Chapter 1
On the Foundations and the Applications of Evolutionary Computing

Pierre Del Moral, Alexandru-Adrian Tantar, and Emilia Tantar

Abstract. Genetic type particle methods are increasingly used to sample from complex high-dimensional distributions. They have found a wide range of applications in applied probability, Bayesian statistics, information theory, and engineering sciences. Understanding rigorously these new Monte Carlo simulation tools leads to fascinating mathematics related to Feynman-Kac path integral theory and their interacting particle interpretations. In this chapter, we provide an introduction to the stochastic modeling and the theoretical analysis of these particle algorithms. We also illustrate these methods through several applications.

1.1 Introduction

Most of population-based algorithms are described in terms of interacting samples evolving in some solution state space. The random samples are also termed solutions, particles, individuals or genotypes. Their time evolution mimics natural selection, physical adaptation, reinforced principles, or some social behavior. For a detailed discussion, and an overview of these classes of evolutionary computing models we refer the reader to the couple of books [8, 115], and references therein.

Pierre Del Moral
Centre INRIA Bordeaux Sud-Ouest, Institut de Mathématiques de Bordeaux,
Université de Bordeaux I, 351 cours de la Libération 33405 Talence cedex,
and Applied Mathematics Department, CMAP École Polyetchnique Paris, France
e-mail: Pierre.Del-Moral@inria.fr

Alexandru-Adrian Tantar · Emilia Tantar
Computer Science and Communications Research Unit, University of Luxembourg
e-mail: {alexandru.tantar,emilia.tantar}@uni.lu

E. Tantar et al. (Eds.): EVOLVE- A Bridge between Probability, SCI 447, pp. 3–89.
springerlink.com

Evolutionary type algorithms are very often presented with limited regard to their rigorous mathematical foundations, without any rigorous analysis of the underlying random evolutionary processes. Their performance often relies on intuition driven by numerical observations, so that evolutionary computing research sometimes falls in obscure inspired paradigms.

In this context, several questions arise: To what extent do we accept numerical evidence as a proof when dealing with complex systems of high risk as nuclear plants, health care management, security or defense systems? What is the sensitivity and the error those paradigms introduce as part of a solution's design? What is the impact and how uncertainties propagate w.r.t. time?

Evolutionary type algorithms can be interpreted in two different ways.

Firstly, population-based algorithms w.r.t. optimization problems can be viewed as a gradient type hole puncher in complex solution state spaces. In this context, the central idea is to use natural evolution mechanisms to improve step by step the population adaptation. For convex optimization problems, the performance and convergence of the algorithm follows from standard analysis of stochastic gradient models. For more complex optimization problems, we expect large population explorations to escape from local minima.

On the other hand, under certain regularity conditions, most of genetic type evolutionary computing algorithms converge towards some particular probability distributions. These target probability measures are often prescribed by distributions on path spaces w.r.t. a series of conditioning events. For regulation problems, and open loop optimal control problems, these two viewpoints can be encapsulated in a single mathematical framework [107, 108, 119].

Answering to all the questions provided above amounts to rigorously analyzing the convergence and the performance of these stochastic hole puncher gradients, or these genetic particle sampling models. The second and rather recent viewpoint is the central theme of this chapter.

We end this introduction with a brief discussion on the origins and the mathematical foundations of genetic type particle models.

Genetic type stochastic models are increasingly used to sample from complex high-dimensional distributions. As we mentioned above, they approximate, as the population size tends to infinty, a given target probability distributions by a large cloud of random samples termed particles. Practically, the particles evolve randomly around the space independently and to each particle is associated a positive potential function. Periodically we duplicate particles with high potentials at the expense of particles with low potentials which die. This intuitive genetic mutation-selection type mechanism appears in numerous applications ranging from nonlinear filtering [22, 36, 52, 45, 38, 68, 67, 69, 89, 118, 120], Bayesian statistics [29, 40, 76, 123], combinatorial counting [3], molecular and polymer simulation [90], rare events simulation [26, 27, 82], quantum Monte Carlo methods [6, 102, 125] and genetic algorithms [47, 48, 87, 104], among others.

From a mathematical point of view, these methods can be interpreted as stochastic numerical approximations of Feynman-Kac measures. These measures represent the distribution of the paths of a reference Markov process, weighted by a collection

of potential functions. These functional models are natural mathematical extensions of the traditional change of probability measures, commonly used in importance sampling. The particle interpretation consists in evolving a population of particles mimicking natural evolution mechanisms. During the mutation stage, the particles evolve independently of one another, according to the same probability transitions as the ones of the reference Markov chain. During the selection stage, each particle evaluates the potential value of its location. The ones with small relative values are killed, while the ones with high relative values are multiplied. The corresponding genealogical tree occupation measure converges, as the population size tends to infinity, to the complete Feynman-Kac distribution on path space.

The origins of stochastic particle simulation certainly start with the seminal paper of N. Metropolis and S. Ulam [126] published in 1949. As explained by these two physicists in the introduction of their pioneering article, the Monte Carlo method is, "essentially, a statistical approach to the study of differential equations, or more generally, of integro-differential equations that occur in various branches of the natural sciences". The links between genetic type particle Monte Carlo models and quadratic type parabolic integro-differential equations have been developed in the beginning of 2000' in the series of articles on continuous time models [51, 52, 54].

The earlier works on heuristic type genetic particle schemes seem to have started in Los Alamos National Labs with works of M.N. Rosenbluth and A.W. Rosenbluth [143], and T.E. Harris and H. Kahn [94]. We also quote the work on artificial life of Nils Aall Barricelli at the Institute for Advanced Study in Princeton [10, 11]. In all of these works, the genetic Monte Carlo scheme is always presented as a natural heuristic resampling type algorithm to generate random population models, to sample molecular conformations, or to estimate high energy particle distributions, without a single convergence estimate to ensure the performance, nor the robustness of the Monte Carlo sampler.

Since the mid 90's, genetic particle algorithms have recorded a dramatic popularity increase due to the proliferation and wide accessibility of powerful computing resources. They are now extensively and routinely used in engineering, machine learning, statistics and physics under sometimes different names, such as: particle filters, bootstrap or genetic filters, population Monte Carlo methods, sequential Monte Carlo models, genetic search models, branching and multi-level splitting particle rare event simulations, condensation models, go-with-the winner, spawning models, walkers population reconfigurations, pruning-enrichment strategies, quantum and diffusion Monte Carlo, rejuvenation models, and many others.

The mathematical foundations, and the performance analysis of all of these discrete generation particle models are rather recent. The first rigorous study in this field seems to be the article published by the first author in 1996 on the applications of particle methods to nonlinear estimation problems [36]. This article provides the first proof of the unbiased property of particle likelihood approximation models (lemma 3 page 12); and adaptive resampling criteria w.r.t. the weight dispersions (see remark 1 on page p.4).

This article also presents the first convergence results for a new class of interacting particle filters, originally presented as heuristic Monte Carlo schemes in the

beginning of the 1990's in three independent schools. A series of classified industrial Research Contracts on tracking and control developed between 1990 and 1993 by the P. Del Moral, J.C. Noyer, G. Rigal, and G. Salut [59, 60, 61, 62, 63, 64], and [23, 65, 66]. The first journal article presenting the heuristic of particle filters is the article by N.J. Gordon, D. Salmond and A.F.M. Smith [89], and the first conference article presenting the heuristic of particle filters is the article by G. Kitagawa [118].

For a more thorough discussion on these models, we refer the reader to [37, 52, 38, 58, 67], as well as in [8, 115, 9, 72, 5, 152], and references therein.

1.1.1 From Evolutionary Computing to Particle Algorithms

Besides a sustained research on evolutionary computing, theoretical support and convergence proofs were until recently regarded as non mandatory. While leading to significant results in practice [34, 8, 115], advances were only derived on intuition and empirical grounds. And while this can be sustained for a real world setup where finite time and resource constraints are imposed, understanding the different paradigms and afferent convergence properties demands an accurate description of the underlying mathematical models.

A first aspect to address is what evolutionary computing is applied for and what information is expected? As later detailed in this section, different classes of problems are considered, e.g. non-linear, non-convex, discrete or continuous, with one or multiple objectives to optimize, highly multimodal, ill-conditioned or with epistatic interactions defined, within dynamic environments or subject to stochastic perturbations (uncertainty). All are finally connected by assumptions implying (i) no asymptotic convergence proof, with no exact solution or reproducible (stochastic) behavior expected under finite time constraints, (ii) nonexistence of a polynomial time alternative approach, e.g. due to a combinatorial explosion of the search space, intractability or exponential increase of the number of local optima, and (iii) exploration (classically) ended with no explicit information on the distribution of the optima, only the best found solution being provided as a result.

At the opposite end, deterministic algorithms (not covered here), e.g. interval methods, branch-and-bound, provide an optimal solution within a finite time and with finite resources, nonetheless requiring an exponentially increasing time (as a function of instance size). With respect to the last assumption, while different evolutionary algorithms enclose by definition intrinsic support for estimating normalizing constants, observing ancestral and genealogical structures or convergence towards target distributions, in most cases this information is not regarded as relevant and discarded. And, while those aspects fall by excellence in the domain of particle algorithms, commonly referred to, for example, as particle filters, sequential, diffusion or quantum Monte Carlo, and with a focus for estimating distribution laws, evolutionary and particle algorithms, with marginal exceptions, follow identical conceptual lines.

From the probabilistic point of view, genetic type particle algorithms are a natural class of Monte Carlo methods for sampling complex high-dimensional probability distributions and estimating their normalizing constants. As we already mentioned, this class of algorithms approximate a given sequence of target probability measures by a large cloud of random samples termed particles (equivalent of individuals in evolutionary algorithms). The particles evolve randomly in the solution space (mutation and free exploration). A positive potential/fitness function is associated to each particle. Periodically, the particles with high potential value are duplicated at the expense of particles with low potentials which are discarded (selection and replacement).

An overview of the evolutionary computing domain is offered in the following, to no extent exhaustive, only in order to highlight connection points with particle algorithms. For the remainder of this section, let us consider a simplified scenario where, given an arbitrary deterministic, static black-box function, the optimal solution (or approximation of) is demanded. Having as sole assumption that the function can be sampled within the entire definition domain, i.e. with no other information on the nature of the function, continuous or discrete, and disregarding the encoding of solutions, e.g. fixed or variable size array of binary, integer or real values, Gray coding (reflected binary code), graphs, trees, cellular, messy, direct or indirect encoding, a straightforward approach would be to draw samples until some termination criterion is met. Except the simplification, this portrays the basic idea of a simple Monte Carlo algorithm. Extending this direction, a first axis of discussion leads to single-solution based exploration paradigms. Note that while this particular class is commonly referred to as local search algorithms this only relates to exploration being conducted by sampling from the neighborhood of a single solution, evolved in iterative manner, and does not automatically imply a limitation for the algorithm's exploration capabilities. A second part of the discussion focuses on algorithms that simultaneously evolve a set of independent solutions while ensuring a balance or trade-off between local and global exploration, later moving to hybrid and parallel aspects.

1.1.1.1 Local Search Algorithms

As a general classification, *direct* and *indirect* search methods can be considered [141]. The first class, also denominated as *zero order methods*, only relies on a direct sampling of the objective function, with no partial derivatives, i.e. no analytic or numeric gradient employed. Initially introduced in the work of Hooke and Jeeves [105], the *direct search* denomination offers an explicit delimitation from higher complexity methods: "*the phrase implies our preference, based on experience, for straightforward search strategies which employ no techniques of classical analysis* [...]". Examples include Hill-Climbing, Nelder-Mead [153, 135, 121], where the exploration is conducted by a set of perturbations applied to a randomly generated initial simplex, Solis and Wets [151], including self-adaptive mechanisms, Tabu Search [83, 84], Variable Neighborhood Search [131, 132], Guided

Local Search [161], Iterated Local Search [13, 124] or Simulated Annealing [117], developed as a generalization of Metropolis Monte Carlo [127]. The latter class can be further divided into *first* and *second order methods*, using the first, respectively second or higher order derivatives (analytic or numerical approximations of) for guiding the exploration process. Examples include steepest descent, Conjugate Gradient [100, 101, 77, 78, 139, 150] and second order methods like the Limited Memory BFGS [17, 79, 88, 149, 137, 140] or the Adaptive Simulated Annealing [112, 111, 109, 110, 113].

1.1.1.2 Set of Solutions Based Algorithms

A reference in the evolutionary computing domain, Genetic Algorithms (GAs), developed through the work of Holland [104], define a structure that inspired and set the bases for different other paradigms. The approach relies on several distinct stages as follows: (i) initialization – a set of initial solutions (chromosomes in the case of GAs) are randomly sampled, forming a population, (ii) selection – a subset of the best fit solutions is constructed, (iii) recombination and perturbation – new samples are drawn by applying several operators on the previously selected solutions, e.g. crossover and mutation operators, and (iv) replacement – least fit solutions in the initial population are replaced, the algorithm iterating steps (ii) to (iv) until a termination criterion is met.

While no in-depth details will be provided here, it may be worthwhile mentioning that different strategies and operators were proposed and analyzed for all stages of a genetic algorithm. Examples of mutation and recombination operators include (i) diversification oriented or mutation constructions [128] like bit-flip or swap operators, polynomial transforms, (non) uniform, Gaussian or Cauchy distribution based, as well as (ii) intensification, or crossover operators [97, 95, 98, 99, 96]. This latter class was extensively investigated, leading to operators with one or multiple cutting or intersection points, uniform, arithmetic, geometric, Wright's heuristic, linear BGA, α, β-blend, simulated binary crossover, fuzzy recombination or dynamic operators. At the same time, multiple offspring operators were studied, with only the best two offspring solutions out of all finally selected, e.g. linear or the min-max arithmetic operator. Examples of more advanced operators include higher or adjustable arity operators (multiple parent solutions), including global recombination, gene-pool recombination, linkage evolving operator or the m-tuple mating [70, 16, 148, 134].

From a mathematical point of view, disregarding recombination operators, these methods can be interpreted as stochastic numerical approximations of Feynman-Kac measures, representing the distribution of paths for a reference Markov process weighted by a collection of potential functions. These functional models are natural mathematical extensions of the traditional change of probability measures, commonly used in importance sampling. The particle interpretation, as a direct analogy to genetic algorithms, consists in evolving a population of particles that mimic natural evolution mechanisms. During the transition (mutation) stage, the particles evolve independently one of another, according to the same probability transitions

as the ones of the reference Markov chain. During the selection stage, each particle evaluates the potential value of its location and the ones with small relative values are discarded while the ones with high relative values are multiplied. The corresponding genealogical tree occupation measure converges, as the population size tends to infinity, to the complete Feynman-Kac distribution on path space. As a direct analogy, particle algorithms also rely on transition operators (equivalent of mutation) although imposing to leave the initial distribution measure invariant. No direct equivalent of crossover operators exists however as a straightforward understanding and modeling of a recombination transition does not always make sense or is even possible and coherent in a simulation context. Analogously, different selection strategies were explored, including proportional selection, stochastic universal sampling, tournament, (linear, exponential) ranking, sigma scaling or Boltzmann selection, all with or without elitism or truncation. Additionally, replacement may consider the depletion of the worst, best or most similar individuals (crowding), replacement of parent(s) or of randomly selected individuals. While selection is also considered in particle algorithms, semantics may differ depending on the specific application area, e.g. being referred to as resampling, filtering, absorption, etc., and implicitly encloses replacement. Additional examples and applications are presented in Section 1.5, allowing for a direct analogy between evolutionary computing and particle algorithms.

Extensions of the classical genetic paradigm fostered different axes of study, leading to co-evolution and memetic algorithms [133, 136, 1], e.g. hybrid local vs global exploration strategies, Lamarckian evolution and Baldwin effect, meta and hyper-heuristics [18], cultural algorithms [142], differential evolution [157], swarm intelligence (ant colony, particle swarm, artificial immune systems, etc.) [15, 33, 116, 35], scatter search and path relinking [85, 93], genetic programming [122] (symbolic regression), or evolution strategies [7, 14], among many others. All and each of these paradigms finally led to intense research on, for example, different hybridization strategies at low-level, operator enclosed, or at high-level, as a sequence of heuristics or independently evolving parallel algorithms, different strategies in differential evolution, etc. Furthermore, different approaches like the Covariance Matrix Adaptation Evolution Strategies [92, 91], Estimation of Distribution Algorithms [138] or the Reactive Search [12] (sub-symbolic learning, adaptation and incremental model development) were introduced, exploiting landscape information in the form of second order model approximations, estimations of optima distribution or reinforcement based learning. As a side note, landscape studies developed as a standalone axis of research in an attempt to understand what correlation exists between specific features in an objective function's landscape or definition and the (non) efficiency of the different exploration strategies [159, 160, 154, 155, 162]. Extensive research was also conducted on (self) tuning and adaptive paradigms [103, 73, 71] with applications in ill-conditioned, dynamic and stochastic problems, including online problems or aspects as state dependency and decision making [106]. Additional axes, although out of scope and not detailed in this introduction, include multi-objective evolutionary computing algorithms [31, 30] where a set of best-compromise solutions have to be found (Pareto set and front in the

solution, respectively objective space), and parallel models [2, 21], e.g. multi-start, islands and topological (a)synchronous information exchange models. As a converging trend, an affinity for including or exploiting aspects and techniques from probability and statistics is emerging, making that, except for correspondences with filtering algorithms and sampling, different analogies are possible with applications in tracking, non-linear estimation problems, signal processing or stochastic optimization [47, 48, 40, 38, 58, 67].

1.1.2 Outline of the Chapter

The remainder of this chapter includes a pedagogical introduction to the stochastic modeling and the theoretical analysis of interacting particle algorithms in an effort to shed new light on some interesting links between physical, engineering, statistical and mathematical domains that appear disconnected at first glance. Second, the mathematical concepts and models are now at a point where they provide a very natural and unifying mathematical basis for a large class of Monte Carlo algorithms. To simplify the presentation and to clarify the main ideas behind these stochastic models, we have chosen to restrict the contents of this chapter to finite or countable state space models, avoiding any measure theory irrelevancies. In this simplified framework, we develop a rigorous mathematical analysis only involving vector and matrix operations. We emphasize that all of these particle models and the associated convergence results can be extended to general state-space models, including path-space models and excursion spaces on abstract measurable state spaces. In Section 1.5 several application areas are presented and a detailed description of interacting particle algorithms is provided.

1.2 Basic Notation and Motivation

In this section, we provide some basic notation and some comments on the stochastic models presented in this chapter. First, we mention that probabilistic models are always defined in terms of measures, numerical functions, as well as operators on functions and measures. Besides the fact that measures on finite spaces can be seen as elementary functions and linear operators as simple matrices, in order to provide a rigorous presentation and to facilitate the extensions to more general models we have chosen to keep the probabilistic terminology and the corresponding notation.

Let E be a finite set equipped with a matrix $(Q(x,y))_{x,y\in E}$. A signed measure on a finite set E is a mapping $x \in E \mapsto \mu(x) \in \mathbb{R}$. For any subset $A \subset E$, and any numerical function $x \in E \mapsto f(x)$ we set

$$\mu(A) := \sum_{x\in A} \mu(x) = \sum_{x\in E} \mu(x)\ \mathbb{1}_A(x) \quad \text{and} \quad \mu(f) = \sum_{x\in E} \mu(x) f(x)$$

with the indicator function $\mathbb{1}_A$ of a subset A. The Dirac measure at some point $x \in E$ is the indicator function $\mathbb{1}_x : y \in E \mapsto \mathbb{1}_x(y)$ of the set $\{x\}$. In this slightly abusive notation, we have $\mu(A) = \mu(\mathbb{1}_A)$ and $\mathbb{1}_x(A) = \mathbb{1}_A(x)$. A probability measure is a non negative measure μ such that $\mu(E) = 1$. Given some nonnegative measure μ on E, sometimes we use the proportional relation and we write

$$\nu \propto \mu \quad \text{to define the probability measure } \nu(x) = \mu(x)/\textstyle\sum_{z\in E} \mu(z).$$

The empirical measure associated with a set of N states $(x^1,\ldots,x^N) \in E^N$ is the measure defined by

$$y \in E \mapsto \eta^N(y) := \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{x^i}(y)$$

with $N \geq 1$. By construction, we have

$$\eta^N(f) = \sum_{y\in E} f(y)\eta^N(y) = \frac{1}{N}\sum_{i=1}^{N} f(x^i)$$

We also denote by $Q(f)$ and (μQ) the function $x \mapsto Q(f)(x)$ and the measure $y \mapsto (\mu Q)(y)$ defined below

$$Q(f)(x) := \sum_y Q(x,y)f(y) \quad \text{and} \quad (\mu Q)(y) := \sum_x \mu(x)Q(x,y) \tag{1.1}$$

In this notation, reversing the summation order, we have $\mu(Q(f)) = (\mu Q)(f)$.

For instance, for finite ordered state spaces with cardinality $d \geq 1$ there is no loss of generality to suppose that $E = \{1,\ldots,d\}$. In this case, we can identify measures μ, matrices Q, and functions f by the conventional notation of vector calculus

$$\mu := [\mu(1),\ldots,\mu(d)] \qquad Q := \begin{pmatrix} Q(1,1) & \cdots & Q(1,d) \\ \vdots & \vdots & \vdots \\ Q(d,1) & \cdots & Q_1(d,d) \end{pmatrix} \qquad \mathrm{f} := \begin{pmatrix} f(1) \\ \vdots \\ f(d) \end{pmatrix}$$

In this situation, the formulae (1.1) coincide with the usual matrix operations, with the x-th entry $Q(f)(x)$ of the column vector Qf, and the y-th entry $(\mu Q)(y)$ of the line vector μQ.

Given a sequence of matrices $(Q_n(x,y))_{x,y\in E}$, indexed by the parameter $n \in \mathbb{N}$, we denote by $(Q_1 \ldots Q_n)$ the composition of the matrices Q_p, from $p = 1$ to $p = n$; that is, we have that

$$(Q_1 \ldots Q_n)(x_0,x_n) = \sum_{x_1,\ldots,x_{n-1}\in E} Q_1(x_0,x_1)Q_2(x_1,x_2)\ldots Q_n(x_{n-1},x_n)$$

For time homogeneous matrices $Q_n = Q$, we set $Q^n = (Q_1 \ldots Q_n)$.

A Markov transition is a positive matrix $(M(x,y))_{x,y\in E}$ such that $\sum_y M(x,y) = 1$, for any $x \in E$. These matrices are sometimes called stochastic matrices in the

literature on probability and Markov chains. We say that a measure $\mu(x)$ on E is reversible for a Markov transition $M(x,y)$ if we have for any states $x,y \in E$

$$\mu(x)M(x,y) = \mu(y)M(y,x)$$

We say that a probability measure $\mu(x)$ is invariant for the Markov transition $M(x,y)$ if we have for each $y \in E$

$$\mu(y) = \sum_x \mu(x)M(x,y)$$

Measures, matrices and functions are defined in the same way on more general measurable state spaces E under appropriate well known regularity conditions. We denote respectively by $\mathscr{M}(E)$, and $\mathscr{B}(E)$, the set of all finite signed measures on some measurable space $(E,\mathscr{E})$, and the Banach space of all bounded and measurable functions f equipped with the uniform norm $\|f\|$.

We let $\mu(f) = \int \mu(dx)\, f(x)$, be the Lebesgue integral of a function $f \in \mathscr{B}(E)$, with respect to a measure $\mu \in \mathscr{M}(E)$. We recall that a bounded integral operator M from a measurable space $(E,\mathscr{E})$ into an auxiliary measurable space $(F,\mathscr{F})$ is an operator $f \mapsto M(f)$ from $\mathscr{B}(F)$ into $\mathscr{B}(E)$ such that the functions $x \mapsto M(f)(x) := \int_F M(x,dy)f(y)$ are $\mathscr{E}$-measurable and bounded, for any $f \in \mathscr{B}(F)$. A Markov kernel is a positive and bounded integral operator M with $M(1) = 1$. Given a pair of bounded integral operators (M_1,M_2), we let (M_1M_2) the composition operator defined by $(M_1M_2)(f) = M_1(M_2(f))$. For time homogenous state spaces, we denote by $M^m = M^{m-1}M = MM^{m-1}$ the m-th composition of a given bounded integral operator M, with $m \geq 1$.

