On Improvement of the DE-MC$_{\mathrm{Z}}$ Algorithm with Modes Identification

Aleksandra Skarżyńska[1] and Maciej Romaniuk[2,3][0000−0001−9649−396X]

[1] Faculty of Mathematics and Information Science, Warsaw University of Technology,
Koszykowa 75, 00-662 Warsaw, Poland
aleksandra.sk97@gmail.com

[2] Systems Research Institute PAS, Newelska 6, 01-447 Warszawa, Poland
mroman@ibspan.waw.pl

[3] WIT Academy, Newelska 6, 01-447 Warszawa, Poland

Abstract. The DE-MC$_{\mathrm{Z}}$ algorithm is an improvement of the DE-MC method, which joins the differential evolution with the theory of the Markov chains. It aims to ensure the numerical effectiveness and the convergence speed of the special variant of the Metropolis-Hastings algorithm with the help of an additional, self-adapting initial matrix. In this paper, we add the modes detection procedures to the DE-MC$_{\mathrm{Z}}$ algorithm to increase its abilities in sampling from multimodal target densities. As our numerical experiments suggest, the obtained DE-MC$_{\mathrm{modes}}$ algorithm provides results that give a better fit to the desired target density than the classical approaches.

Keywords: Statistical simulations · Markov chain · Mode detection · Unsupervised learning · Non-parametric model.

1 Introduction

Among other simulation methods, the Monte Carlo (MC) and Markov Chain Monte Carlo (MCMC) algorithms should be mentioned as widely used in both statistics and real-life applications (e.g., [11]). But the generation of random variables from various probability distributions is also important in statistical inference based on fuzzy numbers (e.g., [4–6, 10, 12–15]). If we aim to generate a sample from the complex, multidimensional distribution, then the Metropolis-Hastings (abbreviated further as MH) algorithm can be used. Then, the normal distribution is usually applied as a so-called instrumental density for this method. However, some parameters of this kind of distribution (especially its covariance matrix) affect the numerical effectiveness and convergence speed of the MH algorithm.

To overcome this problem, the special approach, known as the DE-MC, was proposed in [2]. The DE-MC method joins the differential evolution (DE) approach with the theory of the Markov chains (MC) and significantly improves the quality of simulations for the MH method. Then, various modifications of the DE-MC algorithm were also introduced (e.g., [1, 16, 18, 19]).

In this paper, we consider an extension of one of such modifications, namely the DE-MC$_\mathrm{Z}$ method [1]. The proposed DE-MC$_\mathrm{modes}$ algorithm combines the DE-MC$_\mathrm{Z}$ approach with a tool discovering an important characteristic of the probability distributions: their possible modes.

The multimodal densities pose significant problems in statistical inference (e.g., [7]) because they can affect our conclusions if only the single mode (instead of more of the existing ones) is properly identified, especially in the case of the simulated samples. Therefore, we continue the idea from [16] to seek the "unusual phenomena" for the considered target probability distribution that could lead to erroneous numerical outputs or unnecessarily long simulations. But, instead of combining the classical DE-MC algorithm with the identification of the outliers, its newer modification DE-MC$_\mathrm{modes}$ is applied together with the modes detection.

We aim to keep the proposed algorithm as straightforward, intuitively appealing, and numerically effective as this is possible, together with preserving the non-parametric assumption and improving the quality of the simulated results. To achieve this, the two-phase approach was proposed. Firstly, the possible modes are identified and stored in the special matrix. Then, this matrix is used to initialize simulations in the second phase. Therefore, our approach "knows more, better, and sooner" about the possible multimodality of the target distribution than its standard counterpart. To check its efficiency, the DE-MC$_\mathrm{modes}$ algorithm was numerically compared with the DE-MC and DE-MC$_\mathrm{Z}$ approaches. It seems that the proposed method improves the quality of the obtained final sample and/or leads to a decrease in the number of necessary iterations.

The paper is organized as follows. In Sect. 2, some basic facts concerning the DE-MC and DE-MC$_\mathrm{Z}$ algorithms are recalled. The proposed modifications for the DE-MC$_\mathrm{modes}$ algorithm are described in Sect. 3.1, together with the numerical analysis of the conducted comparisons in Sect. 3.2. Then, some final remarks are presented in Sect. 4.

2 Standard DE-MC$_\mathrm{Z}$ Algorithm

The Markov Chain Monte Carlo (MCMC) methods are widely applied to generate a statistical sample from some complex target density $f(x)$ with the help of other, simpler density [11]. But a proper selection of this second density (known as the instrumental one) can pose important problems, related to, e.g. the appropriate form of its covariance matrix if the multivariate normal distribution is used.

