PARALLEL METROPOLIS CHAINS WITH COOPERATIVE ADAPTATION

L. Martino[◇], *V. Elvira*[†], *D. Luengo*[‡], *F. Louzada*[◇]

[◇] Institute of Mathematical Sciences and Computing, Universidade de São Paulo, São Carlos (Brazil).
[†] Dep. of Signal Theory and Communic., Universidad Carlos III de Madrid, Leganés (Spain).
[‡] Dep. of Signal Theory and Communic., Universidad Politécnica de Madrid, Madrid (Spain).

ABSTRACT

Monte Carlo methods, such as Markov chain Monte Carlo (MCMC) algorithms, have become very popular in signal processing over the last years. In this work, we introduce a novel MCMC scheme where parallel MCMC chains interact, adapting cooperatively the parameters of their proposal functions. Furthermore, the novel algorithm distributes the computational effort adaptively, rewarding the chains which are providing better performance and, possibly even stopping other ones. These extinct chains can be reactivated if the algorithm considers it necessary. Numerical simulations show the benefits of the novel scheme.

***Index Terms*—** Interacting Parallel MCMC, Adaptive MCMC, cooperative adaptation, Bayesian inference.

1. INTRODUCTION

Markov Chain Monte Carlo (MCMC) algorithms [1] are widely employed in signal processing and communications for Bayesian inference and optimization [2, 3, 4, 5, 6]. They draw random samples from a complicated multidimensional target probability density function (pdf), $\pi(\mathbf{x})$ with $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^d$, generating a Markov chain which converges to $\pi(\mathbf{x})$. The performance, i.e., the speed of the convergence, depends strictly on the choice of a suitable proposal function $q(\mathbf{x})$, and more specifically, on the discrepancy between $q(\mathbf{x})$ and $\pi(\mathbf{x})$.

Speeding up the convergence has motivated an intense research activity. On the one hand, one active research line considers the design of an adaptive proposal density within MCMC techniques. Several schemes have been developed in order to tune online the parameters of the proposal density, learning them from the previously generated samples [7, 8, 5, 6]. On the other hand, the use of several parallel chains instead of a single long chain has been studied for different reasons. First of all, employing parallel chains allows the use of different proposal pdfs. Moreover, another important motivation is the interest in the implementation of MCMC techniques within a parallel architecture [9, 10, 11]. Finally, a third reason is to speed up the exploration of the state space [12, 4, 13, 14, 11]. Several works in the literature are focused on producing an interaction among the different parallel chains [15, 7, 13, 14, 16]. The exchange of information among the chains can yield jointly two related benefits: produce a faster convergence of the chains to the target (e.g., reallocating "lost" chains around a mode of the target [11]), and help the cooperative exploration of the state space (for instance, generating a repulsion among the chains during a certain number of iterations [14]). As a consequence, the combined use of interacting parallel chains and adaptive proposal pdfs is considered of great interest in the literature [7].

In this work, we propose a novel scheme, called *parallel adaptive independent Metropolis* (PAIM), involving parallel MCMC chains which exchange information in order to adapt online their proposal densities. Namely, the interaction among the chains is carried out by a cooperative adaptation of the proposal functions. Each MCMC chain employs a proposal pdf, independent of the previous state of the chain, which are formed by a mixture of two densities, each one determined by two parameters: a mean vector and covariance matrix. All the parameters are updated using empirical estimators (as in [8, 5]) applied to the complete set or only to a subset of the previously generated states. On the one hand, the first component of each mixture aims to provide a global adaptation, the complete set of states is used. On the other hand, the second component of each mixture is adapted considering only a subset of the previously generated states in order to learn local features of the target pdf. These subsets of states are built using a simple clustering-type strategy. This generates a cooperative adaptation, preventing that different proposals cover the same region, and allowing them to cover different modes of the target function, for instance. Furthermore, the novel algorithm is able to identify the proposal functions better located, and to allocate more computational effort to the corresponding chains. Indeed, PAIM adapts the number of iterations of every chain, stopping those chains which are using a badly located proposal. PAIM is also able to turn back on certain chains when it is necessary.

This work has been partially supported by the Spanish government through the DISSECT (TEC2012-38058-C03-01) project, the BBVA Foundation through the MG-FIAR project, by the Grant 2014/23160-6 of São Paulo Research Foundation (FAPESP) and by the Grant 305361/2013-3 of the National Council for Scientific and Technological Development (CNPq).

