Evol. Intel. (2010) 3:79–101
DOI 10.1007/s12065-010-0040-1

RESEARCH PAPER

Recombination operators and selection strategies for evolutionary Markov Chain Monte Carlo algorithms

Madalina M. Drugan · Dirk Thierens

Received: 23 February 2010 / Revised: 25 June 2010 / Accepted: 2 July 2010 / Published online: 21 July 2010

Abstract *Markov Chain Monte Carlo* (MCMC) methods are often used to sample from intractable target distributions. Some MCMC variants aim to improve the performance by running a population of MCMC chains. In this paper, we investigate the use of techniques from Evolutionary Computation (EC) to design population-based MCMC algorithms that exchange useful information between the individual chains. We investigate how one can ensure that the resulting class of algorithms, called *Evolutionary MCMC* (EMCMC), samples from the target distribution as expected from any MCMC algorithm. We analytically and experimentally show—using examples from discrete search spaces—that the proposed EMCMCs can outperform standard MCMCs by exploiting common partial structures between the more likely individual states. The MCMC chains in the population interact through recombination and selection. We analyze the required properties of recombination operators and acceptance (or selection) rules in EMCMCs. An important issue is how to preserve the detailed balance property which is a sufficient condition for an irreducible and aperiodic EMCMC to converge to a given target distribution. Transferring EC techniques to population-based MCMCs should be done with care. For instance, we prove that EMCMC algorithms with an elitist acceptance rule do not sample the target distribution correctly.

M. M. Drugan (✉) · D. Thierens
Department of Information and Computing Sciences,
Utrecht University, PO. Box 80.089,
3508 TB Utrecht, The Netherlands
e-mail: madalina@cs.uu.nl

D. Thierens
e-mail: dirk@cs.uu.nl

Keywords Evolutionary Markov chain Monte Carlo · Detailed balance · Recombination · Acceptance rules

1 Introduction

Markov Chain Monte Carlo (MCMC) is a framework of algorithms for sampling from complicated distributions. The use of MCMC in Machine Learning has recently been advocated by [1]. Usually, a single MCMC is run until it converges to the stationary distribution. To improve their efficiency, some MCMC variants consist of a population of chains that interact by exchanging useful information and at the same time preserve the MCMC convergence characteristics at the population level. In this paper, we are particularly interested in techniques that use multiple interacting chains in parallel as opposed to a single chain.

The stochastic process of Evolutionary Computation (EC) and MCMC algorithms is basically similar: both are Markov chains with fixed transition matrices between individual states, for instance transition matrices given by mutation and recombination operators for EC and by perturbation operators for MCMC. Furthermore, both Metropolis-Hastings—a subclass of MCMCs—and EC algorithms have a selection step, the acceptance rule, to propagate good individuals to the next generation. There are also many differences induced by the different scope of these algorithms: EC is used for optimization and MCMC is used for sampling. Additionally, MCMC uses a single chain whereas EC algorithms use a population of individuals that interact. Motivated by the common points of these two algorithms, we have previously discussed the Evolutionary MCMC (EMCMC) framework which aims to improve the efficiency of standard MCMC algorithms [7, 8]. EMCMC is a population-based MCMC that

exchanges information between the individual chains such that at population level it is still an MCMC.

In general, it is not straightforward to integrate interaction between chains, like recombination or selection, into population based MCMCs and to preserve the convergence to the target distribution. To ease proving that EMCMCs converge to the stationary distribution the individuals generated with recombinative operators (an alternation between mutation and recombination operators) should be all accepted or all rejected [8, 16] with a so called *coupled acceptance rule*. Note the difference between this coupled acceptance and the popular selection strategies in EC; the coupled acceptance rule is selective at the family (i.e., the set of children generated by a set of parents) level whereas the selection strategies are selective at individual level that is one of the children competes against one of its parents. Using the standard MH acceptance rule where only one of the multiple children generated from multiple parents is accepted/rejected is a straightforward alternative algorithm [27]. It is interesting to note that Mahfoud and Goldberg [17] also obtained good results for *Simulated Annealing* (SA) [14] algorithms where one child competes against one of the parents. However, such a recombinative EMCMC does not fit in the standard framework of Metropolis-Hastings algorithms. Some alternative solutions proposed previously restrict the proposal distributions that generate new individuals by generating only one child at the time from a family of parents [3, 5, 15, 23, 24]. For example, [15] proposed an EMCMC algorithm that uses a population based univariate distribution to sample from likely Bayesian network structures. Other algorithms, for example some population-based adaptive MCMCs [9] and sequential Monte Carlo [6], relax the Markov property at the price of more difficult convergence properties and usage by practitioners.

In this paper, we theoretically and experimentally study various recombination operators and their interaction with acceptance rules resulting into *EMCMCs* with a required target distribution. We investigate the properties of several popular recombination operators in GAs (i.e., uniform recombination) when integrated in the EMCMC framework. We show that the individuals that interact in generating candidate individuals should also interact in the acceptance rule to sample from the target distribution. Acceptance rules that are directly derived from the EC's selection strategies are more useful for optimization than for sampling. The sampled distribution is skewed compared with the target distribution: the fit states of the search space are amplified and the less fit states are diminished.

We propose a general method that corrects the target distribution of a recombinative EMCMC that does not sample from the intended distribution. This technique simply considers the recombinative EMCMC as the proposal distribution and the generated children are all accepted/rejected with a coupled acceptance rule. In this way we postpone the acceptance or rejection of all children with the hope that the recombinative EMCMC generates fit individuals that will increase the chance that children are accepted and, consequently, that the algorithm converges faster to the target distribution. This method has theoretical value constructing a correction term with which the sampled distribution should be multiplied to transform it into the target distribution.

We compare in practice the performance of various recombinative and non-recombinative EMCMCs with the standard and the population-based MCMC. When comparing (E)MCMCs we respond to three questions: (1) how useful are EMCMCs when compared with MCMCs, (2) how useful are the recombinative operators and (3) what is the difference in performance between EMCMCs using the standard MH acceptance rule selective at individual level and EMCMCs using the coupled acceptance rules. The recombinative operators chosen are the most popular operators in EC: discrete space uniform recombination and uniform mutation. As a consequence, the theory and the practical examples are formulated for the discrete space (E)MCMCs. We also mention that it is straightforward to extend these results to the continuous space (E)MCMCs.

For our first experiment we analytically compare the algorithms an a toy example such that the exact performance of algorithms is calculated from all the transitions between all the states of an (E)MCMCs. In the second experiment we calculate the Kullback-Leiber distance between the target distribution and the distribution output by an algorithm after a finite number of steps on a relatively small size *binary quadratic programming problem* (BQP) to exactly compute the target distribution. The next experiment is on a larger size BQP where we can compare the performance of (E)MCMCs using only graphical (and more imprecise) tests. Note that BQP is related to the popular mathematical problem in statistical mechanics known as the Ising model [10]. The obtained results show that recombination improves the mixing of the EMCMC especially when the standard MH acceptance rule is used with recombination.

1.1 Outline of the paper

Section 2 presents some basic knowledge of MCMC algorithms and introduces the notation used in the rest of the paper. For an in depth study on MCMCs we refer the reader to [12]. In Sect. 3 the EMCMC framework is presented. In Sect. 4 we investigate several recombination operators and their desired properties for EMCMCs. Section 5 proposes and analyzes various MH acceptance rules and the properties of the resulting EMCMCs when

combined with the recombinative operators. We also establish rules to design recombinative EMCMCs for sampling and optimization. In Sect. 6 we analytically investigate the discussed EMCMCs on a toy problem and experimentally test them on two BQP problem instances. Section 7 concludes and discusses the results of the paper.

2 Background: MCMC framework

MCMC is a general framework to generate samples $X^{(t)}$ from a probability distribution $P(\cdot)$ while exploring its so-called countable ℓ-dimensional state (or search) space E using a *Markov chain*. We assume the state space is compact. MCMC does not sample directly from $P(\cdot)$, but only requires that it can be evaluated within a multiplicative constant $P(X) = \hat{P}(X)/Z$, where Z is a normalization constant and $\hat{P}(\cdot)$ the unnormalized target distribution. A discrete time Markov chain is a stochastic process $(X^{(0)}, X^{(1)}, \ldots)$ with the property that the probability distribution for the state $X^{(t)}$ given all previous values $(X^{(0)}, X^{(1)}, \ldots, X^{(t-1)})$ only depends on $X^{(t-1)}$. Mathematically, we can write

$$P(X^{(t)} \mid X^{(0)}, X^{(1)}, \ldots, X^{(t-1)}) = P(X^{(t)} \mid X^{(t-1)})$$

We call $P(X^{(t)} \mid X^{(t-1)})$ the *transition matrix* of the Markov chain. A *homogeneous* Markov chain in addition, has a time-independent transition matrix. In the following we only consider homogeneous Markov chains, unless specified otherwise. Aperiodicity excludes for instance that certain points can only be reached at even times. For any starting point a Markov chain with a finite state-space converges to a unique invariant distribution if it is irreducible and aperiodic. A Markov chain is called irreducible if, and only if, every state can be reached from every other state in a finite number of steps.

A sufficient, but not necessary, condition to ensure that the given distribution $P(\cdot)$ is the stationary distribution is that it satisfies the *detailed balance condition* [1]. A MCMC satisfies the detailed balance condition if, and only if, the probability to move from X to Y multiplied by the probability to be in X is equal to the probability to move from Y to X multiplied by the probability to be in Y:

$$P(Y \mid X) \cdot P(X) = P(X \mid Y) \cdot P(Y)$$

2.1 Metropolis-Hastings algorithms

Many MCMC algorithms are Metropolis-Hastings (MH) algorithms [13, 18]. Since we cannot sample directly from the distribution $P(\cdot)$, MH algorithms consider a simpler distribution $Q(\cdot \mid \cdot)$, called the *proposal distribution* to generate the next state of a MCMC chain. $Q(Y \mid times;^{(t)})$ generates the candidate state Y from the current state $X^{(t)}$, and the new state Y is accepted with probability:

$$\alpha(Y \mid X^{(t)}) = \min\left(1, \frac{\hat{P}(Y) \cdot Q(X^{(t)} \mid Y)}{\hat{P}(X^{(t)}) \cdot Q(Y \mid X^{(t)})}\right)$$

If the candidate state is accepted, the next state becomes $X^{(t+1)} = Y$. Otherwise, $X^{(t+1)} = X^{(t)}$. For finite search spaces, the transition probability $K(Y \mid X^{(t)})$ for arriving in Y when the current state is $X^{(t)}$, where $X^{(t)} \neq Y$, is

$$K(Y \mid X^{(t)}) = Q(Y \mid X^{(t)}) \cdot \alpha(Y \mid X^{(t)})$$

The rejection probability is,

$$K(X^t \mid X^t) = 1 - \sum_{Y', Y' \neq X^{(t)}} Q(Y' \mid X^{(t)}) \cdot \alpha(Y' \mid X^{(t)})$$

An MH algorithm is aperiodic, since the chain can remain in the same state with a probability greater than 0, and by construction it satisfies the detailed balance condition,

$$\hat{P}(X^{(t)}) \cdot K(Y \mid X^{(t)}) = \hat{P}(Y) \cdot K(X^{(t)} \mid Y)$$

If, in addition, the chain is irreducible, then it converges to the stationary distribution $P(\cdot)$. The rate of convergence depends on the relationship between the proposal distribution and the target distribution: the closer the proposal distribution is to the stationary distribution, the faster the chain converges. A popular Metropolis-Hastings algorithm is the *Metropolis algorithm* where the proposal distribution is *symmetrical* $Q(Y \mid X(t)) = Q(X(t) \mid Y)$ and the acceptance rule becomes

$$\alpha(Y \mid X^{(t)}) = \min\left(1, \frac{\hat{P}(Y)}{\hat{P}(X^{(t)})}\right)$$

2.2 Mutation

A popular and often used set of irreducible proposal distributions for MH algorithms can be described by a *mutation operator*. We generically denote the proposal distributions resulting from mutation operators with Q_m. We consider a state in the discrete space as a string of ℓ characters, $\mathbf{X} = (X_1, X_2, \ldots, X_\ell)$. The h-th position in $\mathbf{X}$ is called the *locus* of X_h, where $1 \leq h \leq \ell$, and the value of X in the locus h is called an allele. Each position (or *locus*) h in an individual $\mathbf{X}$ is instantiated with an *allele* $X_h \in E(X_\cdot)$, where $E(X_\cdot)$ is the multi-set of all possible values of $X_\cdot$.

The *uniform mutation operator* randomly changes every value of each variable of the current state with a non-zero probability, called the mutation rate [8, 16, 17, 23]. The bigger the uniform mutation rate, the bigger the jump in the search space of the child state from the parent state. Q_m denotes the *uniform mutation proposal distribution*. When the context is not ambiguous, we simply refer to it as

mutation. The uniform mutation operator defines an irreducible, symmetric and stationary proposal distribution [8].

In the sequel, the *uniform mutation transition matrix*, K_m, proposes candidate individuals with Q_m and accepts them with the MH acceptance rule. The uniform mutation transition matrix, K_m, defines an irreducible MH algorithm which converges to its stationary distribution [8].

2.3 Multiple independent chains (MICs)

When we talk about the performance of an MCMC, we refer to how well an MCMC is mixing or how "fast" it converges to the target distribution. We say that an MCMC is mixing "well" if it rapidly traverses the search space and, at the same time, accurately samples the target distribution. Note that the mixing concept in *MCMC* is not related to the mixing of building blocks in the EC literature.

In an attempt to improve the mixing behavior of MCMCs one could make use of multiple chains that run independently (MICs). The chains are started at different initial states and their output is observed at the same time. It is hoped that this way a more reliable sampling of the target distribution $P(\cdot)$ is obtained. It is important to note that no information exchange between the chains takes place.

Recommendations in the literature are conflicting regarding the efficiency of multiple independent chains. Yet there are at least theoretical advantages of multiple independent chains MCMC for establishing its convergence to $P(\cdot)$ [12]. Let's consider a large dimensional distribution where an MCMC takes a long time to find a relevant region of the search space and to escape from it to search for other relevant regions. Then, the time necessary for a long MCMC can be larger than just starting multiple MCMCs spread over the search space sampling in different regions. However, MIC converges only after all the component MCMC chains have converged.