A method to overcome this problem was proposed in [2], as the **Differential Evolution Markov Chain** (DE-MC) algorithm. It joins the differential evolution (DE) approach (e.g., [8, 9]) with the Markov chain (MC) theory. In this method, N chains are simulated in parallel, and a state of the i-th chain is given by a d-dimensional vector $\boldsymbol{x}_i$. Then, these vectors are members of a population $\boldsymbol{X}$ and rows of an $N \times d$ matrix, where $N > d$.

Firstly, the primary population is independently drawn from some initial d-dimensional distribution. Then, a new state $\boldsymbol{x}_i^*$ of each chain is sequentially updated using

$$\boldsymbol{x}_i^* = \boldsymbol{x}_i + \gamma(\boldsymbol{x}_{r_1} - \boldsymbol{x}_{r_2}) + \epsilon, \tag{1}$$

where ϵ is generated from a symmetric distribution with small variance and unbounded support, $\gamma > 0$ is a tuning parameter, and $\boldsymbol{x}_{r_1}, \boldsymbol{x}_{r_2}$ are randomly selected from $\boldsymbol{X}$ devoid of $\boldsymbol{x}_i$. With (1), the Markov chain that reaches the whole state space is constructed (contrary to the classical DE scheme [2]). If $N(0, b \cdot \mathbf{1}^d)$, i.e., the d-dimensional normal distribution with zeros for the expected value and variance b, is used to generate ϵ, then $b > 0$ is usually some small value compared with that of the target distribution [1] to ensure necessary properties of the respective MC (see also [11]). Next, $\boldsymbol{x}_i^*$ is accepted as a new state with the probability

$$p(\boldsymbol{x}_i, \boldsymbol{x}_i^*) = \begin{cases} \min\left\{\frac{f(\boldsymbol{x}_i^*)}{f(\boldsymbol{x}_i)}, 1\right\} & \text{if } f(\boldsymbol{x}_i) > 0 \\ 1 & \text{if } f(\boldsymbol{x}_i) = 0 \end{cases} \tag{2}$$

or the i-th chain remains in its previous state $\boldsymbol{x}_i$, where $f(\boldsymbol{x})$ is our target density for the stationary probability distribution in this MCMC scheme (see also [11]).

To overcome the necessary assumption $N > d$, a modified version of the DE-MC algorithm was proposed in [1]. In this method, known as the **DE-MC$_{\mathbf{Z}}$** algorithm, the special matrix $\boldsymbol{Z}$ contains the current and past states of the chains, and $\boldsymbol{X}$ stores the values of the current population. The matrix $\boldsymbol{Z}$ is initialized with a sample of size M_0 of d-dimensional starting vectors, and then sequentially updated after each K (known as the thinning rate) repetitions of the main loop (i.e., the generations of the matrix $\boldsymbol{X}$) with vectors from $\boldsymbol{X}$. Instead of proposing new values from $\boldsymbol{X}$ as in (1), all values from the matrix $\boldsymbol{Z}$ are used, respectively (see Algorithm 1).

The user-defined stop condition for the above-described procedure (line 4 in Algorithm 1) can be related to the pre-specified number of all steps, some convergence measure for the obtained mean or other statistic of the output like the $\hat{R}$ Gelman-Rubin statistic.

3 Modified DE-MC$_{\mathbf{modes}}$ Algorithm

Continuing the idea from [16], we improved the DE-MC$_{\mathrm{Z}}$ algorithm with the modes detection and selection steps to obtain its new variant – the DE-MC$_{\mathrm{modes}}$ method (see Sect. 3.1). This modified approach was numerically compared with the classical DE-MC and DE-MC$_{\mathrm{Z}}$ algorithms. The examples of these simulations are described in Sect. 3.2. Other results are available upon request.

3.1 Introduced Modifications

The proposed DE-MC$_{\mathrm{modes}}$ method consists of two main steps (see Algorithm 2):

1. During the first phase, the specially modified version of the DE-MC$_{\mathrm{Z}}$ approach starting from the matrix $\boldsymbol{Z}^*$ with the generated initial population

Algorithm 1 DE-MC$_\mathrm{Z}$ standard algorithm [1]

```
Require: Parameters of the algorithm: initial population, $M_0, N, K, b, d, \gamma, f(\boldsymbol{x})$
Ensure: The matrix $\boldsymbol{Z}$ with the current and past states of the chain
Generate $M_0$ members of the initial population and store them in the matrix $\boldsymbol{Z}$
Copy $N$ first vectors of $\boldsymbol{Z}$ to $\boldsymbol{X}$
$M \leftarrow M_0, j \leftarrow 0$
while stop condition is not fulfilled do
    for $k = 1, 2, \ldots, K$ do
        $j \leftarrow j + 1$
        for $i = 1, 2, \ldots, N$ do
            Take the $i$-th row $\boldsymbol{x}_i$ from $\boldsymbol{X}$
            Select randomly $\boldsymbol{z}_{r_1}, \boldsymbol{z}_{r_2}$ from $\boldsymbol{Z}$ other than $\boldsymbol{x}_i$
            Generate $\epsilon \sim N(0, b \cdot \mathbf{1}^d)$
            $\boldsymbol{x}_i^* = \boldsymbol{x}_i + \gamma(\boldsymbol{z}_{r_1} - \boldsymbol{z}_{r_2}) + \epsilon$
            Set $\boldsymbol{x}_i = \begin{cases} \boldsymbol{x}_i^* & \text{with probability } p(\boldsymbol{x}_i, \boldsymbol{x}_i^*) \\ \boldsymbol{x}_i & \text{with probability } 1 - p(\boldsymbol{x}_i, \boldsymbol{x}_i^*) \end{cases}$
        Append $\boldsymbol{X}$ to $\boldsymbol{Z}$, $M \leftarrow M + N$
return $\boldsymbol{Z}$
```

