong Kong License

An Evolutionary Monte Carlo Algorithm for Identifying Short Adjacent Repeats in Multiple Sequences

Jin Xu[1], Qiwei Li[2], Xiaodan Fan[3], Victor O. K. Li[1], and Shuo-Yen Robert Li[2]
[1]*Department of Electrical and Electronic Engineering*
The University of Hong Kong, Pokfulam, Hong Kong Island, Hong Kong
[2]*Department of Information Engineering,* [3]*Department of Statistics*
The Chinese University of Hong Kong, Sha Tin, New Territories, Hong Kong
xujin@eee.hku.hk, lqw008@ie.cuhk.edu.hk, xfan@sta.cuhk.edu.hk, vli@eee.hku.hk, bobli@ie.cuhk.edu.hk

***Abstract*—Evolutionary Monte Carlo (EMC) algorithm is an effective and powerful method to sample complicated distributions. Short adjacent repeats identification problem (SARIP), i.e., searching for the common sequence pattern in multiple DNA sequences, is considered as one of the key challenges in the field of bioinformatics. A recently proposed Markov chain Monte Carlo (MCMC) algorithm has demonstrated its effectiveness in solving SARIP. However, high computation time and inevitable local optima hinder its wide application. In this paper, we apply EMC to parallelize the MCMC algorithm to solve SARIP. Our proposed EMC scheme is implemented on a parallel platform and the simulation results show that, compared with the conventional MCMC algorithm, EMC not only improves the quality of final solution but also reduces the computation time.**

***Keywords*-Evolutionary Monte Carlo, parallel tempering, repetitive pattern, short adjacent repeats, sequence motif**

I. Introduction

Short tandem repeats (STR) analysis has been drawing extensive attention in the field of forensics and bioinformatics since the late 1990s. A tandem repeat is a segment containing two or more contiguous, approximate copies of a pattern of nucleotides [1]. The pattern width of STR ranges from 2 to 16 base pairs (bp). An example of STR would be

$$\ldots\overbrace{TGGCA}\overbrace{TGGCA}\overbrace{TGGCA}\ldots,$$

where the sequence pattern (also called repeat unit) $TGGCA$ repeats three times. STR can be used for genetic fingerprinting [2]. Recent research efforts have also studied their association with genetic diseases [3]. For example, Huntington's disease, which affects muscle coordination, is the result of explosive growth in the copy number of a trinucleotide pattern CAG [4].

We generalize STR by introduce short adjacent repeats (SAR) that allows gaps between neighboring repeat units. Such inter-unit insertions frequently occur due to errors in genetic manipulations and mutations during the evolutionary process. As a result, it is possible to have a sequence

$$\ldots\overbrace{TGGCA}CC\overbrace{TGGCA}T\overbrace{TGGCA}\ldots$$

with the insertions of CC and T in nature. In other words, STR is a special type of SAR where the length of any gap is zero. Note that, besides the application in bioinformatics, the SAR identification problem (SARIP) can also be used in data analysis such as recognizing repetitive patterns in speeches, texts or images. In this paper, we mainly focus on the problem of identifying SAR in multiple DNA sequences.

Evolutionary Monte Carlo (EMC) algorithm, proposed by Liang and Wong [5] and further developed by Goswami and Liu [6], can be considered as a method for evolving a population of Markov chain Monte Carlo (MCMC) chains in an effort to explore multi-modal multivariate distributions. It can also be regarded as an extension of parallel tempering (PT) algorithm, or replica exchange, originated by Swenden and Wang [7], and then extended by Geyer [8]. EMC and PT have been successfully applied to solve hard computational problems in many fields including biology, chemistry, physics, engineering and material science [9]. The key idea of EMC and PT is to execute multiple independent replicas simultaneously under different conditions. Usually the condition difference is defined by temperature, which tunes the smoothness of the target distribution. The systems with high temperature are generally able to sample a wide range of energy landscape, while those at low temperature have the ability to explore the "local details". In the simulation process, replicas at neighboring temperature levels are allowed to exchange at certain frequency according to the Metropolis-Hastings rule. By virtue of the good balance between exploration and exploitation, EMC is able to achieve good performance, especially in complex systems with rough energy landscapes, where canonical Monte Carlo methods are easily trapped in the local free energy minima.