Since the chains do not interact, MIC is at the population level an MCMC with transition probabilities equal to the product of component chains transition probabilities, or

$$K(\mathbf{X}^{(t+1)} \mid \mathbf{X}^{(t)}) = \prod_{i=1}^{N} K(\mathbf{x}_i^{(t+1)} \mid \mathbf{x}_i^{(t)})$$

where $\mathbf{X}^{(t)} = (\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_N^{(t)})$. If the MCMCs have detailed balance, are irreducible and aperiodic, then MIC inherits these properties and it converges, at the population level, to the product of their target distributions, $P_1(\cdot) \times \cdots \times P_N(\cdot)$, where $P_i(\cdot)$ is the target distribution of the i-th chain.

3 EMCMC framework

EMCMCs use a population of chains that allow interactions between the individuals under the assumption that individuals in the current population exchange information that helps the EMCMC to sample the desired distribution. Note that, in EMCMCs, the population is a multi-set of individual states rather than a collection of MCMCs: the current individual states depend on several states from the previous population. Now the sample at time t is the population $\mathbf{X}^{(t)} = (\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_N^{(t)})$ of N states (or individuals) $\mathbf{x}_\cdot^{(t)}$.

Definition 1 An *evolutionary Markov chain Monte Carlo* (EMCMC) algorithm is a population MCMC that exchanges information between individual states such that, at the population level, the EMCMC is an MCMC.

Similarly to an MCMC, the main goal of an EMCMC is to sample from a given distribution, $P(\cdot)$. Ideally, an MCMC algorithm generates individuals directly from the target distribution. Unfortunately, we do not know where the most likely—or equivalently, the most fit—individual states can be found in the search space. Furthermore, MCMCs can poorly "mix" when individual states are disproportionately proposed with their probability. A standard MCMC, for example, generates individuals with some mutation proposal distribution (e.g., the uniform mutation proposal distribution Q_m) that does not have any knowledge of the sampled distribution. A method to speed up the mixing is to propose individuals using proposal distributions that are "close" to the target distribution. For that, we can use recombination operators that exploit the common structure of the parents. Sampling from a distribution implies that the more fit individuals are more often generated than less fit ones. As a consequence, the commonalities of more likely individuals are used by recombination to create other more likely individuals. Intuitively, such a proposal distribution approximates better the target distribution than a proposal distribution that does not make any assumption about the generated individuals, like uniform mutation. In this perspective, the recombination operators adapt the proposal probabilities to generate an individual from the current population. Note that, the allowed types of proposal distribution are the ones that preserve the Markov chain property at the population level: we can only use the information in the current population for generating new individuals.

3.1 Recombination operators in EMCMCs

We call EMCMCs that use recombination to exchange information between individuals *recombinative EMCMCs*.

Definition 2 A recombination operator used as proposal distribution of an EMCMC generates one or more children from two or more parents using some function that is independent of the EMCMCs' sampled distribution. Each

generation, the population is uniform randomly grouped in disjunct families of few (i.e., two, three) individuals such that each individual belongs to exactly one family. All the chains from an EMCMC eventually interact in population recombinations. We call *recombination proposal distribution*, Q_r, the distribution defined by the recombination probabilities at the population level.

It is important to note that at the individual family level, the proposal probabilities of recombination are not stationary since they depend on the family members with which they are grouped. At population level, however, the recombination proposal distribution generating the next population from the current one is stationary.

We only consider recombination operators that are respectful—this is, the common substructures of the parents are inherited by the offspring [20]. With respectful recombination the common parts of strings are protected against disruption.

An important aspect of any recombination operator is to establish whether it is symmetrical or not: for non symmetrical recombinations, we have to compute the proposal probabilities, whereas for symmetrical operators we can simply use the Metropolis algorithm. In Sect. 4.1 we design and investigate several recombination operators that generate symmetrical proposal distributions and in Sect. 4.2 we give examples of recombination operators that generate non-symmetrical distributions. We focus on the most popular type of recombination operators in GAs that swap alleles between two or more parents with some probability to generate one or more children. Since respectful recombination by definition is reducible [8], in Sect. 4.3 we combine recombination with mutation to obtain irreducible proposal distributions following the simple mathematical rules of mixtures and cycles [8].

3.2 The MH acceptance rules

The recombination operators usually have no information about how fit the individuals in the current and proposed population are. Then, like for the standard MCMCs, we need acceptance rules to sample from the target distribution. Detailed balance is a sufficient, but not a necessary condition, for an irreducible aperiodic EMCMC to converge to a desired target distribution $P(\cdot)$. By definition, MH algorithms are aperiodic and have detailed balance. Most EMCMCs are irreducible MH algorithms—by use of mutation—and apply recombination in the proposal distribution.

In Sect. 5.1 we propose an EMCMC where individuals are generated with recombinative proposal distributions and the parents and children are competing in a Metropolis-Hasting acceptance rule. Such an EMCMC has detailed balance if and only if the individuals that interact through recombination also interact in the acceptance rule. We further call these acceptance rules where two or more chains interact the *coupled acceptance rule*. We prove that such an algorithm is ergodic—that is irreducible and aperiodic—with the stationary distribution $P_1(\cdot)\times\cdots\times P_N(\cdot)$, where $P_i(\cdot)$ is the target distribution of the i-th chain. However, such a coupled acceptance rule has a negative effect on the performance of an EMCMC. If some children are fit individuals but the others are not, this acceptance rule can reject "good" individuals whereas the standard MH acceptance rule will always accept them.

We investigate the convergence properties of recombinative EMCMCs using variations of the Metropolis-Hasting acceptance rule. In Sect. 5.2 we prove that the recombinative population-based MCMCs that accept/reject each candidate state using the standard Metropolis acceptance rule does not have detailed balance. Its advantage is that the probability of accepting at least one individual of this EMCMC is larger than the acceptance probability of an EMCMC using the coupled acceptance rule. In Sect. 5.3 an example of an MH acceptance rule derived from the elitist replacements selection strategy [25] is designed. The sampled distribution is even more skewed towards probable states and the acceptance probability of one individual is even larger. In Sect. 5.4 we propose and analyze a methodology, we call it *nested EMCMC*. It corrects the sampled distributions of skewed EMCMCs by accepting/rejecting all the individuals generated with the EMCMCs with the coupled acceptance rule. This nested EMCMC has detailed balance even though the initial EMCMC does not.

4 Recombinative proposal distributions for EMCMCs

In this section we propose and analyze various recombinative proposal distributions and their properties for EMCMCs that sample from the desired target distribution.

4.1 Symmetrical recombinations

In EMCMCs, the symmetry is obtained by preserving the distance between the parents and their children. For example, the distance between N children is equal with the distance between the N parents that generate the children, or the distance between a parent and its child is constant as compared with the distance between two other individuals in the population.

4.1.1 N parents generate N children

When the distance, i.e. Hamming distance, between the generated children is the same as the distance between their parents, the recombination operator is symmetrical.

Proposition 1 *Consider N parents uniform randomly chosen without replacement from the current population,* $\left\{\mathbf{x}_i^{(t)},\ldots,\mathbf{x}_{i+N-1}^{(t)}\right\}$, *and an associated distance metric* $\Delta : E^2 \rightarrow IR$ *with*

$$\Delta(\mathbf{x}_i^{(t)},\ldots,\mathbf{x}_{i+N-1}^{(t)}) = \sum_{j,k|j\neq k} \Delta(\mathbf{x}_j^{(t)},\mathbf{x}_k^{(t)})$$

where $\Delta(\mathbf{x}_j^{(t)},\mathbf{x}_k^{(t)}) = \Delta(\mathbf{x}_k^{(t)},\mathbf{x}_j^{(t)})$. *Let the recombination operator where N candidate individuals,* $\{\mathbf{y}_i,\ldots,\mathbf{y}_{i+N-1}\}$, *are generated by swapping alleles of parents such that the corresponding proposal probability* $S_r(\mathbf{x}_i^{(t)},\ldots,\mathbf{x}_{i+N-1}^{(t)} \mid \mathbf{y}_i,\ldots,\mathbf{y}_{i+N-1})$ *is a function of the distance between parents such that the distance between parents is equal with the distance between their children*

$$\Delta(\mathbf{x}_i^{(t)},\ldots,\mathbf{x}_{i+N-1}^{(t)}) = \Delta(\mathbf{y}_i,\ldots,\mathbf{y}_{i+N-1})$$

then S_r is symmetrical.

Proof The probability to generate children from their parents is S_r is a distance function f: $E^2 \rightarrow \mathbb{R}$ between parents

$$\begin{aligned} &S_r(\mathbf{x}_i^{(t)},\ldots,\mathbf{x}_{i+N-1}^{(t)} \mid \mathbf{y}_i,\ldots,\mathbf{y}_{i+N-1}) \\ &\quad = f(\Delta(\mathbf{x}_i^{(t)},\ldots,\mathbf{x}_{i+N-1}^{(t)})) = f(\Delta(\mathbf{y}_i,\ldots,\mathbf{y}_{i+N-1})) \\ &\quad = S_r(\mathbf{y}_i,\ldots,\mathbf{y}_{i+N-1} \mid \mathbf{x}_i^{(t)},\ldots,\mathbf{x}_{i+N-1}^{(t)}) \end{aligned}$$

Thus, this recombination is symmetrical. □

Note that if the number of children is different from N, in general, the symmetry condition does not hold. We discuss such examples in the next section.

The swapping recombinations, often used in EMCMCs and the standard GAs, are particular cases of the above proposition where the distance between individuals are kept constant by swapping alleles.

Proposition 2 *Recombination proposal distributions which swap parts of individuals in between chains using a uniform distribution are symmetrical, respectful and stationary.*

Proof Since there are equal probabilities to swap alleles (parts) in between parents and in between children, this recombination is symmetrical and the distance between them remains equal. If the parents have the same allele on a locus, so do the children since the swapping does not change the values of alleles. □

We have recombinations which exchange non-common alleles, e.g., uniform crossover, or parts of individuals, e.g., 1 and 2 point crossover [16, 17]. These recombinations are often used only with two parents.

In binary spaces, an example of swapping recombination is *parameterized uniform crossover*, Q_{unif}, which generates two candidate individuals by swapping alleles between two parents with a uniform probability, p_x. Thus, it is impossible to generate children that have other common alleles than their parents. Where the two parents differ, an allele is swapped with the probability p_x and is not swapped with the probability $1 - p_x$. It is interesting to observe that the time complexity to generate two children from two parents with Q_{unif}, like for uniform mutation, is linear with the dimensionality, $\mathcal{O}(\ell)$.

For $p_x = 0.5$, the operator is called uniform crossover and is used with all codings: for strings of bits [16] and for strings of real numbers [12].

4.1.2 Three parents generate one child

In the following, we introduce a general condition to design symmetrical recombinations using three parents which generate one child.

Proposition 3 *Consider three parents uniform randomly chosen without replacement from the current population,* $\left\{\mathbf{x}_i^{(t)},\mathbf{x}_{i+1}^{(t)},\mathbf{x}_{i+2}^{(t)}\right\}$. *The recombination operator where a candidate individual,* $\mathbf{y}_i$, *is generated from the three parents such that the total distance between parents is equal with the total distance between the candidate individual and* $\left\{\mathbf{x}_{i+1}^{(t)},\mathbf{x}_{i+2}^{(t)}\right\}$,

$$\begin{aligned} &\Delta(\mathbf{x}_i^{(t)},\mathbf{x}_{i+1}^{(t)}) + \Delta(\mathbf{x}_i^{(t)},\mathbf{x}_{i+2}^{(t)}) + \Delta(\mathbf{x}_{i+1}^{(t)},\mathbf{x}_{i+2}^{(t)}) \\ &\quad = \Delta(\mathbf{y}_i,\mathbf{x}_{i+1}^{(t)}) + \Delta(\mathbf{y}_i,\mathbf{x}_{i+2}^{(t)}) + \Delta(\mathbf{x}_{i+1}^{(t)},\mathbf{x}_{i+2}^{(t)}) \end{aligned}$$

is symmetrical, where $\Delta : E^2 \rightarrow IN$ *is a distance metric.*

Proof The parent $\mathbf{x}_i^{(t)}$ and the child $\mathbf{y}_i$ are interchangeable; they have the same total distance with the other two parents. Thus, this recombination is symmetrical. □

As an example in the binary space, we propose the *total difference crossover*, Q_{dif}. This type of recombination is imported from real coded EAs [22] and EMCMCs [24]. The new individual, $\mathbf{y}_i$ has the same alleles like $\mathbf{x}_i^{(t)}$ on the positions where the two other parents coincide. On the other positions, we flip the alleles of $\mathbf{x}_i^{(t)}$ with the probability p_x.

Corollary 1 Q_{dif} *is symmetric, respectful and stationary. The time complexity of* Q_{dif}, *like for* Q_{unif}, *is linear with the dimensionality,* $\mathcal{O}(\ell)$.

The xor crossover [23] is a special case of Q_{dif} where the probability to flip a bit is 1 for $\mathbf{x}_i^{(t)}$'s bits where $\mathbf{x}_{i+1}^{(t)}$ and $\mathbf{x}_{i+2}^{(t)}$ disagree.

The main difference between the two symmetrical types of recombination is that one preserves the sum of distances between the three parents when generating a child and the

other preserves the distance between two parents when generating two children.

4.1.3 Family versus population recombination operators

Given the number of chains that interact, we distinguish between *family* and *population* recombinations. Recombining few chains (e.g., two or three chains) is an example of the first approach, while in the latter all chains from the population exchange information. The above recombination proposal distributions are all family recombinations.

We assume that, for family recombination, each generation, the population is uniform randomly grouped in disjunct families such that each individual belongs to exactly one family. All the chains from an EMCMC, eventually, interact in population recombinations. We call *recombination proposal distribution* the distribution defined by the recombination probabilities at the population level. We denote it with Q_r. In the case of an individual at the family level, the proposal probabilities of recombination are not stationary since they depend on the family members with which they are grouped. At population level, the probability to generate with recombination one population from another one is stationary.

For the above family recombinations (e.g., Q_{unif} and Q_{dif}), the time complexity at the population level is linear with the number of individuals in the population: each generation, each individual is randomly paired in exactly one family. The complexity of these recombination proposal distributions at the population level therefore is $\mathcal{O}(\ell \cdot N)$. Note that, at the population level, the complexity of the mutation proposal distribution depends linearly on the number of individuals in the population $\mathcal{O}(\ell \cdot N)$.