We shall slightly abuse the notation and we denote by 0 and 1 the zero and the unit elements in the semi-rings $(\mathbb{R},+,\times)$ and in the set of functions on some state space E. We recall that the gradient ∇f and the Hessian $\nabla^2 f$ of a smooth function $f : \theta = (\theta^i)_{1\leq i\leq d} \in \mathbb{R}^d \mapsto f(\theta) \in \mathbb{R}$ are defined by the functions

$$\nabla f = \left(\frac{\partial f}{\partial \theta^1}, \frac{\partial f}{\partial \theta^2}, \ldots, \frac{\partial f}{\partial \theta^d}\right) \quad \text{and} \quad \nabla^2 f = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 \theta^1} & \frac{\partial^2 f}{\partial \theta^1 \partial \theta^2} & \cdots & \frac{\partial^2 f}{\partial \theta^1 \theta^d} \\ \frac{\partial^2 f}{\partial \theta^2 \partial \theta^1} & \frac{\partial^2 f}{\partial^2 \theta^2} & \cdots & \frac{\partial^2 f}{\partial \theta^2 \theta^d} \\ \vdots & \vdots & \vdots & \\ \frac{\partial^2 f}{\partial \theta^d \partial \theta^1} & \frac{\partial^2 f}{\partial \theta^d \partial \theta^2} & \cdots & \frac{\partial^2 f}{\partial^2 \theta^d} \end{pmatrix}$$

Given a $(d \times d')$ matrix M with random entries $M(i,j)$, we write $\mathbb{E}(M)$ the deterministic matrix with entries $\mathbb{E}(M(i,j))$. We also denote by $(\bullet)_+$, $(\bullet)_-$ and $\lfloor \bullet \rfloor$ respectively the positive, negative and integer part operations. The maximum and minimum operations are denoted respectively by $\vee$ and $\wedge$

$$a \vee b = \max(a,b) \quad \text{and} \quad a \wedge b = \min(a,b)$$

We also use the traditional conventions

$$\left(\sum_{\emptyset}, \prod_{\emptyset}\right) = (0,1) \quad \text{and} \quad \left(\sup_{\emptyset}, \inf_{\emptyset}\right) = (-\infty, +\infty)$$

1.3 Genetic Particle Models

Genetic algorithms are often presented as a random search heuristic that mimics the process of evolution to generate useful solutions to complex optimization problems. The genetic evolution starts with a population of N candidate possible solutions $(\xi_0^1, \ldots, \xi_0^N)$ randomly chosen w.r.t. some distribution $\eta_0(x)$ on some initial finite state space, say E_0, where the coordinates ξ_0^i are also called individuals or genotypes, with $1 \leq N$. In discrete generation models, the genetic evolution is decomposed into two main steps: the selection and the mutation transitions. During the selection-reproduction stage, multiple individuals in the current population $(\xi_n^1, \ldots, \xi_n^N)$ at time $n \in \mathbb{N}$ are stochastically selected based on some problem dependent fitness function G_n that measure the quality of a solution on a given finite solution space E_n. In practice, we choose a random proportion B_n^i of an existing solution ξ_n^i in the current population with a mean value $\propto G_n(\xi_n^i)$ to breed a brand new generation of "improved" solutions $(\widehat{\xi}_n^1, \ldots, \widehat{\xi}_n^N)$. During the mutation step, every selected individual $\widehat{\xi}_n^i$ mutates to a new solution $\xi_{n+1}^i = x$ randomly chosen with a distribution $M_{n+1}(\widehat{\xi}_n^i, x)$ on a possibly different finite solution space E_{n+1}. This generational random process is repeated until some desired termination condition has been reached.

An informal pseudocode description is provided in figure 1.

The question of why these genetic algorithms often succeed at generating high fitness solutions of complex practical problems is not really well understood. Sometimes some researchers say: *"If God uses this natural evolution procedures why I shouldn't use it to solve my problem?"*. More surprisingly, genetic type selection-mutation models are currently used in a variety of application domains, including numerical physics, biology, signal processing, Bayesian statistics, rare event simulation, uncertainty propagation in numerical codes, and many others. In Sequential Monte Carlo literature, the mutation and the selection steps are called the sampling and the resampling transition. In advanced signal processing, particle filters also coincide with these genetic models with mutation-selection stages given by the prediction-updating steps. In Diffusion Monte Carlo methods as well as in Quantum Monte Carlo methods, the mutation and the selection steps are interpreted as the free evolution of walkers and the reconfiguration of the population. In polymer chain simulations, the selection transition is often called pruning. Many other botanical names are given to the selection transition, including cloning, replenish, go with the winner, and many others. For a more thorough discussion on this question with rather detailed bibliographical references, we refer the reader to [52, 38].

One crucial comment is that the size of the population N should be a precision parameter, so that in some sense we solve the problem at hand when N tends to

Algorithm 1.1. Genetic algorithm pseudocode

{Fix some population size (precision of the algorithm) parameter N}
Initialization
ξ_0:= sample N particles, $(\xi_0^i)_{1\leq i\leq N}$ randomly with some given law η_0.
for $k=1$ to n **do**

Selection
for $i=1$ to N **do**
{For each particle}

$$\widehat{\xi}_{k-1}^i := \begin{cases} \xi_{k-1}^i, \text{ with probability } G_{k-1}(\xi_{k-1}^i)/\max_{1\leq j\leq N} G_{k-1}(\xi_{k-1}^j) \quad (1) \\ \tilde{\xi}_{k-1}^i, \textit{ a random variable with law } \sum_{i=1}^N \frac{G_{k-1}(\xi_{k-1}^i)}{\sum_{j=1}^N G_{k-1}(\xi_{k-1}^j)}\, \delta_{\xi_{k-1}^i}, \textit{ otherwise} \quad (2). \end{cases}$$

end for
{We can replace the acceptance probability in the r.h.s. of (1) by the quantity $\varepsilon\, G_{k-1}(\xi_{k-1}^i)$, for any $\varepsilon \geq 0$, such that $\varepsilon\, \max_{1\leq j\leq N} G_{k-1}(\xi_{k-1}^j) \leq 1$. If we choose $\varepsilon = 0$, we simply remove the line (1), so that the selection transition coincides with the proportional/roulette selection}

Transition
for $i=1$ to N **do**
{For each particle}
ξ_k^i:= $F_k(\widehat{\xi}_{k-1}^i, \omega_k^i)$,
{$F_k(.,\omega_k^i)$ designates the perturbation operator generating new candidate solutions. In other words, $\xi_k^i = x$ with probability $M_k(\widehat{\xi}_{k-1}^i, x)$.}
end for
end for

infinity. In other words, when the computational resources $N \to \infty$ the genetic search model should increased its ability to find the desired solution. One way to understand these questions is to analyze the genealogical tree of a given population of individuals. If we interpret the genetic algorithm as a birth and death branching process, then we can trace back in time the whole ancestral line of the individual ξ_n^i at the n-th generation.

$$\xi_{0,n}^i \longleftarrow \xi_{1,n}^i \longleftarrow \ldots \longleftarrow \xi_{n-1,n}^i \longleftarrow \xi_{n,n}^i = \xi_n^i$$

The random state $\xi_{p,n}^i$ represents the ancestor of the individual ξ_n^i at the level p, with $0 \leq p \leq n$, and $1 \leq i \leq N$.

One could expect that this genealogical tree models have different asymptotic behaviors depending on their sampling and on the problem at hand. In fact, in terms of proportions and probability measures we don't have a lot of variability. The random occupation measure of the tree becomes more and more deterministic and we have the following convergence result

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{(\xi^i_{0,n},\xi^i_{1,n},\ldots,\xi^i_{n,n})}(x_0,x_1,\ldots,x_n)=\frac{1}{\mathscr{Z}_n}\left\{\prod_{0\le p<n}G_p(x_p)\right\}\times\mathbb{P}_n(x_0,\ldots,x_n) \tag{1.2}$$

with some normalizing constant $\mathscr{Z}_n$, and the probability distribution of a Markov chain sequence

$$\mathbb{P}_n(x_0,\ldots,x_n):=\eta_0(x_0)M_1(x_0,x_1)\ldots M_n(x_{n-1},x_n)$$

Furthermore, the product of the empirical population mean values of the fitness functions we used in the genetic evolution provides an unbiased estimate of the unknown normalizing constants

$$\mathscr{Z}^N_n:=\prod_{0\le p<n}\frac{1}{N}\sum_{i=1}^{N}G_p(\xi^i_p)\longrightarrow_{N\to\infty}\mathscr{Z}_n$$

These limiting probability measures in the r.h.s. of (1.2) are often called Feynman-Kac measures or Boltzmann-Gibbs distributions in physics and in the applied probability literature. Inversely, suppose that we have to sample from a Feynman-Kac probability measure on some product space and/or we need to compute their normalizing constants. Then, one particle sampling strategy is to run a genetic particle approximation model.

Besides the fact that these rather surprising theoretical results give some insight on the convergence of genetic algorithms and their range of applications, many questions remain to be answered: What is the rate of convergence in the estimates given above? Are they uniform w.r.t. the time parameter? Is it possible to quantify the law of a finite block of individuals? Do we have Central Limit Theorems and exponentially small sub-Gaussian deviation probabilities as in conventional Monte Carlo sampling? What is the interpretation of these limiting probability measures in practical situation and real world concrete problems? How to turn a given complex estimation problem into this probabilistic framework? In this chapter, we provide some answers to these natural questions. In Section 1.4, we provide a brief overview of the connections between abstract positive matrices and genetic type interacting particle models. This section should not be skipped since it contains a series of recipes on matrix models and their particle interpretations to be combined with one another and applied in the application domains discussed in Section 1.5. The displeasure practitioners may get when analyzing these matrix models and their particle interpretations will fade since the genetic type particle approximations of these quantities presented below will provide instantly a collection of powerful simulation tools for the numerical solution of the problem at hand. In Section 1.5, we discuss a series of application domains of genetic particle models, by no means exhaustive. Each application starts with an introduction connecting the results developed in earlier parts with the current description.

1.4 Positive Matrices and Particle Recipes

1.4.1 Positive Matrices and Measures

1.4.1.1 Description of the Models

Let E be a finite set. We consider a collection of matrices $Q_n := (Q_n(x,y))_{x,y\in E}$ with non negative entries $Q_n(x,y) \geq 0$. Given a probability measure η_0 on E, we denote by $\mathbb{Q}_n$ the measure on the product space $E_n := E^{n+1}$ defined for any sequence $(x_p)_{0\leq p\leq n} \in E_n$ of length n by the following formula:

$$\mathbb{Q}_n(x_0,\ldots,x_n) \propto \eta_0(x_0)\; Q_1(x_0,x_1)\; Q_2(x_1,x_2)\; \ldots\; Q_n(x_{n-1},x_n) \tag{1.3}$$

When the matrices $Q_n(x,y)$ are such that $\sum_{y\in E} Q_n(x,y) = 1$, for any $x \in E$, we can interpret $Q_n(x,y)$ as the probability of the transition $X_{n-1} = x \rightsquigarrow X_n = y$ of a given Markov chain X_n. In this situation, we have

$$\mathbb{Q}_n(x_0,\ldots,x_n) = \text{Proba}\,((X_0,\ldots,X_n) = (x_0,\ldots,x_n)) \tag{1.4}$$

Moreover, if we set

$$M_n(x,y) = \text{Proba}\,(X_n = y \mid X_{n-1} = x) \quad \text{and} \quad \eta_0(x) = \text{Proba}\,(X_0 = x)$$

then we find that

$$\begin{aligned}
&\mathbb{Q}_n(x_0,\ldots,x_n)\\
&= \mathbb{P}_n(x_0,\ldots,x_n)\\
&:= \text{Proba}\,((X_0,\ldots,X_n) = (x_0,\ldots,x_n))\\
&= \text{Proba}\,(X_0 = x_0)\,\text{Proba}\,(X_1 = x_1 \mid X_0 = x_0)\ldots\text{Proba}\,(X_n = y \mid X_{n-1} = x)\\
&= \eta_0(x_0)M_1(x_0,x_1)\ldots M_n(x_{n-1},x_n)
\end{aligned}$$

In a variety of applications, we want to approximate the integral type mean values of functions f_n on the product space E_n

$$\mathbb{Q}_n(f_n) = \sum_{x_0,\ldots,x_n} \mathbb{Q}_n(x_0,\ldots,x_n) f_n(x_0,\ldots,x_n) \tag{1.5}$$

as well as their normalizing constants

$$\mathscr{Z}_n := \sum_{x_0,\ldots,x_n} \eta_0(x_0)\; Q_1(x_0,x_1)\; Q_2(x_1,x_2)\; \ldots\; Q_n(x_{n-1},x_n) \tag{1.6}$$

Reducing a bit our initial objective, sometimes we only want to approximate the final time marginals

$$\eta_n(x_n) := \sum_{x_0,\dots,x_{n-1}} \mathbb{Q}_n(x_0,\dots,x_{n-1},x_n) \quad \text{and} \quad \gamma_n(x_n) := \mathscr{Z}_n \times \eta_n(x_n)$$
$$(\Rightarrow \gamma_n(1) = \mathscr{Z}_n) \tag{1.7}$$

1.4.1.2 Path Space Models

From the pure mathematical point of view, for path space models these marginal measure models are equivalent to the model defined in (1.3). More precisely, if we set

$$\forall n \geq 0 \qquad x_n := (x_{0,n}, x_{1,n}, \dots, x_{n-1,n}, x_{n,n}) \in E_n$$

then we find that

$$\begin{aligned}\mathbb{Q}_n(x_n) &= \mathbb{Q}_n(x_{0,n}, x_{1,n}, \dots, x_{n-1,n}, x_{n,n}) \\ &\propto \eta_0(x_{0,n})\, Q_1(x_{0,n}, x_{1,n})\, Q_2(x_{1,n}, x_{2,n}) \,\dots\, Q_n(x_{n-1,n-1}, x_{n,n}) \\ &\propto \sum_{x_{n-1}\in E_{n-1}} \mathbb{Q}_{n-1}(x_{n-1})\, \mathscr{Q}_n(x_{n-1}, x_n)\end{aligned}$$

with the matrices

$$\mathscr{Q}_n(x_{n-1}, x_n) := \mathbb{1}_{x_{n-1}}(x_{0,n}, x_{1,n}, \dots, x_{n-1,n}) \times Q_n(x_{n-1,n}, x_{n,n}) \tag{1.8}$$

This implies that $\mathbb{Q}_n(x_n)$ is the n-th marginal of the extended measure on the product of the path spaces $\prod_{0\leq p\leq n} E_p$ defined by

$$\mathbb{Q}_n^{(\text{path})}(x_0,\dots,x_n) \propto \eta_0(x_0)\mathscr{Q}_1(x_0,x_1)\mathscr{Q}_2(x_1,x_2)\dots\mathscr{Q}_n(x_{n-1},x_n)$$

for any $x_p \in E_p$, with $0 \leq p \leq n$.

In the case of Markov transitions $Q_n = M_n$ discussed in (1.4), we have

$$\begin{aligned}\mathscr{Q}_n(x_{n-1}, x_n) &= \mathscr{M}_n(x_{n-1}, x_n) \\ &:= \mathbb{1}_{x_{n-1}}(x_{0,n}, x_{1,n}, \dots, x_{n-1,n}) \times M_n(x_{n-1,n}, x_{n,n})\end{aligned}$$

In other words, $\mathscr{M}_n(x_{n-1}, x_n)$ is the Markov transition of the historical process

$$\mathscr{X}_n = (X_0, \dots, X_n)$$

of the Markov chain X_n with transitions M_n; that is, we have that

$$\text{Proba}\,(\mathscr{X}_n = x_n \mid \mathscr{X}_{n-1} = x_{n-1})$$

$$\text{Proba}\,((X_0,\ldots,X_n) = (x_{0,n},\ldots,x_{n,n}) \mid (X_0,\ldots,X_{n-1}) = (x_{0,n-1},\ldots,x_{n-1,n-1}))$$

$$= \mathbb{1}_{x_{n-1}}(x_{0,n},x_{1,n},\ldots,x_{n-1,n}) \times \text{Proba}\,(X_n = x_{n,n} \mid X_{n-1} = x_{n-1,n})$$

$$= \mathbb{1}_{x_{n-1}}(x_{0,n},x_{1,n},\ldots,x_{n-1,n}) \times M_n(x_{n-1,n},x_{n,n})$$

Finally, in this situation we observe that

$$\mathbb{Q}_n(x_n) = \mathbb{Q}_n(x_{0,n},x_{1,n},\ldots,x_{n-1,n},x_{n,n}) = \mathbb{P}_n(x_{0,n},x_{1,n},\ldots,x_{n-1,n},x_{n,n}) = \mathbb{P}_n(x_n)$$

with

$$\mathbb{P}_n(x_{0,n},x_{1,n},\ldots,x_{n-1,n},x_{n,n}) = \eta_0(d_{0,n})M_1(x_{0,n},x_{1,n})\ldots M_n(x_{n-1,n},x_{n,n})$$

Another useful state space enlargement allows to work on "transition type" state spaces E^2. These models are defined as follows. For any time $n \geq 0$, we set

$$x_n := (x_{n-1,n},x_{n,n}) \in E^2 \quad \text{and} \quad \mathscr{Q}_n(x_{n-1},x_n) := \mathbb{1}_{x_{n-1,n-1}}(x_{n-1,n})\; Q_n(x_{n-1,n},x_{n,n})$$

We also use the convention $\eta_0(x_0) = \eta_0(x_{-1,0},x_{0,0}) = \eta_0(x_{0,0})$, for $n = 0$. In this notation, for any sequence x_n with $x_{n-1,n} = x_{n-1,n-1}$, we have

$$\eta_0(x_0)\mathscr{Q}_1(x_0,x_1)\mathscr{Q}_2(x_1,x_2)\ldots\mathscr{Q}_n(x_{n-1},x_n)$$

$$\propto \eta_0(x_{0,0})Q_1(x_{0,0},x_{1,1})Q_2(x_{1,1},x_{2,2})\ldots Q_n(x_{n-1,n-1},x_{n,n})$$

1.4.2 Interacting Particle Models

1.4.2.1 Genetic Population Evolution

This section is concerned with the design of a genetic particle approximation of the measures η_n introduced in (1.7). One universal way to associate a genetic population evolution model to positive matrices $Q_n(x_{n-1},x_n)$ is to use the decomposition

$$Q_n(x_{n-1},x_n) = G_{n-1}(x_{n-1}) \times M_n(x_{n-1},x_n) \tag{1.9}$$

with the Markov transition M_n and the potential function G_{n-1}

$$M_n(x_{n-1},x_n) := \frac{Q_n(x_{n-1},x_n)}{\sum_{x_n} Q_n(x_{n-1},x_n)} \quad \text{and} \quad G_{n-1}(x_{n-1}) = \sum_{x_n} Q_n(x_{n-1},x_n) \tag{1.10}$$

We let X_n be a Markov chain with initial distribution η_0 and Markov transitions M_n. By the definition of the measure $\mathbb{Q}_n$, the integral type formula (1.5) has the

following probabilistic interpretation

$$\mathbb{Q}_n(f_n) \propto \sum_{x_1,\ldots,x_n} f_n(x_0,\ldots,x_n)\left\{\prod_{0\leq p<n} G_p(x_p)\right\}\times\left\{\eta_0(x_0)\prod_{1\leq p\leq n} M_p(x_{p-1},x_p)\right\}$$
$$\propto \mathbb{E}\left[f_n(X_0,\ldots,X_n)\left\{\prod_{0\leq p<n} G_p(X_p)\right\}\right] \tag{1.11}$$

with the normalizing constant

$$\mathscr{Z}_n := \mathbb{E}\left[\prod_{0\leq p<n} G_p(X_p)\right]$$

For unit potential functions $G_n(x)=1$, the model resumes to the Markov transitions model $Q_n = M_n$ discussed in (1.4). In this context, we clearly have that $\mathscr{Z}_n = 1$ and

$$\mathbb{Q}_n(f_n) = \mathbb{P}_n(f_n) = \mathbb{E}\left[f_n(X_0,\ldots,X_n)\right]$$

In this situation, we can approximate these expectation by sampling N independent copies X_n^i of the Markov chain X_n and using the traditional Monte Carlo empirical estimates

$$\mathbb{P}_n^N(f_n) := \frac{1}{N}\sum_{i=1}^{N} f_n(X_0^i,\ldots,X_n^i)$$

In more general situations, the potential functions G_n may change radically the probability mass distributions of the measures $\mathbb{Q}_n$. For instance, for indicator functions $G_n = 1_A$, we have

$$\mathscr{Z}_n := \mathbb{E}\left[\prod_{0\leq p<n} G_p(X_p)\right] = \text{Proba}\left(X_p\in A,\ \forall 0\leq p<n\right)$$

and

$$\mathbb{Q}_n = \text{Law}\left((X_0,\ldots,X_n)\mid X_p\in A,\ \forall 0\leq p<n\right)$$

as soon as $\mathscr{Z}_n > 0$. In this situation, the probability measure $\mathbb{Q}_n$ only charges random trajectories that remains in the set A, for any time $0\leq p<n$. This situation is discussed in some details in section 1.5.1.1 dedicated to particle absorption models. We already mention that in this case we often have

$$\text{Proba}\left(X_p\in A,\ \forall 0\leq p<n\right)\to_{n\uparrow\infty} 0$$

so that it becomes more and more unlikely that a random sample copy of X_n remains in the set A for all times $0\leq p<n$ during a large horizon n. when all the N independent copies X_n^i have left the desired A, we have

$$\frac{1}{N}\sum_{i=1}^{N} f_n(X_0^i,\ldots,X_n^i)\prod_{0\leq p<n} G_p(X_p^i)=0$$

To avoid this "technical" problem, we use a genetic type acceptance rejection sampling scheme. To avoid some unnecessary technical discussion, we further assume that the functions G_n take values in $]0,1]$. As we shall see, this condition can be removed. Practically, we approximate the desired target measures $\eta_n(x_n)$ by a large cloud of N random samples also termed particles ξ_n^i, with $1\leq i\leq N$; that is, at any time $n\geq 0$, we have that

$$\eta_n^N(x):=\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{\xi_n^i}(x)\longrightarrow_{N\to\infty}\eta_n(x) \tag{1.12}$$

The particle system $\xi_n:=\left(\xi_n^i\right)_{1\leq i\leq N}$ is a simple genetic sampling model combing a mutation transition and a selection transition

$$\xi_n:=(\xi_n^i)_{1\leq i\leq N}\xrightarrow{\quad selection\quad}\widehat{\xi}_n:=(\widehat{\xi}_n^i)_{1\leq i\leq N}\xrightarrow{\quad mutation\quad}\xi_{n+1} \tag{1.13}$$

During the mutation transition, every individual performs a local random move according to the Markov transition M_n. During the selection step, every individual evaluates its potential value $G_n(\xi_n^i)$, with $1\leq i\leq N$. For every index i, with a probability $G_n(\xi_n^i)$, we set $\widehat{\xi}_n^i=\xi_n^i$, otherwise we replace ξ_n^i by a fresh new individual $\widehat{\xi}_n^i=\xi_n^j$ randomly chosen in the whole population with a probability $\propto G_n(\xi_n^j)$.

For more general potential functions, we can replace the acceptance rate $G_n(\xi_n^i)$ of any individual ξ_n^i by any acceptance rate of the form $\varepsilon_n(\xi_n)G_n(\xi_n^i)$ with some $\varepsilon_n(\xi_n)\geq 0$ that may depend on the whole population and s.t. $\varepsilon_n(\xi_n)G_n(\xi_n^i)\in[0,1]$. For instance, we can chose $\varepsilon_n=0$ or $\varepsilon_n(\xi_n)=1/\max_{1\leq i\leq N}G_n(\xi_n^i)$. In the first case, the selection transition is often called the "proportional selection". In the second case, we notice that the best fit individuals are always accepted. In all cases, the acceptance probability of an individual ξ_n^i is proportional to $G_n(\xi_n^i)$. In the further development of this chapter, we write that the acceptance probability is $\propto G_n(\xi_n^i)$. We end this section with a second important remark:

Let us suppose that Q_n has the following form

$$Q_n(x_{n-1},x_n)=K_n(x_{n-1},x_n)\times H_n(x_n)$$

for some Markov transition K_n and some potential function H_n. In this situation, the decomposition (1.10) is met with

$$M_n(x_{n-1},x_n):=\frac{K_n(x_{n-1},x_n)H_n(x_n)}{\sum_{x_n}K_n(x_{n-1},x_n)H_n(x_n)}\ \text{ and }\ G_{n-1}(x_{n-1})=\sum_{x_n}K_n(x_{n-1},x_n)H_n(x_n)$$

In practice, the numerical solving of the sum in the r.h.s. equation and the computational cost of sampling a mutation $x_{n-1}\rightsquigarrow x_n$ with the transition M_n can be prohibitive. One way to solve these problems is to sample a sequence of transitions

$\xi_{n-1}^i \rightsquigarrow \zeta_n^{i,j}$, with $1 \leq j \leq N'$ using the Markov transition K_n so that for any $1 \leq i \leq N$ we have

$$K_n^{N'}(\xi_{n-1}^i, x_n) = \frac{1}{N'} \sum_{i=1}^{N'} \mathbb{1}_{\zeta_n^{i,j}}(x_n) \simeq_{N' \uparrow \infty} K_n(\xi_{n-1}^i, x_n)$$

We let $M_n^{N'}$ and $G_{n-1}^{N'}$ be the Markov transitions and the potential functions defined as M_n and G_{n-1}, replacing K_n by $K_n^{N'}$ in the above equations; that is, we have

$$M_n^{N'}(\xi_{n-1}^i, x_n) := \sum_{j=1}^{N'} \frac{H_n(\zeta_n^{i,j})}{\sum_{k=1}^{N'} H_n(\zeta_n^{i,k})} \mathbb{1}_{\zeta_n^{i,j}}(x_n) \quad \text{and} \quad G_{n-1}^{N'}(\xi_{n-1}^i) = \frac{1}{N'} \sum_{j=1}^{N'} H_n(\zeta_n^{i,j})$$

Another strategy is to observe that the measures (1.3) are given by

$$\mathbb{Q}_n(x_0, \ldots, x_n) \propto \left[\eta_0(x_0)\, Q_1'(x_0, x_1)\, Q_2'(x_1, x_2)\, \ldots\, Q_n'(x_{n-1}, x_n)\right]\, H_n(x_n)$$

with

$$Q_n'(x_{n-1}, x_n) = H_{n-1}(x_{n-1})\, K_n(x_{n-1}, x_n)$$

and the convention $H_0 = 1$, for $n = 1$. In this interpretation, we approximate the measures

$$\mathbb{Q}_n'(x_0, \ldots, x_n) \propto \eta_0(x_0)\, Q_1'(x_0, x_1)\, Q_2'(x_1, x_2)\, \ldots\, Q_n'(x_{n-1}, x_n)$$

using the particle occupation measures defined above (with mutation transitions K_n and selection fitness functions H_n) weighted by the function $H_n(x_n)$

$$\sum_{i=1}^{N} \frac{H_n(\xi_n^i)}{\sum_{j=1}^{N} H_n(\xi_n^j)} \mathbb{1}_{\xi_n^i}(x) \longrightarrow_{N \to \infty} \eta_n(x)$$

1.4.2.2 Particle Normalizing Constants

In this section, we present an unbiased particle approximation of the normalizing constants $\mathscr{Z}_n$ introduced in (1.6). Using the decomposition

$$\mathbb{Q}_n(x_0, \ldots, x_n) = \frac{\mathscr{Z}_{n-1}}{\mathscr{Z}_n}\, \mathbb{Q}_{n-1}(x_0, \ldots, x_{n-1})\, Q_n(x_{n-1}, x_n)$$

we find the following matrix formulae

$$\eta_n = \frac{\mathscr{Z}_{n-1}}{\mathscr{Z}_n}\, \eta_{n-1} Q_n \implies \frac{\mathscr{Z}_n}{\mathscr{Z}_{n-1}} = \eta_{n-1} Q_n(1) = \eta_{n-1}(G_{n-1}) \quad \text{and} \quad \gamma_n = \gamma_{n-1} Q_n$$

Now, it is also easily checked that

$$\gamma_n(1) = \gamma_{n-1} Q_n(1) = \gamma_{n-1}(1)\, \eta_{n-1} Q_n(1) = \cdots = \prod_{0 \leq p < n} \eta_p Q_{p+1}(1) = \prod_{0 \leq p < n} \eta_p(G_p)$$

Mimicking the r.h.s. multiplicative formula, an N-particle approximation of the normalizing constants $\mathscr{Z}_n$ is given by the following unbiased estimates

$$\mathscr{Z}_n^N := \prod_{0\leq p<n} \eta_p^N(G_p) \longrightarrow_{N\to\infty} \mathscr{Z}_n = \prod_{0\leq p<n} \eta_p(G_p)$$

and for any $x \in E$

$$\gamma_n^N(x) := \mathscr{Z}_n^N \ \eta_n^N(x) \longrightarrow_{N\to\infty} \gamma_n(x) := \mathscr{Z}_n \ \eta_n(x) \tag{1.14}$$

Furthermore, the particle estimate $\gamma_n^N(x)$ is unbiased in the sense that for any $x \in E$, and for any $n \geq 0$, we have

$$\mathbb{E}\left(\gamma_n^N(x)\right) = \gamma_n(x)$$

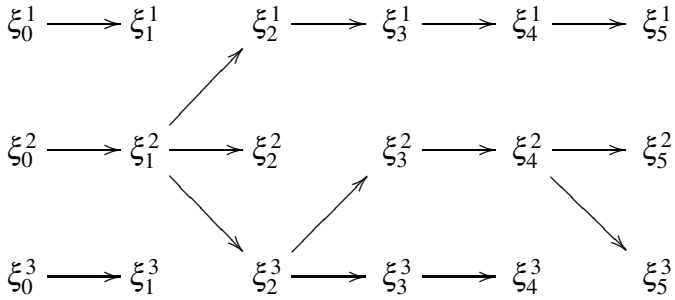
The proof of this property is not so obvious. To our knowledge, in the context of nonlinear filtering this property has first been proved in [36]. See also [52] and [38] for more general models.