Due to its population-based nature, EMC is quite suitable to be performed on parallel machines. Replicas in PT can be executed concurrently on different processors which communicate with each other from time to time. Meanwhile, with the advance of multicore CPU and the improvement of the network environment, more computation resources are readily available, rendering parallel and distributed computing a trend in the future. Thus, in this paper, we implement

978-1-4244-8305-1/10/$26.00 ©2010 IEEE

EMC on a parallel platform to improve the quality of the solution as well as reduce the computation time.

The rest of the paper is organized as follows. In Section II, we describe the generative model of SARIP as well as the adopted canonical MCMC algorithm. Section III addresses our proposed EMC model. Experimental results are reported and discussed in Section IV. Finally, in Section V, we summarize the paper and give suggestions for future investigation.

II. Problem Description

In this section, a brief overview of SARIP is presented in two parts: generative model and proposed MCMC algorithm. For more detailed description of this problem, see [10], [11].

A. Generative Model

Given a set of N DNA sequences with different lengths $(L_n)_{N\times 1}$, $\mathbf{R} = (R_n)_{N\times 1} = \left((r_{n,l})_{1\times L_n}\right)_{N\times 1}, r_{n,l} \in \{A,T,C,G\}$, each of which is embedded with a repeat segment in a homogeneous background, our goal is to find the most probable location and structure for each repeat segment. We denote by $\mathbf{A} = (a_n)_{N\times 1}$ and $\mathbf{S} = (\mathbf{s}_n)_{N\times 1}$ the sets of repeat segment starting positions and structures, respectively. Each $\mathbf{s}_n$ is a base-$(G+1)$ numeral vector $(g_{n,\omega})_{1\times(\Omega_n-1)}, 0 \le g_{n,\omega} \le G$ where Ω_n is the copy number, G is the maximal allowed gap length, and $g_{n,\omega}$ is the gap length between the ω-th repeat unit and the $(\omega+1)$-th repeat unit. All repeat units with predetermined pattern width J are instances sampled from the motif matrix $\Theta = (\theta_{k,j})_{4\times J}, k \in \{A,T,C,G\}$, where $\theta_{k,j}$ specifies the probability of generating the nucleotide k at the j-th position. The background distribution can be written as $\Phi = (\phi_k)_{4\times 1}, k \in \{A,T,C,G\}$, where ϕ_k specifies the probability of generating the nucleotide k at a non-unit position. For ease of presentation, Figure 1 (adapted from [11]) shows an example of schematic diagram of the model, where Ω is the maximal allowed copy number for all repeat segments. In this example, there are 5 sequences with corresponding length equal to $L_n, 1 \le n \le 5$. For each repeat segment, as represented by the gray area, the corresponding starting position and structure are shown on top of it. In order to avoid a trans-dimensional model, we fill the Ω_n-th to the Ω-th elements with the trivial variable -1 for each $\mathbf{s}_n$. The background area is painted in white with dotted borderline. Each white square with solid borderline represents a gap with length 1.