4.2 Non-symmetrical proposal distributions

We investigate two non-symmetric recombinations where the alleles are exchanged between parents but, this time, the distance between parents and children is not preserved

4.2.1 The masked recombination

This recombination swaps the alleles between two parents like the parameterized recombination but it generates one child instead of two. Then, the distance between parents, in general, is not same with the distance between the child and one of the parents. Thus, the recombination is not symmetrical. We call this recombination the *masked recombination*, Q_{mask}.

A child $\mathbf{y}_i$ is generated from a parent, $\mathbf{x}_i^{(t)}$, and a mask, $\mathbf{x}_{i+1}^{(t)}$. The common alleles of $\mathbf{x}_i^{(t)}$ and $\mathbf{x}_{i+1}^{(t)}$ are passed to $\mathbf{y}_i$, but the non-common alleles are flipped in $\mathbf{x}_i^{(t)}$ with the rate p_x. Note that this crossover and the parameterized uniform crossover have the same probabilities to generate one child. But, Q_{unif} is symmetrical and Q_{mask} is not symmetrical, because Q_{mask} generates only one child. Consequently, we have to compute the probabilities to generate a candidate individual with Q_{mask} in the acceptance rule of the MH algorithm. Q_{mask} also resembles Q_{dif} where two parents are identical. However, by replacing the identical parents with the child in the candidate generation, the symmetry condition does not hold.

Proposition 4 *Q_{mask} is reducible and stationary. Consider that from a parent $\mathbf{x}_i^{(t)}$ and a mask $\mathbf{x}_{i+1}^{(t)}$ we generate a child, $\mathbf{y}_i$ with Q_{mask}. Then, Q_{mask} is non-symmetrical. The time complexity to generate a child with Q_{mask} is linear with the string size $\ell, \mathcal{O}(\ell)$.*

Proof Let's consider that $\mathbf{x}_i^{(t)} \neq y_i$ because bits are flipped on some positions. In those positions, the mask $\mathbf{x}_{i+1}^{(t)}$ and the child $\mathbf{y}_i$ have the same values, whereas $\mathbf{x}_i^{(t)}$ and $\mathbf{x}_{i+1}^{(t)}$ do not. Then, it is impossible to generate $\mathbf{x}_i^{(t)}$ from $\mathbf{y}_i$ and $\mathbf{x}_{i+1}^{(t)}$. The rest of the properties follow directly. □

4.2.2 Recombination using probabilistic models

This recombination builds a probabilistic model of the parents to generate the children. It is analogous with the operator that generates individuals for the estimation distribution (EDA) algorithms applied in Evolutionary Computation for solving optimization problems [19].

We propose the *tree frequencies probabilistic recombination*, Q_{tree}, closely related with the probabilistic model of Baluja [2]. Unlike the previous recombination operators where an allele is generated only given the alleles on the same position, Q_{tree} considers the dependencies between two positions in the population using the Chow and Liu [4] algorithm.

In the following, we describe the algorithm we use to generate individuals with Q_{tree}. This algorithm constructs from the population of current individuals a tree with maximum entropy using a mutual information function. The entropy describes the level of uncertainty in a statistical variable. Here, the frequencies of the alleles in a position define a statistical variable for that position. Mutual information captures the extent to which two statistical variables are dependent. This algorithm keeps adding dependencies between variables based upon their mutual information under the constraint of building a tree (e.g., there are no cyclic paths between variables). The higher the mutual information is, the sooner the algorithm tries to add the dependency in the tree.

A root for this tree is chosen at random from the set of positions. The allele for the root position is chosen based

on its frequencies in the current population. We iteratively generate the other alleles based on their dependency with an allele—called parent—which was already instantiated in the tree. If h is the root of the tree, then the allele $\mathbf{y}_{ih}$ is generated using the distribution $N(\mathbf{y}_{ih})/N$, where $N(\mathbf{y}_{ih})$ is the number of alleles $\mathbf{y}_{ih}$ in the current population. Otherwise, if h has the parent h_1 in the tree, then the allele $\mathbf{y}_{ih}$ is generated with the probability $\frac{N(\mathbf{y}_{ih},\mathbf{y}_{ih_1})}{N(\mathbf{y}_{ih_1})}$ where $\mathbf{y}_{ih_1}$ is the allele already generated in position h_1, and $N(\mathbf{y}_{ih},\mathbf{y}_{ih_1})$ is the number of individuals in the current population that have allele $\mathbf{y}_{ih}$ on position h and allele $\mathbf{y}_{ih_1}$ in position h_1.

We observe that Q_{tree} is the most expensive recombinative proposal distribution we have investigated for EMCMC. Unlike the other discrete space recombinations, Q_{tree} exploits some relationships between dimensions: it computes the dependencies between two positions in order to construct the tree of maximum entropy and to assign a value to an allele. Then, the generation of an allele on a position also depends on the alleles on another position.

Proposition 5 *Q_{tree} is respectful, non-symmetrical, stationary and biases the exploration according to the non-linear correlations between dimensions. The computational complexity to generate a child with Q_{tree} is $\mathcal{O}(\ell^2 \cdot N)$, where l is the dimensionality and N the size of the population.*

Proof When an individual is generated with Q_{tree} and replaces a parent, some allele frequencies can increase at the cost of the others. The computational complexity of this operator is given by building the maximum log-likelihood tree. Chow and Liu [4] show that this is $\mathcal{O}(\ell^2 \cdot N)$. □

Q_{tree} is a generalization of Laskey and Myers [15]'s recombination proposal distribution; when generating an allele, they consider only the frequencies of the alleles on the same position and not also on the other positions as Q_{tree} does. Therefore, their recombination, unlike Q_{tree}, does not exploit the relationships between dimensions.

4.3 Irreducible recombinative proposal distributions

Since respectful recombination by definition is reducible, in the following, we study how to combine it with mutation to obtain irreducible proposal distributions. We combine the proposal distributions following the same simple mathematical rules as for transition distributions. We study the properties (like symmetry and irreducibility) of the resulting proposal distributions. We show some examples where the properties of the component proposal distributions are inherited by the complex proposal distribution. However, in general, we have to check the properties for each distribution.

We combine mutation and recombination in mixtures and cycles.

Definition 3 A *mixture of proposal distributions* is a probabilistic sum of proposal distributions where each step one distribution is selected according to some constant positive probability. A *cycle of proposal distributions* is the product of proposal distributions where in each step one distribution is used in turn in a specific order.

4.3.1 Mixtures

Proposition 6 *In a mixture of proposal distributions, if one distribution is irreducible, then the mixture is irreducible. A mixture is symmetrical if the component distributions are symmetrical. A mixture is stationary if all component distributions are stationary.*

Proof If one distribution is irreducible, then there exists a non-zero probability to generate any population from any other population. The rest of the properties follows directly. □

For example, the following mixture

$$Q_{m+r} = (1 - p_r) \cdot Q_m + p_r \cdot Q_r$$

is irreducible when $p_r < 1$, and symmetrical when the recombination is symmetrical. Note that the above operator is equivalent to recombination, $Q_{m+r} = Q_r$, for $p_r = 1$; then, like recombination, Q_{m+r} is reducible. Note that the computational cost of a mixture of proposal distributions is driven by the most expensive component proposal distribution. Furthermore, a mixture exploits some relationships between dimensions if a component proposal distribution does.

4.3.2 Cycles

Unlike for mixtures, for cycles, there are no rules for irreducibility or symmetry. They have to be checked for each cycle. Cycles of proposal distributions are common for the standard GAs where one considers first mutation and then recombination, $Q_{m\times r}$, or first recombination and then mutation, $Q_{r\times m}$.

$$Q_{m\times r} = Q_r \times Q_m; \quad Q_{r\times m} = Q_m \times Q_r$$

In general, since two matrices usually do not commute, $Q_{m\times r}$ and $Q_{r\times m}$ are non-symmetrical.

Proposition 7 *$Q_{m\times r}$ and $Q_{r\times m}$, are symmetrical for any recombination that swaps alleles* [17]. *$Q_{m\times dif}$ and $Q_{dif\times m}$ are non-symmetrical. $Q_{m\times r}$ and $Q_{r\times m}$ are irreducible*

and stationary. If the recombination Q_r is symmetrical, we have

$$Q_{m\times r}(\mathbf{Y} \mid \mathbf{X}^{(t)}) = Q_{r\times m}(\mathbf{X}^{(t)} \mid \mathbf{Y})$$

To ease the reading of the paper, we give the proof for the above proposition in Appendix 1.

Parallel Recombinative Simulated Annealing (PRSA) [17] uses recombination that swaps alleles followed by mutation. Note that it is impractical to compute the probabilities of a cycle: we have to sum over all possible intermediate populations. Therefore, in general, it is impractical to use non-symmetrical cycles. In the following, we show two cycles where the above non-symmetrical recombinations are efficiently combined with uniform mutation directly on each position of an individual.

Consider a parent $\mathbf{x}_i^{(t)}$ and a mask $\mathbf{x}_{i+1}^{(t)}$ chosen at random from the population. Like for Q_{mask}, for the non-common values of the two parents, $\mathbf{x}_i^{(t)}$ is flipped with the probability p_x to generate the child $\mathbf{y}_i$. Unlike for Q_{mask}, for the common parts of these parents, $\mathbf{x}_i^{(t)}$ is flipped with the low probability $1/\ell$ to generate the child $\mathbf{y}_i$. We generate from the mask $\mathbf{x}_{i+1}^{(t)}$ a second child $\mathbf{y}_{i+1}$ with the uniform mutation with the mutation rate p_m. We denote this proposal distribution with $Q_{m\times mask}$ where

$$\begin{aligned} &Q_{m\times mask}(\mathbf{y}_i, \mathbf{y}_{i+1} \mid \mathbf{x}_i^{(t)}, \mathbf{x}_{i+1}^{(t)}) \\ &\quad = Q_{mask}(\mathbf{y}_i \mid \mathbf{x}_i^{(t)}, \mathbf{x}_{i+1}^{(t)}) \times Q_m(\mathbf{y}_{i+1} \mid \mathbf{x}_{i+1}^{(t)}) \end{aligned}$$

In the next proposition we show that $Q_{m\times mask}$, unlike Q_{mask}, can be used with an MH algorithm. Furthermore, although it is a cycle, its computational time is similar with the one of uniform mutation.

Proposition 8 *$Q_{m\times mask}$ is irreducible. $Q_{m\times mask}$ is symmetrical if $p_m = 1/2$ or $p_x = 1/\ell$. If $p_m \neq 1/2$ and $p_x \neq 1/\ell$ then $Q_{m\times mask}$ is non-symmetrical. The time complexity of $Q_{m\times mask}$ is linear with the string size $\ell, \mathcal{O}(\ell)$.*

The prove is given in Appendix 2 to ease the reading of the paper.

Similarly, we combine the tree frequencies probabilistic recombination, Q_{tree}, with the uniform mutation in a cycle to be able to use it with the MH algorithm. We first construct the maximum entropy tree. We choose at random a position, h, which we consider the root, we propose an allele $\mathbf{y}_{ih}$ with the probability $(N(\mathbf{y}_{ih}) + 1)/(N + |\Omega(\mathbf{x}_{..})|)$. Iteratively, we propose an allele $\mathbf{y}_{ih}$ with the probability

$$(N(\mathbf{y}_{ih}, \mathbf{y}_{ih_1}) + 1)/(N(\mathbf{y}_{ih_1})) + |\Omega(\mathbf{x}_{..})|)$$

where the allele on the position h_1, $\mathbf{y}_{ih\_1}$, is already instantiated. We denote this operator with $Q_{m\times tree} = (Q_{tree} + 1/N)/(1 + \Omega(\mathbf{x}_{..})/N)$. Like Q_{tree} and unlike the other proposal distributions, $Q_{m\times tree}$ exploits some relationships between different dimensions.

Proposition 9 *$Q_{m\times tree}$ is irreducible and non-symmetrical. The time complexity to generate an individual with $Q_{m\times tree}$ is $\mathcal{O}(\ell^2 \cdot N)$, where ℓ is the string size and N the population size.*

Proof The proof is immediate. □

In Table 1 we present the operators composed from mutation and/or recombination, their irreducibility, their symmetry, and their number of parents compared with the number of children.

5 MH acceptance rules for recombinative EMCMC

In this section we propose various MH acceptance rules and we discuss the properties of EMCMC algorithms resulting from the interaction between the recombinative operators and these acceptance rules.

5.1 Detailed balance: all children accepted or all rejected

We establish that the EMCMCs that generate individuals with irreducible recombinative proposal distributions and accept/reject them all has detailed balance and the target distribution for this EMCMC.

Theorem 1 *Consider the EMCMC algorithm that proposes $N \geq 2$ children, $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$ from N parents, $\mathbf{X}^{(t)} = (\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_N^{(t)})$ using a irreducible proposal distribution Q that is independent of the target distribution. All children are accepted or all children are rejected with the probability*

Table 1 Properties of several mutation/recombination operators: if they are irreducible or not, symmetrical, and how many children are generated from how many parents

Type op	Op	Irred	Symmetry	Par/child
Mut	Q_m	Irred	Symm	1/1
	Q_{unif}	Red	Symm	2/2
	Q_{dif}	Red	Symm	3/1
	Q_{mask}	Red	Non-symm	2/1
	Q_{tree}	Red	Non-symm	N/1
Mixture cycle	$Q_{m+\ unif}$	Irred	Symm	2/2
	$Q_{m\times unif}$	Irred	Symm	2/2
	$Q_{m\times mask}$	Irred	Non-symm	2/1
	$Q_{m\times tree}$	Irred	Non-symm	N/1

$$\alpha_C(\mathbf{Y} \mid \mathbf{X}^{(t)}) = \min\left(1, \frac{\hat{P}_1(\mathbf{y}_1)\cdot\ldots\cdot\hat{P}_N(\mathbf{y}_N)}{\hat{P}_1(\mathbf{x}_1^{(t)})\cdot\ldots\cdot\hat{P}_N(\mathbf{x}_N^{(t)})}\cdot\frac{Q(\mathbf{X}^{(t)} \mid \mathbf{Y})}{Q(\mathbf{Y} \mid \mathbf{X}^{(t)})}\right)$$

This EMCMC is ergodic with unique stationary distribution $P_1(\cdot)\times\cdots\times P_N(\cdot)$, where $P_i(\cdot)$ is the unique stationary marginal distribution for the ith chain, $\forall i = 1,\ldots, N$.

The prove is given in Appendix 3 to ease the reading of the paper.