tries to identify the modes (defined as local maxima of $f(\boldsymbol{x})$) and append them to the auxiliary matrix $\boldsymbol{V}$. These procedures are applied every K_{modes}-th step to increase the numerical effectiveness of the algorithm and ensure its convergence properties.

2. In the second phase, the new DE-MC$_\mathrm{Z}$ algorithm is started using $\boldsymbol{V}$ as its initial matrix $\boldsymbol{Z}$ (see Algorithm 1). As its stop condition, the $\hat{R}$ Gelman-Rubin statistic (e.g., [3, 16, 17]) is used with the threshold R close to one (as indicated in the literature). When the estimated value of $\hat{R}$ is lower than the selected R, the final output is returned.

The main aim of these modifications is to improve the effectiveness and quality of the generated samples in the case of complex target densities, especially the multimodal ones. Additionally, the value of γ in (1) is dynamically changed to easily reach the possible modes, especially if they are far from each other.

Contrary to the standard DE-MC$_\mathrm{Z}$ approach, the generated initial population is stored in the matrix $\boldsymbol{Z}^*$, which also contains the previous and current states of the preliminary phase (steps 1–15 in Algorithm 2). Then, the detection of the modes is done after each K_{modes} repetition of the main loop related to updating the matrix $\boldsymbol{X}$ (steps 7–14 in Algorithm 2), where $\boldsymbol{X}$ consists of the current values of the whole population. To find the possible modes, observations are compared with their k_{modes} neighbors. Their distances are calculated according to the Euclidean metric. Then, these vectors with their $2n_{\mathrm{modes}}$ closest (in the same metric) neighbors are added to $\boldsymbol{V}$ (step 16 in Algorithm 2). To avoid the possibility that the neighbors selected in this way are too far from the considered mode, a special threshold was introduced. Its value is related to the standard deviation of $f(\boldsymbol{x})$ for all vectors in the current population. Next, the matrix $\boldsymbol{V}$ is

Algorithm 2 DE-MC$_{\text{modes}}$ modified algorithm

```
Require: Parameters of the algorithm: initial population, $M_0, N, N^*, K, K^*, K_{\text{modes}}, L, b, d, \gamma, f(\boldsymbol{x}), n_{\text{modes}}, k_{\text{modes}}$
Ensure: The matrix $\boldsymbol{Z}$ with the current and past states of the chain
Generate $M_0$ members of the initial population and store them in the matrix $\boldsymbol{Z}^*$
Copy $N^*$ first vectors of $\boldsymbol{Z}^*$ to $\boldsymbol{X}$
$M^* \leftarrow M_0, j \leftarrow 0$
for $l = 1, 2, \ldots, L$ do
    for $k = 1, 2, \ldots, K^*$ do
        $j \leftarrow j + 1$
        for $i = 1, 2, \ldots, N^*$ do
            Take the $i$-th row $\boldsymbol{x}_i$ from $\boldsymbol{X}$
            Select randomly $\boldsymbol{z}^*_{r_1}, \boldsymbol{z}^*_{r_2}$ from $\boldsymbol{Z}^*$ other than $\boldsymbol{x}_i$
            Generate $\epsilon \sim N(0, b \cdot \mathbf{1}^d)$
            $\boldsymbol{x}^*_i = \boldsymbol{x}_i + \gamma(\boldsymbol{z}^*_{r_1} - \boldsymbol{z}^*_{r_2}) + \epsilon$
            Set $\boldsymbol{x}_i = \begin{cases} \boldsymbol{x}^*_i & \text{with probability } p(\boldsymbol{x}_i, \boldsymbol{x}^*_i) \\ \boldsymbol{x}_i & \text{with probability } 1 - p(\boldsymbol{x}_i, \boldsymbol{x}^*_i) \end{cases}$
        Append $\boldsymbol{X}$ to $\boldsymbol{Z}^*$, $M^* \leftarrow M^* + N^*$
        if $j$ is divisible by $K_{\text{modes}}$ then
            Find the modes in $\boldsymbol{X}$ and append them to $\boldsymbol{V}$
$\boldsymbol{Z} \leftarrow \boldsymbol{V}$
Copy $N$ last vectors of $\boldsymbol{Z}$ to $\boldsymbol{X}$
Apply steps 3–14 of Algorithm 1
```