For Bayesian inference of these independent parameters $\mathbf{A}$, $\mathbf{S}$, Θ, and Φ, we first specify prior knowledge for each parameter, write the complete data likelihood $P(\mathbf{R}|\mathbf{A},\mathbf{S},\Theta,\Phi)$ and derive the joint posterior probability $P(\mathbf{A},\mathbf{S},\Theta,\Phi|\mathbf{R})$ via Bayes' rule. Then, a collapsing technique [12] is adopted to reduce the dimensionality of the solution space by integrating out nuisance parameters Θ and Φ. Lastly, in this $N\Omega$-dimension space, we explore the target

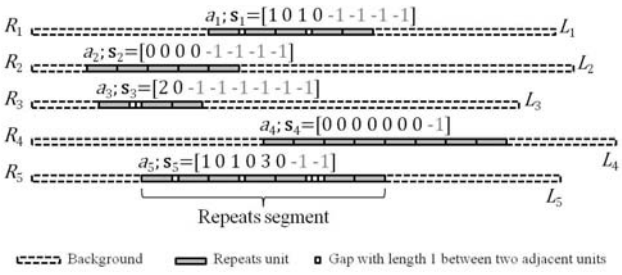

Figure 1. Schematic diagram of the model under the setting $N = 5$ and $\Omega = 9$

distribution $P(\mathbf{A},\mathbf{S}|\mathbf{R})$ that can give us a whole probability density landscape. The particular $\mathbf{A}$ and $\mathbf{S}$ corresponding to the maximum a posteriori (MAP) of the target distribution is considered to be the best solution for the input data set $\mathbf{R}$.

B. MCMC Algorithm

The key idea of addressing this optimization problem is to use Metropolis-in-Gibbs scheme [13], [14] as shown in Table I (adapted from [11]). The MCMC algorithm proceed through iterations after initialization, each of which updates a_n and $\mathbf{s}_n$ one sequence after another in ascending order from 1 to N. Within each iteration, to update the repeat segment starting position a_n and structure $\mathbf{s}_n$ for the n-th sequence, we pretend that the starting positions and structures of the remaining $N-1$ repeat segments are known, and we stochastically predict a_n and $\mathbf{s}_n$. More specifically, we use the given information, $\mathbf{A}_{[-n]}$ and $\mathbf{S}_{[-n]}$, to estimate the current 'motif matrix' $\hat{\Theta}_n$ and 'background distribution' $\hat{\Phi}_n$ so as to determine new a_n via Gibbs sampling and new $\mathbf{s}_n$ via Metropolis-Hastings sampling sequentially. Here, $\mathbf{A}_{[-n]}$ denotes the set $\mathbf{A}$ excluding the element a_n and $\mathbf{S}_{[-n]}$ denotes the set $\mathbf{S}$ excluding the element $\mathbf{s}_n$. Intuitively, the more accurate the estimated motif matrix $\hat{\Theta}_n$ and background distribution $\hat{\Phi}_n$ constructed in the predictive update step, the more accurate the determination of a_n and $\mathbf{s}_n$ in the following sampling steps, and vice versa.

Table I
The Schematic Procedure of the MCMC Algorithm

Step 1:	Initialize $\mathbf{A}$ and $\mathbf{S}$;
Step 2:	for n from 1 to N do
	2.1: Predictive update $\hat{\Theta}_n$ and $\hat{\Phi}_n$ via $\mathbf{A}_{[-n]}$ and $\mathbf{S}_{[-n]}$;
	2.2: Sample and update a_n via $P(a_n\|\mathbf{s}_n,\hat{\Theta}_n,\hat{\Phi}_n,\mathbf{R})$;
	2.3: Sample and update $\mathbf{s}_n$ via $P(\mathbf{s}_n\|a_n,\hat{\Theta}_n,\hat{\Phi}_n,\mathbf{R})$;
Step 3:	Repeat Step 2 until monitoring convergence;

Gibbs sampling [15] is employed to renew each a_n. Conditional on the current values of all other parameters $\mathbf{A}_{[-n]}$ and $\mathbf{S}$, we first break down the sequence R_n into overlapping segments of fixed length $\Omega_n J + \sum_{\omega=1}^{\Omega_n-1} g_{n,\omega}$, then calculate the corresponding probability of generating