1.4.3 Genealogical and Ancestral Structures

1.4.3.1 Genealogical Trees

The aim of this section is to use the genealogical tree structure of the genetic population model defined in the previous sections to approximate the measures $\mathbb{Q}_n$ defined in (1.3).

Running back in time, we can trace back the complete ancestral lines of the individuals and the time evolution of the genealogical tree model associated with the genetic algorithm described above. For instance, a realization of that tree for $N = 3$ individuals and $n = 4$ iterations is given by



One way to encode the ancestral line of a current individual, say ξ_5^2, is to write $\xi_{p,5}^2$ with the level index $0 \leq p \leq 5$ of the ancestor; with the convention $\xi_{5,5}^2 = \xi_5^2$, for $p = 5$. In this notation, we obtain $N = 3$ ancestral lines, and the line associated with the i-th current individual ξ_n^i is the random vector in the product space E_n given by

$$\left(\xi_{0,n}^i, \xi_{1,n}^i, \xi_{2,n}^i, \ldots, \xi_{n-1,n}^i, \xi_{n,n}^i\right)$$

An N-particle approximation of $\mathbb{Q}_n$ is given by the occupation measure of these ancestral lines

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{(\xi^i_{0,n},\xi^i_{1,n},\ldots,\xi^i_{n,n})}(x_0,\ldots,x_n)\longrightarrow_{N\to\infty}\mathbb{Q}_n(x_0,\ldots,x_n) \tag{1.15}$$

Furthermore, the evolution model of the genealogical N ancestral lines

$$\xi^i_n:=(\xi^i_{0,n},\xi^i_{1,n},\ldots,\xi^i_{n,n})$$

with $1\leq i\leq N$, coincides with the genetic model defined in (1.13) on product state spaces $E_n=E_n$ with matrices $\mathscr{Q}_n(x_{n-1},x_n)$ defined in (1.8). In the path space notation used in (1.8), the convergence result (1.15) takes the following form

$$\frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{\xi^i_n}(x_n)\longrightarrow_{N\to\infty}\mathbb{Q}_n(x_n)$$

for any $x_n:=(x_{0,n},x_{1,n},\ldots,x_{n-1,n},x_{n,n})\in E_n$; and this property coincides with (1.12).

Once again, this state space enlargement property is not really obvious. To our knowledge, this property has first been proved in [53].

1.4.3.2 Complete Ancestral Trees

To simplify the presentation, sometimes we denote by Ξ_n the sequence of complete genealogical trees $\xi_p:=(\xi^i_p)_{1\leq i\leq N}$, from the origin $p=0$, up to time $p=n$; that is, we set

$$\Xi_n=(\Xi_{0,n},\Xi_{1,n},\ldots,\Xi_{n,n}):=(\xi_0,\ldots,\xi_n)\in\prod_{0\leq p\leq n}E_p^N$$

with

$$\Xi_{p,n}=(\Xi^i_{p,n})_{1\leq i\leq N}=(\xi^i_p)_{1\leq i\leq N}\quad\text{and}\quad\Xi^i_{p,n}:=\xi^i_p:=\left(\xi^i_{0,p},\xi^i_{1,p},\ldots,\xi^i_{p,p}\right)$$

for any $0\leq p\leq n$.

Combining this observation with the exchangeability of the particle system and with the unbiased property of the unnormalized measures (1.14) discussed above, we conclude that for any function f_n on the product space E_n

$$\mathbb{E}\left(\eta^N_n(f_n)\,\overline{\mathscr{Z}}^N_n\right)=\mathbb{E}\left(f_n\left(\xi^1_n\right)\,\overline{\mathscr{Z}}^N_n\right)=\mathbb{Q}_n(f_n)\quad\text{with}\quad\overline{\mathscr{Z}}^N_n:=\mathscr{Z}^N_n/\mathscr{Z}_n$$

and with the first ancestral line $\xi^1_n=\left(\xi^1_{0,n},\xi^1_{1,n},\ldots,\xi^1_{n,n}\right)$ of the genealogical tree. In other words, we have that

$$\mathbb{E}\left(\overline{\mathscr{Z}}^N_n\mid\xi^1_n=x_n\right)\times\text{Proba}(\xi^1_n=x_n)=\mathbb{Q}_n(x_n)$$

for any $x_n := (x_{0,n}, x_{1,n}, \ldots, x_{n,n}) \in E_n$. In addition, the measure defined by

$$\mathbb{T}_n^N(F_n) := \mathbb{E}\left(F_n(\Xi_n)\ \overline{\mathscr{Z}}_n^N\right) \tag{1.16}$$

for any function F_n on the product space $\prod_{0\leq p\leq n} E_p^N$, is a probability measure whose ξ_n^i-marginals on E_n coincide with $\mathbb{Q}_n$, for any $1 \leq i \leq N$.

1.4.4 Complete Genealogical Tree Model

The aim of this section is to use the genealogical tree structure of the genetic population model defined in the previous sections to approximate the measures $\mathbb{Q}_n$ defined in (1.3).

1.4.4.1 Ancestral Lines Occupation Measures

The complete ancestral tree of the genetic model is the set of all the populations of individuals ξ_p, from the origin $p = 0$, up to the final time horizon $p = n$. Notice that these systems contain all the information about the genetic evolution, including the ancestral lines that have disappeared. A basic convergence estimate is the following

$$\frac{1}{N}\sum_{i=1}^N \mathbb{1}_{(\xi_0^i,\xi_1^i,\ldots,\xi_n^i)}(x_0,\ldots,x_n)$$

$$\longrightarrow_{N\to\infty} \eta_0(x_0)\times K_{1,\eta_0}(x_0,x_1)\times K_{2,\eta_1}(x_1,x_2)\times\cdots\times K_{n,\eta_{n-1}}(x_{n-1},x_n)$$

with the stochastic matrices

$$K_{n,\eta_{n-1}}(x,y) = G_{n-1}(x)\ M_n(x,y) + (1-G_{n-1}(x))\ \sum_z \frac{\eta_{n-1}(z)G_{n-1}(z)}{\eta_{n-1}(G_{n-1})}\ M_n(z,y)$$

It is instructive to notice that the selection-mutation Markov transition $\xi_{n-1} \rightsquigarrow \xi_n$ defined in (1.13) is given by

$$\text{Proba}\left(\xi_n = (x^1,\ldots,x^N) \mid \xi_{n-1}\right) := \prod_{1\leq i\leq N} K_{n,\eta_{n-1}^N}(\xi_{n-1}^i, x^i)$$

We recall that $\eta_{n-1}^N(x) := \frac{1}{N}\sum_{i=1}^N \mathbb{1}_{\xi_{n-1}^i}(x)$ stands for the occupation measures of the genetic population at time $(n-1)$ so that the probability $K_{n,\eta_{n-1}^N}(\xi_{n-1}^i, x^i)$ that an individual $\xi_n^i = x^i$ is given by

$$G_{n-1}(\xi_{n-1}^i)\ M_n(\xi_{n-1}^i, x^i) + (1-G_{n-1}(\xi_{n-1}^i))\ \sum_{j=1}^N \frac{G_{n-1}(\xi_{n-1}^j)}{\sum_{k=1}^N G_{n-1}(\xi_{n-1}^k)}\ M_n(\xi_{n-1}^j, x^i)$$

Without altering the above convergence results, we can replace the fitness function G_n in the selection transition of the genetic model defined in section 1.4.2 by any function of the form $\varepsilon_n G_n$, for any constant $\varepsilon_n \geq 0$ s.t. $\varepsilon_n G_n \in [0,1]$. The selection transition associated with the choice $\varepsilon = 0$ is often called a simple selection, a proportional selection, as well as a multinomial branching model. In this situation, we have the following convergence result

$$\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{(\xi_0^i,\xi_1^i,\ldots,\xi_n^i)}(x_0,\ldots,x_n) \longrightarrow_{N\to\infty} \eta_0(x_0)\times\eta_1(x_1)\times\cdots\times\eta_n(x_n) \tag{1.17}$$

1.4.4.2 Backward Markov Chain Model

Using the matrix formulae given above, we observe that

$$\frac{\eta_{n-1}Q_n(x_n)}{\eta_n(x_n)}\times\frac{\eta_{n-2}Q_{n-1}(x_{n-1})}{\eta_{n-1}(x_{n-1})}\times\cdots\times\frac{\eta_0 Q_1(x_1)}{\eta_1(x_1)} = \frac{\mathscr{Z}_n}{\mathscr{Z}_{n-1}}\times\frac{\mathscr{Z}_{n-1}}{\mathscr{Z}_{n-2}}\times\cdots\times\frac{\mathscr{Z}_1}{\mathscr{Z}_0} = \mathscr{Z}_n$$

From this observation, we readily prove the following backward representation of $\mathbb{Q}_n$

$$\begin{aligned}
&\mathbb{Q}_n(x_0,\ldots,x_n)\\
&= \eta_n(x_n)\times\frac{\eta_{n-1}(x_{n-1})Q_n(x_{n-1},x_n)}{\eta_{n-1}Q_n(x_n)}\ \cdots\ \frac{\eta_1(x_1)Q_2(x_1,x_2)}{\eta_1 Q_2(x_2)}\times\frac{\eta_0(x_0)Q_1(x_0,x_1)}{\eta_0 Q_1(x_1)}\\
&:= \eta_n(x_n)\times\ Q^\star_{n,\eta_{n-1}}(x_n,x_{n-1})\ \cdots\ Q^\star_{2,\eta_1}(x_2,x_1)\times Q^\star_{1,\eta_0}(x_1,x_0)
\end{aligned} \tag{1.18}$$

with the time reversal Markov matrices $Q^\star_{n,\eta_{n-1}}(x_n,x_{n-1})$ defined below

$$Q^\star_{n,\eta_{n-1}}(x_n,x_{n-1}) = \frac{\eta_{n-1}(x_{n-1})Q_n(x_{n-1},x_n)}{\eta_{n-1}Q_n(x_n)}$$

Mimicking formula (1.18) an alternative particle approximation of the measures $\mathbb{Q}_n$ by the following estimate

$$\begin{aligned}
\mathbb{Q}^N_n(x_0,\ldots,x_n) &= \eta^N_n(x_n)\times Q^\star_{n,\eta^N_{n-1}}(x_n,x_{n-1})\cdots Q^\star_{2,\eta^N_1}(x_2,x_1)\times Q^\star_{1,\eta^N_0}(x_1,x_0)\\
&\to_{N\uparrow\infty} \mathbb{Q}_n(x_0,\ldots,x_n)
\end{aligned} \tag{1.19}$$

with the time reversal random matrices $Q^\star_{n,\eta^N_{n-1}}(x_n,x_{n-1})$ defined by

$$Q^\star_{n,\eta^N_{n-1}}(x_n,x_{n-1}) = \frac{\eta^N_{n-1}(x_{n-1})Q_n(x_{n-1},x_n)}{\eta^N_{n-1}Q_n(x_n)} = \sum_{i=1}^{N}\frac{Q_n(\xi^i_{n-1},x_n)}{\sum_{j=1}^{N}Q_n(\xi^j_{n-1},x_n)}\ \mathbb{1}_{\xi^i_{n-1}}(x_{n-1})$$

Once again, for any function f_n on the product space E_n, we have the following unbiased property

$$\mathbb{E}\left(\overline{\mathscr{Z}}_n^N\ \mathbb{Q}_n^N(f_n)\right) = \mathbb{E}\left(\overline{\mathscr{Z}}_n^N\ f_n(\zeta_n)\right) = \mathbb{Q}_n(f_n)$$

where $\zeta_n := (\zeta_{n,n}, \zeta_{n-1,n}, \ldots, \zeta_{1,n}, \zeta_{0,n})$ stands for a backward Markov chain (as well as a complete ancestral line) with distribution $\mathbb{Q}_n^N$. In other words, if we set

$$x_n := (x_{0,n}, x_{1,n}, \ldots, x_{n-1,n}, x_{n,n}) \in E_n \quad \text{with} \quad \overleftarrow{x}_n = (x_{n,n}, x_{n-1,n}, \ldots, x_{1,n}, x_{0,n})$$

then we have that

$$\mathbb{E}\left(\overline{\mathscr{Z}}_n^N \mid \zeta_n = \overleftarrow{x}_n\right) \times \text{Proba}(\zeta_n = \overleftarrow{x}_n) = \mathbb{Q}_n(x_n)$$

In addition, the measures $\mathbb{A}_n^N$ defined by

$$\mathbb{A}_n^N(F_n) := \mathbb{E}\left(F_n(\Xi_n, \zeta_n)\ \overline{\mathscr{Z}}_n^N\right)$$

for any function F_n on the product space $\left\{\left[\prod_{0\le p\le n} E_p^N\right] \times E_n\right\}$, is a probability measure whose ζ_n-marginals on E_n coincide with the measure $\mathbb{Q}_n$, for any $1 \le i \le N$.

1.4.5 Particle Derivation and Conditioning Principles

1.4.5.1 Particle Derivation Models

Besides the fact that the computation of $\mathbb{Q}_n$ and $\mathscr{Z}_n$ comes from a specific estimation problem, in some application areas, the distribution $\eta_0^{(\theta)}$, as well as the matrices $Q_n^{(\theta)}(x,y)$ also depend on some parameter $\theta \in \mathbb{R}^d$, and we want to estimate the gradient $\nabla\mathbb{Q}_n^{(\theta)}$ and $\nabla\mathscr{Z}_n^{(\theta)}$ of the corresponding functions $\theta \mapsto \mathbb{Q}_n^{(\theta)}$ and $\theta \mapsto \mathscr{Z}_n^{(\theta)}$. The computation of these quantities is again connected to the integral type computation w.r.t. the measure $\mathbb{Q}_n^{(\theta)}$, with the following easy to check formulae

$$\nabla \log \mathscr{Z}_n^{(\theta)} = \mathbb{Q}_n^{(\theta)}(\Lambda_n^{(\theta)}) \quad \text{and} \quad \nabla \log \mathbb{Q}_n^{(\theta)} = \Lambda_n^{(\theta)} - \mathbb{Q}_n^{(\theta)}(\Lambda_n^{(\theta)}) \tag{1.20}$$

with the gradient

$$\Lambda_n^{(\theta)} := \nabla\mathbb{L}_n^{(\theta)} \ \text{ of the additive functional } \ \mathbb{L}_n^{(\theta)}(x_0, \ldots, x_n) := \sum_{p=0}^{n} \log Q_p^{(\theta)}(x_{p-1}, x_p)$$

In the above display, we have used the convention $Q_0(x_{-1}, x_0) = \eta_0(x_0)$, for $p = 0$. We also have the correlation representation of the Hessian functions

$$\nabla^2 \log \mathscr{Z}_n^{(\theta)} = \mathbb{Q}_n^{(\theta)} \left[\left(\Lambda_n^{(\theta)} - \mathbb{Q}_n^{(\theta)}(\Lambda_n^{(\theta)}) \right)' \left(\Lambda_n^{(\theta)} - \mathbb{Q}_n^{(\theta)}(\Lambda_n^{(\theta)}) \right) \right] \tag{1.21}$$

and for any f_n on E_n

$$\begin{aligned} \nabla^2 \mathbb{Q}_n^{(\theta)}(f_n) = \mathbb{Q}_n^{(\theta)} & \left[f_n \left(\Lambda_n^{(\theta)} - \mathbb{Q}_n^{(\theta)}(\Lambda_n^{(\theta)}) \right)' \left(\Lambda_n^{(\theta)} - \mathbb{Q}_n^{(\theta)}(\Lambda_n^{(\theta)}) \right) \right] \\ - \mathbb{Q}_n^{(\theta)} & \left[f_n \left(\nabla^2 \mathbb{L}_n^{(\theta)} - \mathbb{Q}_n^{(\theta)}(\nabla^2 \mathbb{L}_n^{(\theta)}) \right) \right] \end{aligned} \tag{1.22}$$

The physical interpretations or the engineering meaning of these rather abstract mathematical objects depend on the application domain they are thought for.

Next, we design particle approximations of these derivative models. We denote by $\eta_n^{(\theta,N)}$ the occupation measures associated with a genetic particle model with mutation transitions and fitness potential function defined by

$$M_n^{(\theta)}(x,y) := Q_n^{(\theta)}(x,y) / \sum_z Q_n^{(\theta)}(x,z) \quad \text{and} \quad G_n^{\theta}(x) := \sum_z Q_{n+1}^{(\theta)}(x,z)$$

We also denote by $\mathbb{Q}_n^{(\theta,N)}$ the random measures on path space defined by

$$\begin{aligned} \mathbb{Q}_n^{(\theta,N)}(x_0,\ldots,x_n) := \eta_n^{(\theta,N)}(x_n) \times & Q^{\star}_{n,\eta_{n-1}^{(\theta,N)}}(x_n,x_{n-1}) \cdots Q^{\star}_{2,\eta_1^{(\theta,N)}}(x_2,x_1) \\ & \times Q^{\star}_{1,\eta_0^{(\theta,N)}}(x_1,x_0) \end{aligned} \tag{1.23}$$

and the corresponding particle normalizing constants

$$\mathscr{Z}_n^{(\theta,N)} := \prod_{0 \le p < n} \eta_p^{(\theta,N)} \left(G_p^{\theta} \right)$$

Mimicking the derivation formulae (1.20) we define the particle derivation of the logarithm of the normalizing constants by

$$\nabla_N \log \mathscr{Z}_n^{(\theta)} := \mathbb{Q}_n^{(\theta,N)}(\Lambda_n^{(\theta)}) \longrightarrow_{N \uparrow \infty} \nabla \log \mathscr{Z}_n^{(\theta)}$$

In the same vein, the particle derivation of the measure $\mathbb{Q}_n^{(\theta)}$ is defined by the following correlation formulae

$$\nabla_N \mathbb{Q}_n^{(\theta)}(f_n) := \mathbb{Q}^{(\theta,N)} \left(f_n \left[\Lambda_n^{(\theta)} - \mathbb{Q}_n^{(N,\theta)}(\Lambda_n^{(\theta)}) \right] \right) \longrightarrow_{N \uparrow \infty} \nabla \mathbb{Q}_n^{(\theta)}(f_n)$$

for any function f_n on E_n with the additive functional

$$\Lambda_n^{(\theta)}(x_0,\ldots,x_n) := \sum_{p=0}^{n} \nabla \log Q_p^{(\theta)}(x_{p-1},x_p)$$

and the convention $Q_0(x_{-1},x_0) = \eta_0(x_0)$ for $p = 0$.

Analogously, we define the particle Hessian functions $\nabla_N^2 \log \mathscr{Z}_n^{(\theta)}$ and $\nabla_N^2 \mathbb{Q}_n^{(\theta)}(f_n)$ replacing in (1.21) and (1.22) the measures $\mathbb{Q}_n^{(\theta)}$ by $\mathbb{Q}_n^{(\theta,N)}$.