Note that the EMCMC resulting from the interaction between the proposal distribution Q and the MH acceptance rule α_C is an MCMC over the N dimensional search space E^N. We denote the transition matrix for this EMCMC algorithm with K_C. The transition probability between a candidate state $\mathbf{Y}$ and the current state $\mathbf{X}^{(t)}$ is $K_C(\mathbf{Y} \mid \mathbf{X}^{(t)}) = \alpha_C(\mathbf{Y} \mid \mathbf{X}^{(t)})\cdot Q(\mathbf{Y} \mid \mathbf{X}^{(t)})$ and the rejection probability is $K_C(\mathbf{X}^{(t)} \mid \mathbf{X}^{(t)}) = 1 - \sum_{Y\neq X^{(t)}} K_C(\mathbf{Y} \mid \mathbf{X}^{(t)})$.

5.1.1 Two examples

The coupled acceptance rule. The coupled acceptance rule α_C [11, 16] considers for acceptance two chains. Two children, $\mathbf{y}_1$ and $\mathbf{y}_2$, generated from two parents, $\mathbf{x}_1^{(t)}$ and $\mathbf{x}_2^{(t)}$, are both accepted or rejected with the coupled acceptance rule $\alpha_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$.

When α_C is associated with one of the irreducible recombinative proposal distributions—for instance $Q_{m\times unif}$ and Q_{m+unif} that generates two children from two parents—according with Theorem 1, the EMCMC algorithm has detailed balance and samples from the target distribution $P_1(\cdot)\times P_2(\cdot)$.

Corollary 2 *Consider the EMCMC algorithm that proposes two children from two parents using an irreducible proposal distribution Q and accepts/rejects the children using the coupled acceptance rule α_C. We denote the corresponding transition matrix with K_C. This EMCMC converges to $P_1(\cdot)\times P_2(\cdot)$, where $P_i(\cdot)$ is the marginal distribution of the i-th chain $i = 1,2$.*

However, in practice, such an acceptance rule is not always desired, since it is not selective at individual level. For example, usually, individuals with higher and lower probabilities are proposed; with α_C the fit individuals can be rejected and the acceptance of less fit individuals depends on the family's fit individuals.

Note that the target distribution of this EMCMC is given by the product of distributions in the MH acceptance rule. By replacing this product with other mathematical functions (e.g., maximum of two values as in the next example), the corresponding EMCMC converges to a different distribution.

The order two statistics acceptance rule To sample from the *order two statistics* distribution

$$P_{2:1}(\cdot,\cdot) = \max\{P(\cdot), P(\cdot)\}$$

we create a variant of the coupled acceptance rule

$$\alpha_{2:1}(\mathbf{y}_i, \mathbf{y}_j \mid \mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)}) = \min\left\{1, \frac{\max(\hat{P}(\mathbf{y}_i), \hat{P}(\mathbf{y}_j))}{\max(\hat{P}(\mathbf{x}_i^{(t)}), \hat{P}(\mathbf{x}_j^{(t)}))}\cdot\frac{Q(\mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)} \mid \mathbf{y}_i, \mathbf{y}_j)}{Q(\mathbf{y}_i, \mathbf{y}_j \mid \mathbf{x}_i^{(t)}, \mathbf{x}_j^{(t)})}\right\}$$

where max is the maximum for the values of two individuals, and $Q(\cdot \mid \cdot)$ is any proposal distribution.

According to Lemma1, an EMCMC that proposes two candidate individuals from two parents and accepts/rejects them both with $\alpha_{2:1}$ has detailed balance. If the proposal distribution is also irreducible, this EMCMC converges to the stationary distribution $P_{2:1}(\cdot,\cdot)$.

5.1.2 Detailed balance at population level

Most EMCMCs use family recombinations where, each generation, all individuals are randomly grouped such that each individual belongs to exactly one group. If the children generated with recombination are all accepted or all rejected with an acceptance rule as suggested in Theorem 1, we obtain *family transition probabilities* with detailed balance. At individual or family level, these transitions are not MCMCs, since their proposal probabilities are not stationary—they depend on how the individuals are grouped. At population level, for all possible groupings of the current population, the transition distribution is stationary. Then, the *population transition probabilities* obtained by combining the family transitions have detailed balance and define an MCMC.

5.2 The standard MH acceptance rule in recombinative EMCMCs

In the following, we investigate the properties of EMCMCs that use irreducible recombinative proposal distributions and the standard MH acceptance rule. Such an EMCMC does not fit in the standard MH framework where the individuals that interact in the proposal distribution also interact in the acceptance rule. For this EMCMC individuals interact in the proposal distribution but children are accepted/rejected individually given only one parent.

To ease the reading, we consider that two children, $\mathbf{y}_1$ and $\mathbf{y}_2$, are generated with a symmetrical proposal distribution Q from two parents $\mathbf{x}_1^{(t)}$ and $\mathbf{x}_2^{(t)}$. Each child is accepted/rejected given one of the parents, randomly chosen without replacement, with the standard Metropolis

acceptance rules, $\alpha(\mathbf{y}_i \mid \mathbf{x}_i^{(t)}) = \min(1, \frac{\hat{P}_i(\mathbf{y}_i)}{\hat{P}_i(\mathbf{x}_i^{(t)})})$. Let's denote with $K_{1.1}$ the resulting transition matrix. The transition probability to accept both children is

$$K_{1.1}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) \cdot \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})$$

The transition probability to accept only one child (i.e., $\mathbf{y}_1$) and to reject the other child is

$$K_{1.1}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = \sum_{\mathbf{y}_2} Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]$$

The rejection probability of both candidate states is

$$K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = \sum_{\mathbf{y}_1, \mathbf{y}_2} Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})] \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]$$

To analyze the behavior of this EMCMC, we compare its transition distribution with K_C, which we showed in Theorem 1 that it converges to the target distribution. We show that even though the acceptance and rejection transition probabilities are similar, K_C samples from the target distribution and $K_{1.1}$ does not.

Proposition 10 *Consider the two transition distributions K_C and $K_{1.1}$, the coupled acceptance rule α_C, the standard Metropolis acceptance rule α as before. Let's further consider two parents $(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$ and their two children $(\mathbf{y}_1, \mathbf{y}_2)$ generated with an irreducible symmetrical proposal distribution Q.*

The probability to accept a child that it fitter than one of its parents, $\hat{P}(\mathbf{y}_1) > \hat{P}(\mathbf{x}_1^{(t)})$, is higher for $K_{1.1}$ than for K_C

$$K_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \leq K_{1.1}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) + K_{1.1}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

The probability to reject a child less fit than one of its parents, $\hat{P}(\mathbf{y}_1) < \hat{P}(\mathbf{x}_1^{(t)})$, is higher for $K_{1.1}$ than for K_C when the second child is more fit than the second parent, $\hat{P}(\mathbf{y}_2) > \hat{P}(\mathbf{x}_2^{(t)})$,

$$K_C(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \leq K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) + K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

The probability to reject a child less fit than one of its parents, $\hat{P}(\mathbf{y}_1) < \hat{P}(\mathbf{x}_1^{(t)})$, is lower for $K_{1.1}$ than for K_C when the second child is less fit than the second parent, $\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})$.

The EMCMC algorithm $K_{1.1}$ has detailed balance if and only if the probability to generate two children from two parents is equal with the probability to generate one child and one parent from the other parent and the other child

$$Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = Q(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{y}_2) \quad (1)$$

If Eq. 1 holds, the algorithm converges to the target distribution $P_1(\cdot) \cdot P_2(\cdot)$.

Again, to ease the reading, we prove this theorem in Appendix 4.

According to the above proposition, an MH algorithm that accepts/rejects with the standard MH acceptance rule some, not all, of the individuals generated with some recombinative proposal distribution does exhibit detailed balance only for some particular types of recombinations.

Equation 1 holds, for example, for uniform mutation distribution Q_m and symmetrical recombination distributions that generate one child [8, 23]. It does not hold for other symmetrical recombinations that generate two or more children, like for example, uniform recombination. With uniform recombination for any four individuals, we have

$$Q_{unif}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \neq Q_{unif}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{y}_2)$$

Unfortunately, if the detailed balance condition does not hold, there is no standard method to know the target distribution.

It is interesting to notice that the MH algorithms generated with $K_{1.1}$ have a higher probability of acceptance of at least one candidate state than the algorithms generated with K_C that accept or reject all individuals at once. As a consequence, for the same proposal distribution, the algorithm determined by $K_{1.1}$ samples faster than an algorithm that uses K_C.

5.3 The elitist coupled acceptance rule

In this section we investigate an acceptance rule inspired from the elitist replacement strategy [25] which does not have detailed balance regardless of the proposal distribution used. Furthermore, we show that the marginal distribution of the generated EMCMC is different from the target distribution being amplified for the fit individuals and diminished for the less fit individuals.

The elitist coupled acceptance rule (ECA) algorithm is a family competitive acceptance rule where the best two solutions from the family of four is kept if at least one of them is a child. Otherwise, when both children have a lower fitness than both their parents, the children can probabilistically replace the parents.

ECA can be viewed as a combination between the elitist replacement rule from regular GAs and the coupled acceptance rule α_C. When compared with the elitist replacement, ECA is more exploratory but less elitist since it still accepts with some probability less fit individuals. When compared with α_C and α acceptance rules, ECA is more elitist but less exploratory. With ECA, if a child and a

parent are the two most fit individual states from two parents and their children, they are *always* accepted whereas with α the other child will be accepted with some probability.

To establish the properties of ECA's target distribution, we compare it with K_C. The probability to escape from the basin of attraction of a peak, as we show in the next paragraphs, is rather poor when compared with the transition distribution K_C generated with the same proposal distribution and the coupled acceptance rule α_C. The transition distribution generated by accepting with ECA the individuals proposed with the irreducible proposal distribution Q is denoted with K_{ECA}. We call $\max_2$ the function returning the two most fit solutions.

We distinguish three cases.

a) *Both children are better or worse than their parents*. Then

$$\{\mathbf{y}_1, \mathbf{y}_2\} = \max_2 \left\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \mathbf{y}_1, \mathbf{y}_2\right\}$$

or

$$\left\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\right\} = \max_2 \left\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \mathbf{y}_1, \mathbf{y}_2\right\}$$

where $\mathbf{y}_1 \neq \mathbf{x}_1^{(t)}$ and $\mathbf{y}_2 \neq \mathbf{x}_2^{(t)}$. The transition probability to accept or reject both children, $\{\mathbf{y}_1, \mathbf{y}_2\}$, proposed with the proposal distribution Q is non-zero only in this case. Then

$$K_{ECA}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot \min\left\{1, \frac{\hat{P}(\mathbf{y}_1) \cdot \hat{P}(\mathbf{y}_2)}{\hat{P}(\mathbf{x}_1^{(t)}) \cdot \hat{P}(\mathbf{x}_2^{(t)})} \cdot \frac{Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)}{Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})}\right\}$$

Note that in this case, the transition probability of *ECA* is equal with the transition probability of an EMCMC using the coupled acceptance,

$$K_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = K_{ECA}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

b) *One of the children and one of the parents are most fit*. Then, for example,

$$\left\{\mathbf{y}_1, \mathbf{x}_1^{(t)}\right\} = \max_2 \left\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \mathbf{y}_1, \mathbf{y}_2\right\}$$

The transition probability to go from $\left\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\right\}$ to $\{\mathbf{y}_1, \mathbf{y}_2\}$ is 0.

$$K_{ECA}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = 0$$

Now, K_C is larger than 0 and K_{ECA} is 0.

c) *Only one parent is replaced by its child*. The proposal probability where only one parent is replaced in the next generation, $K_{ECA}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$, is amplified with the sum over all proposal probabilities that generate a state $\mathbf{y}_2$ such that

$$\left\{\mathbf{y}_1, \mathbf{x}_2^{(t)}\right\} = \max_2 \left\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\right\}$$

Then

$$K_{ECA}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = K_C(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) + \sum_{\mathbf{y}_2, \left\{\mathbf{y}_1, \mathbf{x}_2^{(t)}\right\} = \max_2\left\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\right\}} Q(\mathbf{y}_1, \mathbf{y}_2 \vee \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

For irreducible proposal distributions Q, this EMCMC algorithm is irreducible because any two individuals can be generated from any other two individuals with a non-zero probability in two iterations of the algorithm $T_{ECA^2}(\cdot \mid \cdot) > 0$. Let's assume again that a child $\mathbf{y}_1$ and one of the parents $\mathbf{x}_2^{(t)}$ have the largest probabilities. In one iteration

$$K_{ECA}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) > 0$$

and, for the second iteration, we also have $K_{ECA}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) > 0$.

Following the above observation, we prove that this EMCMC converges to a stationary distribution and also it does not have detailed balance regardless of the proposal distribution. The proof is given in Appendix 5.

Proposition 11 *Consider the EMCMC algorithm that generates candidate individuals using an irreducible proposal distribution Q* and then accepts or rejects them with the ECA acceptance rule. This EMCMC algorithm does not have detailed balance for any non-uniform distribution Q *and converges to a stationary distribution* $\prod_{i=1}^{N} R(\cdot)$.

This algorithm is climbing towards a local optima since it is very probable that a good solution remains a long time in the population to generate better solutions. Only when both children are worse than their parents this algorithm rejects the two candidate individuals with some probability. Otherwise, ECA always accepts at least one child. As a consequence, the probability to accept at least one proposed child is the largest from all the previous acceptance rules. Thus, an algorithm that uses ECA behaves more similar to a standard GA than to a sampling algorithm. As a consequence, the target distribution of ECA is biased towards high regions of $P(\cdot)$: the highest fitness states are sampled more often at the expense of the lower fitness states.

5.4 Nested transition distributions: repairing the detailed balance

In the following, we propose a method to integrate the transition distributions without detailed balance in MH

algorithms with detailed balance. To achieve this, we need to accept or reject all the individuals generated with an MH algorithm without detailed balance.

Definition 4 A *nested EMCMC algorithm* is an EMCMC algorithm where individuals are proposed using a transition distribution, and are further all accepted or all rejected by a coupled MH acceptance rule. A *nested transition distribution* is the transition distribution used as proposal distribution by the nested EMCMC algorithm.

Furthermore, the nested transition distribution that generates individuals with a recombination distribution is itself a recombinative proposal distribution: from two or more parents we propose two or more children.

Proposition 12 *The nested EMCMC algorithm has detailed balance. The nested transition distribution composed by a respectful recombination proposal distribution and an acceptance rule is by itself a respectful recombination proposal distribution.*

Proof The proof is immediate if we consider the nested transition distribution as a proposal distribution and Lemma 1. If parents have identical values at certain positions, then the individuals generated by respectful recombination have—by definition—the same values at those positions. An acceptance rule simply selects from parents and children, therefore, the accepted individuals have the same values on those positions. □

Nested transitions are, usually, non-symmetrical. Thus, we need to compute these probabilities. In Fig. 1, we graphically depict the nested EMCMC framework.