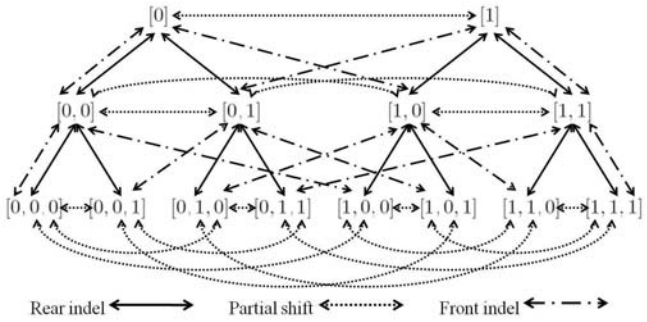

Figure 2. The full state transition diagram under the setting $G = 1$ and $\Omega = 4$.

those matching repeat units within each possible segment, and finally sample the new a_n according to such probabilistic weights. We use Metropolis-Hastings algorithm [16] to update each $\mathbf{s}_n$ because it is extremely difficult to compute the normalization constant of $P(\mathbf{s}_n|a_n, \hat{\Theta}_n, \hat{\Phi}_n, \mathbf{R})$. In order to make the Markov chain ergodic and fast-convergent, we design three types of moves: rear indel (insert a unit behind the current last unit or delete the current last unit), partial shift (make a selected sub-segment shifted left or right), and front indel (insert a unit in front of the current first unit or delete the current first unit). For more details, see [11]. For ease of presentation, Figure 2 (adapted from [11]) shows an example of full state transition diagram for $\mathbf{s}_n$. Since the maximal allowed gap length $G = 1$ and the maximal allowed copy number $\Omega = 4$, each state of $\mathbf{s}_n$ can be described as a 3-dimension binary vector. For convenience, we erase the nuisance variable -1 for all vectors. The three categories of moves are represented by different types of two-way lines. The upwards directed lines mean deleting a unit while the backwards directed lines mean inserting a unit within the repeat segment.

III. Implementation of EMC Algorithm

Generally we sample the points in a target distribution with the expression

$$\psi(x) \propto e^{-E(x)/T}, \quad x \in \Re^d, \quad T > 0$$

where the objective function $E(x)$ is interpreted as the energy of a thermodynamic system, and T is the temperature parameter used to control the shape of the distribution $\psi(x)$. Thereby, the higher the temperature, the "flatter" the target distribution, which means the sampling points can easily get over the energy barriers. As mentioned in Section II, $P(\mathbf{A}, \mathbf{S}|\mathbf{R})$ is our probability density function, and for our problem we would like to find the maximal value in the posterior distribution. However, thermodynamic systems tend to stay in the lowest free energy state, which is accordingly a minimization problem. Thus, we define the energy density function $E(x) = -ln(P(\mathbf{A}, \mathbf{S}|\mathbf{R}))$. In this way, target density function $\psi(x)$ can be written as:

$$e^{-E(x)/T} = e^{-(-lnP(\mathbf{A},\mathbf{S}|\mathbf{R}))/T} = (P(\mathbf{A}, \mathbf{S}|\mathbf{R}))^{1/T}$$

and when $T = 1$, it becomes the original $P(\mathbf{A}, \mathbf{S}|\mathbf{R})$.

Subsequently, the Gibbs sampling for updating $\mathbf{A}$ in [11] should be modified as

$$P(a_n|\hat{\Theta}_n, \mathbf{A}_{[-n]}, \mathbf{S}, \mathbf{R}) \propto (P(\mathbf{R}|\hat{\Theta}_n, \mathbf{A}, \mathbf{S})P(a_n))^{1/T}$$

and the Metropolis-Hasting sampling for updating $\mathbf{S}$ in [11] needs to be rewritten as

$$\begin{aligned}\lambda &= \left(\frac{P(s_n^*|\hat{\Theta}_n, \mathbf{A}, \mathbf{S}_{[-n]}, \mathbf{R})P(s_n; s_n^*)}{P(s_n|\hat{\Theta}_n, \mathbf{A}, \mathbf{S}_{[-n]}, \mathbf{R})P(s_n^*; s_n)}\right)^{1/T} \\ &= \left(\frac{P(\mathbf{R}|\hat{\Theta}_n, \mathbf{A}, \mathbf{S})P(s_n^*)P(s_n; s_n^*)}{P(\mathbf{R}|\hat{\Theta}_n, \mathbf{A}, \mathbf{S})P(s_n)P(s_n^*; s_n)}\right)^{1/T}.\end{aligned}$$