2. GENERAL SETUP

In many different applications [12, 3, 4], it is necessary to draw samples from a complicated d-dimensional target probability density function (pdf), $\bar{\pi}(\mathbf{x}) \propto \pi(\mathbf{x})$, with $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^d$. For this purpose, we consider the use of N parallel Metropolis-Hastings chains [1], each one employing a different independent proposal pdf $\psi_n(\mathbf{x})$, defined as a mixture of two pdfs $q_{1,n}(\mathbf{x})$ and $q_{2,n}(\mathbf{x})$. More specifically, explicitly indicating the parameters of the two components, we have

$$\psi_n(\mathbf{x}) = \frac{1}{2} q_{1,n}(\mathbf{x}|\boldsymbol{\mu}_{1,n}, \mathbf{C}_{1,n}) + \frac{1}{2} q_{2,n}(\mathbf{x}|\boldsymbol{\mu}_{2,n}, \mathbf{C}_{2,n}),$$

for $n = 1, \dots, N$, where $\boldsymbol{\mu}_{i,n}$ represents a d-dimensional mean vector and $\mathbf{C}_{i,n}$ is a $d \times d$ covariance matrix for $i \in \{1, 2\}$. In the novel method, the N parallel chains interact in order to jointly adapt all the parameters, $\boldsymbol{\mu}_{i,n}$ and $\mathbf{C}_{i,n}$ for $i \in \{1, 2\}$ and $\forall n$. The parameters $\boldsymbol{\mu}_{1,n}$ and $\mathbf{C}_{1,n}$, of the first component $q_{1,n}$ of every proposal ψ_n are updated to provide a *global* adaptation, whereas the parameters $\boldsymbol{\mu}_{2,n}$ and $\mathbf{C}_{2,n}$ of $q_{2,n}$ are adapted to extract *local* features of the target pdf.

Furthermore, in the novel scheme, each chain performs a different number of iterations, K_n for $n = 1, \dots, N$. Indeed, the proposed algorithm is also able to determine online a subset of the N chains which are obtaining the best performance. These chains are employed more often than the rest. Namely, the total number of iterations K_n performed by the corresponding chains is increased. Let us denote the total number of desired samples as L, chosen in advance by the user. Then, we have $K_1 + K_2 \dots + K_N = L$, i.e., PAIM is stopped when L samples have been generated.[1]

3. ADAPTIVE PARALLEL INDEPENDENT METROPOLIS ALGORITHM

The *parallel adaptive independent Metropolis* (PAIM) algorithm works on two different time scales. First of all, the index t denotes the current step of the algorithm, with

$$t = 0, \dots, T_{tot}, \tag{1}$$

where the total number of steps T_{tot} is not decided by the user, but automatically tuned by PAIM (we have always $T_{tot} \leq L$). Furthermore, we have have a different iteration index k_n for each chain, such that

$$k_n = 0, \dots, K_n, \tag{2}$$

for $n = 1, \dots, N$. The value of each K_n is also decided by PAIM (recall that $\sum_{n=1}^{N} K_n = L$). At each step, the set $\mathcal{A}_t$ contains the indices corresponding to the active chains. At every t-th step, all the active chains perform one iteration (i.e., if the j-th chain is active, then $k_j = k_j + 1$), whereas the inactive chains remain frozen. For every $t \leq T_{train}$, where T_{train} is chosen by the user, all the chains are active, i.e.,

$$\mathcal{A}_t = \{1, 2, \dots, N\}, \qquad t \leq T_{train}.$$

Whereas for $t > T_{train}$ we have $\mathcal{A}_t \subseteq \{1, 2, \dots, N\}$. The interacting adaptation is performed at any t such that $T_{train} < t < T_{stop}$. We consider the possibility of stopping the adaptation after T_{stop} steps, since the adaptation could jeopardize the ergodicity of the chains. However, the numerical results described in Section 5 show that the algorithm seems to maintain the correct ergodicity properties.