1.4.5.2 Particle Conditioning Models

This section is concerned with some conditional distributions of the measure $\mathbb{Q}_n$ and their particle approximations. First, we observe that

$$\begin{aligned}\mathbb{Q}_p^{(n)}(x_0,\ldots,x_p) &:= \sum_{x_{p+1},\ldots,x_n} \mathbb{Q}_n(x_0,\ldots,x_p,x_{p+1},\ldots,x_n) \\ &= \eta_{p|n}(x_p)\, Q^\star_{p,\eta_{p-1}}(x_p,x_{p-1})\cdots Q^\star_{1,\eta_0}(x_1,x_0) \qquad (1.24)\end{aligned}$$

with the p-th marginal $\eta_{p|n}$ of the measure $\mathbb{Q}_n$ defined by the matrix formula

$$\eta_{p|n} := \eta_n Q^\star_{n,\eta_{n-1}} \cdots Q^\star_{p+1,\eta_p}$$

This clearly implies that

$$\begin{aligned}&\mathbb{Q}_{n|p}((x_{p+1},\ldots,x_n) \mid (x_0,\ldots,x_p)) \\ &:= \mathbb{Q}_n(x_0,\ldots,x_p,x_{p+1},\ldots,x_n)/\mathbb{Q}_p^{(n)}(x_0,\ldots,x_p) \\ &= \frac{1}{\eta_{p|n}(x_p)}\, \eta_n(x_n) Q^\star_{n,\eta_{n-1}}(x_n,x_{n-1})\cdots Q^\star_{p+1,\eta_p}(x_{p+1},x_p)\end{aligned}$$

The distributions $\eta_{p|n}(x_p)$ and $\mathbb{Q}_{n|p}((x_{p+1},\ldots,x_n) \mid (x_0,\ldots,x_p) = \mathbb{Q}_{n|p}((x_{p+1},\ldots, x_n) \mid x_p)$ can be approximated using the complete ancestral tree and replacing in the above formulae the measures η_p by their particle approximations η_p^N, with $0 \le p \le n$:

$$\eta_{p|n}^N := \eta_n^N Q^\star_{n,\eta_{n-1}^N} \cdots Q^\star_{p+1,\eta_p^N} \qquad (1.25)$$

and

$$\mathbb{Q}_{n|p}^N((x_{p+1},\ldots,x_n) \mid x_p) = \frac{1}{\eta_{p|n}^N(x_p)}\, \eta_n^N(x_n) Q^\star_{n,\eta_{n-1}^N}(x_n,x_{n-1})\cdots Q^\star_{p+1,\eta_p^N}(x_{p+1},x_p)$$

We end this section with a probabilistic interpretation of these mathematical objects. First, using (1.9) we find the following decomposition

$$\mathbb{Q}_n(x_0,\ldots,x_n) = \frac{1}{\mathscr{Z}_n} \left\{ \prod_{0\le p<n} G_p(x_p) \right\} \mathbb{P}_n(x_0,\ldots,x_n)$$

with the distribution $\mathbb{P}_n(x_0,\ldots,x_n)$ of the Markov chain sequence with initial condition $\mathrm{Law}(X_0) = \eta_0$ and Markov transitions M_n

$$\begin{aligned}\mathbb{P}_n(x_0,\ldots,x_n) &= \text{Proba}\left((X_0,\ldots,X_n)=(x_0,\ldots,x_n)\right)\\ &= \eta_0(x_0)M_1(x_0,x_1)M_2(x_1,x_2)\cdots M_n(x_{n-1},x_n)\end{aligned}$$

In this interpretation, we have

$$\begin{aligned}\mathbb{Q}_p^{(n)}(x_0,\ldots,x_p) &:= \frac{\mathscr{Z}_{n|p}(x_p)}{\mathscr{Z}_n}\left\{\prod_{0\le q<p} G_q(x_q)\right\}\mathbb{P}_p(x_0,\ldots,x_p)\\ &= \frac{\mathscr{Z}_{n|p}(x_p)}{\mathscr{Z}_{n|p}}\,\mathbb{Q}_p(x_0,\ldots,x_p)\\ &= \frac{\mathscr{Z}_{n|p}(x_p)}{\mathscr{Z}_{n|p}}\,\eta_p(x_p)\,\left[Q^\star_{p,\eta_{p-1}}(x_p,x_{p-1})\cdots Q^\star_{1,\eta_0}(x_1,x_0)\right]\end{aligned}$$

with $\mathscr{Z}_{n|p}$ and $\mathscr{Z}_{n|p}(x_p)$ defined by

$$\mathscr{Z}_{n|p}=\mathscr{Z}_n/\mathscr{Z}_p=\prod_{p\le q<n}\eta_q(G_q)\ \text{ and }\ \mathscr{Z}_{n|p}(x_p):=\mathbb{E}\left(\left\{\prod_{p\le q<n}G_q(x_q)\right\}|X_p=x_p\right)$$

Using (1.24), this implies that

$$\eta_{p|n}(x_p)=\frac{1}{\mathscr{Z}_{n|p}}\,\mathscr{Z}_{n|p}(x_p)\,\eta_p(x_p) \tag{1.26}$$

and

$$\mathbb{Q}_{n|p}((x_{p+1},\ldots,x_n)\mid x_p)=\frac{1}{\mathscr{Z}_{n|p}(x_p)}\left\{\prod_{p\le q<n}G_q(x_q)\right\}\mathbb{P}_{n|p}((x_{p+1},\ldots,x_n)|x_p)$$

with the conditional distribution

$$\begin{aligned}\mathbb{P}_{n|p}((x_{p+1},\ldots,x_n)|x_p) &= \text{Proba}\left((X_{p+1},\ldots,X_n)=(x_{p+1},\ldots,x_n)|X_p=x_p\right)\\ &= M_{p+1}(x_p,x_{p+1})\cdots M_n(x_{n-1},x_n)\end{aligned}$$

Combining (1.25) and (1.26) we also have the following particle approximations

$$\eta_p^N(x)\ \mathscr{Z}^N_{n|p}(x):=\left\{\prod_{p\le q<n}\eta_q^N(G_q)\right\}\ \eta^N_{p|n}(x)\simeq_{N\uparrow\infty}\eta_p(x)\mathscr{Z}_{n|p}(x)$$

1.5 Some Application Domains

1.5.1 Particle Absorption Models

1.5.1.1 Random Walks Confined in a Set

We consider a symmetric random walk X_n on the integers $\mathbb{Z}$ starting at the origin $X_0 = 0$. More formally, we take independent random variables U_n, where $\mathbb{P}(U_n = 1) = \mathbb{P}(U_n = -1) = 1/2$ and we set $X_n = X_0 + \sum_{1 \leq p \leq n} U_p$. We fix $A = \{-a+1, -a+2, ..., a-1\}$, with $a \in \mathbb{N}$. We want to compute the conditional distributions

$$\text{Law}\left((X_0, \ldots, X_n) \;\middle|\; \forall 0 \leq p \leq n,\, X_p \in A\right) \tag{1.27}$$

as well as the quantities

$$\mathscr{Z}_n := \mathbb{P}(\forall 0 \leq p < n,\, X_p \in A)$$

This problem can be solved by simulation using the following particle algorithm. We start with N particles at the origin denoted by $\xi_0^i = 0$, with $i = 1, \ldots, N$. Each of them evolve $\xi_0^i \rightsquigarrow \xi_1^i$ according to one transition of the random walk; more formally, we sample N independent copies $(U_1^i)_{1 \leq i \leq N}$ of the random variables U_1, and we set $\xi_1^i = \xi_0^i + U_1^i$. We denote

$$\eta_1^N(\mathbb{1}_A) = \frac{1}{N} \sum_{1 \leq i \leq N} \mathbb{1}_A(\xi_1^i) = \frac{1}{N} \text{Card}\left\{1 \leq i \leq N \,:\, \xi_1^i \in A\right\}$$

the proportion of points ξ_1^i in the set A. We define from the sample population $\left(\xi_1^i\right)_{1 \leq i \leq N}$ a new population of N individuals $\left(\widehat{\xi}_1^i\right)_{1 \leq i \leq N}$ as follows. For each $i = 1, \ldots, N$, we perform the following operation: if $\xi_1^i \in A$, we set $\widehat{\xi}_1^i = \xi_1^i$. If $\xi_1^i \notin A$, we pick randomly an individual $\tilde{\xi}_1^i$ among those ξ_1^j in the set A and we set $\widehat{\xi}_1^i = \tilde{\xi}_1^i$. In other words, individuals within A do not move, while the individuals outside A are replaced by a randomly chosen individual among those in the set A. It may happen that all individuals ξ_1^i are outside of the set A. In this case, the algorithm stops and we set $\tau^N = 1$ to report the time of this event. If the algorithm has not stopped, we have a new configuration $\left(\widehat{\xi}_1^i\right)_{1 \leq i \leq N}$ of N individuals in the set A. We evolve $\widehat{\xi}_1^i \rightsquigarrow \xi_2^i$ according to one transition of the random walk; that is we sample N independent copies $(U_2^i)_{1 \leq i \leq N}$ of the random variables U_2, we set $\xi_2^i = \widehat{\xi}_1^i + U_2^i$ and we define

$$\eta_2^N(\mathbb{1}_A) = \frac{1}{N} \sum_{1 \leq i \leq N} \mathbb{1}_A(\xi_2^i) = \frac{1}{N} \text{Card}\left\{1 \leq i \leq N \,:\, \xi_2^i \in A\right\}.$$

As before, we define from the sample population $\left(\xi_2^i\right)_{1 \leq i \leq N}$ a new population of N individuals $\left(\widehat{\xi}_2^i\right)_{1 \leq i \leq N}$: individuals within A do not move, while the individuals

outside the desired set are replaced by a randomly chosen individual among those in the set A. If all individuals ξ_2^i fall are outside of the set A, we set $\tau^N = 2$. Iterating this stochastic process, for every time n ($< \tau^N$), we define a sequence of genetic type populations

$$\xi_n := \left(\xi_n^i\right)_{1\leq i\leq N} \in \mathbb{Z}^N \xrightarrow{\text{selection}} \widehat{\xi}_n := \left(\widehat{\xi}_n^i\right)_{1\leq i\leq N} \in \mathbb{Z}^N \xrightarrow{\text{mutation}} \xi_{n+1} \in \mathbb{Z}^N \tag{1.28}$$

This stochastic algorithm can be interpreted as a genetic type model with mutation transitions given by the one of a symmetric random walk and an acceptance-rejection selection type transition associated with the potential indicator type function $\mathbb{1}_A$. Several estimates can be extracted from this interacting sampling algorithm.

First, we mention that the stopping time τ^N tends to infinity as the size of the population $N \to \infty$. More precisely, the probability that the algorithm stops at a given time n tends to zero exponentially fast, as N tends to infinity. More interestingly, the product of the proportions of surviving particles at each time step

$$\mathscr{Z}_n^N := \prod_{0\leq p<n} \eta_p^N(\mathbb{1}_A)$$

is asymptotically a consistent estimate of the quantity $P_n(A)$ and it is unbiased; that is we have

$$\lim_{N\to\infty} \mathscr{Z}_n^N = \mathscr{Z}_n \quad \text{and} \quad \mathbb{E}\left(\mathscr{Z}_n^N\right) = \mathscr{Z}_n \tag{1.29}$$

The convergence on the l.h.s. is an almost sure asymptotic convergence. It can be made precise by non asymptotic estimates including non asymptotic variance estimates and more refined exponential type deviations. If we interpret the selection transition as a birth and death process, then the important notion of the ancestral line of a current individual arises. More precisely, when a particle $\widehat{\xi}_{n-1}^i \longrightarrow \xi_n^i$ evolves to a new location ξ_n^i, we can interpret $\widehat{\xi}_{n-1}^i$ as the parent of ξ_n^i. Looking backwards in time and recalling that the particle $\widehat{\xi}_{n-1}^i$ has selected a site ξ_{n-1}^j in the configuration at time $(n-1)$, we can interpret this site ξ_{n-1}^j as the parent of $\widehat{\xi}_{n-1}^i$ and therefore as the ancestor $\xi_{n-1,n}^i$ at level $(n-1)$ of ξ_n^i. Running back in time we can construct the whole ancestral line

$$\xi_{0,n}^i \longleftarrow \xi_{1,n}^i \longleftarrow \ldots \longleftarrow \xi_{n-1,n}^i \longleftarrow \xi_{n,n}^i = \xi_n^i \tag{1.30}$$

of each current individual. The occupation measures of the corresponding N-genealogical tree model converge as $N \to \infty$ to the conditional distribution (1.27). In a sense to be given, for any function f on the set $\mathbb{Z}^{n+1}$, we have the convergence, as $N \to \infty$,

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^N f(\xi_{0,n}^i, \xi_{1,n}^i, \ldots, \xi_{n,n}^i)\ \mathbb{1}_{\tau^N>n} = \mathbb{E}\left(f(X_0,\ldots,X_n)\ \middle|\ \forall 0\leq p<n,\ X_p\in A\right) \tag{1.31}$$

This convergence result can be refined in various directions. For instance, we can prove that the ancestral lines are "almost" independent with a common distribution given by the limiting conditional distribution. This is often called the propagation of chaos property in applied probability. It refers to the fact that the initial population consists of independent and identically distributed random variables and that this property "propagates" approximately despite the introduction of interactions. Many other results can be derived including the fluctuations and the exponential concentration of the occupation measures of the genealogical tree around the limiting conditional distribution.

Besides the fact that the particle model approximate the (rare event) probabilities (1.29) and the conditional distributions (1.31) in path spaces, it also contains some information about the top of the spectrum of the matrix Q defined below

$$\forall (x,y) \in \{-a,-a+1,...,a-1,a\} \qquad Q(x,y) := G(x)\, M(x,y)$$

with

$$G(x) := \mathbb{1}_A(x) \quad \text{and} \quad M(x,y) = \frac{1}{2}\, \mathbb{1}_{x-1}(y) + \frac{1}{2}\, \mathbb{1}_{x+1}(y)$$

Indeed, if we consider λ to be the top eigenvalue of Q and we denote by h the corresponding eigenvector s.t. $\sum_x h(x) = 1$, then we have

$$\lim_{N\to\infty} \lim_{n\to\infty} \frac{1}{n} \sum_{0\le p\le n} \log \eta_p^N(\mathbb{1}_A) = \log \lambda$$

In addition, the value $h(x)$ coincides with the long time proportion of visits of the algorithm to the state x. In other words, $h(x)$ can be interpreted as the limiting distribution of the individuals within the set A; that is

$$\lim_{N,n\to\infty} \frac{1}{n} \sum_{0\le p\le n} \frac{1}{N} \sum_{1\le i\le N} \mathbb{1}_x(\widehat{\xi}_n^i)\, \mathbb{1}_{\tau^N>n} = h(x) = \lim_{N,n\to\infty} \frac{1}{N} \sum_{1\le i\le N} \mathbb{1}_x(\widehat{\xi}_n^i)\, \mathbb{1}_{\tau^N>n}$$

The particle approximation model discussed above is far from unique. Many other interacting sampling strategies can be introduced by a simple change of probability measure. For instance, we can replace the mutation or the free evolution of the individuals in the previous algorithm by local moves restricted to the desired set A. These mutation type transitions $\widehat{\xi}_{n-1} \rightsquigarrow \xi_n$ can also be seen as transitions of a simple random walk on $\mathbb{Z}$ reflected at the boundaries of the set A. By construction all the individuals ξ_n^i at any time horizon n and for any index $i = 1,\ldots,N$ are in the desired set A.

The corresponding selection transition $\xi_n \rightsquigarrow \widehat{\xi}_n$ is now defined as follows: each individual $\xi_n^i = x$ on the boundary $x \in \partial A = \{-a+1,(a-1)\}$ of the set A has a probability $G(x) := 1/2$ to stay in A, while the other individuals ξ_n^i (which are in the set A) have a probability $G(x) = 1$ to stay in A. The population $\widehat{\xi}_n$ is now defined as follows. For every index i, with a probability $G(\xi_n^i)$, we set $\widehat{\xi}_n^i = \xi_n^i$, otherwise we replace ξ_n^i by a new individual $\widehat{\xi}_n^i = \xi_n^j$ randomly chosen in the whole population with

a probability proportional to $G(\xi_n^j)$. If we now write $\eta_n^N(G) = \frac{1}{N}\sum_{1\leq i\leq N} G(\xi_n^i)$, all the previous particle approximation results (corresponding to $G(x) = \mathbb{1}_A(x)$) remain valid for this new particle algorithm.

1.5.1.2 Feynman-Kac Model

The sampling techniques described in section 1.5.1.1 are far from being restricted to random walks models confined to a set. These strategies apply to a variety of application areas including computational physics, nonlinear filtering, biology, as well as rare event analysis. From the pure mathematical point of view, they correspond to interacting particle approximation models of Feynman-Kac measures in path spaces.

To introduce these models, we recall that the conditional distributions discussed in (1.27) can be represented in terms of the distributions of the free path evolution

$$\begin{aligned}\mathbb{P}_n(x_0,\ldots,x_n) &= \text{Proba}\left((X_0,\ldots,X_n)=(x_0,\ldots,x_n)\right)\\ &= \mathbb{1}_0(x_0)\, M_1(x_0,x_1)\ \ldots\ M_n(x_{n-1},x_n)\end{aligned} \tag{1.32}$$

of the simple random walk starting at the origin with elementary transitions given by the matrix $M_n := (M_n(x,y))_{x,y\in\mathbb{Z}}$ with entries given by

$$M_n(x,y) := \frac{1}{2}\,\mathbb{1}_{x-1}(y) + \frac{1}{2}\,\mathbb{1}_{x+1}(y)$$

More formally, if we set

$$\mathbb{Q}_n(x_0,\ldots,x_n) := \text{Proba}\left((X_0,\ldots,X_n)=(x_0,\ldots,x_n)\ \middle|\ \forall 0\leq p<n,\ X_p\in A\right)$$

then we have

$$\mathbb{Q}_n(x_0,\ldots,x_n) = \frac{1}{\mathscr{Z}_n}\left\{\prod_{0\leq p<n} G_p(x_p)\right\}\mathbb{P}_n(x_0,\ldots,x_n) \tag{1.33}$$

with the indicator potential functions $G_n(x) = \mathbb{1}_A(x)$ and $\mathbb{P}_n(x_0,\ldots,x_n)$ being the distribution of a free path of length n of the symmetric random walk. In (1.33), $\mathscr{Z}_n$ is the normalizing constant given by

$$\mathscr{Z}_n = \mathbb{P}(\forall 0\leq p<n,\ X_p\in A) = \mathbb{E}\left(\prod_{0\leq p<n} G_p(X_p)\right)$$

These path integration type models are called Feynman-Kac measures in reference to Feynman path integral formulation of quantum mechanics where the classical notion of a single deterministic trajectory for a system is replaced by a sum over all possible trajectories weighted by the contributions of all the histories in configuration space.

1.5.1.3 A Killed Markov Chain

The Feynman-Kac measures presented in (1.33) can be regarded as the distribution of the paths of a Markov particle evolving using the Markov transitions M_n in an environment with absorbing obstacles related to potential functions G_n, and starting with some initial distribution $\mathrm{Law}(X_0) = \eta_0$ with $\eta_0(x_0) = \mathbb{1}_0(x_0)$ in (1.32). To be more precise, we consider an auxiliary coffin or cemetery state c and we set $E_c = E \cup \{c\}$. We define an E_c-valued Markov chain X_n^c with two separate killing/exploration transitions:

$$X_n^c \xrightarrow{\text{killing}} \widehat{X}_n^c \xrightarrow{\text{exploration}} X_{n+1}^c \tag{1.34}$$

This killing/exploration mechanism are defined as follows:

- **Killing:** If $X_n^c = c$, we set $\widehat{X}_n^c = c$. Otherwise the particle X_n^c is still alive. In this case, with a probability $G_n(X_n^c)$, it remains in the same site so that $\widehat{X}_n^c = X_n^c$, and with a probability $1 - G_n(X_n^c)$ it is killed and we set $\widehat{X}_n^c = c$.
- **Exploration:** Once a particle has been killed, it can not be brought back to life so if $\widehat{X}_n^c = c$ then we set $\widehat{X}_p^c = X_p = c$ for any $p > n$. Otherwise, the particle $\widehat{X}_n^c \in E$ evolves to a new location $X_{n+1}^c = x$ in E randomly chosen according to the distribution $M_{n+1}(X_n^c, x)$.

In this physical interpretation, the measure $\mathbb{Q}_n$ represent the conditional distributions of the paths of a non absorbed Markov particle. To see this claim, we denote by T the time at which the particle has been killed

$$T = \inf\{n \geq 0\,;\, \widehat{X}_n^c = c\}$$

By construction, we have

$$\begin{aligned}
&\mathrm{Proba}(T > n-1)\\
&= \mathrm{Proba}(\widehat{X}_0^c \in E, \ldots, \widehat{X}_{n-1}^c \in E)\\
&= \int_{E_n} \eta_0(dx_0)\, G_0(x_0)\, M_1(x_0, dx_1)\, \ldots M_{n-1}(x_{n-2}, dx_{n-1}) G_{n-1}(x_{n-1})\\
&= \mathbb{E}\left(\prod_{p=0}^{n-1} G_p(X_p)\right)
\end{aligned}$$

This also shows that the normalizing constants $\mathscr{Z}_n$ represent respectively the probability for the particle to be alive at time $n-1$. In other words, we have that

$$\mathscr{Z}_n = \mathrm{Proba}(T > n-1)$$

Similar arguments yield that the distribution of a particle conditional of being alive at time $n-1$ is

$$\mathbb{Q}_n(x_0, \ldots, x_n) = \mathrm{Proba}\left((X_0^c, \ldots, X_n^c) = (x_0, \ldots, x_n) \mid T > n-1\right)$$

Using (1.18) we also have the following backward representation of $\mathbb{Q}_n$

$$\mathbb{Q}_n(x_0,\ldots,x_n) = \eta_n(x_n) \times Q^{\star}_{n,\eta_{n-1}}(x_n,x_{n-1}) \cdots Q^{\star}_{2,\eta_1}(x_2,x_1) \times Q^{\star}_{1,\eta_0}(x_1,x_0) \tag{1.35}$$

with the time reversal Markov matrices $Q^{\star}_{n,\eta_{n-1}}(x_n,x_{n-1})$ defined below

$$Q^{\star}_{n,\eta_{n-1}}(x_n,x_{n-1}) = \frac{\eta_{n-1}(x_{n-1})Q_n(x_{n-1},x_n)}{\eta_{n-1}Q_n(x_n)}$$

1.5.1.4 A Particle Sampling Model

The particle sampling technique of any distribution $\mathbb{Q}_n$ associated with some Markov transition M_n and some sequence of $[0,1]$-valued potential function G_n on some (countable) state space E is defined as before in terms of a genetic type algorithm with M_n-mutations and G_n-selection type transitions. More precisely, at every time step n, we sample the mutation-selection transitions as follows: during the mutation step, every individual performs a local random move according to the Markov transition M_n. During the selection step, every individual evaluates its potential value $G_n(\xi^i_n)$, with $1 \leq i \leq N$. For every index i, with a probability $G_n(\xi^i_n)$, we set $\widehat{\xi}^i_n = \xi^i_n$, otherwise we replace ξ^i_n be a fresh new individual $\widehat{\xi}^i_n = \xi^j_n$ randomly chosen from the population with a probability proportional to $G_n(\xi^j_n)$.

As in the confinement model (discussed in the previous section), it may happen that all individuals ξ^i_n have a null potential value $G_n(\xi^i_n) = 0$, at some time period n . In this case, the algorithm stops and we set $\tau^N = n$ to report the time of this event. Under some rather weak regularity properties, we also mention that the stopping time τ^N tends to infinity as the size of the population $N \to \infty$.

For any time horizon n and any function f on the set E_n, we have

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} f(\xi^i_{0,n},\xi^i_{1,n},\ldots,\xi^i_{n,n})\, \mathbb{1}_{\tau^N>n} = \sum_{x_0,\ldots,x_n} f(x_0,\ldots,x_n)\, \mathbb{Q}_n(x_0,\ldots,x_n) \tag{1.36}$$

Furthermore, the unbiased approximations of the normalizing constants $\mathscr{Z}_n$ are given by

$$\mathscr{Z}^N_n := \prod_{0\leq p<n} \eta^N_p(G_p) \quad \text{with} \quad \forall n \in \mathbb{N} \quad \eta^N_n(G_n) := \frac{1}{N} \sum_{1\leq i\leq N} G_n(\xi^i_n) \tag{1.37}$$

In addition, mimicking formula (1.41), an alternative particle approximation of the measures $\mathbb{Q}_n$ is defined, replacing the measures η_n by their particle approximations

$$\begin{aligned} \mathbb{Q}^N_n(x_0,\ldots,x_n) &= \eta^N_n(x_n) \times Q^{\star}_{n,\eta^N_{n-1}}(x_n,x_{n-1}) \cdots Q^{\star}_{2,\eta^N_1}(x_2,x_1) \times Q^{\star}_{1,\eta^N_0}(x_1,x_0) \\ &\to_{N\uparrow\infty} \mathbb{Q}_n(x_0,\ldots,x_n) \end{aligned}$$

with the time reversal random matrices $Q^{\star}_{n,\eta^N_{n-1}}(x_n,x_{n-1})$ defined below

$$Q^{\star}_{n,\eta^N_{n-1}}(x_n,x_{n-1}) = \frac{\eta^N_{n-1}(x_{n-1})Q_n(x_{n-1},x_n)}{\eta^N_{n-1}Q_n(x_n)} = \sum_{i=1}^{N} \frac{Q_n(\xi^i_{n-1},x_n)}{\sum_{j=1}^{N} Q_n(\xi^j_{n-1},x_n)} \mathbb{1}_{\xi^i_{n-1}}(x_{n-1})$$

For time homogeneous models $(G_n,M_n) = (G,M)$ associated with a reversible matrix M w.r.t. to some measure μ on E, i.e. $\mu(x)M(x,y) = \mu(y)M(y,x)$, the corresponding particle model also contains information about the top of the spectrum of the matrix Q defined through

$$\forall (x,y) \in E \qquad Q(x,y) := G(x)\, M(x,y)$$

More precisely, if we consider λ to be the top eigenvalue of Q in $\mathbb{L}_2(\mu)$ and we denote by h the corresponding eigenvector s.t. $\sum_x \mu(x)h(x) = 1$, then we have

$$\lim_{N\to\infty} \lim_{n\to\infty} \frac{1}{n} \sum_{0\le p\le n} \log \eta^N_p(G) = \log \lambda$$

as well as

$$\lim_{N,n\to\infty} \frac{1}{n} \sum_{0\le p\le n} \frac{1}{N} \sum_{1\le i\le N} \mathbb{1}_x(\widehat{\xi}^i_n)\ \mathbb{1}_{\tau^N>n} = \mu(x)h(x) = \lim_{N,n\to\infty} \frac{1}{N} \sum_{1\le i\le N} \mathbb{1}_x(\widehat{\xi}^i_n)\ \mathbb{1}_{\tau^N>n}$$

For further details on this subject, we refer the reader to [38, 39, 54] and references therein.

1.5.2 Signal Processing and Bayesian Inference

1.5.2.1 Nonlinear Filtering Problems

We discuss here the application of these particle model to filtering problems. Suppose that at every time step the state of the Markov chain X_n is partially observed according to the following schematic picture

$$\begin{array}{ccccccc} X_0 & \longrightarrow & X_1 & \longrightarrow & X_2 & \longrightarrow & \ldots \\ \downarrow & & \downarrow & & \downarrow & & \\ Y_0 & & Y_1 & & Y_2 & & \ldots \end{array}$$

with some random variables Y_n whose values only depend on the current state of the chain

$$\text{Proba}\,(Y_n = y_n \mid X_n = x_n) := G(x_n,y_n) \tag{1.38}$$

We consider the following pair of events

$A_n(x) := \{(X_0,\ldots,X_n) = (x_0,\ldots,x_n)\}$ and $B_{n-1}(y) := \{(Y_0,\ldots,Y_{n-1}) = (y_0,\ldots,y_{n-1})\}$

The filtering problem consists of computing the conditional distributions of the state variables $A_n(x)$ given the observations $B_n(y)$. By construction, given $A_n(x)$, the random variables Y_n are independent and identically distributed with a distribution given by

$$\text{Proba}\,(B_{n-1}(y)\,|A_n(x)\,) = \prod_{0\le p<n} G(x_p,y_p)$$

By direct application of Bayes' rule we have the following formula

$$\begin{aligned}\text{Proba}\,(A_n(x)\cap B_{n-1}(y)) &= \text{Proba}\,(B_{n-1}(y)\,|A_n(x)\,)\times\text{Proba}\,(A_n(x))\\ &= \left\{\prod_{0\le p<n} G(x_p,y_p)\right\}\,\mathbb{P}_n(x_0,\ldots,x_n)\end{aligned} \tag{1.39}$$

with the distributions of the path sequence $(X_0,\ldots,X_n)$ given by

$$\mathbb{P}_n(x_0,\ldots,x_n) = \text{Proba}\,(X_0 = x_0,\ldots,X_n = x_n)$$

from which we conclude that

$$\text{Proba}\,(A_n(x)\ |\ B_{n-1}(y)\,) = \frac{1}{\mathscr{Z}_n(y)}\,\left\{\prod_{0\le p<n} G(x_p,y_p)\right\}\,\mathbb{P}_n(x_0,\ldots,x_n)$$

with the normalizing constants

$$\mathscr{Z}_n(y) := \text{Proba}(B_{n-1}(y)) = \sum_{x_0,\ldots,x_n}\left\{\prod_{0\le p<n} G(x_p,y_p)\right\}\,\mathbb{P}_n(x_0,\ldots,x_n)$$

These Feynman-Kac formulae express the conditional distributions of the path sequence $(X_0,\ldots,X_n)$ as the distribution $\mathbb{P}_n(x_0,\ldots,x_n)$ of the signal paths $(X_0,\ldots,X_n) = (x_0,\ldots,x_n)$ weighted by the product of the likelihood functions $G(x_p,y_p)$ from the origin $p = 0$ up to time $p = n$.