EMC adopts a sequence of temperature ladder $\{T_1 > T_2 > \cdots > T_N > 0\}$, and in our simulation, we assume $T_N = 1$. Thus, the target density function for each replica x_i can be expressed as $\psi(x_i) \propto e^{-E(x_i)/T_i}$, where $i = 1, 2, \cdots, N$. Consequently, the solution in the composite system can be specified as $X = \{x_1, x_2, \cdots, x_N\}$, and accordingly, the target distribution for the whole system is:

$$\Psi(X) \propto \prod_{i=1}^{N} e^{-E(x_i)/T_i}$$

Replica exchange only happens between neighboring temperature levels ($x_i \leftrightarrow x_{i+1}$). According to the Metropolis-Hastings criterion, this transition is accepted with probability:

$$\begin{aligned}\theta(x_i \leftrightarrow x_{i+1}) &= min\left(1, \frac{\Psi(x_1, \cdots x_{i+1}, x_i, \cdots, x_N)}{\Psi(x_1, \cdots x_i, x_{i+1}, \cdots, x_N)}\right) \\ &= min\left(1, e^{(E(x_i)-E(x_{i+1}))\left(\frac{1}{T_i}-\frac{1}{T_{i+1}}\right)}\right).\end{aligned}$$

Based on the above design, EMC can be guaranteed to follow the reversibility condition (also called "detailed balance" or "time reversibility"). The pseudocode for the EMC algorithm is shown in Table II. Its process includes three stages: initialization, iterations, and the final output stage. In the initialization step, we generate the initial replica $(x_1, x_2, \cdots, x_N)$ on each processor, and set the corresponding temperature for them. Then, the EMC enters the iteration part. Basically, each processor runs the modified MCMC algorithm simultaneously, and with certain frequency, replicas are switched in terms of the Hastings ratio. The results are reported in the final stage.

IV. Experiment

In order to explore the potential of applying EMC to SARIP, we construct synthetic data for testing. In this section, we first illustrate the testing data, and then describe the simulation scenario. Finally, we give and discuss the simulation results.

Table II
THE PSEUDOCODE FOR THE EMC ALGORITHM

1:	Initialize the N replicas $(x_1, x_2, \cdots, x_N)$, and set the temperature ladder $T_1 > T_2 > \cdots > T_N > 0$ with $T_N = 1$;
2:	*While* stopping criterion not met;
3:	*While* replica exchange rate not met;
4:	Perform the modified MCMC algorithm on each processor;
5:	Try to exchange the replicas;
6:	Randomly select an integer variable i from $[1, N-1]$;
7:	Calculate the acceptance probability $\theta(x_i \leftrightarrow x_{i+1})$;
8:	Generate a random value α from the uniform distribution [0,1];
9:	*If* $\alpha < \theta(x_i \leftrightarrow x_{i+1})$;
10:	Swap the replica: $x_i \leftrightarrow x_{i+1}$;
11:	Else
12:	Keep the replica unmovable;
13:	Output the best value of the posterior distribution: $P(\mathbf{A}, \mathbf{S}\|\mathbf{R})$