The adaptation is performed as follows. During the first T_{train} time steps the algorithm simply assigns the new current states, generated at the t-th step, to one chain among the N possible, according to the minimum Euclidean distance between them and the means, $\boldsymbol{\mu}_{2,n}^{(t)}$ for $n = 1, ..., N$. Thus, we allow the method to use a few iterations ($t = 1, \dots, T_{train}$) to collect information about the target, as in [8, 5]. Afterwards, the algorithm adapts all the parameters, $\boldsymbol{\mu}_{i,n}^{(t)}$ and $\mathbf{C}_{i,n}^{(t)}$ for $i \in \{1, 2\}$, and the set of active chains $\mathcal{A}_t$, until $t = T_{stop}$. On the one hand, the parameters $\boldsymbol{\mu}_{2,n}^{(t)}$ and $\mathbf{C}_{2,n}^{(t)}$ are updated using the empirical estimators for the means and covariances considering only the samples assigned to the n-th chain. On the other hand, the parameters $\boldsymbol{\mu}_{1,n}^{(t)}$ and $\mathbf{C}_{1,n}^{(t)}$ are adapted considered all the states generated so far. The chains are turned on or off taking into account the number of samples assigned to the n-th chain. PAIM is detailed below.

1. Initialization:

1.1- *Parameters:* choose the number of chains N and the desired samples L. Select the positive values ϵ, T_{train}, T_{stop},[2] the initial parameters, $\boldsymbol{\mu}_{i,n}^{(0)}$ and $\mathbf{C}_{i,n}^{(0)}$ for $i \in \{1, 2\}$, the initial states $\mathbf{x}_{n,0}$, and the counters $m_n = 1$ for $n = 1, \dots, N$. Define the initial set of the active chains as $\mathcal{A}_0 = \{1, 2, \dots, N\}$.

1.2- *Indices:* set $\ell = 0$, $t = -1$, and $k_n = 0$ for $n = 1, \dots, N$.

2. MH steps:

2.1- Set $t = t + 1$, $\mathcal{Z} = \emptyset$ and $i = 0$.

2.2- For all the active chains, i.e., for all the indices $j \in \mathcal{A}_t$:

(a) Sample $\mathbf{x}'$ from the j-th proposal pdf,

$$\mathbf{x}' \sim \psi_j^{(t)}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^{2} q_{i,j}^{(t)}(\mathbf{x}|\boldsymbol{\mu}_{i,j}^{(t)}, \mathbf{C}_{i,j}^{(t)}).$$

[1] Note that we are including all the states in the "burn-in" periods of the N chains. However, they can be discarded if desired.

[2] We recall that it is necessary to set $T_{train} < T_{stop}$.

(b) Accept $\mathbf{x}_{j,k_j+1} = \mathbf{x}'$ with probability

$$\alpha = \min\left[1, \frac{\pi(\mathbf{x}')\psi_j^{(t)}(\mathbf{x}_{j,k_j})}{\pi(\mathbf{x}_{j,k_j})\psi_j^{(t)}(\mathbf{x}')}\right]. \qquad (3)$$

Otherwise, set $\mathbf{x}_{j,k_j+1} = \mathbf{x}_{j,k_j}$.

(c) Set $\mathcal{Z} = \mathcal{Z} \cup \{\mathbf{z}_{i+1} = \mathbf{x}_{j,k_j+1}\}$, and $i = i+1$.

(d) Set $\boldsymbol{\theta}_{\ell+1} = \mathbf{x}_{j,k_j+1}$, $k_j = k_j + 1$ and $\ell = \ell + 1$.

(e) *Stop condition:* If $\ell > L$ then stop and return $\{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_L\}$.

3. Assignment (if $t < T_{stop}$)**:**

3.1- For all the vectors $\mathbf{z}_r \in \mathcal{Z}$, i.e., for $r = 1, \ldots, |\mathcal{Z}|$:

(a) Find the closest mean to $\mathbf{z}_i$ (w.r.t. the Euclidean distance), i.e., find the index

$$n^* = \arg\min_n ||\boldsymbol{\mu}_{2,n}^{(t)} - \mathbf{z}_r||_2^2. \qquad (4)$$

(b) Set $\mathbf{s}_{n^*,m_{n^*}+1} = \mathbf{z}_i$ and $m_{n^*} = m_{n^*} + 1$.

4. Adaptation (if $T_{train} < t < T_{stop}$)**:**

4.1- Set $\mathcal{A}_{t+1} = \emptyset$.