If we fix the observation sequence $Y_n = y_n$ and set

$$G_n(x_n) := G(x_n,y_n)$$

then we find that these measures have exactly the same form as the one presented in (1.33). We can also rewrite these conditional distributions as follows

$$\begin{aligned}\mathbb{Q}_n(x_0,\ldots,x_n) &= \text{Proba}\,(A_n(x)\ |\ B_{n-1}(y)\,)\\ &\propto \left\{\prod_{0\le p<n} G_p(x_p)\right\}\,\underbrace{\left\{\eta_0(x_0)\prod_{1\le p\le n} M_p(x_{p-1},x_p)\right\}}_{\mathbb{P}_n(x_0,\ldots,x_n)}\\ &= \eta_0(x_0)Q_1(x_0,x_1)Q_2(x_1,x_2)\ldots Q_n(x_{n-1},x_n)\end{aligned} \tag{1.40}$$

with the positive matrices $Q_n(x_{n-1},x_n)$ defined for any $n \geq 1$ by

$$Q_n(x_{n-1},x_n) := G_{n-1}(x_{n-1})\, M_n(x_{n-1},x_n)$$

The corresponding particle approximations defined in section 1.5.1.4 are often referred to as particle filters in signal processing and statistics [36, 37, 52, 38, 67]. These particle algorithms can also be used to approximate the log-likelihood functions using (1.37); that is the log-likelihood

$$L_n(y) := \log \mathscr{Z}_n(y)$$

is approximated using

$$L_n^N(y) := \log \mathscr{Z}_n^N(y) = \sum_{0\leq p<n} \log \eta_p^N(G_p).$$

1.5.2.2 Smoothing Estimation Models

Smoothing problems consist of estimating some values of the signal X_p at some time p, given s series of observations $Y_q = y_q$, with $0 \leq q \leq n$, and $p \leq n$. One strategy is to estimate the whole signal path sequence $(X_0,\ldots,X_n)$ given the observations from the origin, up to the time horizon n. The conditional distributions on path space defined in section 1.5.2.1 can be estimated using three methods:

- the genealogical tree evolution of the particle filters;
- the particle backward Markovian interpretation of conditional distributions;
- the particle conditional distributions of the noise of the signal.

These three methods are described below.

- The genealogical tree evolution of the particle filters. To describe with some precision these models, let E be the finite state space of the signal, and let $E_n = E^{(n+1)}$. These N particle approximations on path spaces coincide with a simple genetic type evolution model with N path-valued particles

$$\xi_n^i := \left(\xi_{0,n}^i, \xi_{1,n}^i, \ldots, \xi_{n,n}^i\right) \quad \text{and} \quad \widehat{\xi}_n^i := \left(\widehat{\xi}_{0,n}^i, \widehat{\xi}_{1,n}^i, \ldots, \widehat{\xi}_{n,n}^i\right) \in E_n$$

During the selection stage, with a probability $G(\xi_{n,n}^i, y_n)$ every path-valued individual ξ_n^i stays in the same place $\widehat{\xi}_n^i = \xi_n^i$; otherwise, we replace ξ_n^i be a new individual $\widehat{\xi}_n^i = \xi_n^j$ randomly chosen among the individuals ξ_0^j with a probability proportional to its weight $G(\xi_{n,n}^i, y_n)$. This mechanism is intended to favor more likely signal path sequences. During the mutation transition, $\widehat{\xi}_n^i$ evolves randomly to a new path sequence

$$\begin{aligned}\xi_{n+1}^i &= ((\xi_{0,n+1}^i,\ldots,\xi_{n,n+1}^i),\xi_{n+1,n+1}^i) = ((\widehat{\xi}_{0,n}^i,\ldots\ldots,\widehat{\xi}_{n,n}^i),\xi_{n+1,n+1}^i) \in E_{n+1} \\ &= (E_n \times E)\end{aligned}$$

If $\widehat{\xi}_n^i = x_n$, then $\xi_{n+1,n+1}^i$ is a random variable that takes the value x with the distribution $\text{Proba}(X_{n+1} = x_{n+1} | X_n = x_n)$. As usual, for any function f on $E_n = E^{(n+1)}$ and any time horizon n, we have

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} f(\xi_{0,n}^i, \xi_{1,n}^i, \ldots, \xi_{n,n}^i) = \sum_{x_0,\ldots,x_n} \mathbb{Q}_n(x_0,\ldots,x_n) f(x_0,\ldots,x_n)$$

- Particle backward Markov models. An alternative approach is to use the backward representation (1.18) of the conditional distribution $\mathbb{Q}_n$ defined in (1.40)

$$\mathbb{Q}_n(x_0,\ldots,x_n) = \eta_n(x_n) \times Q^\star_{n,\eta_{n-1}}(x_n,x_{n-1}) \cdots Q^\star_{2,\eta_1}(x_2,x_1) \times Q^\star_{1,\eta_0}(x_1,x_0) \tag{1.41}$$

with the time reversal Markov matrices $Q^\star_{n,\eta_{n-1}}(x_n,x_{n-1})$ defined below:

$$Q^\star_{n,\eta_{n-1}}(x_n,x_{n-1}) = \frac{\eta_{n-1}(x_{n-1}) Q_n(x_{n-1},x_n)}{\eta_{n-1} Q_n(x_n)} = \frac{\eta_{n-1}(x_{n-1}) G_{n-1}(x_{n-1}) M_n(x_{n-1},x_n)}{\sum_x \eta_{n-1}(x) G_{n-1}(x) M_n(x,x_n)}$$

Replacing the measures η_n by their particle estimates η_n^N, we define the particle approximation of $\mathbb{Q}_n$ by setting

$$\begin{aligned} \mathbb{Q}_n^N(x_0,\ldots,x_n) &= \eta_n^N(x_n) \times Q^\star_{n,\eta_{n-1}^N}(x_n,x_{n-1}) \cdots Q^\star_{2,\eta_1^N}(x_2,x_1) \times Q^\star_{1,\eta_0^N}(x_1,x_0) \\ &\to_{N\uparrow\infty} \mathbb{Q}_n(x_0,\ldots,x_n) \end{aligned}$$

with the time reversal random matrices $Q^\star_{n,\eta_{n-1}^N}(x_n,x_{n-1})$ defined by

$$\begin{aligned} Q^\star_{n,\eta_{n-1}^N}(x_n,x_{n-1}) &= \frac{\eta_{n-1}^N(x_{n-1}) Q_n(x_{n-1},x_n)}{\eta_{n-1}^N Q_n(x_n)} \\ &= \sum_{i=1}^{N} \frac{G_{n-1}(\xi_{n-1}^i,) M_n(\xi_{n-1}^i,x_n)}{\sum_{j=1}^N G_{n-1}(xi_{n-1}^j) M_n(\xi_{n-1}^j,x_n)} \mathbb{1}_{\xi_{n-1}^i}(x_{n-1}) \end{aligned}$$

- Particle approximations of the noise of the signal. We further assume that the signal process given by recursive equations on some finite state space E of the following form

$$X_n := F_n(X_{n-1},U_n) \tag{1.42}$$

with some independent random variables U_n, and with distribution ν_n independent of X_0 on the finite set $\mathscr{U}$. If we consider the following events

$$C_n(u) = \{(X_0,(U_0,\ldots,U_n)) = (x_0,(u_1,\ldots,u_n))\}$$

then we find that

$$\text{Proba}\,(C_n(u)\,|B_{n-1}(y)\,)=\frac{1}{\mathscr{Z}_n(y)}\left\{\prod_{0\le p<n}G(\mathscr{X}_p^{(x_0,u)},y_p)\right\}\mathbb{P}_n(x_0,(u_1,\ldots,u_n)) \tag{1.43}$$

where $\mathscr{X}_n^{(x_0,u)}$ stands for the solution of the discrete generation system (1.42) associated with a given realization $(u_n)_{n\ge1}$ and some initial condition x_0. The function G is the likelihood function defined in (1.38).

In the semigroup formulation, $\mathscr{X}_n^{(x_0,u)}$ is a function of the initial state and the control sequence $(u_1,\ldots,u_n)$; that is, we have that $\mathscr{X}_n^{(x_0,u)}=\phi_n(x_0,(u_1,\ldots,u_n))$ for some function ϕ_n from $(E\times\mathscr{U}^n)$ into E. For any $n\ge0$, we set

$$H_n(x_0,(u_1,\ldots,u_n)):=G\,(\phi_n(x_0,(u_1,\ldots,u_n)),y_n)$$

In this notation we have

$$\text{Proba}\,(C_n(u)\,|B_{n-1}(y)\,)=\frac{1}{\mathscr{Z}_n(y)}\left\{\prod_{0\le p<n}H_p(x_0,(u_1,\ldots,u_p))\right\}\mathbb{P}_n(x_0,(u_1,\ldots,u_n))$$

As above, the N particle approximation of these probability measures on control sequences is again described by genetic evolution models with N path-valued particles

$$\begin{aligned}\xi_n^i&:=\left(\xi_{0,n}^i,\xi_{1,n}^i,\ldots,\xi_{n,n}^i\right)\\ \widehat{\xi}_n^i&:=\left(\widehat{\xi}_{0,n}^i,\widehat{\xi}_{1,n}^i,\ldots,\widehat{\xi}_{n,n}^i\right)\in E_n:=(E\times\mathscr{U}^n)\end{aligned}$$

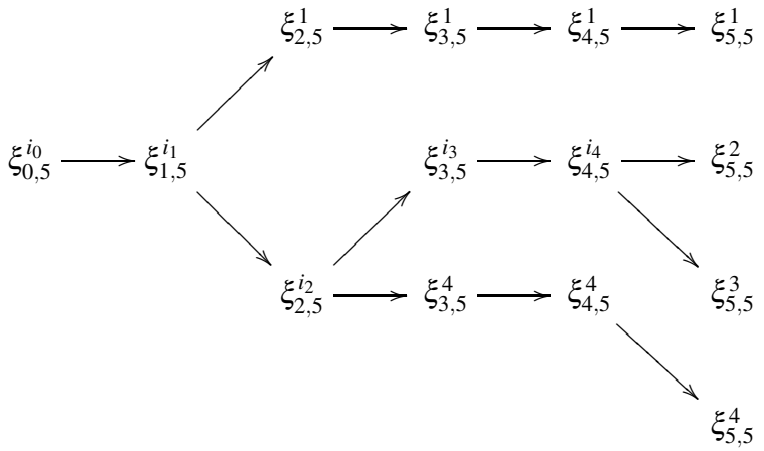
During the selection stage, with a probability $H_n(\xi_n^i)$ every path-valued individual stays in the same place $\widehat{\xi}_n^i=\xi_n^i$; otherwise, we replace ξ_n^i by a new individual $\widehat{\xi}_n^i=\xi_n^j$ randomly chosen among the individuals ξ_0^j with a probability proportional to its weight $H_n(\xi_n^i)$. This mechanism is intended to favor more likely noise sequences w.r.t. the observations. During the mutation transition, to every selected signal-noise sequence $\widehat{\xi}_n^i$ we add randomly new possible values of the noise at time $(n+1)$; that is, we set

$$\begin{aligned}\xi_{n+1}^i&=((\xi_{0,n+1}^i,\ldots,\xi_{n,n+1}^i),\xi_{n+1,n+1}^i)\\&=((\widehat{\xi}_{0,n}^i,\ldots\ldots,\widehat{\xi}_{n,n}^i),\xi_{n+1,n+1}^i)\in E_{n+1}=(E_n\times\mathscr{U})\end{aligned} \tag{1.44}$$

where $\xi_{n+1,n+1}^i$ is a random variable with distribution ν_n on $\mathscr{U}$. Various asymptotic estimates can be derived. For instance, for any function f on $E_n=E^n$ and any time horizon n, we have

$$\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^N f(\xi_{0,n}^i,\xi_{1,n}^i,\ldots,\xi_{n,n}^i)=\sum_{x_0,\ldots,x_n}\mathbb{Q}_n(x_0,(u_1,\ldots,u_n))\;f(x_0,(u_1,\ldots,u_n)) \tag{1.45}$$

In other words, the occupation measures of the genealogical tree evolution, like the one illustrated below for $(N,n)=(4,5)$



with any $i_0 \in \{1,2,3,4\}$, $i_2 \in \{2,3,4\}$, $i_3 \in \{2,3\}$, $i_4 \in \{2,3\}$, represent the conditional distribution of $(X_0,(U_1,U_2,U_3,U_4,U_5))$ w.r.t. the observations (Y_0,Y_1,Y_2,Y_3,Y_4), in terms of the more likely initial condition $\xi^{i_0}_{0,5}$ and the four more likely signal-noise sequences $(\xi^i_{1,5},\xi^i_{2,5},\xi^i_{3,5},\xi^i_{4,5},\xi^i_{5,5})_{i=1,2,3,4}$.

1.5.2.3 Approximate Bayesian Computation

Approximate Bayesian computation (*abbreviate ABC*) techniques are Bayesian inference methods currently used to evaluate posterior distributions without having to calculate likelihoods. For instance, in biology applications and more particularly in predictive bacteriology and food risk analysis, the observations of a kinetic biological complex system are given by counting bacteria individuals after successive dilutions of a food sample coming from an in vitro culture [74, 75, 80, 81]. Of course, this experimental observation process is often modeled by a series of Poisson type dependent random variables but the computation of the likelihood function often requires successive summations over the set of all the integers. In this situation likelihood functions are computationally intractable or very costly to estimate.

One of the central ideas of ABC methods is to replace the evaluation of the likelihood function by a simulation-based procedure of the observation process coupled with a numerical comparison between the observed and simulated data. This strategy is rather well known in particle filtering literature, see for instance [45, 46, 44]. In the same manner, these additional levels of simulation-based approximations can also be extended to compute the posterior distribution of fixed parameters in hidden Markov chain models. In signal processing literature, these ABC type particle models are sometimes called convolution particle filters, see for instance [19, 20, 144].

First, we notice that the transition probabilities of the signal-observation Markov chain $\mathscr{X}_n := (X_n,Y_n)$ are given by

$$\text{Proba}\,(\mathscr{X}_n=(x_n,y_n)\mid \mathscr{X}_n=(x_{n-1},y_{n-1}))=M_n(x_{n-1},x_n)\times G(x_n,y_n) \qquad (1.46)$$

with the likelihood function G defined in (1.38) and the Markov transitions of the chain X_n

$$\text{Proba}\,(X_n = x_n \mid X_n = x_{n-1}) = M_n(x_{n-1}, x_n)$$

Suppose that at every time step the state of the pair signal-observation Markov chain $\mathscr{X}_n := (X_n, Y_n)$ is partially observed according to the following schematic picture

$$\mathscr{X}_0 := \begin{bmatrix} X_0 \\ \downarrow \\ Y_0 \end{bmatrix} \longrightarrow \mathscr{X}_1 := \begin{bmatrix} X_1 \\ \downarrow \\ Y_1 \end{bmatrix} \longrightarrow \mathscr{X}_2 := \begin{bmatrix} X_2 \\ \downarrow \\ Y_2 \end{bmatrix} \longrightarrow \ldots$$
$$\begin{matrix} \downarrow \\ Y_0^{\varepsilon} \end{matrix} \qquad \begin{matrix} \downarrow \\ Y_1^{\varepsilon} \end{matrix} \qquad \begin{matrix} \downarrow \\ Y_2^{\varepsilon} \end{matrix} \qquad \ldots$$

with some random variables Y_n^{ε} whose values only depend on the second component Y_n of the current state (X_n, Y_n) of the chain

$$\text{Proba}\,(Y_n^{\varepsilon} = y_n \mid (X_n, Y_n) = (x_n, z_n)\,) := G^{\varepsilon}(z_n, y_n)$$

We further assume that the likelihood function $G^{\varepsilon}(z,y)$ is a Markov transition indexed by some parameter $\varepsilon \in [0,1]$ s.t. $\lim_{\varepsilon \to 0} G^{\varepsilon}(z,y) = \mathbb{1}_y(z)$. When the state space of the observation process is equipped with some neighborhood system, we can take

$$G^{\varepsilon}(z,y) = \frac{1}{\text{Card}(\mathscr{V}_{\varepsilon}(z))}\ \mathbb{1}_{\mathscr{V}_{\varepsilon}(z)}(y)$$

where $\mathscr{V}_{\varepsilon}(z)$ is a collection of neighborhoods of the point z s.t. $\mathscr{V}_{\varepsilon}(z) \to_{\varepsilon \to 0} \{z\}$. For instance, if the observation state space is equipped with some distance function d we can take $\mathscr{V}_{\varepsilon}(z) = \{y \,:\, d(z,y) \leq \varepsilon\}$. In this situation, given the the current state of the chain (X_n, Y_n), the observation Y_n^{ε} is randomly chosen in the set $\mathscr{V}_{\varepsilon}(Y_n)$.

Using (1.46) we prove that the distribution $\mathbf{P}_n((x_0,y_0),\ldots,(x_n,y_n))$ of the signal-observation paths

$$(\mathscr{X}_0,\ldots,\mathscr{X}_n) := ((X_0,Y_0),\ldots,(X_n,Y_n)) = ((x_0,y_0),\ldots,(x_n,y_n))$$

is given by

$$\begin{aligned} \mathbf{P}_n((x_0,y_0),\ldots,(x_n,y_n)) &= \text{Proba}\,(A_n(x) \cap B_n(z)) \\ &= \left\{ \prod_{0 \leq p \leq n} G(x_p, z_p) \right\} \mathbb{P}_n(x_0,\ldots,x_n) \end{aligned}$$

with the pair of events

$$A_n(x) := \{(X_0,\ldots,X_n) = (x_0,\ldots,x_n)\} \quad \text{and} \quad B_n(z) := \{(Y_0,\ldots,Y_n) = (z_0,\ldots,z_n)\}$$

and the distribution $\mathbb{P}_n(x_0,\ldots,x_n) := \text{Proba}\,(A_n(x))$ of the paths $(X_0,\ldots,X_n) = (x_0,\ldots,x_n)$. We consider the following events

$$\mathbf{A}_n((x,z)) := A_n(x) \cap B_n(z) \quad \text{and} \quad \mathbf{B}_n^\varepsilon(y) := \{(Y_0^\varepsilon, \ldots, Y_n^\varepsilon) = (y_0, \ldots, y_n)\}$$

As in section 1.5.2.1, the filtering problem defined above consists of computing the conditional distributions of the state variables $\mathbf{A}_n(x,z)$ given the observations $\mathbf{B}_n^\varepsilon(y)$. By construction, given $\mathbf{A}_n(x,z)$, the random variables Y_n^ε are independent and identically distributed with a distribution given by

$$\text{Proba}\,(\mathbf{B}_n^\varepsilon(y)\,|\mathbf{A}_n(x,z)\,) = \prod_{0\le p\le n} G^\varepsilon(z_p, y_p) \quad \longrightarrow_{\varepsilon\to 0} \mathbb{1}_{(y_0,\ldots,y_n)}(z_0,\ldots,z_n)$$

from which we conclude that

$$\begin{aligned}\text{Proba}\,(\mathbf{A}_n(x,z) \mid \mathbf{B}_n^\varepsilon(y)\,) &= \frac{1}{\mathscr{Z}_n^\varepsilon(y)} \left\{ \prod_{0\le p\le n} G^\varepsilon(z_p, y_p) \right\} \mathbf{P}_n((x_0,z_0),\ldots,(x_n,z_n)) \\ &\to_{\varepsilon\downarrow 0} \text{Proba}\,(A_n(x) \cap B_n(y))\end{aligned}$$

with the normalizing constants

$$\begin{aligned}\mathscr{Z}_n^\varepsilon(y) &:= \text{Proba}(\mathbf{B}_n^\varepsilon(y)) = \sum_{(x_0,z_0),\ldots,(x_n,z_n)} \left\{ \prod_{0\le p\le n} G^\varepsilon(z_p, y_p) \right\} \mathbf{P}_n((x_0,z_0),\ldots,(x_n,z_n)) \\ &\to_{\varepsilon\downarrow 0} \text{Proba}(B_n(y))\end{aligned}$$

As in section 1.5.2.1 these posterior distributions have exactly the same form as the one presented in (1.33). Notice that in this situation, at every time step n the stochastic model consists of N-particle samples $\xi_n^i := (\xi_n^{i,1}, \xi_n^{i,2})$ with a signal component $\xi_n^{i,1}$ and the corresponding observation component $\xi_n^{i,2}$, with $1 \le i \le N$. Given a series of observations $(y_n)_{n\ge 0}$, the conditional distributions defined above are approximated by the N-empirical measures of the particle model

$$\eta_n^N := \frac{1}{N} \sum_{i=1}^N \mathbb{1}_{(\xi_n^{i,1}, \xi_n^{i,2})} \longrightarrow_{N\to\infty} \text{Proba}\left(\mathbf{A}_n(x,z) \;\middle|\; \mathbf{B}_{n-1}^\varepsilon(y)\right)$$

and an unbiased estimate of the normalizing constants $\mathscr{Z}_n^\varepsilon(y)$ is given by

$$\mathscr{Z}_n^{\varepsilon,N}(y) := \prod_{0\le p\le n} \frac{1}{N} \sum_{i=1}^N G^\varepsilon(\xi_p^{i,2}, y_p) \longrightarrow_{N\to\infty} \mathscr{Z}_n^\varepsilon(y) = \text{Proba}(\mathbf{B}_n^\varepsilon(y))$$

1.5.3 Interacting Kalman Filters

1.5.3.1 A Brief Introduction to Kalman Filters

We consider a $\mathbb{R}^{p+q}$-valued Markov chain (X_n, Y_n) defined by the recursive relations

$$\begin{cases} X_n = A_n X_{n-1} + B_n W_n\,, & n \geq 1 \\ Y_n = C_n X_n + D_n V_n\,, & n \geq 0 \end{cases} \tag{1.47}$$

for some $\mathbb{R}^{d_w}$ and $\mathbb{R}^{d_v}$-valued independent random sequences W_n and V_n, independent of X_0, and some matrices A_n, B_n, C_n, D_n with appropriate dimensions. We further assume that W_n and V_n are centered Gaussian random sequences with covariance matrices R_n^v, R_n^w and X_0 is a Gaussian random variable in $\mathbb{R}^p$ with a mean $\widehat{X}_0^-$ and covariance matrix $\widehat{P}_0^-$. In the further development of this section we shall denote by $\mathscr{N}(m,R)$ a Gaussian distribution in a d-dimensional space $\mathbb{R}^d$ with mean vector $m \in \mathbb{R}^d$ and covariance matrix $R \in \mathbb{R}^{d\times d}$

$$\mathscr{N}(m,R)(dx) = \frac{1}{(2\pi)^{d/2}\sqrt{|R|}}\ \exp\left[-2^{-1}(x-m)R^{-1}(x-m)'\right]\,dx$$

Using this notation, we have

$$\mathrm{Law}(X_n\,|Y_0,\ldots,Y_{n-1}) = \mathscr{N}(\widehat{X}_n^-, P_n^-) \ \text{ and } \ \mathrm{Law}(X_n \mid Y_0,\ldots,Y_{n-1},Y_n) = \mathscr{N}(\widehat{X}_n, P_n)$$

The synthesis of the conditional mean and covariance matrices is carried out using the traditional Kalman-Bucy recursive equations

$$\left(\widehat{X}_n^-, P_n^-\right) \xrightarrow{\text{updating}} \left(\widehat{X}_n, P_n\right) \xrightarrow{\text{prediction}} \left(\widehat{X}_{n+1}^-, P_{n+1}^-\right) \tag{1.48}$$

The updating and the prediction step are given below

[**Updating**] $\qquad \widehat{X}_n = \widehat{X}_n^- + \mathbf{G}_n\,(Y_n - C_n\widehat{X}_n^-) \quad \text{and} \quad P_n = P_n^- - \mathbf{G}_n C_n P_n^-$

with the gain matrix $\mathbf{G}_n = P_n^- C_n'(C_n P_n^- + D_n R_n^v D_n')^{-1}$, and

[**Prediction**] $\qquad \widehat{X}_{n+1}^- = A_{n+1}\widehat{X}_n \quad \text{and} \quad P_{n+1}^- = A_{n+1}\,P_n\,A_{n+1}' + B_{n+1}\,R_{n+1}^w\,B_{n+1}'$

Proof. The proof of the updating recursion equation is based on the fact that

$$\widehat{X}_n := \widehat{X}_n^- + \mathbf{G}_n\,(Y_n - \widehat{Y}_n^-) \quad \text{with} \quad \widehat{Y}_n^- = \mathbb{E}(Y_n|Y_0,\ldots,Y_{n-1}) = C_n\widehat{X}_n^-$$

Since $\mathbb{E}((X_n - \widehat{X}_n)(Y_n - \widehat{Y}_n^-)') = 0$, we find $\mathbb{E}((X_n - \widehat{X}_n^-)(Y_n - \widehat{Y}_n^-)') = \mathbf{G}_n\,\mathbb{E}((Y_n - \widehat{Y}_n^-)(Y_n - \widehat{Y}_n^-)')$, from which we find the gain matrix. Finally using the decomposition $X_n - \widehat{X}_n = (X_n - \widehat{X}_n^-) + (\widehat{X}_n^- - \widehat{X}_n)$ and by symmetry argument we conclude that

$$\begin{aligned} P_n &= P_n^- - \mathbb{E}((\widehat{X}_n^- - \widehat{X}_n)(\widehat{X}_n^- - \widehat{X}_n)') \\ &= P_n^- - \mathbf{G}_n\mathbb{E}((Y_n - \widehat{Y}_n^-)(Y_n - \widehat{Y}_n^-)')\mathbf{G}_n' = P_n^- - \mathbf{G}_n C_n P_n^- \end{aligned}$$

The proof of the prediction recursion is rather elementary. The first assertion is clear. The second one comes from the fact that

$$\begin{aligned} P_{n+1}^{-} &= \mathbb{E}((A_{n+1}(X_n - \widehat{X}_n) + B_{n+1}W_{n+1})(A_{n+1}(X_n - \widehat{X}_n) + B_{n+1}W_{n+1})') \\ &= A_{n+1}\, P_n\, A'_{n+1} + B_{n+1}\, R^w_{n+1}\, B'_{n+1} \end{aligned}$$

It is also useful to observe that

$$\mathrm{Law}(Y_n \mid Y_0, \ldots, Y_{n-1}) = \mathscr{N}(C_n \widehat{X}_n^{-}, C_n\, P_n^{-}\, C'_n + R^v_n)$$

We prove this claim using the fact that, given $(Y_0, \ldots, Y_{n-1})$, the current observation takes the form

$Y_n = C_n \tilde{X}_n + D_n V_n$ with some variable $\tilde{X}_n$ s.t. $\mathrm{Law}\left(\tilde{X}_n \mid Y_0, \ldots, Y_{n-1}\right) := \mathscr{N}(\widehat{X}_n^{-}, P_n^{-})$.