A. Testing Data

DNA sequences are unbranched polymers with several thousands of nucleotide bases, and the corresponding input data are quite huge. Thus, for SARIP, the energy landscape is likely to have multiple local optima. Meanwhile, as mentioned previously, the advantage of EMC lies in its ability to help the solution escape from the local optima. Therefore, we test the proposed EMC on the synthetic DNA sequences, which are generated with many local optima. The main parameters for the test case are as follows: $N = 5$, $L_n = 2000, 1 \leq n \leq 5$, $G = 2$, $J = 9$, $\Omega = 9$, $\Phi = \begin{bmatrix} 0.25 & 0.25 & 0.25 & 0.25 \end{bmatrix}^T$ and $\Theta =$

$$\begin{bmatrix} 0.85 & 0.07 & 0.07 & 0.07 & 0.07 & 0.80 & 0.07 & 0.07 & 0.85 \\ 0.05 & 0.80 & 0.07 & 0.07 & 0.07 & 0.07 & 0.07 & 0.80 & 0.05 \\ 0.05 & 0.07 & 0.80 & 0.06 & 0.06 & 0.07 & 0.06 & 0.07 & 0.05 \\ 0.05 & 0.06 & 0.06 & 0.80 & 0.80 & 0.06 & 0.80 & 0.06 & 0.05 \end{bmatrix}.$$

Figure 3 shows the locations of multiple segments (marked as blocks) within each sequence. For simplicity, we are mainly interested in locating the most probable segment (marked as red block) within each sequence.

Figure 3. Location of Multiple Segments within Each Sequence

B. Simulation Setup

In both EMC and PT, one of the crucial steps is to establish the temperature ladder. In our problem, we assume the heat of the system is constant, and thus, a geometric progression [17] can be utilized to approximate the temperature ladder set $\{T_1 > T_2 > \cdots > T_N > 0\}$. As discussed in Section III, the lowest temperature T_N is set to 1.000. For the highest temperature, we assign the value of 2.500 to T_1, which is big enough to produce a "flatter" energy landscape. Other temperatures $T_2, \cdots, T_{N-1}$ can be calculated via

$$T_j = T_1 \mu^{j-1}, \quad \mu = \sqrt[N-1]{\frac{T_1}{T_N}}$$

In the preliminary simulation, we use four processors, and the temperature ladder can be configured as $\{1.000, 1.357, 1.841, 2.500\}$. Meanwhile, taking into account the "detailed balance", each processor should do the same number of evaluations before the replica exchange. Thus, for convenience, EMC is parallelized in a synchronized manner. The replica exchange between neighboring temperature lever is triggered every 10 iterations. When the exchange is accepted, we swap the configurations, namely, the starting positions $\mathbf{A}$ and the segment structures $\mathbf{S}$.

Since EMC is a stochastic algorithm, different runs may yield diverse results. Thus, we repeat each scenario 20 times, and record the average value. Our EMC algorithm is coded in C++, and the simulations are conducted on a cluster of computers with an Intel Core Quad 2.66GHz CPU and 4G RAM connected in a Ethernet. The replica exchanges among processors are realized by the MPICH2 [18], which is high-performance and widely used implementation of the Message Passing Interface (MPI). This software library is fairly flexible and convenient for the parallel design.

C. Analysis of Results

MAP [10] is adopted as the metric to weigh the quality of the solution ($\mathbf{A}$ and $\mathbf{S}$). The higher the MAP, the better the solution we obtain. Since it is extremely difficult to compute the normalization constant of the posterior probability, without influencing on the final results, we employ the unnormalized natural logarithm posterior probability.

Figure 4 shows the comparison of MAP values between EMC and MCMC algorithms for different number of iterations. Note that on these two curves, the MAP of each dot is the average value from 20 independent runs. It can be observed that MAP obtained from EMC are higher than those from MCMC algorithm through the whole iteration process, and the difference reaches the peak when these two algorithms converge. As mentioned above, the energy landscape in our problem is rugged with many local optima. MCMC can easily get stuck in the local minimum as shown in Figure 4, while EMC is able to jump out of the energy barrier by swapping replicas at different temperature levels. We can also observe that MCMC converges faster than EMC, although its final solutions are much worse than those of EMC. This is because EMC explores a wider range of the solution space, thus requiring many more iterations.