used as the initial population for the second phase of this approach. The aim is to "populate" the main part of the algorithm with starting points that are close to possible modes. Please note, that because of the random and approximate nature of the above-described procedure, we can not precisely state the size of the respective sample stored in $\boldsymbol{V}$ (i.e. if there is only one vector or more there, see, e.g., Fig. 3e). But even one vector can be important as "almost the best" starting point. If necessary, one can add more vectors to $\boldsymbol{V}$ originated from, e.g., $\boldsymbol{Z}^*$ or even $\boldsymbol{X}$.

To faster reach the possible modes, the parameter γ is changed every certain number of steps and then restored to its primary value. This modification allows for "jumping" between the modes (especially if they are far from each other) but preserves the convergence properties of the Markov chain.

3.2 Numerical Analysis

The proposed algorithm was numerically compared with its classical counterparts, the DE-MC and DE-MC$_{\mathrm{Z}}$ methods, using procedures written in R. In this section, we present results concerning different target multimodal densities. Other examples and graphs are available upon request.

In the first example, $f(\boldsymbol{x})$ was a mixture of two normal densities with modes that are rather distant from each other, namely

$$1/6 \cdot N(-8, 1) + 5/6 \cdot N(38, 1), \tag{3}$$

where $N(\mu, \sigma)$ stands for the normal distribution with the mean μ and standard deviation σ. The algorithms were started with the initial population generated from $N(15, 2)$ and parameters $N^* = N = 20, M_0 = 25, K^* = 5, K = 10, \gamma = 2.38/\sqrt{2}, R = 1.1, L = 200, n_{\text{modes}} = 2, k_{\text{modes}} = 4, b = 0.01$. The values of these parameters were chosen with the thumb rule based on the previous experiments [16] or obtained directly from the literature (e.g., such a value of γ is considered in [2] as a sensible choice motivated by comparison with the RWNM – random walk with a normal jumping distribution) to ensure some balance between the numerical effectiveness of the algorithm and quality of the output. It seems that the results are not very sensitive to values of these parameters coming from a reasonable range but additional experiments are still necessary.

The scaling parameter was changed to $\gamma = 1$ every 20 iterations of the algorithm to help reach the distant local maxima of (3), the possible modes were compared with their four neighbors, and the found modes with these 4 neighboring vectors were added to $\boldsymbol{V}$.

The DE-MC approach converged after $j = 1449$ iterations, the DE-MC$_{\text{Z}}$ needed even more steps $j = 2299$, and DE-MC$_{\text{modes}}$ required only 1399 iterations. Moreover, the estimated density for the last iteration of the DE-MC$_{\text{modes}}$ algorithm was clearly closer to the target function (see Fig. 1c) than in the case of the DE-MC (see Fig. 1a) and DE-MC$_{\text{Z}}$ (see Fig. 1b) methods, especially for the left-hand side mode. In these graphs, the estimated density is denoted by the thin, blue line, and the target function – by the thick, purple one. The estimated means for all algorithms converged to the proper expected value (see Fig. 1d), but the DE-MC$_{\text{Z}}$ (denoted by the blue line in this graph) exhibited the slowest convergence in comparison with the DE-MC (the green line) and DE-MC$_{\text{modes}}$ (the orange line, respectively). The above-mentioned colors of the graphs have the same meaning in the following figures.

The convergence of the means was also analyzed, as the mean is one of the most important descriptive statistic of the sample and also the ultimate goal in many of the MCMC schemes.

When the initial population was generated from $N(0, 1)$ instead of $N(15, 2)$, as described previously, the effectiveness of the DE-MC$_{\text{modes}}$ was even more clear. The DE-MC$_{\text{modes}}$ required 699 iterations to fulfill the previously specified stop condition, and the DE-MC$_{\text{Z}}$ did not converge even after 3000 steps. The parameter γ was changed to 5 every 10 iterations for both methods. Moreover, the estimated density for the last iteration of the DE-MC$_{\text{modes}}$ (see Fig. 2a) is significantly better fitted to $f(\boldsymbol{x})$ than for the DE-MC$_{\text{Z}}$ (see Fig. 2b).