5.4.1 Examples of nested EMCMCs

Correcting $K_{1.1}$. Consider the *nested EMCMC* that uses as proposal distribution the nested transition distribution, $K_{1.1}$ where two candidate individuals are proposed from two parents with some recombinative proposal distribution, Q, and each child competes against one of the parents randomly chosen from the population with a standard MH acceptance rule. The candidate individuals proposed with $K_{1.1}$ are, at their turn, accepted with the coupled acceptance rule, α_C. The nested EMCMC's transition distribution is

$$K_{nEMCMC} = K_{1.1} \cdot \alpha_C = (Q \cdot A \cdot A) \cdot \alpha_C$$

where the coupled acceptance rule is

$$\alpha_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = \min\left\{1, \frac{\hat{P}(\mathbf{y}_1)\cdot\hat{P}(\mathbf{y}_2)}{\hat{P}(\mathbf{x}_1^{(t)})\cdot\hat{P}(\mathbf{x}_2^{(t)})} \cdot \frac{K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)}{K_{1.1}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})}\right\}$$

We observe that the nested EMCMC eliminates the influence of the proposal distribution on $K_{1.1}$'s target distribution with the coupled acceptance rule, α_C.

In the following proposition, we express K_{nEMCMC} as a function of $K_{1.1}$ and the proposal distribution Q. The proof of this proposition is in Appendix 6.

Proposition 13 *Consider that the symmetrical proposal distribution Q generates $\mathbf{y}_1$ and $\mathbf{y}_2$ from $\mathbf{x}_1^{(t)}$ and $\mathbf{x}_2^{(t)}$. If both children are different from their parents, $\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\} \neq \{\mathbf{y}_1, \mathbf{y}_2\}$, the nested transition distribution is*

$$K_{nEMCMC}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = K_{1.1}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

If only one child is different from its parent, $\mathbf{y}_1 \neq \mathbf{x}_1^{(t)}$, then

$$K_{nEMCMC}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = K_{1.1}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot \min\left\{1, \frac{q + \sum_{\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})} Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]}{q + \sum_{\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})} Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]}\right\}$$

where

$$q = Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) = Q(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

Otherwise, if both children are rejected,

$$K_{nEMCMC}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = 1 - \sum_{\mathbf{y}_1 \neq \mathbf{x}_1^{(t)}, \mathbf{y}_2 \neq \mathbf{x}_2^{(t)}} K_{nEMCMC}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

From the above proposition, we note that the difference between the two transition distributions, K_{nEMCMC}, which

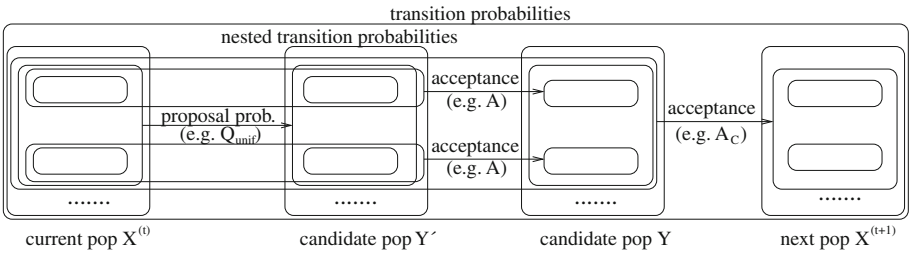

Fig. 1 Nested EMCMC framework: a candidate population $\mathbf{Y}'$ is proposed with some proposal distribution Q from the current population $\mathbf{X}^t$ and some children are accepted with some MH acceptance rule. These accepted children and the parents that are not replaced form the candidate population $\mathbf{Y}$ compete against $\mathbf{X}^t$ such that the resulting EMCMC has detailed balance

samples from the target distribution, and $K_{1.1}$, which does not sample from the target distribution, is given by the correction term from Eq. 2. In other words, $K_{1.1}$ has to be multipled with the above correction term to sample from the target distribution. If the irreducible proposal distribution Q has the property that

$$Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) = Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

for any $\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \mathbf{y}_1$ and $\mathbf{y}_2$, the correction term is 1. In this case, according with both Proposition 13 and 10, $K_{1.1}$ has detailed balance and converges to the target distribution. Note that the probability of acceptance of at least one candidate individual with K_{nEMCMC} is smaller than with $K_{1.1}$ and larger than with K_C. Furthermore, $K_{1.1}$, as proposal distribution, is not symmetrical and to use it in K_{nEMCMC}, we have to compute the impractical correction term.

$\mathbf{K_C}$ as nested proposal distribution. The coupled transition distribution K_C is invariant for the nested method. The proof of this proposition is in Appendix 7.

Proposition 14 *Consider that the symmetrical proposal distribution Q generates $\mathbf{y}_1$ and $\mathbf{y}_2$ from $\mathbf{x}_1^{(t)}$ and $\mathbf{x}_2^{(t)}$. The nested transition distribution is*

$$K_{nEMCMC}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = K_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

The coupled transition distribution K_C does not need a correction term to converge to the target distribution.

6 Three experimental tests

We compare the performance of recombinative and non-recombinative population-based (E)MCMCs on two functions: a toy problem, the *hyper-geometrical* distribution, which we use to analytically compare the performance of the algorithms and a larger problem the *binary quadratic programming problem* (BQP). We show that recombinative EMCMCs can outperform the standard MCMCs. Furthermore, we show that the algorithms that use the coupled acceptance rule α_C are less efficient than the algorithms that use the standard acceptance rule α.

We compare the performance of five MCMCs: a single chain MCMC, two non-recombinative MCMCs with two recombinative EMCMCs. We take the size of population $N = 2$.

1. *MCMC*: one single chain MCMC that proposes new states with Q_m with the mutation rate $p_m = 1/\ell$ and accepts (rejects) them using the Metropolis acceptance rule α.
2. *MIC*: 2 independent MCMCs that propose new states with Q_m with the mutation rate $p_m = 1/\ell$ and accept (reject) them using the Metropolis acceptance rule α.
3. *mut*+α_C: a non-recombinative population-based MCMC that proposes each generation 2 new states with the same Q_m and accepts (rejects) all of them using the coupled acceptance rule α_C.
4. *rEMCMC*: generates two individuals with a cycle between Q_m and parameterized uniform recombination, Q_{unif}, with $p_r = 50\%$, and then accepts them with the standard Metropolis acceptance rule α.
5. *rEMCMC*+α_C: generates two individuals with a cycle between Q_m and Q_{unif} and then accepts them with the coupled acceptance rule α_C.

As shown in previous sections, the target distribution of the three population based EMCMCs– *MIC*, *mut*+α_C and *rEMCMC*+α_C—is $\prod_N P(\cdot)$ and the target distribution of single chain *MCMC* is $P(\cdot)$. The sampled distribution of *rEMCMC* is not the target distribution but, as the experimental results show it, it approximates quite well $P(\cdot)$ for large search spaces and a small number of samples.

6.1 Sampling from the hyper-geometrical function

To compare MH algorithms analytically, we compute the second largest eigenvalue of the transition matrices of the corresponding (E)MCMCs. Note that the second eigen value should be small to mix well.

6.1.1 The tested distribution

A *hyper-geometric distribution* (Hyper) is

$$\hat{P}(\mathbf{x}) = \begin{cases} h_2 \cdot \frac{w - \Delta(\mathbf{x}, \mathbf{x}_0)}{w} & \text{if } \Delta(\mathbf{x}, \mathbf{x}_0) < w \\ 0.01 & \text{if } \Delta(\mathbf{x}, \mathbf{x}_0) = w \\ h_1 \cdot \frac{\Delta(\mathbf{x}, \mathbf{x}_0) - w}{\ell - w} & \text{otherwise} \end{cases}$$

with ℓ the string size, w the number of bits-1 in the individuals with the lowest value 0.01, individual $\mathbf{x}_0$ with all bits equal to 0 is the second largest peak h_2, and the individual with all bits equal to 1 is the largest peak h_1. We set $\ell = 8$ and $h_1 = 1$.

6.1.2 Results

In the first experiment, see Fig. 2a and b, we vary the distance of the lowest valued states to the optimum, $w = \{1, 2, 3\}$ and we set the value of the second largest state to $h_2 = 0.75$. In this case, the local and global optimum have a close value and we vary their basin of attraction: the greater the distance from the local optimum, the smaller the basin of attraction of the global optimum. Second, we vary h_2 from 0.25 to 0.75 with a step size of 0.25 and we set $w = 3$. In this case, the optimum is isolated and its importance is decreasing with the height of the second largest peak. In Fig. 2a and c we show results for

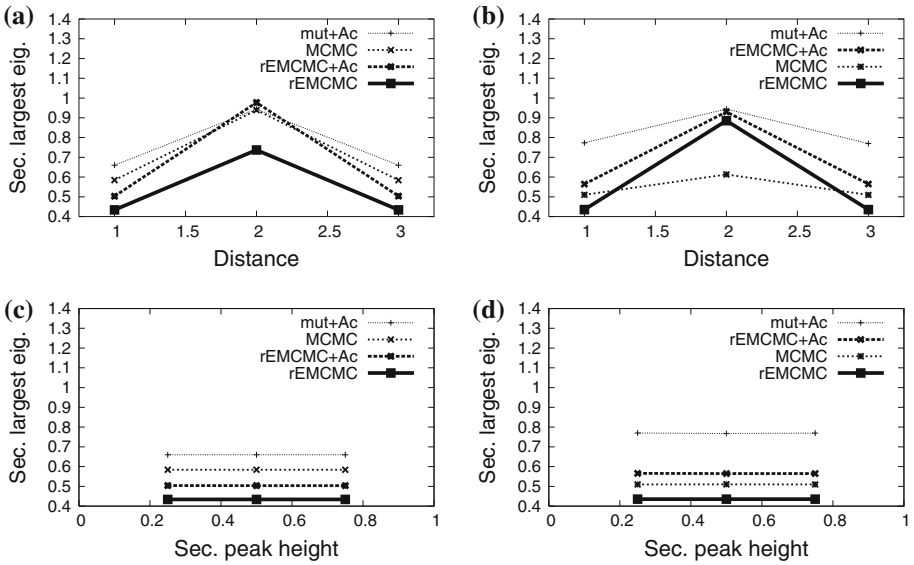

Fig. 2 Second largest eigenvalues for Hypergeometrical function on 8 bits, that is two blocks each of 4 bits, where **a**,**b** $w = \{1, 2, 3\}$ and the peak heights are set to $h_1 = 1$ and $h_2 = 0.75$, and **c**,**d** $h_2 = \{0.25, 0.5, 0.75\}$ and the distance to the highest peak is set to $w = 3$

high mutation and swapping rates, 0.5; in Fig. 2b and d we have low mutation and swapping rates 0.125. We set the low mutation rate for the cycle $Q_m \times Q_{unif}$ to 0.125. Again, we reduce the computation costs by grouping individuals with the same number of ones and zeros in one individual because these individuals have the same fitness value and therefore, the same acceptance probability. The eigenvalues for *MIC* and *MCMC* are the same because the two MCMCs have the same acceptance rule and proposal distribution. Thus, we have chosen to show results only for one of the two algorithms.

In Fig. 2a and b, for $w = 2$, the basin of attraction is equal for the two peaks. Then, we obtain the highest eigenvalues, and thus the worst performance, for all the four algorithms. Here we have the largest amount of low fit states that separate two narrow regions with fit individuals; a random sampler, see *MCMC* with mutation rate of 0.5 in Fig. 2b, is the best algorithm since it covers a large area with low equal values in short time.

For the other values of w, the basin of attraction of one of the peaks is wider than the basin of attraction of the other peak; the narrower one region is, the harder to find and sample it. For $w = \{1,3\}$ we have the lowest eigenvalues and, furthermore, the highest difference between the algorithms. The non-recombinative (E)MCMCs do well because the narrow peak is reduced now to one point. The recombinative EMCMCs do better than the non-recombinative (E)MCMCs with the same acceptance rule because recombination generates with higher probability more fit individuals by combining the two building blocks of this function, In Fig. 2c and d, we observe that the performance of all the (E)MCMC algorithms varies very little with the height of the second largest peak h_2. Thus, these eigenvalues are (approximatively) the same with the eigenvalues for $w = \{1,3\}$ from Fig. 2a and b.

To conclude this example, we observe that due to the structure of the problem recombinative EMCMCs have provably a better performance than the non-recombinative EMCMCs. The performance of MCMCs are diminished by the coupled acceptance rule α_C; *MCMC* is sampling more efficient than *mut*+α_C and *rEMCMC* is better than *rEMCMC*+α_C. The mutation rate greatly influences the performance of non-recombinative MCMCs; a high mutation rate decreases the performance of the algorithm. The swapping probability influences less the efficiency of the recombinative EMCMCs. *rEMCMC* and *rEMCMC*+α_C perform best for high swapping probabilities, whereas *MCMC* and *mut*+α_C perform best for low mutation rates.

6.2 Sampling from BQP

In the following, we have performed experiments with the *binary quadratic programming problem* (BQP) to show, on a more elaborated example, that recombination is useful for sampling. The fitness function of an individual x is $f(x) = \sum_{j=1}^{\ell} \sum_{k=1}^{\ell} F[j][k] \cdot x[j] \cdot x[k]$, where $F[j][k]$ is the element on the j-th row and on the k-th column of a matrix F of integers, both positive and negative and $\mathbf{x}$ a binary string (e.g., $\mathbf{x}_\cdot$ is 0 or 1). Then, F's size is $\ell \cdot \ell$.

The interaction between two or more positions of the BQP problem depends on the matrix F's density, which is defined as the number of non-zero elements divided by the number of total elements in the matrix. The density is then between 0 and 1, where 0 means no interaction between positions and 1 means maximum interaction—that is every position depends on every other position. For our experiments, we generate random matrices with density 0.1.

These fitness values are positive and negative. However, the (unnormalized) probabilities of a distribution can be only greater than 0. Therefore, we add to all the fitness

values a fixed positive integer *transl*; every value that now is equal or below 0 is assigned with the value 0.01. The unnormalized probabilities are $\hat{P}(x) = f(x) - transl$, when $f(x) > transl$ and, otherwise, $\hat{P}(x) = 0.01$.