Table III compares the average and standard deviation values of the MCMC and EMC algorithms. After 600 iterations, MAP obtained by EMC exceeds the MCMC's, which is reached by MCMC at around 1000 iterations. In other words, to acquire the same quality of solution, EMC only needs half of the number of iterations as that of MCMC. Equivalently, EMC only requires half of the computation time as that of MCMC ignoring the communication overhead. Actually, by analyzing the computation time of MCMC and EMC in Table IV, we discover that communication overhead

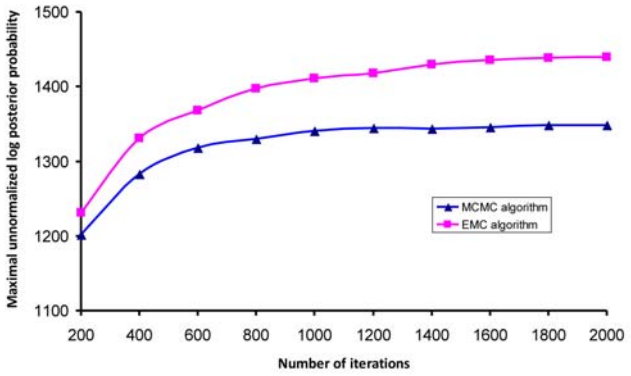

Figure 4. The Comparison of MAP between EMC and MCMC Algorithms for Different Number of Iterations

Table III
AVERAGE AND STANDARD DEVIATION VALUES OF EMC AND MCMC ALGORITHMS

No. of iterations	200	400	600	800	1000
AVG_MCMC	1202.1	1283.3	1318.4	1329.8	1340.7
AVG_EMC	1230.7	1330.7	1367.5	1397.0	1411.0
SD_MCMC	101.2	92.0	63.4	62.6	75.6
SD_EMC	84.3	65.1	43.5	45.9	39.4
No. of iterations	1200	1400	1600	1800	2000
AVG_MCMC	1344.1	1343.2	1345.4	1348.6	1348.1
AVG_EMC	1418.2	1429.3	1435.8	1438.4	1439.2
SD_MCMC	43.5	53.0	57.2	49.1	77.2
SD_EMC	40.8	41.9	33.5	39.4	36.7

occupies less than 10% of the total running time. Thus, EMC can also reduce the computation time when compared with MCMC. Moreover, all of the standard deviations of EMC are smaller than those of MCMC, which means EMC is more robust than MCMC. This is of great significance in practice, as we would not like the algorithm to have to run many times to get a relatively good solution.

Table IV
COMPARISON OF COMPUTATION TIME BETWEEN MCMC AND EMC ALGORITHMS

No. of iterations	200	400	600	800	1000
MCMC	2.949	7.150	12.023	16.476	21.053
EMC	3.209	8.009	12.74	17.645	22.791
No. of iterations	1200	1400	1600	1800	2000
MCMC	26.838	30.083	37.831	39.450	48.333
EMC	27.950	34.924	38.330	43.209	51.413

V. CONCLUSION

EMC or PT have been developed as useful tools to solve optimization problems in science and engineering. The basic idea behind EMC and PT is to properly set a series of temperatures, based on which replicas explore the solution space in various degree. Information exchange between replicas enables one in lower temperature to have the opportunity to surmount the barrier. The contributions of this paper are mainly in two parts: (1) An EMC model was proposed to solve SARIP, parallelizing the MCMC algorithm by introducing the parameter temperature, and by doing this, the MCMC was reconstructed as an EMC; (2) The EMC was then deployed on a parallel platform MPICH2 to handle a case of SARIP, which has multiple local optima. The simulation results show that with appropriate design of EMC, it not only enhances the quality of final solution but also cut the computation time. Additionally, its robustness makes EMC suited to be deployed in practice.

For our future investigations, in order to get a comprehensive understanding of EMC for SARIP, more accurate temperature ladder designs can be tried, such as the feedback scheme [19]. In addition, other operators such as mutation and crossover may be incorporated into the EMC. We may also employ more processors to see if further progress can be made.