4.2- For $n = 1, \ldots, N$, update the mean vectors,

$$\begin{aligned} \boldsymbol{\mu}_{1,n}^{(t+1)} &= \widehat{\boldsymbol{\mu}}^{(t+1)} = \frac{1}{\ell}\sum_{i=1}^{\ell} \boldsymbol{\theta}_i, \\ \boldsymbol{\mu}_{2,n}^{(t+1)} &= \frac{1}{m_n}\sum_{i=1}^{m_n} \mathbf{s}_{n,i}, \end{aligned} \qquad (5)$$

update the covariance matrices,

$$\begin{aligned} \mathbf{C}_{1,n}^{(t+1)} &= \widehat{\mathbf{C}}^{(t+1)} = \frac{1}{\ell-1}\sum_{i=1}^{\ell}(\boldsymbol{\theta}_i - \widehat{\boldsymbol{\mu}}^{(t+1)})^2 + \epsilon\mathbf{I}_d \\ \mathbf{C}_{2,n}^{(t+1)} &= \frac{1}{m_n-1}\sum_{i=1}^{m_n}(\mathbf{s}_{n,i} - \boldsymbol{\mu}_{2,n}^{(t+1)})^2 + \epsilon\mathbf{I}_d, \end{aligned} \qquad (6)$$

where $\mathbf{I}_d$ is $d \times d$ unit matrix. Moreover, if

$$a_n = \left\lceil N\frac{m_n}{\sum_{j=1}^{N} m_j}\right\rceil > 0, \qquad (7)$$

then set $\mathcal{A}_{t+1} = \mathcal{A}_{t+1} \cup \{n\}$ (where $\lceil a \rceil$ denotes the smallest integer larger or equal than a, with $a \in \mathbb{R}$). Otherwise, if $a_n = 0$ the n-th chain is deactivated, i.e., not used at the next iteration.

5. Repeat from step 2.

4. FURTHER CONSIDERATIONS ABOUT PAIM

Observe that the set $\mathcal{Z}$ contains all the new states generated in one specific step. This is refreshed at the beginning of every step t. Since the number of active chains is variable, the cardinality of $\mathcal{Z}$ is also changing with t. Moreover, we have denoted with $\mathbf{s}_{n,i}$ the i-th state assigned to the n-th chain. The counter m_n indicates the number of states associated to the n-th chain. Note also that for all $t > T_{train}$ we have

$$\boldsymbol{\mu}_{1,n}^{(t)} = \widehat{\boldsymbol{\mu}}^{(t)}, \quad \mathbf{C}_{1,n}^{(t)} = \widehat{\mathbf{C}}^{(t)}, \quad \text{for} \quad n = 1, \ldots, N.$$

Namely, all the functions $q_{1,n}$ are updated using the empirical estimators of the mean and covariance of the target obtained from all the generated samples.

It is important to remark that PAIM is able to distribute the computational efforts efficiently. Indeed, let us consider the initial use of a huge number N of parallel chains, with proposal pdfs localized randomly over the state space. All the chains using a badly located proposal pdf would be quickly deactivated, i.e., only the chains with proposal pdfs located close to high probability regions would survive. However, the algorithm is also able to start up again certain chains if, after some steps, new states have been assigned to them. Finally note that, in the description of the algorithm, the parameters are updated using a block procedure, but efficient recursive update formulas can be employed (e.g., see [5]), so that PAIM can be efficiently applied in high dimensional problems.

5. NUMERICAL RESULTS

We consider a bi-dimensional "banana-shaped" target distribution [8], which is a benchmark function commonly used in the literature. Mathematically, it is given by

$$\bar{\pi}(x_1, x_2) \propto \exp\left(-\frac{1}{2\eta_1^2}\left(4 - Bx_1 - x_2^2\right)^2 - \frac{x_1^2}{2\eta_2^2} - \frac{x_2^2}{2\eta_3^2}\right),$$

where, we have set $B = 10$, $\eta_1 = 4$, $\eta_2 = 5$, and $\eta_3 = 5$. The goal is to estimate the expected value $E[\mathbf{X}]$, where $\mathbf{X} = [X_1, X_2] \sim \bar{\pi}(x_1, x_2)$. We compute the true value $E[\mathbf{X}] \approx [-0.4845, 0]^\top$ approximately by using an exhaustive deterministic numerical method (with an extremely thin grid), in order to obtain the mean square error (MSE) of PAIM and the corresponding independent parallel MH chains with the same initial parameters but no adaptation.