We slight abuse the notation and we denote by $\mathscr{N}(m,R)(x)$ the density of a Gaussian distribution $\mathscr{N}(m,R)(dx) = \mathscr{N}(m,R)(x)dx$ w.r.t. the Lebesgue measure dx. In this notation, the density $p_n(y_0, \ldots, y_n)$ of the random sequence of observation $(Y_0, \ldots, Y_n)$ evaluated at the random observation path $(Y_0, \ldots, Y_n)$ is given by

$$p_n(Y_0, \ldots, Y_n) = \prod_{k=0}^{n} \mathscr{N}(C_k \widehat{X}_k^{-}, C_k\, P_k^{-}\, C'_k + R^v_k)(Y_k)$$

In Bayesian inference literature, this formula is sometimes written in the following form

$$p_n(Y_0, \ldots, Y_n) = p_n(Y_n \,|Y_0, \ldots, Y_{n-1}) \times p_{n-1}(Y_0, \ldots, Y_{n-1}) = \prod_{k=0}^{n} p_k(Y_k \,|Y_0, \ldots, Y_{k-1}).$$

1.5.3.2 Interacting Kalman Filters

We consider a Markov chain Θ_n taking values in some finite state space E, and a collection of matrices $A_n(\theta), B_n(\theta), C_n(\theta), D_n(\theta)$ indexed by $\theta \in E$, and of the same dimension as the matrices (A_n, B_n, C_n, D_n) introduced in (1.47)indexed We let (Θ_n, X_n, Y_n) be the $(E \times \mathbb{R}^{p+q})$-valued Markov chain defined by the same recursive relations as in (1.47)

$$\begin{cases} X_n = A_n(\Theta_n)X_{n-1} + B_n(\Theta_n)W_n\,, & n \geq 1 \\ Y_n = C_n(\Theta_n)X_n + D_n(\Theta_n)V_n\,, & n \geq 0 \end{cases} \tag{1.49}$$

Arguing as above, given a realization of the chain $\Theta = (\Theta_n)_{n\geq 0}$, we have

$$\begin{aligned} \mathrm{Law}(X_n \,|\Theta,\, Y_0, \ldots, Y_{n-1}) &= \mathscr{N}(\widehat{X}_n^{\Theta,-}, P_n^{\Theta,-}) \\ \mathrm{Law}(X_n \mid \Theta,\, Y_0, \ldots, Y_{n-1}, Y_n) &= \mathscr{N}(\widehat{X}_n^{\Theta}, P_n^{\Theta}) \end{aligned}$$

with some parameters $(\widehat{X}_n^{\Theta,-}, P_n^{\Theta,-})$ and $(\widehat{X}_n^{\Theta}, P_n^{\Theta})$ that can be computed using the same Kalman recursions given above by replacing the matrices (A_n, B_n, C_n, D_n) by the matrices $(A_n(\Theta_n), B_n(\Theta_n), C_n(\Theta_n), D_n(\Theta_n))$. We observe that $(\widehat{X}_n^{\Theta,-}, P_n^{\Theta,-})$ only

depends on the random sequence $(\Theta_0,\ldots,\Theta_n)$ so that

$$\mathscr{N}(C_n(\Theta_n)\widehat{X}_n^{\Theta,-},C_n(\Theta_n)\,P_n^{\Theta,-}\,C_n'(\Theta_n)+R_n^{\nu})(Y_n):=G_{n,Y_n}(\Theta_0,\ldots,\Theta_n) \tag{1.50}$$

Therefore, the density $p_n((y_0,\ldots,y_n)\mid(\theta_0,\ldots,\theta_n))$ of the random sequence of observation $(Y_0,\ldots,Y_n)$ evaluated at the random observation path $(Y_0,\ldots,Y_n)$ and given a realization of the parameters $(\Theta_0,\ldots,\Theta_n)=(\theta_0,\ldots,\theta_n)$ is given by

$$p_n((Y_0,\ldots,Y_n)|\,(\theta_0,\ldots,\theta_n))=\prod_{k=0}^{n}G_{k,Y_k}(\theta_0,\ldots,\theta_n) \tag{1.51}$$

If we denote by $\mathbb{P}_n(\theta_0,\ldots,\theta_n)$ the probability measure of the sequence of random parameters $(\Theta_0,\ldots,\Theta_n)$, then using Bayes' rule we find that the probability measure

$$\mathbb{Q}_n(\theta_0,\ldots,\theta_n):=\frac{1}{\mathscr{Z}_{n,Y}}\left\{\prod_{0\le k<n}G_{k,Y_k}(\theta_0,\ldots,\theta_n)\right\}\mathbb{P}_n(\theta_0,\ldots,\theta_n) \tag{1.52}$$

(with some normalizing constant $\mathscr{Z}_{n,Y}$) coincides with the conditional distribution of the random sequence $(\Theta_0,\ldots,\Theta_n)$ given the observations $(Y_0,\ldots,Y_{n-1})$; that is, we have that

$$\mathbb{Q}_n=\mathrm{Law}\left((\Theta_0,\ldots,\Theta_n)\mid(Y_0,\ldots,Y_{n-1})\right)$$

The corresponding particle approximations on the set of sequences are often referred as particle methods in path space in signal processing literature and Bayesian inference studies (see for instance [38, 53, 67], and references therein).

1.5.4 Stochastic Optimization Algorithms

1.5.4.1 Interacting MCMC Models

We present now a genetic type particle strategy for sampling random states according to a sequence of probability measures on some finite state space E given by

$$\mu_n(x)=\frac{1}{\lambda(G_n)}\,G_n(x)\,\lambda(x)\quad\text{with}\quad G_n(x):=G_{n-1}(x)\times g_{n-1}(x)=\prod_{0\le p<n}g_p(x)$$

where $\lambda(x)$ is a probability measure and g_n is a collection of positive functions on E.

The interacting particle sampler of these measures is defined as follows. We start with a population of N independent individuals $\xi_0:=\left(\xi_0^i\right)_{1\le i\le N}$ randomly chosen in E according to μ_0. We perform a selection transition $\xi_0\rightsquigarrow\widehat{\xi}_0:=\left(\widehat{\xi}_0^i\right)_{1\le i\le N}$ using the potential functions g_0. More precisely, every individual evaluates its potential value $g_0(\xi_0^i)$. For every index i, with a probability $g_0(\xi_0^i)$, we set $\widehat{\xi}_0^i=\xi_0^i$,

otherwise we replace ξ_0^i by a new individual $\widehat{\xi}_0^i = \xi_0^j$ randomly chosen in the whole population with a probability proportional to $g_0(\xi_0^j)$. During the mutation transition $\widehat{\xi}_0 \rightsquigarrow \xi_1 := \left(\xi_1^i\right)_{1\leq i\leq N}$, every selected individual $\widehat{\xi}_0^i$ performs a local random move $\widehat{\xi}_0^i \rightsquigarrow \xi_1^i$ (independently of one another) according to the Markov transition P_1 associated with an MCMC sampler with invariant measure μ_1. Then, we perform a selection transition $\xi_1 \rightsquigarrow \widehat{\xi}_1 := \left(\widehat{\xi}_1^i\right)_{1\leq i\leq N}$ using the fitness functions g_1. After this selection stage we mutate each selected individual using the Markov transition P_2 associated with an MCMC sampler with invariant measure μ_2, and so on. Iterating these transitions, we define a simple genetic model with mutations transitions P_n and selection fitness functions g_n:

$$\xi_n := \left(\xi_n^i\right)_{1\leq i\leq N} \in E^N \xrightarrow{\text{selection}} \widehat{\xi}_n := \left(\widehat{\xi}_n^i\right)_{1\leq i\leq N} \in E^N \xrightarrow{\text{mutation}} \xi_{n+1} \in E^N \tag{1.53}$$

This algorithm belongs to the class of sequential Monte Carlo samplers proposed in [40]. Many convergence results can be established. For instance, under some weak regularity conditions we can show that for any $1 \leq q \leq N$, and any time horizon $n \geq 0$, the first q random samples $(\xi_n^i)_{1\leq i\leq q}$ among N are almost independent and identically distributed with the desired target measure μ_n; that is, we have that

$$\sum_{x^1,\ldots,x^q} \left|\text{Proba}\left(\xi_n^1 = x^1,\ldots,\xi_n^q = x^q\right) - \mu_n(x^1)\cdots\mu_n(x^q)\right| \leq c(n)\min\left(\frac{q^2}{N}, \sqrt{\frac{q}{N}}\right)$$

and some finite constant $c(n) < \infty$. We also have that for any $x \in E$ and any $n \geq 0$

$$\lim_{N\to\infty} \frac{1}{N}\sum_{1\leq i\leq N} \mathbb{1}_{\xi_n^i}(x) = \mu_n(x) \quad \text{and} \quad \mathscr{Z}_n^N := \prod_{0\leq p<n} \eta_p^N(g_p) \longrightarrow_{N\to\infty} \mathscr{Z}_n$$

1.5.4.2 Interacting Monte Carlo Markov Chains

Suppose we want to compute the global minima of a given non negative cost function V on some finite state space E equipped with the counting measure $\lambda(x) := \frac{1}{\text{Card}(E)}$. From the probabilistic point of view, this problem amounts to sampling random states according to the Boltzmann-Gibbs distributions associated with a large inverse temperature parameter β and given

$$\mu_\beta(x) := \frac{1}{\mathscr{Z}_\beta}\, e^{-\beta V(x)}\, \lambda(x) \quad \text{with} \quad \mathscr{Z}_\beta := \sum_x e^{-\beta V(x)}\, \lambda(x)$$

There is no loss of generality to assume that $\inf_x V(x) = 0$ and for any state $x \notin V_0 := V^{-1}(\{0\})$, $V(x) \geq \delta$ for some $\delta > 0$. It follows that we have

$$\text{Card}(V_0) \leq \mathscr{Z}_\beta \leq \text{Card}(V_0) + \text{Card}(V_0^c)\, e^{-\beta\delta} \to_{\beta\uparrow\infty} \text{Card}(V_0)$$

and therefore

$$\lim_{\beta\to\infty}\mu_\beta(x)=\mu_\infty(x):=\mathbb{1}_{V_0}(x)/\mathrm{Card}(V_0)$$

This simple observation shows that sampling according to μ_β is roughly equivalent to randomly sampling an unknown state variable with minimal cost. For very large state spaces, it is typically impossible to sample from μ_β directly.The celebrated simulated annealing algorithm to sample from μ_∞ consists of sampling approximately from a sequence of distributions μ_{β_n} where β_n is a non-decreasing sequence going to ∞. The rationale is that it is "easier" to sample from μ_β when β is small; if $\beta=0$ then μ_0 is the uniform counting measure on E from which it is trivial to sample. For $\beta_n>0$, we sample approximately from each intermediate distribution μ_{β_n} using Markov chain Monte Carlo (MCMC) sampling techniques; that is we select a transition matrix $M_{\beta_n}=\left(M_{\beta_n}(x,y)\right)_{x,y\in E}$ with left eigenvector μ_{β_n} associated with the eigenvalue 1, that is

$$\sum_x \mu_{\beta_n}(x)M_{\beta_n}(x,y)=\mu_{\beta_n}(y)$$

The probabilistic interpretation of the above equation is as follows: pick randomly a state x with distribution $\mu_{\beta_n}(x)$ and take a random transition $x\rightsquigarrow y$ from the distribution $M_{\beta_n}(x,y)$, then the probability of being at state y is again $\mu_{\beta_n}(y)$. The literature on MCMC methods discusses numerous choices of transitions M_{β_n} satisfying this property. The most famous is the Metropolis-Hastings transition associated to a symmetric transition matrix $K(x,y)=K(y,x)$ and defined by

$$M_{\beta_n}(x,y)$$

$$=K(x,y)\ \min\left(1,e^{-\beta_n(V(y)-V(x))}\right)+\left(1-\textstyle\sum_z K(x,z)\ \min\left(1,e^{-\beta_n(V(z)-V(x))}\right)\right)\ \mathbb{1}_x(y)$$

Using the fundamental ergodic theorem for regular Markov chains, starting from any initial state x_0, the n-th step of a run of the Markov chain with transitions M_{β_n} has a probability very close to $\mu_{\beta_n}(y)$ of being at the site y, for a large n. Practically, we select β_1 and we run the chain starting at $X_0=x_0$ for a large enough number of runs n_1 such that the law of the state X_{n_1} is close to μ_{β_1}

$$X_0=x_0\xrightarrow{M_{\beta_1}}X_1\xrightarrow{M_{\beta_1}}\ldots\xrightarrow{M_{\beta_1}}X_{n_1}\text{ with }n_1\text{ large enough s.t. Law}(X_{n_1})\simeq\mu_{\beta_1}$$

Notice that the choice of n_1 depends on β_1: the larger β_1 is, the "peakier" μ_{β_1} is and the larger n_1 is. When the chain is stabilized, we choose a $\beta_2>\beta_1$ and we run the chain starting at X_{n_1} for a new large enough number of time steps n_2 such that the law of the state $X_{n_1+n_2}$ is close to μ_{β_2}

$$X_{n_1}\xrightarrow{M_{\beta_2}}X_{n_1+1}\xrightarrow{M_{\beta_2}}\ldots\xrightarrow{M_{\beta_2}}X_{n_1+n_2}\text{ with }n_2\text{ large enough s.t. Law}(X_{n_1+n_2})\simeq\mu_{\beta_2}$$

The theoretical "optimal" inverse temperature parameter ensuring convergence in some sense of the Markov chain to μ_∞ is logarithmic. This amounts to saying that we change by one unit the parameter β on every time interval with exponential length. This is unrealistic from a practical point of view.

We present now an alternative particle strategy for sampling random states according to the sequence of measures μ_{β_n} associated with a given non decreasing sequence of inverse temperature parameters β_n. We suppose that $\beta_0 = 0$ so that μ_{β_0} coincides with the uniform counting measure on the set E. We start with N independent individuals $\xi_0 := \left(\xi_0^i\right)_{1\leq i\leq N}$ randomly chosen in E according to μ_{β_0}. We perform a selection transition $\xi_0 \rightsquigarrow \widehat{\xi}_0 := \left(\widehat{\xi}_0^i\right)_{1\leq i\leq N}$ using the potential functions G_0 defined by

$$G_0(x) = \exp\left(-(\beta_1 - \beta_0)V(x)\right)$$

In other words, every individual evaluates its potential value $G_0(\xi_0^i)$. For every index i, with a probability $G_0(\xi_0^i)$, we set $\widehat{\xi}_0^i = \xi_0^i$, otherwise we replace ξ_0^i by a new individual $\widehat{\xi}_0^i = \xi_0^j$ randomly chosen in the whole population with a probability proportional to $G_0(\xi_0^j)$. During the mutation step $\widehat{\xi}_0 \rightsquigarrow \xi_1 := \left(\xi_1^i\right)_{1\leq i\leq N}$, every selected individual $\widehat{\xi}_0^i$ performs a local random move $\widehat{\xi}_0^i \rightsquigarrow \xi_1^i$ (independently of one another) according to the Markov transition M_{β_1}. Then, we perform another selection transition $\xi_1 \rightsquigarrow \widehat{\xi}_1 := \left(\widehat{\xi}_1^i\right)_{1\leq i\leq N}$ using the fitness functions G_1 defined below:

$$G_1(x) = \exp\left(-(\beta_2 - \beta_1)V(x)\right)$$

After this selection stage we mutate each selected individual using the Markov transition M_{β_2}, and so on. Iterating these transitions, we define a simple genetic model with mutation transitions M_{β_n} and selection fitness functions G_n:

$$\xi_n := \left(\xi_n^i\right)_{1\leq i\leq N} \in E^N \xrightarrow{\text{selection}} \widehat{\xi}_n := \left(\widehat{\xi}_n^i\right)_{1\leq i\leq N} \in E^N \xrightarrow{\text{mutation}} \xi_{n+1} \in E^N \tag{1.54}$$

This algorithm was first proposed in [40]. A variety of convergence results can be established for this algorithm. For instance, for any function f on E and any time horizon, we have

$$\lim_{N\to\infty} \frac{1}{N} \sum_{1\leq i\leq N} f(\xi_n^i) = \sum_x \mu_{\beta_n}(x)\, f(x)$$

In addition, if we set $\eta_n^N(G_n) := \frac{1}{N}\sum_{1\leq i\leq N} G_n(\xi_n^i)$, the unbiased N-particle approximation $\mathscr{Z}_{\beta_n}^N$ of the normalizing constants $\mathscr{Z}_{\beta_n}$ is given by

$$\mathscr{Z}_{\beta_n}^N := \prod_{0\leq p<n} \eta_p^N(G_p) \longrightarrow_{N\to\infty} \mathscr{Z}_{\beta_n}$$

1.5.4.3 Combinatorial Counting and Sampling

Suppose we want to compute the cardinality of a given subset A of some finite state space E equipped with the counting measure $\lambda(x) := \frac{1}{\mathrm{Card}(E)}$. Once again, from a probabilistic point of view, this problem is equivalent to computing the normalizing constant of the following Boltzmann-Gibbs distribution

$$\mu_A(x) := \frac{1}{\mathscr{Z}_A}\ \mathbb{1}_A(x)\ \lambda(x) \quad \text{with} \quad \mathscr{Z}_A := \sum_x\ \mathbb{1}_A(x)\ \lambda(x)$$

To sample from μ_A and compute $\mathscr{Z}_A$, the idea consists of selecting a judicious sequence of decreasing subsets A_n in such a way that it is easy to sample states in A_n starting from the set A_{n-1}. We suppose that $A_0 = E$ so that μ_{A_0} coincides with the uniform counting measure on the set E. The algorithm is thus very similar to the one described previously for optimization. For any set A_n, we introduce an MCMC transition matrix $M_{A_n} = (M_{A_n}(x,y))_{x,y\in E}$ with left eigenvector μ_{A_n} associated with the eigenvalue 1, that is

$$\sum_x \mu_{A_n}(x) M_{A_n}(x,y) = \mu_{A_n}(y)$$

A simple Metropolis-Hasting type transition associated with a symmetric transition matrix $K(x,y) = K(y,x)$ is given by

$$M_{A_n}(x,y) = K(x,y)\ \mathbb{1}_{A_n}(y) + \left(1 - \sum_z K(x,z)\ \mathbb{1}_{A_n}(z)\right)\ \mathbb{1}_x(y)$$

The N-particle stochastic algorithm is defined as follows. We start with N independent random individuals $\xi_0 := \left(\xi_0^i\right)_{1\leq i\leq N}$ randomly chosen in E with μ_{A_0}. We perform a selection transition $\xi_0 \rightsquigarrow \widehat{\xi}_0 := \left(\widehat{\xi}_0^i\right)_{1\leq i\leq N}$ using the fitness functions $G_0 = \mathbb{1}_{A_1}$. In other words, every individual in the set A_1 stays in the same place $\widehat{\xi}_0^i = \xi_0^i$, otherwise we replace ξ_0^i by a fresh new individual $\widehat{\xi}_0^i = \xi_0^j$ randomly chosen among the individuals $\xi_0^j \in A_1$. When no individuals ξ_0^j are in the set A_1, the algorithm stops and we set $\tau^N = 0$. Assuming that $\tau^N > 0$, during the mutation step $\widehat{\xi}_0 \rightsquigarrow \xi_1 := \left(\xi_1^i\right)_{1\leq i\leq N}$, every selected individual $\widehat{\xi}_0^i$ performs a local random move $\widehat{\xi}_0^i \rightsquigarrow \xi_1^i$ (independently of one another) in the set A_1 according to the Markov transition M_{A_1}. Then, we perform another selection transition $\xi_1 \rightsquigarrow \widehat{\xi}_1 := \left(\widehat{\xi}_1^i\right)_{1\leq i\leq N}$ using the fitness functions $G_1 = \mathbb{1}_{A_2}$. When no individuals ξ_1^j are in the set A_2, the algorithm stops and we set $\tau^N = 1$. After this selection stage we mutate each selected individual using the Markov transition M_{A_2}, and so on. For any function f on E and any time horizon n, we have

$$\lim_{N\to\infty}\frac{1}{N}\sum_{1\leq i\leq N} f(\xi_n^i)\mathbb{1}_{\tau^N>n}=\sum_x \mu_{A_n}(x)\,f(x)$$

In addition, if we set $\eta_n^N(G_n):=\frac{1}{N}\sum_{1\leq i\leq N}G_n(\xi_n^i)$, the proportion of individuals in A_{n+1} after the n-th mutation, the unbiased N-particle approximation $\mathscr{Z}_{A_n}^N$ of the normalizing constants $\mathscr{Z}_{A_n}$ is given by

$$\mathscr{Z}_{A_n}^N:=\prod_{0\leq p<n}\eta_p^N(G_p)\longrightarrow_{N\to\infty}\mathscr{Z}_{A_n}=\mathrm{Card}(A_n)/\mathrm{Card}(E)$$

1.5.4.4 Genetic Search Algorithms

We consider an energy function or a cost criteria $V:\ x\in E\mapsto(x)$ on some finite state space E where we assume $\inf_x V(x)=0$ without loss of generality. The objective is to find the global minima points $x^\star\in E$ s.t. $V(x^\star)=\inf_{x\in E}V(x)$. Let $V^\star$ denote the set of these points. We describe in Section 1.5.4.2 an interacting particle algorithm to solve this problem which relies on interacting simulated annealing type chains. We present here the more standard genetic algorithm with mutation and proportional selection.

To construct this algorithm, we introduce a collection of Markov transitions $M_n(x,y)$ from E into itself. This collection of transition matrices represents the probability $M_n(x,y)$ that a individual at site x evolves to a new state x during the n-th mutation transition.