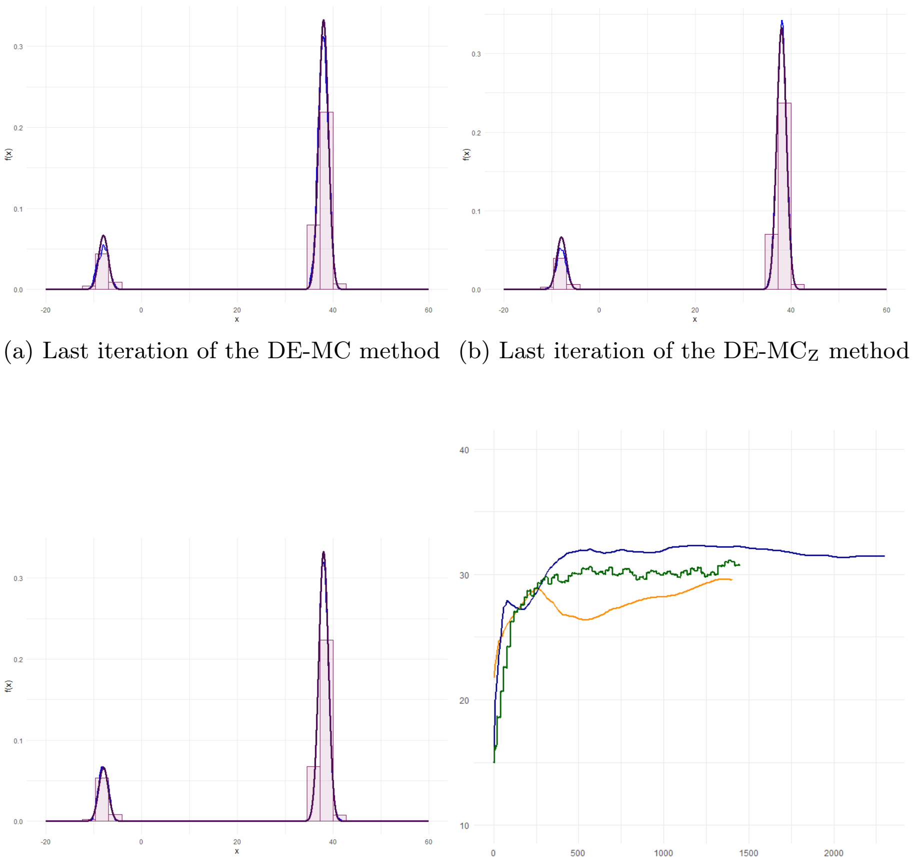
In the second example, a mixture of two 2-dimensional normal distributions

$$0.6 \cdot N(\mu_1, \Sigma_1) + 0.4 \cdot N(\mu_2, \Sigma_2) \tag{4}$$

with

$$\mu_1 = [10, 10], \Sigma_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \mu_2 = [25, 25], \Sigma_2 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 2 \end{bmatrix}, \tag{5}$$

was used. In this case, the modes were rather close to each other, but they differed concerning their heights (see Fig. 3a). During the simulations, the parameters



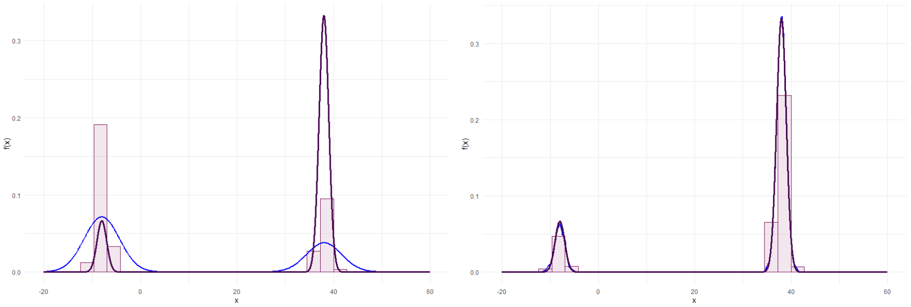
(a) Last iteration of the DE-MC method

(b) Last iteration of the DE-MC$_{\mathrm{Z}}$ method

(c) Last iteration of the DE-MC$_{\mathrm{modes}}$ method

(d) Convergence of the means

Fig. 1. Comparison of the numerical results for the mixture of two normal distributions for $N(15,2)$ starting distribution.



(a) Last iteration of the DE-MC$_Z$ method (b) Last iteration of the DE-MC$_{\text{modes}}$ method

Fig. 2. Comparison of the numerical results for the mixture of two normal distributions for $N(0,1)$ starting distribution.

$N^* = 30, N = 20, M_0 = 35, K^* = 5, K = 5, \gamma = 2.38/\sqrt{2}, R = 1.05, L = 400, n_{\text{modes}} = 0, k_{\text{modes}} = 8$ were set. To better identify the modes in this 2-dimensional setting, each possible mode was compared with its 8 neighbors. Because there were problems with fulfilling the stop condition for all of the algorithms, the simulations were conducted for 3000 iterations. The DE-MC$_{\text{modes}}$ gave the best fit to the target density, which was especially seen for the Y axis (see Figs. 3b–3d). Also the modes were properly identified in the obtained matrix $\boldsymbol{V}$ (see Fig. 3e).