In this section, we show that recombination can improve sampling. We first discuss the experimental methodology available to measure and compare the performance of EMCMCs. Second, we show experimental results on a BQP problem on 20 bits. By expanding the target distribution, we are able to compute the distance between this distribution and the true distribution. At last, we show results on a larger search space, for $l = 100$ bits. Unlike for the previous example, we are not able to expand this distribution, and therefore we are constrained to use less precise methods to assess the performance of (E)MCMCs. For both experiments, we compare the five (E)MCMC algorithms described above: three non-recombinative (E)MCMCs—that are one long chain *MCMC*, *MIC* and *mut*+α_C—and two recombinative EMCMCs—that are *rEMCMC* and *rEMCMC*+α_C.

6.2.1 Experimental methodology

To assess the efficiency of various EMCMCs we focus on monitoring how fast an MCMC is mixing and how well the samples spread over the entire target distribution after a fixed and rather small number of generated individuals. There is no generally acknowledged methodology on measuring how "close" a set of samples generated with a real-coded MCMC is to the true target distribution. Wolpert and Lee [26] argue that a good approach is to use the Kullback-Leibler (KL) distance between an approximation of the sampled distribution and a discrete approximation of the true distribution.

To measure the speed with which an algorithm samples the search space, Roberts et al. [21] recommend to monitor the acceptance probability of an algorithm. They analytically and experimentally study the behavior of a standard MCMC using a normal distributed mutation with fixed and equal variances in all dimensions. The target distribution is a multivariate normal distribution with standard deviation of 1.0 in all dimensions and no correlations. They conclude that a very high or very low acceptance rate of the MCMC indicates slow mixing, and a good acceptance rate is between 0.2 and 0.5. A high acceptance rate and a high performance (e.g., the KL-measure close to 0) indicates a well performing algorithm that mixes fast. Analytically computing the optimal acceptance probability is only feasible for very simple target and proposal distributions and when using the Metropolis acceptance rule. Here, we restrict ourselves to experimentally monitoring the acceptance probability.

For the tested recombinative EMCMCs, we have good performance (e.g., KL distance) even for very high acceptance rates that shows that recombination can improve the mixing of MCMCs. Furthermore, we show that algorithms with similar acceptance probabilities can have rather different performance.

6.2.2 A 20 bits BQP

For the first experiment, we set the string length to $\ell = 20$ and $transl = 50$. Since the F's density is 0.1, only 40 elements of F have non-zero values. The non-zero integers are generated at random from the interval $[-100, 100]$. For the generated matrix F, we have found the maximum fitness value 146; when this is translated, the maximum unnormalized probability value is 196. We group the individuals with the same value to generate the histogram and we also store the number of individuals with the same value.

We set the population size $N = 20$. Each generation, all individuals are randomly coupled in $N/2$ pairs such that each individual belongs to exactly one pair. We have performed experiments for various mutation rates (from 0.05 to 0.5) and swapping probabilities for the uniform recombination (from 0.05 to 0.5). With each algorithm, we generate 20,000 individuals; our measurements are averaged over 50 runs. We throw away the first 10,000 generated individuals to diminish the impact of the starting points over the performance of the algorithms. This is called the *burn-in* period. Thus, in total, we sample 10,000 "useful" individuals from which we generate Table 2 and the graphs from Fig. 3.

The search space is 2^{20}. By expanding the target distribution, we are able to compare the frequencies of samples generated with the tested (E)MCMCs with their value in the true distribution. In Table 2, we compute the KL distance and acceptance ratio for the five (E)MCMCs. Mann-Whitney nonparametric two-sided test with significance level $p < 0.05$ is used to verify if KL distances of the five tested algorithms are sampled from different distributions. The algorithms have statistical significantly different output except with *mut*+α_C and *MCMC*.

The best algorithm, with the significantly lowest KL distance and the highest acceptance ratio, is the recombinative

Table 2 Efficiency of (E)MCMCs for a BQP on 20 bits: the KL distances and acceptance probabilities

Alg.	KL dist	Accept prob
Mut+α_C	$(1.47 \pm 0.36) \cdot 10^{-4}$	0.16 ± 0
MCMC	$(1.29 \pm 0.41) \cdot 10^{-4}$	0.26 ± 0.01
rEMCMC+α_C	$(0.86 \pm 0.28) \cdot 10^{-4}$	0.53 ± 0
MIC	$(0.73 \pm 0.12) \cdot 10^{-4}$	0.29 ± 0
rEMCMC	$(0.57 \pm 0.07) \cdot 10^{-4}$	0.74 ± 0.01

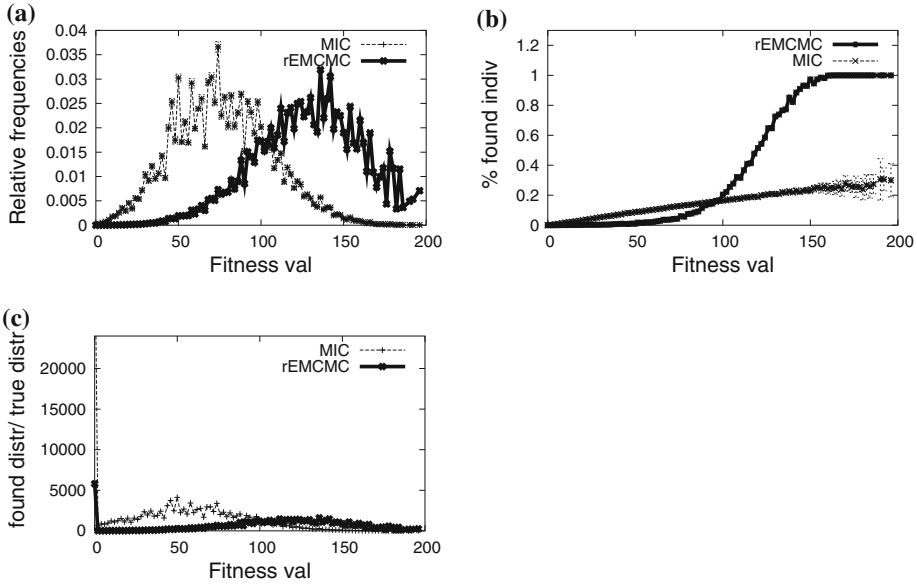

Fig. 3 **a** The frequencies and **b** the percentage of solutions found for each fitness value for *MIC* and *rEMCMC* on BQP on 20 bits; **c** how many times the sampled frequencies differ from the true distribution

EMCMC, *rEMCMC*. The only difference between *MIC*, the algorithm with second best KL distance, and *rEMCMC* is that *rEMCMC* uses recombination and *MIC* does not. Further, we observe that the two recombinative EMCMCs, that are *rEMCMC* and *rEMCMC*+α_C, have a higher acceptance probability than the three non-recombinative EMCMCs, that are *MCMC*, *MIC* and *mut*+α_C. That indicates that the recombinative proposal distribution $Q_m \times Q_{unif}$, by exploiting the commonalities of the search space, is a "better" proposal distribution than Q_m.

Furthermore, as we already observed in the analytical experiments, the coupled acceptance rule α_C has a negative influence over both recombinative and non-recombinative EMCMCs. Even though using α_C, the recombinative *rEMCMC*+α_C has the third best KL distance and the second acceptance ratio, whereas *mut*+α_C is the worst algorithm of all. We explain the good behavior of *rEMCMC*+α_C by synchronizing the individuals in the family with the uniform recombination: children that inherit the common parts of their parents have similar fitness with the parents and the algorithm accepts more individuals. In opposition, uniform mutation independently proposes two individuals in random directions; then, if one of the candidates has very low fitness, there is a big probability that both children are rejected. As a consequence, *mut*+α_C has a low acceptance rate and, thus, performance.

In accordance with the analytical results from the previous section, we observe that *MIC*, by using populations of MCMC chains has a lower KL distance than the standard *MCMC*. Note that the acceptance ratio for these two algorithms is the same, but their KL distance quite different.

In Fig. 3, we show experimental results for the two most performant (E)MCMCs presented in the previous section: *MIC* and *rEMCMC*. By using recombination, *rEMCMC* is a better sampler than *MIC* is: in frequencies, *rEMCMC*, see Fig. 3a, is closer to the true target distribution than *MIC* is. If *rEMCMC* samples with predilection in the high values of the target distribution, in opposition, *MIC* typically samples the low fit individuals. Furthermore, *rEMCMC* finds more higher probable solutions than *MIC*, see Fig. 3b. Overall, in Fig. 3c, we notice that the distribution sampled with *rEMCMC* is closer to the true distribution than *MIC* is. These results are in concordance with the ones in Table 2 from which we conclude that *rEMCMC* is the most performant algorithm for this particular problem by proposing individuals with recombination.

6.2.3 A 100 bits BQP

We now show that recombination can improve mixing on BQP with string size $\ell = 100$, for which it is impractical to generate the target distribution. In this experiment, we cannot compute the KL distance. Furthermore, we do not know the maximum value of this function or if the values are uniformly distributed in some interval.

For this experiment, we compare all the five MCMCs as before, but we show the results only for the best two algorithms *MIC* and *rEMCMC*. Note that these two algorithms are the best performant algorithms in all three experiments.

Assuming, that the unnormalized values of the distribution are in a very large range we group our samples is using the individual's number of ones to compare two (E)MCMCs algorithms. Given this grouping, we compute the frequencies, Fig. 4a, and the mean value, Fig. 4b, for each such a group.

Again, we set the density of F to 0.1; thus, approximately 1,000 elements of F are non-zero. We generate these non-zero integers with a uniform random distribution

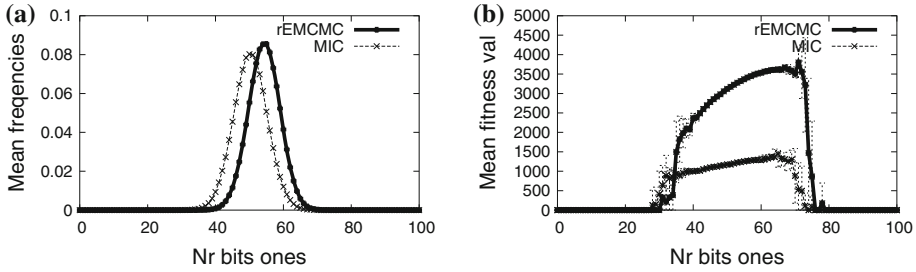

Fig. 4 **a** The frequencies and **b** the average fitness of the found solutions for each number of ones for *MIC* and *rEMCMC* on BQP on 100 bits

from the interval $[-100, 100]$. To generate a distribution with positive values, we set $transl = 1{,}250$. We set population size $N = 100$ and, each algorithm we run 50 times. With each algorithm, we generate 100,000 individuals which we throw away, and we use the next 100,000 generated individuals. Again, we vary the mutation rate and swapping rate from 0.05 and 0.5. In Fig. 4, we show results for *MIC*'s mutation rate 0.2, and *rEMCMC*'s swapping probability 0.5 and mutation rate 0.01. Then, *MIC* has an acceptance rate of 30%, whereas *rEMCMC* has an acceptance rate of 78%. We mention that we have performed experiments with various mutation and swapping probabilities but the results are not very different from the ones we currently show.

In Fig. 4a, we notice that *rEMCMC* samples slightly more individuals with a higher number of ones than *MIC* does. Except that, the distributions sampled by *MIC* and *rEMCMC* are similar and both are sampling especially from individuals with half number of zeros and ones indicating that the target distribution is symmetrically distributed around these individuals. Despite that, the mean values of the sampled individuals are remarkably larger for *rEMCMC* than for *MIC*. It seems that 100,000 individuals are not enough for *MIC*'s burn in whereas for *rEMCMC* it is. We also have performed experiments with single chain *MCMC*; we mention that the mean values are worse than of *MIC*. We explain that by the shape of this BQP: a lot of peaks with many low fit individuals. We therefore consider that *MIC* mixes slower than *rEMCMC*: $N = 100$ is not large enough to cover the number of these peaks and thus *MIC* will always have the problem to escape from these peaks to find the other useful ones. To sample the same amount of individuals, an increase in population size must be combined with a decrease in the *MCMC*'s time to run. The less time we allow an *MCMC* to run, the worse an *MCMC* samples from the search space and eventually, when population size goes to infinity, *MIC* is just a random sampler.

7 Conclusions and discussion

We discussed aspects from the Evolutionary MCMC framework, a class of population based MCMC algorithms that exchange useful information by using recombination and selection. The main issue for EMCMC algorithms is to improve the performance of the sampling process, or the convergence time to a desired distribution. Detailed balance is a straightforward and sufficient, but not necessary, condition for an irreducible and aperiodic EMCMC to converge to a given distribution.

We aim to increase the efficiency of MCMCs by the use of recombination. Recombination operators can generate "good" proposal distributions that exploit the structure of the search space such that EMCMCs using it converge faster to the target distribution. Of course, when the search space has no structure or the structure is not correctly matched with the recombination operator, the recombinative proposal distribution will offer no advantage and will most likely be as efficient as a uniform randomly generated distribution, or even worse in the worst case.

We proposed various recombinative proposal distributions on discrete spaces and we studied how to integrate them into EMCMCs with detailed balance. Since we consider only respectful recombinations, which are reducible, we have to combine recombination with mutation in order to obtain irreducible EMCMCs. We focus on discrete space recombinations and study the properties of discrete space EMCMCs resulting from the interaction of recombinative proposal distributions and MH acceptance rules. The analytical and experimental results show that EMCMCs can outperform the standard MCMC sampling algorithms by using recombination operators.

We have proposed and investigated various MH acceptance rules derived from EC's selection strategies. In order to obtain a recombinative EMCMC with detailed balance, the children proposed by the recombination operator need to be all accepted or all rejected with the coupled acceptance rule.

Both analytical investigations and experimental tests show that the recombinative EMCMC in which a child individually competes against a parent in the standard MH acceptance rule is the best sampler. In the experimental section, for very large search spaces and small number of samples, this EMCMC, *rEMCMC*, samples high regions of the search space faster than an EMCMC using the coupled acceptance rule. Thus, in short time, *rEMCMC* approximates the desired distribution better. However, there is no theoretical guarantee that *rEMCMC* converges to the target distribution. We also proposed the nested EMCMCs that individually accept or reject fitted states with a EMCMC

without detailed balance. Even though the nested EMCMCs on the proposed unbalanced EMCMC have theoretical value by indicating a correction term of the sampled distribution, its computation is impractical.