We consider $N \in \{5, 10, 50, 100\}$ chains with Gaussian proposals, $q_{i,n}(\mathbf{x}|\boldsymbol{\mu}_{i,n}^{(0)}, \mathbf{C}_{i,n}^{(0)})$ for $i \in \{1, 2\}$ and $n = 1, \ldots, N$. The initial means and the initial states are chosen randomly at each run. More specifically, we have $\mathbf{x}_{n,0} \sim \mathcal{U}([-15, -15] \times [-15, 15])$ and $\boldsymbol{\mu}_{i,n}^{(0)} \sim \mathcal{U}([-15, -15] \times [-15, 15])$. The initial covariance matrices are $\mathbf{C}_{i,n}^{(0)} = [\sigma^2 \quad 0; \ 0 \quad \sigma^2]^\top$ with $\sigma = 10$. We consider $L = 5000$ total number of samples. For PAIM, we test $T_{train} \in \{1, 10, 20\}$,

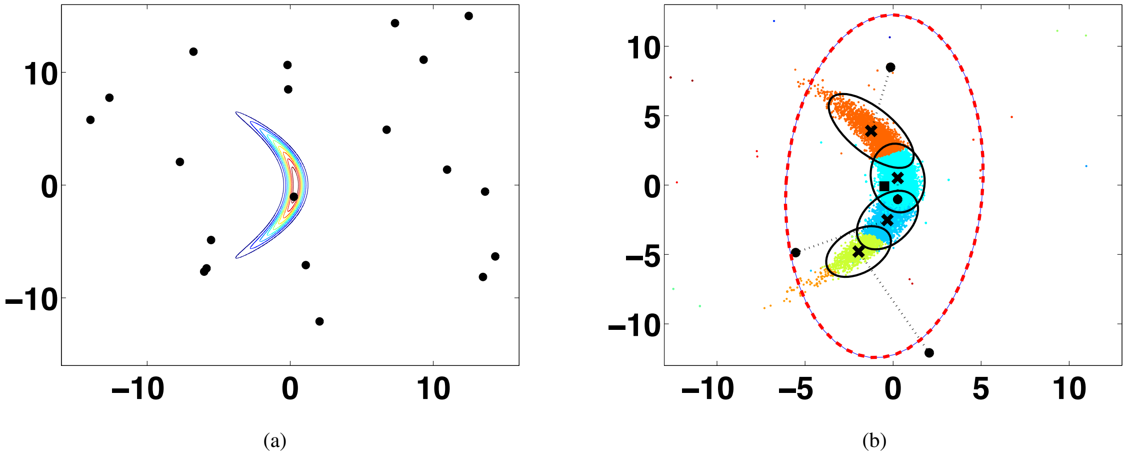

Fig. 1. **(a)** Contour plot of the banana-shaped target $\bar{\pi}$, and the initial means $\boldsymbol{\mu}_{2,n}^{(0)}$ (circles) with $N = 20$. **(b)** We show the generated samples (dots), the initial (circles) and final means (x-marks) of the second component $q_{2,n}$ of the proposal of the final active chains. The final covariance ellipsoids are also depicted. The final mean of the first component of the proposals, $\boldsymbol{\mu}_{1,n}^{(T_{stop})} = \widehat{\boldsymbol{\mu}}^{(T_{stop})}$, is shown with a square jointly with the corresponding covariance ellipsoid with dashed line.

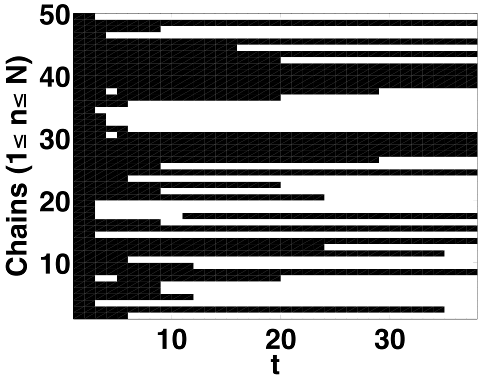

Fig. 2. The indices of the active chains (denoted with black marks) at each step t of PAIM in one specific run, setting $N = 50$, $T_{train} = 2$, $L = 1000$. In this run, the final number of active chains is 16 and $T_{tot} = 38$.