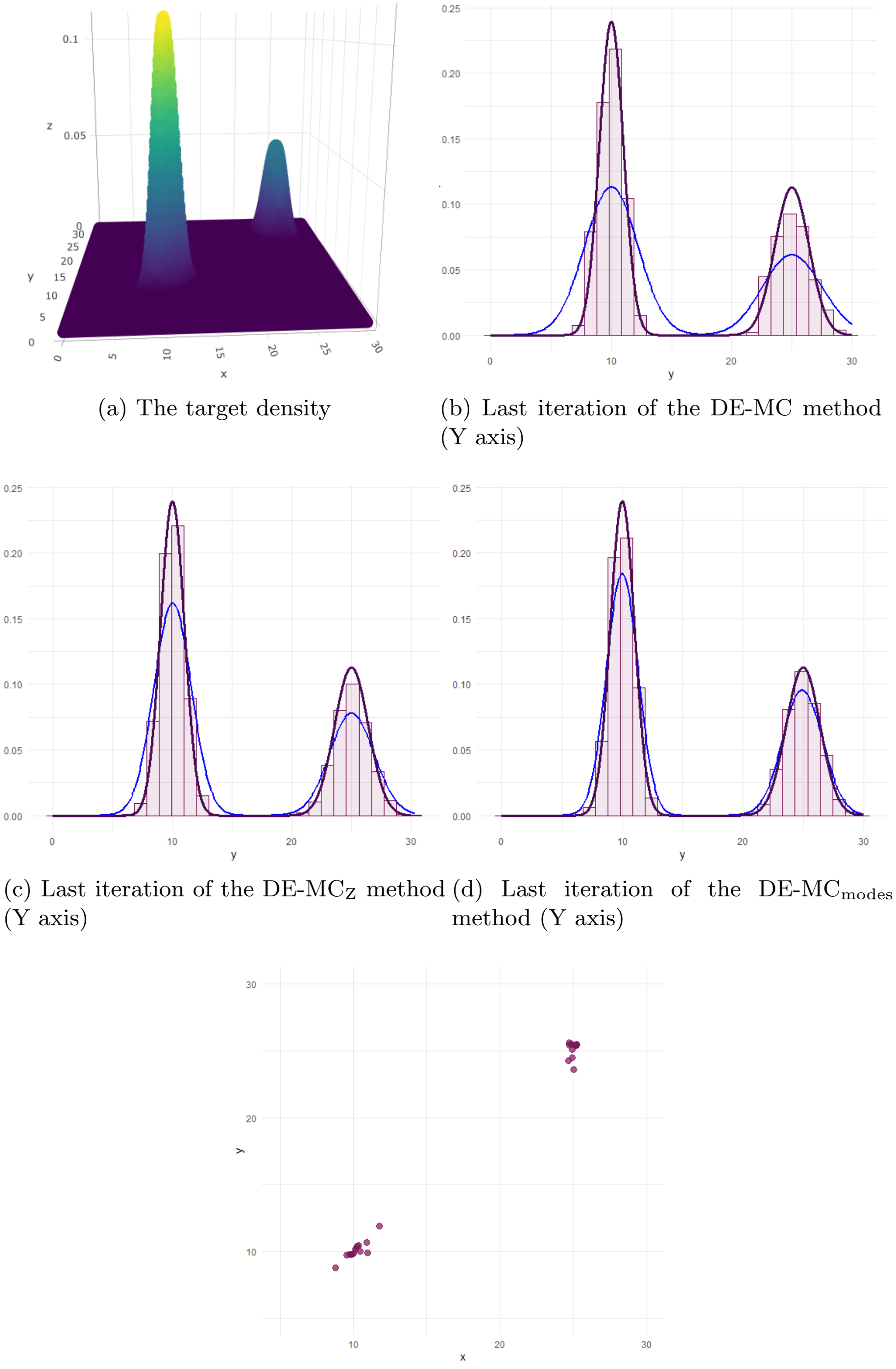
The third example concerned the mixture of three logistic distributions

$$0.4 \cdot \text{Log}(-23, 1.5) + 0.3 \cdot \text{Log}(-14, 1.6) + 0.3 \cdot \text{Log}(28, 1.2), \tag{6}$$

where $\text{Log}(\mu, s)$ stands for the logistic distribution with the location μ and scale parameter s. In this case, the two modes were close to each other, and the third one – more distant from them. The algorithms were started from $N(0,1)$ with the parameters $N^* = N = 20, M_0 = 25, K^* = 5, K = 10, \gamma = 2.38/\sqrt{2}, R = 1.05, L = 200, n_{\text{modes}} = 0, k_{\text{modes}} = 2$. The parameter γ was not altered during iterations. The DE-MC method did not fulfill the stop condition and was called a halt after the specified 7000 iterations, the DE-MC$_Z$ converged sooner, after 4049 steps, but the DE-MC$_{\text{modes}}$ required more iterations – 6749 in this case. However, the results of the DE-MC$_{\text{modes}}$ approach showed their best fit to the target density, and all of the existing three modes were clearly identified (see Figs. 4a–4c). In Fig. 4d, the convergence of the means can be observed. It seems that the DE-MC$_{\text{modes}}$ method had greater flexibility in jumping among the modes.