Finally, we also discussed a recombinative EMCMC without detailed balance but that can be useful for optimization purposes. It is a straightforward extension of the elitist replacement in an MH acceptance rule: two parents compete against two children and the best two from the four are selected for the next generation. This EMCMC can be used only for optimization since it is sampling mainly from the fittest regions of the sampled distribution at the expense of the less fit regions. Its disadvantage is that it can get stuck for a long time in good, but isolated, modes of the sampled distribution.

We conclude that one should be careful with adopting recombination and selection operators from EC into population-based MCMC framework. Population-based techniques that are suited for optimization can be less suitable for sampling and vice-versa. To have a positive impact on the sampling performance of interacting MCMC chains, recombination and selection techniques need to follow some design principles as outlined in this paper.

Appendix 1: Proof for Proposition 7

$Q_{m\times r}$ and $Q_{r\times m}$ are irreducible because they have non-zero probabilities to go from any population to any other population, $Q_{m\times r} > 0$ and $Q_{r\times m} > 0$.

When Q_r is symmetric, we have

$$Q_{m\times r}(Y|\mathbf{X}^{(t)}) = \sum_{\mathbf{Y}'\in E} Q_m(\mathbf{Y}'|\mathbf{X}^{(t)})\cdot Q_r(Y|\mathbf{Y}') = \sum_{\mathbf{Y}'\in E} Q_r(\mathbf{Y}'|Y)\cdot Q_m(\mathbf{X}^{(t)}|\mathbf{Y}') = Q_{r\times m}(\mathbf{X}^{(t)}|Y)$$

$Q_{r\times m}$ and $Q_{m\times r}$ are symmetrical for recombinations that swap alleles because mutation generates the alleles which differ in the two populations and recombination swaps them or vice-versa.

By means of an example, we prove that $Q_{dif\times m}$ is non-symmetrical. Consider the current population of bits $\mathbf{X}^{(t)} = \{0, 1, 0\}$ and the candidate population $Y = \{1, 1, 1\}$, the mutation rate of 1/3, and, for simplicity, the xor operator. We compute the probability to generate Y from $\mathbf{X}^{(t)}$ with uniform mutation and then with xor recombination and the inverse probability to generate $\mathbf{X}^{(t)}$ from Y.

Let's consider all possible parent choices for xor. With the xor recombination, given the distance $\Delta(0,1)$ between the first two bits, we generate 1 from the third bit of the current population 0; the intermediate population is now $\mathbf{Y}' = \{0,1,1\}$. The distance between the second and the third bit is also $\Delta(1,0)$, and thus the intermediate population is again $\mathbf{Y}' = \{1,1,0\}$. Since the distance between first and second bits of the current population is $\Delta(0,0)$, we generate 1 from 1 and the intermediate population is $\mathbf{Y}' = \{0,1,0\}$. When we mutate the intermediate populations, we have $Q_m(1,1,1 \mid 0,1,0) = (1/3)^2 \cdot 2/3$ and $Q_m(1,1,1 \mid 1,1,0) = (2/3)^2 \cdot 1/3$. Computing in a similar manner the inverse probability, for all possible intermediate populations, we have $Q_{dif\times m}(Y \mid \mathbf{X}^{(t)}) = 1/3 \cdot ((1/3)^2 \cdot 2/3 + 2 \cdot (2/3)^2 \cdot 1/3) = 10/81$.

To generate $\mathbf{X}^{(t)}$ from Y with $Q_{dif\times}m$, we mutate Y to $\mathbf{Y}' = \{0,1,1\}$ and then swap with the xor operator the last bit given the difference between the first two bits resulting in $\mathbf{X}^{(t)}$. Similarly, we mutate Y to $\mathbf{Y}' = \{1,1,0\}$ and swap the first bit of $\mathbf{Y}'$ or we mutate into $\mathbf{Y}' = Y$ and do not swap the middle bit with xor since the difference between the first and the last bit is 0. We then have $Q_{diff\times m}(\mathbf{X}^{(t)} \mid Y). = 1/3 \cdot (2 \cdot (2/3)^2 \cdot 1/3 + 1/3 \cdot (2/3)^2) = 4/81$.

We conclude that $Q_{dif\times m}$ is not symmetrical since $Q_{diff\times m}(\mathbf{X}^{(t)} \mid Y) \neq Q_{dif\times m}(Y \mid \mathbf{X}^{(t)})$. □

Appendix 2: Proof for Proposition 8

$Q_{m\times mask}$ is irreducible, since is has $Q_{m\times mask}(\cdot \mid \cdot) > 0$. If $p_x = 1/\ell$, the $Q_{m\times mask}$ is equivalent with the mutation operator, since all alleles in the parents can be flipped with the probability $1/\ell$. Then $Q_{m\times mask}$ is symmetric.

For $p_m = 1/2$, we uniformly random generate the child $\mathbf{y}_{i+1}$ from the mask $\mathbf{x}_{i+1}^{(t)}$. Then, $Q_{m\times mask}$ is symmetric since the common and uncommon parts of the parents and the children are randomly generated.

By means of an example, we show that $Q_{m\times mask}$ is non-symmetrical for other values of p_m and p_x. Consider $\mathbf{x}_i^{(t)} = \mathbf{x}_{i+1}^{(t)} = 0$ and $\{\mathbf{y}_i, \mathbf{y}_{i+1}\} = \{1,0\}$. When $\mathbf{y}_i = 1$ and $\mathbf{y}_{i+1} = 0$, the probability to generate $\mathbf{y}_i$ is $1/\ell$, and the probability to generate $\mathbf{y}_{i+1}$ is $1 - p_m$. The inverse probability is $Q_{m\times mask}(\mathbf{x}_i^{(t)}, \mathbf{x}_{(i+1)}^{(t)} \mid \mathbf{y}_i, \mathbf{y}_{i+1}) = (1 - p_x) \cdot (1 - p_m)$. When $\mathbf{y}_i = 0$ and $\mathbf{y}_{i+1} = 1$, the probability to generate $\mathbf{y}_i$ is $1 - 1/\ell$ and the probability to generate $\mathbf{y}_{i+1}$ is p_m. The reverse probability is now $p_x \cdot p_m$. Then $Q_{m\times mask}(\mathbf{y}_i, \mathbf{y}_{i+1} \mid \mathbf{x}_i^{(t)}, \mathbf{x}_{i+1}^{(t)}) = (1 - p_m)/\ell + (1 - 1/\ell) \cdot p_m$ and $Q_{m\times mask}(x(t)_i, x^(t)_(i+1) \mid y_i, y_{i+1}) = (1 - p_x) \cdot (1 - p_m) + p_x \cdot p_m$. We now have that if $p_x \neq 1/\ell$ and $p_m \neq 1/2$, then $Q_{m\times mask}$ is non-symmetrical. □

Appendix 3: Proof for Theorem 1

We consider that the EMCMC resulting from the interaction between transition matrix Q and the (generalized)

Metropolis acceptance rule is an EMCMC over the N dimensional search space E^N. For ease of exposure and without loss of generality, let's consider populations of two individuals $N = 2$. Two children $\{\mathbf{y}_1, \mathbf{y}_2\}$ that are generated with some irreducible and symmetrical proposal distribution Q from two parents $\left\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\right\}$. Then $Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)$.

The Metropolis acceptance rule in this case is $\alpha_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = \min(1, \frac{\hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)}{\hat{P}_1(\mathbf{x}_1^{(t)})\cdot\hat{P}_2(\mathbf{x}_2^{(t)})})$. The transition matrix we denote with K_C. The transition probability that two children $\mathbf{y}_1$ and $\mathbf{y}_2$ are generated and both are accepted is

$$K_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) = Q(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) \cdot \alpha_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})$$

The rejection transition probability that both children are rejected is

$$\sum_{\{\mathbf{y}_1,\mathbf{y}_2\}\neq\left\{\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}\right\}} Q(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) \cdot [1 - \alpha_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})]$$

Let's assume without loss of generality that $\frac{\hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)}{\hat{P}_1(\mathbf{x}_1^{(t)})\cdot\hat{P}_2(\mathbf{x}_2^{(t)})} < 1$. Then,

$$\begin{aligned}\alpha_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) &= \min(1, \frac{\hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)}{\hat{P}_1(\mathbf{x}_1^{(t)})\cdot\hat{P}_2(\mathbf{x}_2^{(t)})}) \\ &= \frac{\hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)}{\hat{P}_1(\mathbf{x}_1^{(t)})\cdot\hat{P}_2(\mathbf{x}_2^{(t)})}\end{aligned}$$

and $\alpha(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2) = 1$.

We now show that the detailed balance condition holds

$$\begin{aligned}&\hat{P}_1(\mathbf{x}_1^{(t)})\cdot\hat{P}_2(\mathbf{x}_2^{(t)})\cdot K_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{y}_1^{(t)},\mathbf{x}_2^{(t)}) \\ &= \hat{P}_1(\mathbf{x}_1^{(t)})\cdot\hat{P}_2(\mathbf{x}_2^{(t)})\cdot Q(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})\cdot\alpha_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) \\ &= \hat{P}_1(\mathbf{x}_1^{(t)})\cdot\hat{P}_2(\mathbf{y}_2^{(t)})\cdot Q(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{y}_1^{(t)},\mathbf{x}_2^{(t)})\cdot\frac{\hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)}{\hat{P}_1(\mathbf{x}_1^{(t)})\cdot\hat{P}_2(\mathbf{x}_2^{(t)})} \\ &= \hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)\cdot Q(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) \\ &= \hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)\cdot Q(\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)} \mid \mathbf{y}_1,\mathbf{y}_2)\cdot 1 \\ &= \hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)(\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)} \mid \mathbf{y}_1,\mathbf{y}_2)\cdot\alpha_C(\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)} \mid \mathbf{y}_1,\mathbf{y}_2) \\ &= \hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)\cdot K_C(\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)} \mid \mathbf{y}_1,\mathbf{y}_2)\end{aligned}$$

The marginal transition probability to generate $\mathbf{x}_1^{(t)}$ from $\mathbf{y}_1$ when summing over the variables of the second chain is

$$K_C(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) = \sum_{\mathbf{x}_2^{(t)},\mathbf{y}_2} \hat{P}_2(\mathbf{x}_2)\cdot K_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})$$

The stationary marginal distribution of the first chain is $\hat{P}(\cdot)$

From the above equations we infer

$$\begin{aligned}\hat{P}_1(\mathbf{y}_1) &= \sum_{\mathbf{y}_2}\sum_{\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}} K_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})\cdot\hat{P}_1(\mathbf{y}_1^{(t)})\cdot\hat{P}_2(\mathbf{x}_2^{(t)}) \\ &= \sum_{\mathbf{y}_2}\sum_{\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}} K_C(\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)} \mid \mathbf{y}_1,\mathbf{y}_2)\cdot\hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2) \\ &= \sum_{\mathbf{y}_2}\hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)\cdot\sum_{\mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}} K_C(\mathbf{y}_1^{(t)},\mathbf{x}_2^{(t)} \mid \mathbf{y}_1,\mathbf{y}_2) \\ &= \hat{P}_1(\mathbf{y}_1)\cdot\sum_{\mathbf{y}_2}\hat{P}_2(\mathbf{y}_2)\cdot 1 = \hat{P}_1(\mathbf{y}_1)\end{aligned}$$

where we have used

$$\begin{aligned}&\hat{P}_1(\mathbf{y}_1)\cdot\hat{P}_2(\mathbf{y}_2)\cdot K_C(\mathbf{y}_1^{(t)},\mathbf{x}_2^{(t)} \mid \mathbf{y}_1,\mathbf{y}_2) \\ &\quad = \hat{P}_1(\mathbf{x}_1^{(t)})\cdot\hat{P}_2(\mathbf{y}_2^{(t)})\cdot K_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{y}_1^{(t)},\mathbf{x}_2^{(t)})\end{aligned}$$

We conclude that the marginal target distribution for the i-th chain is $P_i(\cdot)$ and that this EMCMC algorithm has the stationary distribution $P_1(\cdot)\times P_2(\cdot)$.

The EMCMC algorithm is irreducible since the proposal distribution Q is irreducible. This algorithm is aperiodic since the Metropolis algorithm, by construction is aperiodic. We conclude that this EMCMC is ergodic with the stationary distribution $P_1(\cdot)\times P_2(\cdot)$, where $P_i(\cdot)$ is the marginal target distribution of the i-th chain. □

Appendix 4: Proof of Proposition 10

Consider two parents $\mathbf{x}_1^{(t)}$ and $\mathbf{x}_2^{(t)}$ and their two generated children $\mathbf{y}_1$ and $\mathbf{y}_2$, and the coupled acceptance α_C that accepts/rejects both states and the probability to accept/reject one or both children with $\alpha_{1.1}$. At first, we prove the first inequality from Proposition 10

$$\begin{aligned}&K_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) \\ &\quad \le K_{1.1}(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) + K_{1.1}(\mathbf{y}_1,\mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})\end{aligned}$$

The right side of this inequation can be rewriten as

$$\begin{aligned}&K_{1.1}(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) + K_{1.1}(\mathbf{y}_1,\mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) \\ &\quad = Q(\mathbf{y}_{1,2} \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})\cdot\alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})\cdot\alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)}) \\ &\qquad + Q(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})\cdot\alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})\cdot[1 - \alpha(\mathbf{y}_2 \mid \mathbf{y}_2^{(t)})] \\ &\quad = Q(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})\cdot\alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) = Q(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{y}_1^{(t)},\mathbf{x}_2^{(t)})\end{aligned}$$

because $\hat{P}(\mathbf{y}_1) > \hat{P}(\mathbf{x}_1^{(t)})$ and, thus $\alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) = 1$. The left side of the inequality 3 can be rewritten as

$$\begin{aligned}&K_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)}) \\ &\quad = Q(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})\cdot\alpha_C(\mathbf{y}_1,\mathbf{y}_2 \mid \mathbf{x}_1^{(t)},\mathbf{x}_2^{(t)})\end{aligned}$$

The inequality 3 holds since

$$\alpha_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \leq 1$$

We now prove that the rejection probability for a less fit child, $\hat{P}(\mathbf{y}_1) < \hat{P}(\mathbf{x}_1^{(t)})$, is larger for $K_{1.1}$ than for K_C when the second child is more fit than the second parent, $\hat{P}(\mathbf{y}_2) > \hat{P}(\mathbf{x}_2^{(t)})$. Then

$$K_C(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \leq K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) + K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

The right side of the inequality 4 is

$$\begin{aligned} &K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid {}_1^{(t)}, \mathbf{x}_2^{(t)}) + K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \\ &\quad = Q({}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})] \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})] \\ &\qquad + Q(\mathbf{y}_1, \mathbf{y}_2 \mid {}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})] \cdot \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)}) \\ &\quad = Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})] \end{aligned}$$

Rewriting the left side of the inequality 4

$$\begin{aligned} &K_C(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \\ &\quad = Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})] \end{aligned}$$

The inequality 4 holds since

$$\begin{aligned} &K_C(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid {}_1^{(t)}, \mathbf{x}_2^{(t)}) - K_{1.1}(\mathbf{x}_1^{(t)}, {}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \\ &\quad = [1 - \alpha_C(\mathbf{y}_1, \mathbf{y}_2 \mid {}_1^{(t)}, \mathbf{x}_2^{(t)})] - [1 - \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})] \\ &\quad = \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) - \alpha_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \\ &\quad = \frac{\hat{P}(\mathbf{y}_1)}{\hat{P}(\mathbf{x}_1^{(t)})} - \frac{\hat{P}(\mathbf{y}_1) \cdot \hat{P}(\mathbf{y}_2)}{\hat{P}({}_1^{(t)}) \cdot \hat{P}(\mathbf{x}_2^{(t)})} = \frac{\hat{P}({}_1)}{\hat{P}(\mathbf{x}_1^{(t)})} \cdot \left[1 - \frac{\hat{P}(\mathbf{y}_2)}{\hat{P}(\mathbf{x}_2^{(t)})}\right] \leq 0 \end{aligned}$$

Following the same line of reasoning, the rejection probability that a less fit child, $\hat{P}(\mathbf{y}_1) < \hat{P}(\mathbf{x}_1^{(t)})$, is lower for $K_{1.1}$ than for K_C when the second child is less fit than the second parent, $\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})$ follows directly.