setting $\epsilon = 0.4$ and $T_{stop} = \infty$ (i.e., we never stop the adaptation). The results are averaged over 500 independent simulations for each combination of parameters. PAIM always provides a smaller mean square error (MSE) in the estimation of $E[\mathbf{X}]$ (averaging the two components) with respect to the corresponding independent parallel chains (IPCs). Table 1 shows the percentage of reduction in MSE obtained using PAIM with different values of N and T_{train}, which ranges from 22 % up to 60 % approximately. Indeed, in this example the minimum train period ($T_{train} = 1$) provides the best results. Figure 1(a) shows the initial means of the second components of the proposals in PAIM, i.e., $\boldsymbol{\mu}_{2,n}^{(0)}$, and the contour plot of the target $\bar{\pi}$. Figure 1(b) depict the initial (circles) and final (x-marks) configurations of the means of the second components of the final active proposals, i.e., $\boldsymbol{\mu}_{2,n}^{(0)}$ and $\boldsymbol{\mu}_{2,n}^{(T_{stop})}$, obtained in one specific run (setting $N = 20$ and $L = 5 \times 10^4$). The final mean $\boldsymbol{\mu}_{1,n}^{(T_{stop})} = \widehat{\boldsymbol{\mu}}^{(T_{stop})}$ of the first component common to all the proposals is depicted with a square. The figures also show with solid line the covariance ellipsoids, corresponding to the $\approx 90\%$ of the probability mass, of the second components of the proposals of the final active chains. The covariance ellipsoid corresponding to the first common component of the proposal pdfs is displayed with a dashed line. Figure 2 shows the indices corresponding to the active chains, as a function of t, in one specific run ($N = 50$ and $L = 10^3$).

Table 1. Percentage of reduction in the MSE obtained using PAIM, with respect to IPCs.

T_{train}	$N = 5$	$N = 10$	$N = 50$	$N = 100$
1	51.34%	58.05%	63.78%	60.31%
10	46.95%	41.73%	44.23%	35.65%
20	29.81%	35.51%	33.29%	22.33%

6. CONCLUSIONS

In this work, we have introduced a new interacting parallel MCMC scheme, where a cooperative adaptation of the proposal densities is performed. Furthermore, the computational effort is efficiently distributed by the the novel method among the set of parallel chains, according to their performance.

7. REFERENCES

[1] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer, 2004.

[2] A. Doucet and X. Wang, "Monte Carlo methods for signal processing," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 152–170, Nov. 2005.

[3] W. J. Fitzgerald, "Markov chain Monte Carlo methods with applications to signal processing," *Signal Processing*, vol. 81, no. 1, pp. 3–18, January 2001.

[4] A. Jasra, D. A. Stephens, and C. C. Holmes, "On population-based simulation for static inference," *Statistics and Computing*, vol. 17, no. 3, pp. 263–279, 2007.

[5] D. Luengo and L. Martino, "Fully adaptive Gaussian mixture Metropolis-Hastings algorithm," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 6148–6152.

[6] L. Martino, J. Read, and D. Luengo, "Independent doubly adaptive rejection Metropolis sampling within Gibbs sampling," *IEEE Transactions on Signal Processing*, vol. 63, no. 12, pp. 3123–3138, 2015.

[7] R. Craiu, J. Rosenthal, and C. Yang, "Learn from thy neighbor: Parallel-chain and regional adaptive MCMC," *Journal of the American Statistical Association*, vol. 104, no. 448, pp. 1454–1466, 2009.

[8] Heikki Haario, Eero Saksman, and Johanna Tamminen, "An adaptive Metropolis algorithm," *Bernoulli*, vol. 7, no. 2, pp. 223–242, April 2001.

[9] B. Calderhead, "A general construction for parallelizing Metropolis-Hastings algorithms," *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, vol. 111, no. 49, pp. 17408–17413, 2014.

[10] P. Jacob, C. P. Robert, and M. H. Smith, "Using parallel computation to improve Independent Metropolis-Hastings based estimation," *Journal of Computational and Graphical Statistics*, vol. 3, no. 20, pp. 616–635, 2011.

[11] L. Martino, D. Luengo V. Elvira, J. Corander, and F. Louzada, "Orthogonal parallel MCMC methods for sampling and optimization," *arXiv:1507.08577*, 2015.

[12] J. Corander, M. Ekdahl, and T. Koski, "Parallel interacting MCMC for learning of topologies of graphical models," *Data Mining and Knowledge Discovery*, vol. 17, no. 3, pp. 431–456, 2008.

[13] L. Martino, V. Elvira, D. Luengo, A. Artés-Rodríguez, and J. Corander, "Orthogonal MCMC algorithms," *IEEE Workshop on Statistical Signal Processing (SSP)*, pp. 364–367, June 2014.

[14] L. Martino, V. Elvira, D. Luengo, A. Artés-Rodríguez, and J. Corander, "Smelly parallel MCMC chains," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4070–4074, 2015.

[15] R. Casarin, R. V. Craiu, and F. Leisen, "Interacting multiple try algorithms with different proposal distributions," *Statistics and Computing*, vol. 23, no. 2, pp. 185–200, 2013.

[16] L. Martino, V. Elvira, D. Luengo, and J. Corander, "Layered adaptive importance sampling," *arXiv:1505.04732*, May 2015.