4 Conclusion

To generate a random sample from a complex density, various algorithms were suggested in the literature. Such samples are then used, e.g., in statistical infer-



(a) The target density

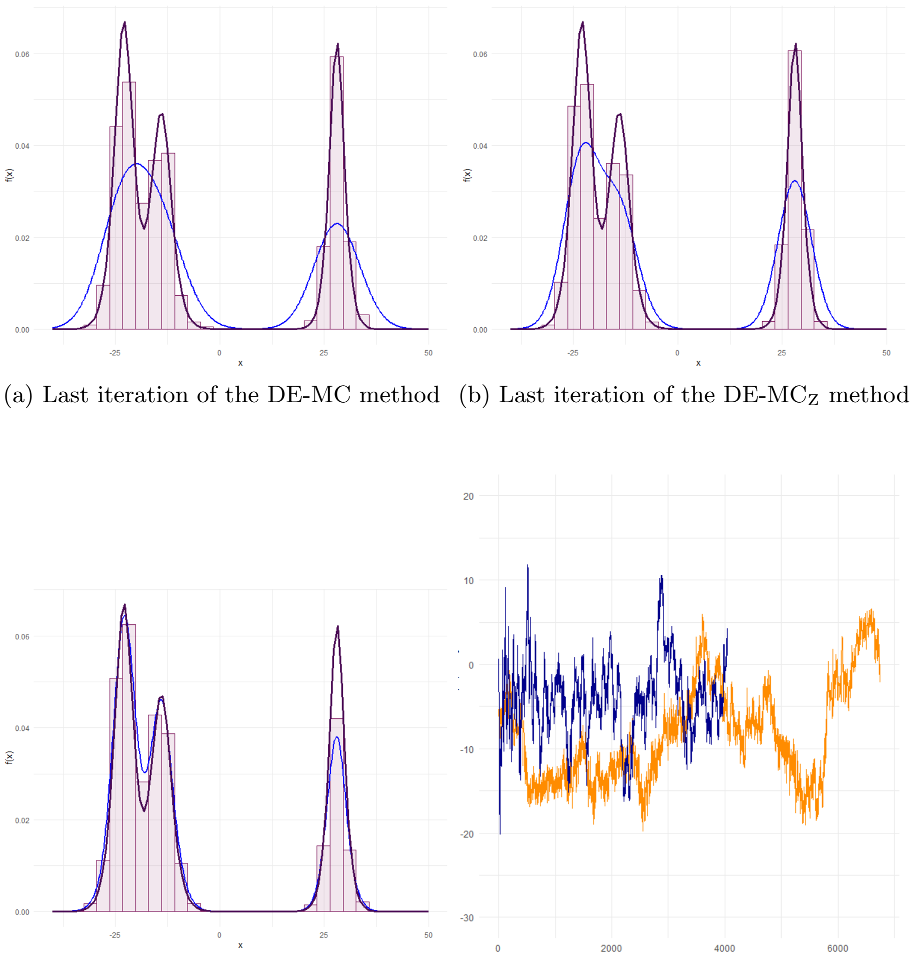
(b) Last iteration of the DE-MC method (Y axis)

(c) Last iteration of the DE-MC$_Z$ method (Y axis)

(d) Last iteration of the DE-MC$_{\text{modes}}$ method (Y axis)

(e) The modes identified in the DE-MC$_{\text{modes}}$ method

Fig. 3. Comparison of the numerical results for the mixture of two 2-dimensional normal distributions.



(a) Last iteration of the DE-MC method (b) Last iteration of the DE-MC$_{\mathrm{Z}}$ method

(c) Last iteration of the DE-MC$_{\mathrm{modes}}$ method (d) Convergence of the means

Fig. 4. Comparison of the numerical results for the mixture of three logistic distributions.

ence problems including crisp or fuzzy data. To overcome issues related to multimodal target densities, we improved the DE-MC$_Z$ method with the modes detection and selection steps. The introduced two-phase method, the DE-MC$_{\text{modes}}$ approach, was numerically compared with its classical counterparts – the DE-MC and DE-MC$_Z$ algorithms. It seems that the DE-MC$_{\text{modes}}$ method generates samples that are closer to the desired target distribution and converges faster. Obviously, further experiments are still necessary, e.g., the probability distributions with almost "merging" or some "artificial" modes can be numerically analyzed. Equipping the DE-MC methods with other statistical tools to identify the existing "unusual phenomena" of the sampling distributions can be also fruitful in future research.