We now show that the EMCMC defined by $K_{1.1}$ has detailed balance if and only if

$$Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)})$$

The detailed balance should hold only in the case that two different children are proposed but only one of them is accepted and the other is rejected

$$\begin{aligned} &\hat{P}_1(\mathbf{x}_1^{(t)}) \cdot \hat{P}_2(\mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})] \\ &\quad \cdot Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = \hat{P}_1(\mathbf{y}_1) \cdot \hat{P}_2(\mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{x}_1^{(t)} \mid \mathbf{y}_1) \\ &\quad \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})] \cdot Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) \end{aligned}$$

Or the above equation holds if and only if

$$Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)})$$

If the detailed condition holds and the proposal distribution is irreducible and symmetrical, the EMCMC is ergodic and converge to the target distribution $P_1(\cdot) \times P_2(\cdot)$. □

Appendix 5: Proof of Proposition 11

We show that ECA does not have detailed balance for any non-uniform distribution. Let's consider three states, $\mathbf{y}_1, \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}$ such that $\hat{P}(\mathbf{y}_1) > \hat{P}(\mathbf{x}_2^{(t)}) > \hat{P}(\mathbf{x}_1^{(t)})$. In our discussion from Sect. 5.3 we show that

$$\begin{aligned} K_{ECA}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) &= K_C(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \\ &+ \sum_{\mathbf{y}_2, \{\mathbf{y}_1, \mathbf{x}_2^{(t)}\} = \max_2\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\}} Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \end{aligned}$$

If ECA has detailed balance, then

$$\begin{aligned} &Q(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot K_{ECA}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \\ &\quad = Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) \cdot K_{ECA}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) \end{aligned}$$

Because Q is symmetrical we further have

$$K_{ECA}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = K_{ECA}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{x}_2^{(t)})$$

K_C is also symmetrical, so further we have that

$$\begin{aligned} &\sum_{\mathbf{y}_2, \{\mathbf{y}_1, \mathbf{x}_2^{(t)}\} = \max_2\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\}} Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \\ &\quad = \sum_{\mathbf{y}_2, \{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\} = \max_2\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\}} Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) \end{aligned}$$

The above equation does not hold since for the first sum requires that

$$\left\{\mathbf{y}_1, \mathbf{x}_2^{(t)}\right\} = \max_2\left\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\right\}$$

and for the second sum that

$$\left\{\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\right\} = \max_2\left\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}\right\}$$

We conclude that K_{ECA} does not have detailed balance for any Q.

When Q is irreducible, this algorithm is irreducible since it can generate any state from any other state with a non-zero probability. Therefore, the algorithm is also aperiodic since the K_C is aperiodic and thus the target distribution of ECA exists. □

Appendix 6: Proof of Proposition 13

The proof is split in three parts, corresponding with the three equations in the proposition.

a) *Both children are different from their parents*. Then

$$K_{1.1}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) \cdot \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})$$

The reverse transition probability is

$$K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2) = Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2) \cdot \alpha(\mathbf{x}_1^{(t)} \mid \mathbf{y}_1) \cdot \alpha(\mathbf{x}_2^{(t)} \mid \mathbf{y}_2)$$

We now have

$$\frac{K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)}{K_{1.1}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, {}_2^{(t)})} = \frac{Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2) \cdot \alpha(\mathbf{x}_1^{(t)} \mid {}_1) \cdot \alpha(\mathbf{x}_2^{(t)} \mid \mathbf{y}_2)}{Q(\mathbf{y}_1, {}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) \cdot \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})} = 1 \cdot \frac{\alpha(\mathbf{x}_1^{(t)} \mid {}_1)}{\alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})} \cdot \frac{\alpha(\mathbf{x}_2^{(t)} \mid \mathbf{y}_2)}{\alpha({}_2 \mid \mathbf{x}_2^{(t)})}$$

where Q is symmetrical and thus $Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2) = Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$. By replacing the definition of acceptance rule α, we have

$$\frac{\alpha(\mathbf{x}_1^{(t)} \mid \mathbf{y}_1)}{\alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})} = \frac{\hat{P}(\mathbf{x}_1^{(t)})}{\hat{P}(\mathbf{y}_1)}$$

and

$$\frac{\alpha(\mathbf{x}_2^{(t)} \mid \mathbf{y}_2)}{\alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})} = \frac{\hat{P}(\mathbf{x}_2^{(t)})}{\hat{P}(\mathbf{y}_2)}$$

The coupled acceptance probability for the nested acceptance probability is now

$$\alpha_C(y_1, y_2 \mid x_1^{(t)}, x_2^{(t)}) = \min\left\{1, \frac{\hat{P}(\mathbf{y}_1) \cdot \hat{P}(\mathbf{y}_2)}{\hat{P}(\mathbf{x}_1^{(t)}) \cdot \hat{P}(\mathbf{x}_2^{(t)})} \cdot \frac{K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)}{K_{1.1}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})}\right\} = \min\left\{1, \frac{\hat{P}(\mathbf{y}_1) \cdot \hat{P}(\mathbf{y}_2)}{\hat{P}(\mathbf{x}_1^{(t)}) \cdot \hat{P}(\mathbf{x}_2^{(t)})} \cdot \frac{\hat{P}(\mathbf{x}_1^{(t)})}{\hat{P}(\mathbf{y}_1)} \cdot \frac{\hat{P}(\mathbf{x}_2^{(t)})}{\hat{P}(\mathbf{y}_2)}\right\} = 1$$

The nested transition probability now is

$$K_{nEMCMC}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = K_{1.1}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

b) *One child is different from its parent*. For example $\mathbf{y}_2$ is rejected and $\mathbf{y}_1$ is accepted. Then

$$K_{1.1}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = Q(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) + \sum_{\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})} Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)}) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]$$

The reverse transition probability is

$$K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) = Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{x}_1^{(t)} \mid \mathbf{y}_1) + \sum_{\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})} Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) \cdot \alpha(\mathbf{x}_1^{(t)} \mid \mathbf{y}_1) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]$$

We now have

$$\frac{K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{x}_2^{(t)})}{K_{1.1}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})} = \frac{\alpha(\mathbf{x}_1^{(t)} \mid \mathbf{y}_1)}{\alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})} \cdot \frac{q + \sum_{\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})} Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]}{q + \sum_{\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})} Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]}$$

where $q = Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) = Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$. Now, the coupled acceptance is

$$\alpha_C(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = \min\left\{1, \frac{\hat{P}(\mathbf{y}_1)}{\hat{P}(\mathbf{x}_1^{(t)})} \cdot \frac{K_{1.1}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{x}_2^{(t)})}{K_{1.1}(\mathbf{y}_1, \mathbf{x}_2^{(t)} \mid vecx_1^{(t)}, \mathbf{x}_2^{(t)})}\right\} = \min\left\{1, \frac{q + \sum_{\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})} Q(\mathbf{x}_1^{(t)}, \mathbf{y}_2 \mid \mathbf{y}_1, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]}{q + \sum_{\hat{P}(\mathbf{y}_2) < \hat{P}(\mathbf{x}_2^{(t)})} Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) \cdot [1 - \alpha(\mathbf{y}_2 \mid \mathbf{x}_2^{(t)})]}\right\}$$

where $\frac{\hat{P}(\mathbf{y}_1)}{\hat{P}(\mathbf{x}_1^{(t)})} = \frac{\alpha(\mathbf{y}_1 \mid \mathbf{x}_1^{(t)})}{\alpha(\mathbf{x}_1^{(t)} \mid \mathbf{y}_1)}$. The second equation from the proposition now follows directly.

c) *Both children are rejected*. Then

$$K_{nEMCMC}(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}) = 1 - \sum_{\mathbf{y}_1, \mathbf{y}_2 \neq \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}} K_{nEMCMC}(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

□

Appendix 7: Proof of Proposition 14

The coupled acceptance probability for the nested acceptance probability is now

$$\alpha_C(y_1, y_2 \mid x_1^{(t)}, x_2^{(t)}) = \min\left\{1, \frac{\hat{P}(\mathbf{y}_1) \cdot \hat{P}(\mathbf{y}_2)}{\hat{P}(\mathbf{x}_1^{(t)}) \cdot \hat{P}(\mathbf{x}_2^{(t)})} \cdot \frac{K_C(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)}{K_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})}\right\} = \min\left\{1, \frac{\hat{P}(\mathbf{y}_1) \cdot \hat{P}(\mathbf{y}_2)}{\hat{P}(\mathbf{x}_1^{(t)}) \cdot \hat{P}(\mathbf{x}_2^{(t)})} \cdot \frac{Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)}{Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})} \cdot \frac{\alpha_C(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)}{\alpha_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})}\right\} = 1$$

where Q is symmetrical

$$Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2) = Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})$$

and

$$\frac{\alpha_C(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)}{\alpha_C(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})} = \frac{Q(\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)} \mid \mathbf{y}_1, \mathbf{y}_2)}{Q(\mathbf{y}_1, \mathbf{y}_2 \mid \mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)})}$$

The equation in the proposition follows directly. □

References

1. Andrieu C, de Freitas N, Doucet A, Jordan M (2003) An introduction to MCMC for machine learning. Mach Learn 50:5–43
2. Baluja S, Davies S (1997) Using optimal dependency-trees for combinational optimization. In: Proceedings of international conference of machine learning (ICML'97). Morgan Kaufmann, San Francisco, pp 30–38
3. Campillo F, Rakotozafy R, Rossi V (2009) Parallel and interacting Markov chain Monte Carlo algorithm. Math Comput Simul 79(12):3424–3433
4. Chow CK, Liu CN (1968) Approximating discrete probability distributions with dependence trees. IEEE Trans Inf Theory 14:462–467
5. Corander J, Ekdahl M, Koski Timo (2008) Parallel interacting MCMC for learning of topologies of graphical models. Data Min Knowl Discov 17(3):431–456
6. Doucet A, de Freitas N, Gordon N, (eds) (2001) Sequential Monte Carlo methods in practice. Springer, Berlin
7. Drugan MM, Thierens D (2004) Evolutionary Markov chain Monte Carlo. In: Artificial evolution, LNCS 2936. Springer, Berlin, pp 63–76
8. Drugan MM, Thierens D (2005) Recombinative EMCMC algorithms. In: Proceeding of IEEE congress of evolutionary computation, CEC'05. IEEE Press, Piscataway, pp 2024–2031
9. Fearnhead P (2008) Special issue: adaptive Monte Carlo methods. Stat Comput 18(4):341–480
10. Geraci J (2008) A new connection between quantum circuits, graphs and the ising partition function. Quantum Inf Process 7(5):227–242
11. Geyer CJ (1991) Markov Chain Monte Carlo maximum likelihood. In: Computing science and statistics: proceeding of the 23rd symposium on the interface. pp 156–163
12. Gilks WR, Richardson S, Spiegelhalter DJ, editors (1996) Markov Chain Monte Carlo in practice. Chapman & Hall, London
13. Hastings WK (1970) Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57:97–109
14. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
15. Laskey KB, Myers JW (2003) Population Markov Chain Monte Carlo. Mach Learn 50:175–196
16. Liang F, Wong WH (2000) Evolutionary Monte Carlo: applications to C_p model sampling and change point problem. In: Statistica sinica. pp 317–342
17. Mahfoud SW, Goldberg DE (1995) Parallel recombinative simulated annealing: a genetic algorithm. Parallel Comput 21:1–28
18. Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH , Teller E (1953) Equation of state calculations by fast computing machines. J Chem Phys 21:1087–1092
19. Pelikan M, Goldberg DE, Lobo F (2002) A survey of optimization by building and using probabilistic models. Comput Optim Appl 21:5–20
20. Radcliffe NJ (1991) Forma analysis and random respectful recombination. In: Proceeding of the fourth international conference on genenetic algorithms. pp 222–229. Morgan Kaufmann
21. Roberts GO, Gelman A, Gilks WR (1997) Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann Appl Probab 7(1):110–120
22. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11:341–359
23. Strens MJA (2003) Evolutionary MCMC sampling and optimization in discrete spaces. In: Proceeding of international conference of machine learning (ICML'03). pp 736–743
24. ter Braak CJF (2006) Genetic algorithms and Markov chain Monte Carlo: differential evolution Markov Chain makes Bayesian computing easy. Stat Comput 16(3):239–249
25. Thierens D, Goldberg DE (1994) Elitist recombination: an integrated selection-recombination GA. In Proceeding of the congress on computational intelligence. pp 508–512
26. Wolpert DH, Lee CF (2005) Adaptive metropolis sampling and optimization with product distributions. Technical report, NASA Ames Research Center
27. Zhang BT, Cho DY (2001) System identification using evolutionary Markov Chain Monte Carlo. J Syst Archit 47:587–599