ACKNOWLEDGMENT

This work was supported in part by the Strategic Research Theme of Information Technology of The University of Hong Kong.

REFERENCES

[1] G. Benson, "Tandem repeats finder: a program to analyze DNA sequences," *Nucleic Acids Research*, vol. 27, no. 2, pp. 573–580, 1999.

[2] J. M. Butler, C. M. Ruitberg, and D. J. Reeder, "STRBase: A Short Tandem Repeat DNA Internet-accessible Database," in *Proc. The Eighth International Symposium on Human Identification*, 1997, p. 38–47.

[3] R. R. Sinden, "Trinucleotide Repeats Biological Implication of the DNA Structure Associated with Disease-causing Triplet Repeats," *Human Genetics*, vol. 64, pp. 346–353, 2000.

[4] Huntington's Disease Collaborative Research Group, "A Novel Gene containing A Trinucleotide Repeat that is Expanded and Unstable on Huntington's Disease Chromosomes," *Cell*, vol. 72, pp. 971–983, 1993.

[5] F. Liang and W. H. Wong, "Evolutionary Monte Carlo: Applications to C_p Model Sampling and Change Point Problem," *Statistica Sinica*, vol. 10, pp. 317–342, 2000.

[6] G. Goswami and J. S. Liu, "On Learning Strategies for Evolutionary Monte Carlo," *Statistics and Computing*, vol. 17, pp. 23–38, 2007.

[7] R. H. Swendsen and J.-S. Wang, "Replica Monte Carlo Simulation of Spin Glasses," *Physical Review Letters*, vol. 57, pp. 2607–2609, 1986.

[8] C. J. Geyer and E. A. Thompson, "Annealing Markov Chain Monte Carlo with Application to Ancestral Inference," *Journal of the American Statistical Association*, vol. 90, pp. 909–220, 1995.

[9] D. J. Earl and M. W. Deem, "Parallel Tempering: Theory, Applications, and New Perspectives," *Physical Chemistry Chemical Physics*, vol. 7, pp. 3910–3916, 2005.

[10] Q. Li, T. Liang, S.-Y. R. Li, and X. Fan, "Bayesian Approach for Identifying Short Adjacent Repeats in Multiple Sequences," in *Proc. 11th International Conference on Bioinformatics and Computational Biology*, 2010, pp. 255–261.

[11] Q. Li, X. Fan, T. Liang, and S.-Y. R. Li, "MCMC Algorithms for Detecting Short Adjacent Repeats in Multiple Sequences," Available as technical report on https://sites.google.com/site/liqiwei2000/research/journal-papers.

[12] J. S. Liu, A. F. Neuwald, and C. E. Lawrence, "Bayesian Models for Multiple Local Sequence Alignment and Gibbs Sampling Strategies," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1156–1170, 1995.

[13] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*, New York, United States: Springer-Verlag, 2001.

[14] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin, *Bayesian Data Analysis*, New York, United States: Chapman & Hall / CRC, 2004.

[15] A. E. Gelfand and A. F. M. Smith, "Sampling-based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, vol. 85, pp. 398–409, 1990.

[16] K. Hastings, "Monte Carlo Sampling Methods using Markov Chains and Their Applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[17] C. Predescu, M. Predescu, and C. Ciobanu, "The Incomplete Beta Function Law for Parallel Tempering Sampling of Classical Canonical Systems," *Journal of Chemical Physics*, vol. 120, pp. 4119, 2009.

[18] MPICH2: High-performance and Widely Portable MPI [Online]. Available: http://www.mcs.anl.gov/research/projects/mpich2/index.php.

[19] H. G. Katzgraber, S. Trebst, D. A. Huse, and M. Troyer, "Feedback-optimized Parallel Tempering Monte Carlo," *Journal of Statistical Mechanics*, vol. 2006, P03018, 2006.