The genetic algorithm with N individuals is defined as follows. We start with N independent random individuals $\xi_0:=\left(\xi_0^i\right)_{1\leq i\leq N}$ randomly chosen in E with some distribution η_0. We perform a proportional type selection transition $\xi_0\rightsquigarrow\widehat{\xi}_0:=\left(\widehat{\xi}_0^i\right)_{1\leq i\leq N}$ using the potential functions $G_0\left(\xi_0^i\right)=\exp\left(-\beta_0 V\left(\xi_0^i\right)\right)$, where $\beta_0\geq 0$ is an inverse temperature parameter. In other words, with probability $G_0(\xi_0^i)$ every individual stays in the same place $\widehat{\xi}_0^i=\xi_0^i$; otherwise, we replace ξ_0^i by a new individual $\widehat{\xi}_0^i=\xi_0^j$ randomly chosen among the individuals ξ_0^j with a probability proportional to its weight $G_0(\xi_0^i)$. Formally, we set

$$\widehat{\xi}_0^i=\varepsilon_0^i\ \xi_0^i+\left(1-\varepsilon_0^i\right)\ \widetilde{\xi}_0^i$$

where ε_0^i stands for a sequence of independent $\{0,1\}$-valued Bernoulli random variables with distributions

$$G_0(\xi_0^i):=\mathrm{Proba}\left(\varepsilon_0^i=1\mid\xi_0\right)=1-\mathrm{Proba}\left(\varepsilon_0^i=0\mid\xi_0\right)$$

and $\widetilde{\xi}_0:=\left(\widetilde{\xi}_0^i\right)_{1\leq i\leq N}$ are independent, identically distributed and $\left\{\xi_0^j,\ 1\leq j\leq N\right\}$-valued random variables with common distributions given for any index $1\leq i\leq N$ by

$$\forall 1 \leq j \leq N \qquad \text{Proba}\left(\widetilde{\xi}_0^i = \xi_0^j \mid \xi_0\right) = G_0(\xi_0^j)/\sum_{1\leq j\leq N} G_0(\xi_0^j)$$

During the mutation step $\widehat{\xi}_0 \rightsquigarrow \xi_1 := \left(\xi_1^i\right)_{1\leq i\leq N}$, every selected individual $\widehat{\xi}_0^i$ performs a local random move $\widehat{\xi}_0^i \rightsquigarrow \xi_1^i$ (independently of one another) according to the Markov transition M_1. Then, we perform another proportional type selection transition $\xi_1 \rightsquigarrow \widehat{\xi}_1 := \left(\widehat{\xi}_1^i\right)_{1\leq i\leq N}$ using the potential functions $G_1\left(\xi_1^i\right) = \exp\left(-\beta_1 V\left(\xi_1^i\right)\right)$, where $\beta_1 \geq 0$ is another inverse temperature parameter, and so on. We define in this way a sequence of genetic type populations $\xi_n, \widehat{\xi}_n$, as in (1.28) and the corresponding genealogical tree model (1.30) associated with the ancestral lines $\left(\xi_{p,n}^i\right)_{0\leq p\leq n}$ of every i-th individuals after the n-th mutation. In the same way, running back in time we have the whole ancestral line

$$\widehat{\xi}_{0,n}^i \longleftarrow \widehat{\xi}_{1,n}^i \longleftarrow \ldots \longleftarrow \widehat{\xi}_{n-1,n}^i \longleftarrow \widehat{\xi}_{n,n}^i = \widehat{\xi}_n^i \tag{1.55}$$

of every i-th individual after the n-th selection.

For any function f on E_n and any time horizon n, we can prove that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^N f(\widehat{\xi}_{0,n}^i, \widehat{\xi}_{1,n}^i, \ldots, \widehat{\xi}_{n,n}^i) = \frac{\mathbb{E}\left(f_n(X_0,\ldots,X_n)\ \exp\left(-\sum_{0\leq p\leq n}\beta_p\, V(X_p)\right)\right)}{\mathbb{E}\left(\exp\left(-\sum_{0\leq p\leq n}\beta_p\, V(X_p)\right)\right)}$$

In other words, the proportion of paths $(\widehat{\xi}_{0,n}^i, \widehat{\xi}_{1,n}^i, \ldots, \widehat{\xi}_{n,n}^i)$ taking some value $(x_0,\ldots,x_n)$ is given by

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^N \mathbb{1}_{(x_0,\ldots,x_n)}(\widehat{\xi}_{0,n}^i, \widehat{\xi}_{1,n}^i, \ldots, \widehat{\xi}_{n,n}^i) = \frac{1}{\mathscr{Z}_{n+1}}\ \exp\left(-\sum_{0\leq p\leq n}\beta_p\, V(x_p)\right)\ \mathbb{P}_n(x_0,\ldots,x_n)$$

with the probability of a free evolution path involving only mutation transitions

$$\mathbb{P}_n(x_0,\ldots,x_n) = \eta_0(x_0)M_1(x_0,x_1)\ldots M_n(x_{n-1},x_n)$$

where $\mathscr{Z}_{n+1}$ is a normalizing constant.

Suppose that every free evolution path has the same chance to be sampled, in the sense that

$$\mathbb{P}_n(x_0,\ldots,x_n) = \mathbb{P}_n(y_0,\ldots,y_n)$$

for any admissible pair of paths $(x_0,\ldots,x_n)$ and $(y_0,\ldots,y_n)$. This condition is satisfied if η_0 is the uniform counting measure on E and the mutation transitions $M_n(x,y)$ correspond to local random choices of the same number of neighbors, starting from any state x. In this case, for any admissible path $(x_0,\ldots,x_n)$ we have that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^N \mathbb{1}_{(x_0,\ldots,x_n)}(\widehat{\xi}_{0,n}^i, \widehat{\xi}_{1,n}^i, \ldots, \widehat{\xi}_{n,n}^i) = \frac{1}{\mathscr{Z}'_n}\ \exp\left(-\sum_{0\leq p\leq n}\beta_p\, V(x_p)\right)$$

for some normalizing constant $\mathscr{Z}'_n$. When the inverse temperature parameter β_p increases the r.h.s. probability mass quantity only charges admissible paths $(x_0,\ldots,x_n)$ that minimize the path potential function

$$\mathscr{V}_n(x_0,\ldots,x_n) = \inf_{(y_0,\ldots,y_n)} \sum_{0\leq p\leq n} V(y_p)$$

In other words at low temperatures, the ancestral lines of the simple genetic model described above converge to the uniform measure on all the paths $(x_0,\ldots,x_n)$ of length n that minimize the energy function $\mathscr{V}_n$. For time homogenous mutation transitions associated with stochastic matrices $M_n(x,y) = M(x,y)$ satisfying the following condition for some integer $m \geq 1$ and any pair $(x,y) \in E^2$

$$M(x,x) > 0 \quad \text{and} \quad M^m(x,y) \geq \varepsilon M^m(x,z)$$

we also have the convergence result

$$\lim_{n\to\infty} \lim_{N\to\infty} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{V^\star}(\widehat{\xi}^i_n) = 1$$

as soon as $\beta_n = C\ \log(n+1)$ for some constant C that depends on m and on the oscillations of the function V. This convergence result is also true for $\beta_n = C\ (n+1)^\alpha$, with any $\alpha \in]0,1[$, as soon as the above condition is met for $m = 1$.

Further details on these concentration properties can be found in [55]. Related convergence results for fixed population sizes can be found in [24]. To give a flavor of these results, let us suppose that the mutation transitions $M_n(x,y)$ also depend on the inverse temperature parameter and

$$M_n(x,y) \to_{n\to\infty} \mathbb{1}_x(y) \quad \text{as} \quad \beta_n \uparrow \infty$$

Intuitively speaking, the genetic mutations become rare transitions at low temperatures. In this situation, we can prove that there exists a "critical population size" $N^\star$ that depends on the energy function as well as on the free evolution model such that

$$\forall N \geq N^\star \qquad \lim_{n\to\infty} \text{Proba}\left(\forall 1 \leq i \leq N \quad \widehat{\xi}^i_n \in V^\star\right) = 1$$

1.5.5 Analysis of Convergence under Uncertain Behavior

The following analysis focuses on a particular class of genetic type algorithms for which it is assumed that operators have a nonzero probability of erroneous or uncertain behavior. A direct example may be found in practice for distributed environments where remote nodes carry part of the steps of the algorithm and where nodes are prone to processing or communication errors and malicious behavior. Different

questions arise in this context on the influence of erroneous (or abnormal) operation on the convergence of the algorithm.

In this work we do not concentrate on proving that we have a probability one in reaching the optimal solution when time goes to infinity, given a fixed population size, as these results are already present in the literature [145, 146], but rather on bounding the probability that the obtained results are within a certain error threshold. Another line of research is concerned with results using Feynman-Kac representations, focusing on the asymptotic stability and uniform convergence of genetic algorithms [47]. Note that in the following we will address convergence in finite spaces.

Finally, connections to dynamic optimization or in the presence of uncertainties could be made by considering noise or time dependent external factors as being an integrated part of how the operators function.

Let $(X_n)_{n\geq 0}$, $X_n \in E$, be a Markov chain, with E being an arbitrary space, for which a transition kernel is given as

$$M(x,dy) = \mathbb{P}(X_n \in dy | X_{n-1} = x).$$

Assumption A1: There exists ν, a probability measure over E, $\lambda > 0$, $m \geq 1$ s. t.

$$\mathbb{P}(X_m \in dx | X_0 = x_0) \geq \lambda \nu(dx)$$

Example 1.5.1. *Let $E = \{x_1, x_2, \ldots, x_d\}$ be a finite space and $M(x,y) \geq \delta > 0$ a Markov transition $\left(M(x,y) \geq \delta d \times \frac{1}{d}\right)$. Having $\nu(y) = \frac{1}{d}$ a uniform measure over E and by denoting with $\lambda = \delta d$, we obtain that*

$$M(x,y) = \mathbb{P}(X_1 = y | X_0 = x) \geq \lambda \nu(y).$$

Example 1.5.2. *Let $E = \{x_1, x_2, \ldots, x_d\}$ be a finite space and $M^m(x,y)$ a Markov transition involving the application of the M kernel m times, with $M^m(x,y) \geq \delta > 0 \Leftrightarrow M^m(x,y) \geq \delta d \times \frac{1}{d}$. By denoting as previously $\lambda = \delta d$, we obtain thus*

$$M^m(x,y) = \mathbb{P}(X_m = y | X_0 = x) \geq \lambda \ \ \nu(y).$$

Under **Assumption (A1)**, it is well known that there exists an unique probability measure π s.t. $\pi M = \pi$. This measure π is said to be an invariant measure.

Remark 1.5.3. *We further assume that E is finite and **Assumption (A1)** is met for $m = 1$, and some measure ν s.t. $\nu(x) > 0$ for any $x \in E$. We also let π be the invariant measure for the Markov chain of transition M. In this case, for any $x \in E$ we clearly have that:*

$$\pi(x) = \sum_y \pi(y) M(y,x) > 0, \forall x$$

The same goes for the case involving m successive transition steps (see Example 1.5.2.):

$$\pi(x) = \sum_y \pi(x) M^m(x,y) > 0, \forall x, \text{ as } M^m(x,y) > 0, \forall x.$$

For more general state spaces E, for any measurable subset $A \subset E$ we have

$$\pi(A) = \int_A \pi(dx) = \int \pi(dx) M^m(x,A) > 0, \forall x, \text{ as } M^m(x,A) > 0, \forall x.$$

Notation: We let $\pi^n = \frac{1}{n} \sum_{0 \leq p < n} \delta_{X_p}$ be the occupation measure of the Markov chain $(X_p)_{p \geq 0}$ at time n, starting at some initial state, say $X_0 = x_0$. For any bounded measurable function f on E, we set

$$\pi^n(f) = \int f(x)\ \pi^n(dx) = \frac{1}{n} \sum_{0 \leq p < n} f(X_p) \quad \text{and} \quad \pi(f) = \int f(x)\ \pi(dx)$$

In order to obtain stronger bounds we will base our further investigations on the result presented in [86], adjusted to our context, i.e. $\|f\| = 1$ and using the aforementioned notation.

Theorem 1.5.4. (Glynn and Ormoneit [86]) *Under the conditions of **Assumption (A1)**, for any bounded measurable function f s.t. $\|f\| = 1$, and for any $n > 2m/(\lambda\varepsilon)$ and $\varepsilon > 0$, we have that*

$$\mathbb{P}\left(\pi^n(f) - \mathbb{E}(\pi^n(f)) \geq \varepsilon\right) \leq e^{-\frac{\lambda^2 (n\varepsilon - 2m/\lambda)^2}{2nm^2}}$$

When considering the absolute value, it is implied moreover that

$$\mathbb{P}\left(|\pi^n(f) - \mathbb{E}(\pi^n(f))| \geq \varepsilon\right) \leq 2 \times e^{-\frac{\lambda^2}{2m^2}\left(\varepsilon - \frac{2m}{\lambda n}\right)^2} \tag{1.56}$$

Example 1.5.5. *As a direct application of this result, for $m = 1$ and $\forall n > 2/(\lambda\varepsilon)$, we have that*

$$\mathbb{P}\left(\pi^n(f) - \mathbb{E}(\pi^n(f)) \geq \varepsilon\right) \leq e^{-\frac{\lambda^2}{2n}\left(n\varepsilon - \frac{2}{\lambda}\right)^2} = e^{-\frac{\lambda^2 n}{2}\left(\varepsilon - \frac{2}{\lambda n}\right)^2}$$

We further assume that **Assumption (A1)** is met for some parameters $\lambda > 0$ and $m \geq 1$, and some probability measure ν on E. We notice that

$$\mathbb{E}\left(\frac{1}{n} \sum_{0 \leq p < n} f(X_p)\right) = \frac{1}{n} \sum_{0 \leq p < n} \mathbb{E}(f(X_p)) = \frac{1}{n} \sum_{0 \leq p < n} M^p(f)(x_0).$$

On the other hand, under our assumptions it is well known that

$$\sup_{x_0, y_0} |M^p(f)(x_0) - M^p(f)(y_0)| \leq c_1(m)\ e^{-c_2(m)\ p}$$

for any $p \geq 0$, and any measurable function f on E s.t. $\|f\| = 1$, for some nonnegative and finite constants $c_1(m)$ and $c_2(m)$ whose values only depend on the parameters λ and m. For a detailed proof of these inequalities we refer the reader to [38].

Thus, recalling that $\pi = \pi M^p$, for any $p \geq 0$, the following relations can be derived:

$$\mathbb{E}(\pi^n(f)) - \pi(f) = \frac{1}{n} \sum_{0 \leq p < n} [M^p(f)(x_0) - \pi(f)] = \frac{1}{n} \sum_{0 \leq p < n} [M^p(f)(x_0) - \pi M^p(f)]$$

so that

$$\begin{aligned} |\mathbb{E}(\pi^n(f)) - \pi(f)| &= \frac{1}{n} \left| \sum_{0 \leq p < n} [M^p(f)(x_0) - \pi M^p(f)] \right| \\ &\leq c_1(m) \times \frac{1}{n} \sum_{0 \leq p < n} e^{-c_2(m)\ p} \leq c_3(m)/n \end{aligned}$$

for some constant

$$c_3(m) \leq c_1(m)/(1 - e^{-c_2(m)})$$

Considering these results with **Theorem** 1.5.4. one can conclude that

$$\begin{aligned} \mathbb{P}(|\pi^n(f) - \pi(f)| \geq \varepsilon) &\leq \mathbb{P}(|\pi^n(f) - \mathbb{E}(\pi^n(f))| + |\mathbb{E}(\pi^n(f)) - \pi(f)| \geq \varepsilon) \\ &\leq \mathbb{P}(|\pi^n(f) - \mathbb{E}(\pi^n(f))| \geq \varepsilon - c_3(m)/n) \\ &\leq 2e^{-\frac{\lambda^2 n}{2}(\varepsilon - [c_3(m) + 2/\lambda]/n)^2} \end{aligned} \tag{1.57}$$

for any $n \geq 1$ and any $\varepsilon > 0$ such that $\varepsilon > [c_3(m) + 2/\lambda]/n$. We summarize the above discussion with the following corollary:

Corollary 1.5.6. *Under the conditions of **Assumption (A1)**, for any bounded measurable function f s.t. $\|f\| = 1$, any $\varepsilon > 0$ and any $n > [c_3(m) + 2/\lambda]/\varepsilon$, we have the exponential concentration inequality*

$$\mathbb{P}(|\pi^n(f) - \pi(f)| \geq \varepsilon) \leq 2e^{-\frac{\lambda^2 n}{2}(\varepsilon - [c_3(m) + 2/\lambda]/n)^2}$$

with some finite constant $c_3(m) \leq c_1(m)/(1 - e^{-c_2(m)})$.

1.5.5.1 Application in Optimization and Archive Models

As already mentioned, the existing results are mainly intended on the study of the behavior in an optimization environment, focusing on the limiting behavior [145, 146] as well as the limit probability distribution over populations as depicted in [147]. Further results, see for instance [130], deduce properties of the stationary distribution of the Markov chain associated with the evolutionary process by con-

structing a quotient chain associated with the original chain. The advantage offered by the herein depicted new result resides in the fact that it is based on an assumption that concerns the overall transitions, without being bounded to specific types of operators (mutation, selection) and without requiring any additional stronger assumptions. In fact, it provides the means of building specific transition operators that need only to satisfy the conditions from assumption (A1).

In order to apply the results presented in the previous section, in an optimization context, we consider a finite state space E for which $|E| = d$ and an objective function V having the set of optimal solutions defined as $V^* = \{x|V(x) = \inf(V)\} = \{x_1^*, x_2^*, \ldots, x_{d^*}^*\}$.

The aforementioned existing convergence results from [145, 146] were applied in an optimization context and considered the study of the behavior of the algorithm in limit, when time tends ot infinity. In the current case we consider the probability of deviation from the invariant measure to be bounded by a positive vlaue ε and establish bounds on these probability.

Let us also assume π an invariant measure such that $\forall i = \{1, 2, \ldots, d^*\}$, $\pi(x_i^*) > 0$ and $\pi(V^*) = \sum_{i=1}^{d^*} \pi(x_i^*)$. Let us further consider the optimization context modeled as

$$\begin{cases} X_n^* & = Argmin\{V(X_0), V(X_1), \ldots, V(X_n)\}, \\ V(X_n^*) & = \min\left(V(X_{n-1}^*), V(X_n)\right), \end{cases} \tag{1.58}$$

where $X_{n-1}^* \in V^*$ is the equivalent of having $\frac{1}{n}\sum_{i=0}^{n-1} \mathbb{1}_{V^*}(X_i) > 0$.

When applying the previously obtained results from equation (1.57), for $f = \mathbb{1}_{V^*}$ and by considering the measure π on V^*, the following holds:

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=0}^{n-1} \mathbb{1}_{V^*}(X_i) - \pi(V^*)\right| \geq \varepsilon\right) \leq 2e^{-\frac{\lambda^2 n}{2}(\varepsilon - [c_3(m)+2/\lambda]/n)^2} \tag{1.59}$$

for any $n > [c_3(m) + 2/\lambda]/\varepsilon$. Adopting an opposite perspective, i.e. for the probability of having a deviation smaller than a given threshold, the following expression is derived:

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=0}^{n-1} \mathbb{1}_{V^*}(X_i) - \pi(V^*)\right| < \varepsilon\right) \geq 1 - 2e^{-\frac{\lambda^2 n}{2}(\varepsilon - [c_3(m)+2/\lambda]/n)^2} \tag{1.60}$$

for any $n > [c_3(m) + 2/\lambda]/\varepsilon$. At the same time, without any loss of generality, we consider $\varepsilon = \varepsilon' \pi(V^*)$, with $\varepsilon' \in [0, 1[$, which leads to

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=0}^{n-1} \mathbb{1}_{V^*}(X_i) - \pi(V^*)\right| < \varepsilon\right) \leq \mathbb{P}\left(\frac{1}{n}\sum_{i=0}^{n-1} \mathbb{1}_{V^*}(X_i) \geq \pi(V^*) - \varepsilon\right) \tag{1.61}$$

where the right part of the expression can be rewritten as follows:

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=0}^{n-1}\mathbb{1}_{V^*}(X_i) \geq \pi(V^*)(1-\varepsilon')\right) \geq 1-2\times e^{-\frac{\lambda^2 n}{2}\left(\varepsilon'\pi(V^*)-[c_3(m)+2/\lambda]/n\right)^2} \tag{1.62}$$

for any $n > [c_3(m)+2/\lambda]/\varepsilon'\pi(V^*)$. Nonetheless, given that the following stands, in relation with the above result, we conclude the following

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=0}^{n-1}\mathbb{1}_{V^*}(X_i) \geq \pi(V^*)(1-\varepsilon')\right) \leq \mathbb{P}(X_n^* \in V^*) \tag{1.63}$$

and

$$\mathbb{P}(X_n^* \in V^*) \geq 1-2\times e^{-\frac{\lambda^2 n}{2}\left(\varepsilon'\pi(V^*)-[c_3(m)+2/\lambda]/n\right)^2}$$

for any $0 < \varepsilon' < 1$ and any $n > [c_3(m)+2/\lambda]/\varepsilon'\pi(V^*)$. For instance, taking $\varepsilon' = 1/2$ we find that

$$n > 4\left(c_3(m)+2/\lambda\right)/\pi(V^*) \Rightarrow \pi(V^*)/2 - [c_3(m)+2/\lambda]/n > \pi(V^*)/4$$

From these observations, we obtain the following theorem:

Theorem 1.5.7. *Under the conditions of **Assumption (A1)**, we have that*

$$\forall n > 4\frac{c_3(m)+2/\lambda}{\pi(V^*)}, \qquad \mathbb{P}(X_n^* \in V^*) \geq 1-2e^{-n\,(\lambda\pi(V^*))^2/32}$$

The above result clearly shows that convergence is attained exponentially fast as $n \to \infty$. As a consideration for application in practice, if $\pi(V^*)$ or λ are close to zero, a large value is required for n, i.e. the algorithm needs a large number of iterations in order to converge. The current result reaches generality as it provides clear bounds on the probability that the evolutionary algorithm modeled as a Markov process, approaches the actual global optima of the optimization problem, without focusing on limit properties when time goes to infinity and without considering the absence of mutation/selection [129].

1.5.5.2 Bounds on Perturbed Processes

The current section aims at quantifying the error that a given stochastic perturbation has on the behavior of a genetic algorithm. Perturbations are considered to be induced, for example, as a result of external stochastic factors affecting the transition kernel and/or the selection kernel based on the use of a potential function. While the semantics of what exactly *perturbed behavior* means are widely open, e.g. some functional error, malicious behavior, etc., we will only consider that, with

some known probability, the operators behave in some different manner than what is expected. In order to model this behavior we consider a genetic algorithm for which the transition kernel and the potential function are given by M_n, respectively G_n. A perturbed version of the algorithm, in the limits of the previous terms, is considered to be defined on M_n^ε and G_n^ε, where, with some fixed probability ε, the behavior of the transition kernel, for example, is different than what the M_n kernel models, while with probability $1-\varepsilon$, the M_n kernel applies. An analogous definition is considered for G_n^ε, i.e. with probability ε the potential of a solution is given by G_n^ε, otherwise, with probability $1-\varepsilon$, being given by G_n. Examples may be found in practice, e.g. algorithms executed across volatile resources or with the support of external, unreliable participants that offer or share computational power, and where, due to failures or malicious behavior, the way different operators act can not be ensured – a brief outline and discussion is offered by the end of this section.

A question that one may ask is, knowing that M_n and M_n^ε are comparable up to some constant, what impact on convergence does the M_n^ε transition kernel have, i.e. is there a significant difference between $\eta_n^{\varepsilon,N}(f)$ and $\eta_n^N(f)$, do the algorithms converge to similar or comparable results? A similar remark can be raised by observing the effect G_n^ε has on convergence. An analysis of both cases is presented in the following, within some assumptions on the relative difference of M_n^ε and M_n, respectively of G_n^ε and G_n.