References

1. ter Braak, C.J.F., Vrugt, J.A.: Differential Evolution Markov Chain with snooker updater and fewer chains. Statistics and Computing **18**(4), 435–446 (2008). https://doi.org/10.1007/s11222-008-9104-9
2. ter Braak, C.J.F.T.: A Markov Chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces. Statistics and Computing **16**(3), 239–249 (2006). https://doi.org/10.1007/s11222-006-8769-1
3. Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., Rubin, D.: Bayesian Data Analysis. Chapman and Hall/CRC (2013). https://doi.org/10.1201/b16018
4. Grzegorzewski, P., Hryniewicz, O., Romaniuk, M.: Flexible bootstrap for fuzzy data based on the canonical representation. International Journal of Computational Intelligence Systems **13**(1), 1650–1662 (2020). https://doi.org/10.2991/ijcis.d.201012.003
5. Grzegorzewski, P., Romaniuk, M.: Bootstrap methods for epistemic fuzzy data. International Journal of Applied Mathematics and Computer Science **32**(2), 285–297 (2022). https://doi.org/10.34768/amcs-2022-0021
6. Grzegorzewski, P., Romaniuk, M.: Bootstrap methods for fuzzy data. Lecture Notes in Networks and Systems **338 LNNS**, 28–47 (2022). https://doi.org/10.1007/978-3-030-95929-6_3
7. Jasra, A., Holmes, C.C., Stephens, D.A.: Markov Chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. Statistical Science **20**(1), 50–67 (2005). https://doi.org/10.1214/088342305000000016
8. Opara, K., Arabas, J.: Comparison of mutation strategies in differential evolution - a probabilistic perspective. Swarm and Evolutionary Computation **39**, 53–69 (2018). https://doi.org/10.1016/j.swevo.2017.12.007
9. Opara, K.R., Arabas, J.: Differential evolution: A survey of theoretical analyses. Swarm and Evolutionary Computation **44**, 546–558 (2019). https://doi.org/10.1016/j.swevo.2018.06.010
10. Parchami, A., Grzegorzewski, P., Romaniuk, M.: Statistical simulations with LR random fuzzy numbers. Statistical Papers (2024). https://doi.org/10.1007/s00362-024-01533-5, (in print)
11. Robert, C.P., Casella, G.: Monte Carlo Statistical Methods. Springer-Verlag, Berlin, Heidelberg (2005)

12. Romaniuk, M.: Analysis of the insurance portfolio with an embedded catastrophe bond in a case of uncertain parameter of the insurer's share. Advances in Intelligent Systems and Computing **524**, 33–43 (2017). https://doi.org/10.1007/978-3-319-46592-0_3
13. Romaniuk, M.: On some applications of simulations in estimation of maintenance costs and in statistical tests for fuzzy settings. In: Steland, A., Rafajłowicz, E., Okhrin, O. (eds.) Stochastic Models, Statistics and Their Applications. pp. 437–448. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-28665-1_33
14. Romaniuk, M., Grzegorzewski, P.: Resampling fuzzy numbers with statistical applications: FuzzyResampling package. The R Journal **15**, 271–283 (2023). https://doi.org/10.32614/RJ-2023-036
15. Romaniuk, M., Hryniewicz, O.: Discrete and smoothed resampling methods for interval-valued fuzzy numbers. IEEE Transactions on Fuzzy Systems **29**(3), 599–611 (2021). https://doi.org/10.1109/TFUZZ.2019.2957253
16. Rychlik, K.M., Romaniuk, M.: Improved DE-MC algorithm with automated outliers detection. In: Massanet, S., Montes, S., Ruiz-Aguilera, D., González-Hidalgo, M. (eds.) Fuzzy Logic and Technology, and Aggregation Operators. pp. 701–712. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-39965-7_58
17. Vats, D., Knudson, C.: Revisiting the Gelman-Rubin diagnostic (2018). https://doi.org/10.48550/ARXIV.1812.09384, https://arxiv.org/abs/1812.09384
18. Vrugt, J.A., ter Braak, C.F.: DREAM(D): an adaptive Markov Chain Monte Carlo simulation algorithm to solve discrete, noncontinuous, and combinatorial posterior parameter estimation problems. Hydrology and Earth System Sciences **15**(12), 3701–3713 (2011). https://doi.org/10.5194/hess-15-3701-2011
19. Vrugt, J.A., ter Braak, C., Diks, C., Robinson, B.A., Hyman, J.M., Higdon, D.: Accelerating Markov Chain Monte Carlo simulation by Differential Evolution with self-adaptive randomized subspace sampling. International Journal of Nonlinear Sciences and Numerical Simulation **10**(3), 273–290 (2009). https://doi.org/10.1515/IJNSNS.2009.10.3.273