We recall that the total variance distance is defined by

Definition 1.5.8. (Total variance distance)

$$\|\mu-\nu\|_{tv} = \sup_{f:\omega(f)\leq 1} |\mu(f)-\nu(f)|$$

Given a positive and bounded potential function G_n on E, we start by introducing the mappings $(\phi_n)_{n\geq 1}$, $(\psi_{G_n})_{n\geq 0}$, respectively $(\phi_n)_{n\geq 1}^\varepsilon$ and $(\psi_{G_n})_{n\geq 0}^\varepsilon$ from $P(E)$ into itself, with M_n being some Markov transition; ψ_{G_n} can be, for example, a Boltzmann-Gibbs mapping.

$$\begin{cases} \phi_{n+1}^\varepsilon(\eta) &= \Psi_{G_n^\varepsilon}(\eta)M_{n+1}^\varepsilon \\ \phi_{n+1}(\eta) &= \Psi_{G_n}(\eta)M_{n+1} \end{cases} \tag{1.64}$$

In order to quantify the difference in behavior of the two different variants, we would like to estimate $\phi_{n+1}^\varepsilon(\eta)-\phi_{n+1}(\eta)$, which can be further decomposed as

$$\begin{aligned} \phi_{n+1}^\varepsilon(\eta)-\phi_{n+1}(\eta) &= \Psi_{G_n^\varepsilon}(\eta)M_{n+1}^\varepsilon - \Psi_{G_n}(\eta)M_{n+1} \\ &= \Psi_{G_n^\varepsilon}(M_{n+1}^\varepsilon - M_{n+1}) + [\Psi_{G_n^\varepsilon}(\eta)-\Psi_{G_n}(\eta)]M_{n+1} \end{aligned} \tag{1.65}$$

By denoting with $\nu = \Psi_{G_n}(\eta)$, we notice that

$$\begin{aligned} \Psi_{G_n^\varepsilon}(\eta)-\Psi_{G_n}(\eta) &= [\Psi_{G_n^\varepsilon/\ G_n}(\nu)-\nu](f) \\ &= \frac{1}{\nu(G_n^\varepsilon/\ G_n)}\nu\left(\left(\frac{G_n^\varepsilon}{G_n}\right)[f-\nu(f)]\right) \end{aligned} \tag{1.66}$$

Next, we denote the oscillation of a function f by $\omega(f)$ and use it to define two working hypothesis, as described in the following:

Hypothesis H 1. *The Markovian transition kernels M_n^ε and M_n differ up to some constant c_1, the ε probability of having some alternative behavior than what is expected, and the oscillation of f*

$$\|M_n^\varepsilon(f) - M_n(f)\| \leq c_1 \varepsilon \omega(f) \tag{1.67}$$

Example 1.5.9. *A simple example can be constructed by defining M_n^ε as $M_n^\varepsilon = \varepsilon_n K_n + (1-\varepsilon_n)M_n$, with $\varepsilon = \sup_n(\varepsilon_n)$ and $K_n(x,dy) = \delta_n(dy)$, M_n Markov transitions. Following this rationale, the following relation can be inferred:*

$$\begin{aligned} M_n^\varepsilon(f)(x) &= \varepsilon_n K_n(f)(x) + (1-\varepsilon_n)M_n(f)(x) \\ M_n^\varepsilon(f)(x) - M_n(f)(x) &= \varepsilon_n(K_n(f)(x) - M_n(f)(x)) \\ |M_n^\varepsilon(f)(x) - M_n(f)(x)| &\leq \varepsilon_n \left| \iint K_n(x,dy)M_n(x,dz)(f(y)-f(z)) \right| \end{aligned}$$

Nonetheless, as $f(y)-f(z) \leq \omega(f)$, it directly follows that $|M_n^\varepsilon(f)(x) - M_n(f)(x)| \leq \varepsilon_n \omega(f)$ which, when taking $\varepsilon = \sup_n \varepsilon_n$, leads to the following relation:

$$\sup_n |M_n^\varepsilon(f)(x) - M_n(f)(x)| = \|M_n^\varepsilon(f) - M_n(f)\| \leq \varepsilon \omega(f).$$

Hypothesis H 2. *The difference between the potential functions is bounded, meaning:*

$$\|G_n^\varepsilon / G_n\| \leq c_2\ \varepsilon \tag{1.68}$$

which is equivalent to

$$1 - c_2\ \varepsilon \leq G_n^\varepsilon / G_n \leq 1 + c_2\ \varepsilon \tag{1.69}$$

Example 1.5.10. *As a direct example we can take $G_n^\varepsilon = e^{-V_n^\varepsilon}$ and $G_n = e^{-V_n}$, leading to $G_n^\varepsilon / G_n = e^{-(V_n^\varepsilon - V_n)}$. The expression can be rewritten to read $|G_n^\varepsilon/G_n - 1| = |e^{V_n - V_n^\varepsilon} - e^0|$ where the following stands:*

$$|e^{(V_n - V_n^\varepsilon)(x)} - e^0| \leq |(V_n - V_n^\varepsilon)(x)| \times e^{\|V_n\| + \|V_n^\varepsilon\|}$$

Further, knowing that $\|V_n\| < \infty$ and that $\sup_\varepsilon \|V_n^\varepsilon\| < \infty$, there exists a constant v_1 such that $\|V - V^\varepsilon\| \leq \varepsilon v_1$. Next, given that $e^{\|V_n\| + \|V_n^\varepsilon\|} \leq e^{2\max(\|V_n\|,\, \sup_\varepsilon \|V_n^\varepsilon\|)}$, the following relation holds:

$$\|G_n^\varepsilon / G_n - 1\| \leq \varepsilon\ c_2,\ \textit{where}\ c_2 \leq v_1\ e^{2\max(\|V_n\|,\, \sup_\varepsilon \|V_n^\varepsilon\|)}.$$

Example 1.5.11. *We adopt a different perspective, considering that* $V_n^\varepsilon = (1-\varepsilon)V_n + \varepsilon W_n$, *or in a different form, that* $V_n^\varepsilon - V_n = \varepsilon(W_n - V_n)$. *Knowing that* $\sup_n \|W_n\| < \infty$ *and that* $\sup_n \|V_n\| < \infty$, *we obtain* $\|V_n^\varepsilon - V_n\| \leq \varepsilon c_2$, *with* $c_2 = \sup_n \|V_n\| + \sup_n \|W_n\|$. *In analogous manner we have the following:*

$$\|V_n^\varepsilon\| \leq (1-\varepsilon)\sup_n \|V_n\| + \varepsilon \sup_n \|W_n\| \leq \max(\sup_n \|V_n\|, \sup_n \|W_n\|).$$

From the second hypothesis **(H2)**, for any $\varepsilon < \frac{1}{2c_2}$ we obtain that

$$\begin{aligned} [\Psi_{G_n^\varepsilon/\ G_n}(\nu)(f) - \nu](f) &\leq \frac{c_2\varepsilon}{1-c_2\varepsilon}\nu(|f-\nu(f)|) \leq \frac{c_2\varepsilon}{1-c_2\varepsilon}, \\ \|\Psi_{G_n^\varepsilon}(\eta) - \Psi_{G_n}(\eta)\|_{tv} &= \sup_{f\,:\,\omega(f)\leq 1} |\Psi_{G_n^\varepsilon}(\eta)(f) - \Psi_{G_n}(\eta)(f)| \leq \frac{c_2\varepsilon}{1-c_2\varepsilon}, \end{aligned} \tag{1.70}$$

Furthermore, equation (1.65), defining the difference between two mappings, can be rewritten as:

$$\begin{aligned} \|\phi_{n+1}^\varepsilon(\eta) - \phi_{n+1}(\eta)\|_{tv} &= c_1\varepsilon\omega(f) + \frac{c_2\varepsilon}{1-c_2\varepsilon}\omega(M_n(f)) \\ &\leq c_1\varepsilon + \frac{c_2\varepsilon}{1-c_2\varepsilon} \quad \text{as soon as } \omega(f) \leq 1 \\ &\leq (c_1+2c_2)\varepsilon \quad \text{for any } \ \varepsilon < 1/(2c_2). \end{aligned} \tag{1.71}$$

In summary, we have proved the following technical lemma.

Lemma 1.5.12. *In the conditions defined by hypotheses* ***(H1)*** *and* ***(H2)****, for any* $\varepsilon < 1/(2c_2)$ *and any probability measure* η *we obtain that*

$$\|\phi_{n+1}^\varepsilon(\eta) - \phi_{n+1}(\eta)\|_{tv} \leq (c_1+2c_2)\varepsilon$$

Let (γ_n, η_n) be the Feynman-Kac model associated with the potential function G_n and the transition kernel M_n, representing the normalized, respectively unnormalized Feynman-Kac measures. At first we consider defining the sequential update of the flow of distributions $(\eta_n)_{n\geq 0}$ (standard case) and $(\eta_n^\varepsilon)_{n\geq 0}$ (perturbed variant):

$$\eta_n^\varepsilon = \phi_n^\varepsilon(\eta_{n-1}^\varepsilon), \tag{1.72}$$

where η_n is the measure associated with N independent samples of common law $\phi_n(\eta_{n-1})$. We will adopt the convention that for $p=0$, $\eta_0 = \phi_0(\eta_{n-1}^\varepsilon)$.

An additional mapping notation is further considered $\phi_{p,n} = \phi_n \circ \ldots \circ \phi_{p+1}$. For $p=n$ we consider that $\phi_{p,n} = \phi_{n,n} = Id$, the identity mapping, and

$$\phi_{p,n}(\eta_p) = \eta_n. \tag{1.73}$$

For the normalized Feynman-Kac measure, we consider it as defined by

$$\gamma_p Q_{p,n} = \gamma_n \tag{1.74}$$

with $Q_{p,n}$ having the following functional representation

$$Q_{p,n}(f)(x_p) = \mathbb{E}\left(f(X_n) \prod_{k=p}^{n-1} G_k(X_n) | X_p = x_p \right). \tag{1.75}$$

The rationale behind introducing the additional mapping notation is to ease the description of the difference between the empirical measures in the presence of external stochastic factors and in the classical case, defined as follows:

$$\eta_n^\varepsilon - \eta_n = \sum_{p=0}^{n} [\phi_{p,n}(\eta_p^\varepsilon) - \phi_{p,n}(\phi_p(\eta_{p-1}^\varepsilon))] \tag{1.76}$$

The proof of the above has been obtained using a telescoping sum decomposition.

Let us now consider the following regularity properties.

Hypothesis H. There exists some integer $m \geq 1$ and some parameter $\varepsilon > 0$ such that for any $p \geq 1$, any $(x,y) \in E^2$ and any measurable subset A we have that

$$\begin{aligned} M_{p+1},\ldots,M_{p+m}(x,A) &\geq \varepsilon \times M_{p+1},\ldots,M_{p+m}(y,A) \\ g = \sup_p \sup_{x,y} \frac{G_p(x)}{G_p(y)} &< \infty \end{aligned} \tag{1.77}$$

The following contraction inequalities are proved in [38, 42, 43, 49], see also [58] for a more recent development on these stability properties.

Theorem 1.5.13. *We assume **(H)** is met for some parameters (m,ε). In this situation, there exists some $\delta \in]0,1[$ such that for any probability measures (η,ν), and for any $p \leq n$, we have*

$$\|\phi_{p,n}(\eta) - \phi_{p,n}(\nu)\|_{tv} \leq c\ (1-\delta)^{(n-p)/m} \|\eta - \nu\|_{tv}. \tag{1.78}$$

for some finite constant $c < \infty$ whose values do not depend the parameters (p,n,η,ν).

Replacing η with η_n^ε and ν with η_n, we find that

$$\begin{aligned} \|\eta_n^\varepsilon - \eta_n\|_{tv} &\leq \sum_{p=0}^{n} c(1-\delta)^{\frac{n-p}{m}} \|\phi_p^\varepsilon(\eta_{p-1}^\varepsilon) - \phi_p(\eta_{p-1}^\varepsilon)\|_{tv} \\ &\leq c\ (c_1 + 2c_2)\varepsilon \times \sum_{p\geq 0} (1-\delta)^{\frac{p}{m}} \end{aligned} \tag{1.79}$$

for any $\varepsilon < 1/(2c_2)$. The second implication comes from Lemma 1.5.12, under the conditions **(H1)** and **(H2)**. We conclude that

$$\sup_{n\geq 0} \|\eta_n^\varepsilon - \eta_n\|_{tv} \leq c(\delta)\varepsilon \quad \text{with} \quad c(\delta) \leq c\ (c_1 + 2c_2)/(1-(1-\delta)^{1/m}) \tag{1.80}$$

We recall that we consider a population of N individuals and the occupation measures of the population for the two variants (standard and perturbed) are approximated as follows:

$$\begin{aligned}\eta_n^{\varepsilon,N} &= \frac{1}{N}\sum_{i=1}^{N}\delta_{\xi_n^{\varepsilon,i}},\ with \begin{cases} M_n^{\varepsilon}\ mutation \\ G_n^{\varepsilon}\ selection \end{cases} \\ \eta_n^{N} &= \frac{1}{N}\sum_{i=1}^{N}\delta_{\xi_n^{i}},\ with \begin{cases} M_n\ mutation \\ G_n\ selection \end{cases}\end{aligned} \tag{1.81}$$

Hypothesis $\mathbf{H}^{\varepsilon}$ There exists some integer $m \geq 1$ and some parameter $\varepsilon' > 0$ such that for any $p \geq 1$, any $(x,y) \in E^2$, and $\varepsilon > 0$, and any measurable subset A we have that

$$\begin{aligned}\sup_{\varepsilon\geq 0}\sup_{p}\sup_{x,y}\frac{G_p^{\varepsilon}(x)}{G_p^{\varepsilon}(y)} &< \infty \\ M_{p+1}^{\varepsilon},\ldots,M_{p+m}^{\varepsilon}(x,A) &\geq \varepsilon'\, M_{p+1}^{\varepsilon},\ldots,M_{p+m}^{\varepsilon}(y,A)\end{aligned} \tag{1.82}$$

Under the assumptions **(H)** and $(\mathbf{H}^{\varepsilon})$, based on the Proposition 2.9 from [52] (see also [43, 58]) it can be deduced that $(\forall p \geq 1)$ and $(\forall f : \omega(f) \leq 1)$

$$\begin{aligned}\sup_{n\geq 0}\mathbb{E}\left(|\eta_n^N(f)-\eta_n(f)|^p\right)^{\frac{1}{p}} &\leq c(p)/\sqrt{N} \\ \sup_{n\geq 0}\mathbb{E}\left(|\eta_n^{N,\varepsilon}(f)-\eta_n^{\varepsilon}(f)|^p\right)^{\frac{1}{p}} &\leq c(p)/\sqrt{N}\end{aligned} \tag{1.83}$$

for some finite constant $c(p)$. This implies that

$$\mathbb{E}\left(|\eta_n^N(f)-\eta_n^{\varepsilon,N}(f)|^p\right)^{\frac{1}{p}} \leq \frac{2c(p)}{\sqrt{N}} + c(\delta)\varepsilon \leq c(p,\delta)\left(\frac{1}{\sqrt{N}}+\varepsilon\right) \tag{1.84}$$

for some finite constant $c(p,\delta) \leq \max\left(2c(p), c(\delta)\right)$.

Some exponential estimates can also be deduced using Bernstein-type martingales inequalities or alternatively by employing the Hoeffding's inequality [58, 57]. Under the assumptions **(H1)** and **(H2)** for any measurable function f, s.t. $\|f\| \leq 1$, for any $x > 0$ and any $N \geq 1$, when considering c_1 as being a finite constant related to *the bias* of the particle model and c_2 a constant related to the variance of the method, we have

$$\mathbb{P}\left((\eta_n^N - \eta_n)(f) \leq \frac{c_1}{N}(1+x+\sqrt{x}) + \frac{c_2}{\sqrt{N}}\sqrt{x}\right) \geq 1 - e^{-x} \tag{1.85}$$

$$\mathbb{P}\left(\left|\eta_n^N(f) - \eta_n(f)\right| \leq \frac{c_1}{N}(1+x+\sqrt{x}) + \frac{c_2}{\sqrt{N}}\sqrt{x}\right) \geq 1 - 2e^{-x}. \tag{1.86}$$

Next, by taking

$$(\eta_n^{N,\varepsilon} - \eta_n^N)(f) = (\eta_n^{N,\varepsilon} - \eta_n^{\varepsilon})(f) + (\eta_n^{\varepsilon} - \eta_n)(f) + (\eta_n - \eta_n^N)(f)$$

where the middle term is bounded with respect to ε up to some constant b_1, i.e. $(\eta_n^{\varepsilon} - \eta_n)(f) \leq b_1\varepsilon$, and the relation in Equation (1.86) (with respect to the first and third terms), we obtain the following concentration inequalities

$$\mathbb{P}\left((\eta_n^{N,\varepsilon} - \eta_n^N)(f) \leq 2\left(\frac{c_1}{N}(1+x+\sqrt{x}) + \frac{c_2}{\sqrt{N}}\sqrt{x}\right) + b_1\varepsilon\right) \geq 1 - 2e^{-x}$$

$$\mathbb{P}\left(\left|(\eta_n^{N,\varepsilon} - \eta_n^N)(f)\right| \leq 2\left(\frac{c_1}{N}(1+x+\sqrt{x}) + \frac{c_2}{\sqrt{N}}\sqrt{x}\right) + b_1\varepsilon\right) \geq 1 - 4e^{-x}.$$

Direct applications of this result could find a way into, for example, distributed desktop computing. For a few notorious examples, one can refer to the Seti@Home (Search for Extraterrestrial Intelligence) [4], Leiden Classical (desktop computer grid dedicated to general classical dynamics) [158], Rosetta@Home (protein folding, design, and docking) [156] or MilkyWay@Home (highly accurate model of our galaxy) [32]. A common part all those projects have is the use of desktop resources offered by (anonymous) users all over the world. As underlying principles of how the workload is managed, the aspects hereafter need to be considered, within the limits of the herein results, i.e. genetic algorithms like structure:

- a high complexity of the problem to deal with, e.g. sampling a large conformational space or running a computationally intensive analysis over enormous amounts of data, surpassing the power provided by classical resources like clusters or even grids; a second important element is the use of spare computational cycles (or specified amount), allowing to give a meaning to otherwise wasted energy and computing power while simultaneously contributing to an advance of our knowledge on important scientific problems;
- the problem allows a decomposition into independent sub-tasks that can be independently processed – each of the participating users only needs to deal with such sub-tasks (locally installed clients) and does not see the complete picture the problem draws; as a straightforward example, this may be the equivalent of receiving a set of instances from a server (input data), processing and last, sending back the results.

While this (simplified) design has several advantages, it is not difficult to understand that it is also subject to several issues that direct to security or data integrity aspects. Errors, as a result of network or processing faults, may lead to data loss or corruption. At the same time, malicious behavior, while unlikely, can not be excluded – if

reasons exist for malware and viruses, why would this particular context be any different? A question one may raise is how errors can be controlled or how results can be legitimated. While different approaches exist, the result presented in this section implies that an exponential decrease of the probability of having a difference above a given threshold can be obtained.

1.5.5.3 Weak Bounds for Behavior under Perturbation

We are modeling in the following a stochastic behavior that can occur for evolutionary algorithms. Let a candidate solution be modeled as a Markov chain $X_n = \begin{pmatrix} \varepsilon_n \\ Y_n \end{pmatrix}$ taking values in the product space $(\{0,1\} \times E)$. The parameter ε_n represents the stochastic factor, modeling the presence or absence of an uncertain behavior. We further assume that ε_n is a sequence of independent Bernoulli random variables with common law

$$\mathbb{P}(\varepsilon_n = 0) = (1 - \mathbb{P}(\varepsilon_n = 1)) = p$$

Let us now consider that the behavior involving no external stochastic factor and the absence of uncertainty is modeled through a value of zero attributed to the stochastic marker as $(\varepsilon_0, \ldots, \varepsilon_n) = (0, \ldots, 0)$. Given a realization $(\varepsilon_p)_{0 \leq p \leq n} = (u_p)_{0 \leq p \leq n}$, the second component Y_n forms a Markov chain with transitions M_{n+1,u_n} that depends on the parameter u_n, the initial random variable Y_0 is also distributed w.r.t. some probability measure η_{0,u_0} that depends on u_0.

We consider the space of possible values for the stochastic marker as Ω_n and let

$$\Omega_n^0 = \{\forall 0 \leq p \leq n \quad , \varepsilon_p = 0\}, \tag{1.87}$$

and $(\Omega_n^0)^C$ the complementary set. For a given function f, we use the following notation

$$f_n^{(0)}(y) = f_n \begin{pmatrix} 0 \\ y \end{pmatrix} \quad \text{and} \quad f_n^{(1)}(y) = f_n \begin{pmatrix} 1 \\ y \end{pmatrix}$$

We also set $G_n \begin{pmatrix} 0 \\ y \end{pmatrix} = G_n^{(0)}(y)$ and consider the uniform norm be given by $\|f\| = \sup_{u \in \{0,1\}, y \in E} \left| f \begin{pmatrix} u \\ y \end{pmatrix} \right|$.

Given the number of transitions that the algorithm is subject to, given by n, the normalized Feynman-Kac measure associated with the perturbed process behavior is given by

$$\gamma_n(f) = \mathbb{E}\left(f(X_n) \prod_{k=0}^{n-1} G_k(X_k) \right) \tag{1.88}$$

We let $\gamma_n^{(0)}$ be the Feynman-Kac measure defined by

$$\gamma_n^{(0)}\left(f_n^{(0)}\right) = \mathbb{E}\left(f_n^{(0)}(Y_n^{(0)}) \prod_{k=0}^{n-1} G_n^{(0)}(Y_k^{(0)}) \right) \tag{1.89}$$

where $Y_n^{(0)}$ stands for the Markov chain with transitions $M_{n+1,0}$ and initial distribution $\eta_{0,0}$.

The relation defined in equation (1.88) can be further decomposed in two cases according to the complete absence of external stochastic factors and the presence of perturbations.

$$\begin{aligned}\gamma_n(f) =& \mathbb{E}\left(f(X_n) \prod_{k=0}^{n-1} G_k(X_k) \mathbb{1}_{\Omega_n^0} \right) + \mathbb{E}\left(f(X_n) \prod_{k=0}^{n-1} G_k(X_k) \mathbb{1}_{(\Omega_n^0)^C} \right) \\ =& \gamma_n^{(0)}\left(f_n^{(0)}\right) \mathbb{P}(\Omega_n^0) + \mathbb{E}\left(f(X_n) \prod_{k=0}^{n-1} G_k(X_k) \mathbb{1}_{(\Omega_n^0)^C} \right).\end{aligned} \tag{1.90}$$

From equation 1.90 we can further derive

$$\begin{aligned}|\gamma_n(f) - \gamma_n^{(0)}\left(f_n^{(0)}\right)| \leq& |\gamma_n^{(0)}\left(f_n^{(0)}\right)| \left(1-(1-p)^{n+1}\right) + \left| \mathbb{E}\left(f(X_n) \prod_{k=0}^{n-1} G_k(X_k) \mathbb{1}_{(\Omega_n^0)^C} \right) \right| \\ \leq& |\gamma_n^{(0)}\left(f_n^{(0)}\right)| |(1-p)^{n+1} - 1| + \|f\| \prod_{k=0}^{n-1} \|G_k\| (1 - \mathbb{P}(\Omega_n^0)) \\ \leq& (1-(1-p)^{n+1}) \left(|\gamma_n^{(0)}\left(f_n^{(0)}\right)| + \|f\| \prod_{k=0}^{n-1} \|G_k\| \right)\end{aligned} \tag{1.91}$$

This implies that

$$|\gamma_n(f) - \gamma_n^{(0)}\left(f_n^{(0)}\right)| \leq (1-(1-p)^{n+1})\, c(n)$$

for some constant $c(n) \leq 2\|f\| \prod_{k=0}^{n-1} \|G_k\|$.

In practice, when subject to a corrupted computing environment (e.g. involving malicious/cheating behavior or faults of the hardware material) this result provides a quantitative measure of the fault-tolerance accepted by the system. This can be useful in assesing the level of accuracy of the results.

1.5.6 Rare Events Stochastic Models

1.5.6.1 Calibration and Uncertainty Propagation

Modern computers are capable of simulating complex physical and engineering systems. Nevertheless, formalized mathematical models are rarely certain and error-free. For instance, the physical environment is often too complex to formalize

perfectly, and all the different physical scales are difficult to capture with high precision. In addition, the reliability and the accuracy of computational approximation models often relies on complex calibration processes combined with the dispersion analysis of inputs and other sources of randomness.

Given some reference physical observation we would like to calibrate the model parameters so that the outputs simulated by some numerical code coincide with this reference data, or at least behave as much as possible as these physical observations.

In another context, with a successfully calibrated model one may be also interested in computing the probability that simulation outputs belong to some critical event; that is, to find the law of the input parameters and the sources of randomness leading to such events.

This couple of important issues can be formulated in terms of a classical input-output transformation

$$\underbrace{\text{Inputs} = I}_{\substack{\text{sources of randomness}\\ \text{uncertainty representations}\\ \text{tuning parameters}\\ \text{unknown kinetic parameters}}} \longrightarrow [\textbf{Black-box simulation model}] \longrightarrow \underbrace{\text{Outputs } O = C(I)}_{\substack{\text{physical, biological or forecasting predictions}\\ \text{partial differential equation profiles}\\ \text{mechanical forces}\\ \text{hydrodynamic profiles}}}$$

The prototype of question arising in practice is the following. We are given a desired domain, say $\mathscr{O}$, in the space of the outputs, and we want to estimate both the probability that the outputs fails into this set and the distribution of the inputs leading to these outputs; that is, we want to compute the following quantities

$$\text{Proba}(O \in \mathscr{O}) \quad \text{and} \quad \text{Law}(I \mid O \in \mathscr{O})$$

- The set $\mathscr{O}$ represents some critical event with a very small occurrence probability, say 10^{-9}. In this context, we are interested in computing rare event probabilities as well as the distribution of the random sources leading to this critical regime. The conditional distribution provides all the statistical information on the different contributions of the input parameters and the random sources on "the desired" critical rare event.
- The domain $\mathscr{O}$ is related to some distance-like criteria that measure the adequacy of some output profile with some reference data or some observations delivered by some sensors. In this context, we are interested in computing the chance that some collection of models may reach a given precision w.r.t. the physical data. Furthermore, we are also often interested in calibrating the numerical code with the selection of the most accurate input parameters that achieve a given precision w.r.t. the data.
- The couple of situations discussed above can be combined. For instance, we may be interested in computing the probability of a critical rare event given some observations, or reversely the law of some input parameters given some observations as well as some critical event.

Of course, the choice of the stochastic particle algorithm that solves these three questions is far from being unique. The design of an appropriate Monte Carlo simulation model strongly depends on the physical problem at hand, including the nature

of the numerical code, the quantification of the sources of randomness, the specification of the inputs and the outputs profiles.

In the following we consider the case where some fixed critical level exists, given by a value h and the rare event of interest is the fact that the output value passes the maximal value h

The inputs are represented by a function like d-dimensional vector $I = (I^1, \ldots, I^d)$ and the set $\mathscr{O}$ represents a failure region. The outputs of the numerical code are represented by a function like d'-dimensional vector $O = (O^1, \ldots, O^{d'})$. In this situation, we are given some critical threshold value h and a random event of interest is the fact that the forces acting on the provided structure get above this maximal value; the corresponding probabilistic quantities of interest are given below

$$\text{Proba}\left(\sup_{1\leq t\leq d'} |O^i| \geq h\right) \quad \text{and} \quad \text{Law}\left((I^1,\ldots,I^d) \mid \sup_{1\leq t\leq d'} |O^i| \geq h\right)$$

Our next objective is to relate these questions to the probabilistic model of combinatorial counting and sampling presented in section 1.5.4.3. To this end, let us assume that the random input parameters I are distributed according to some probability measure λ on some finite or countable state space; that is we have that

$$\text{Proba}(I = x) = \lambda(x)$$

We also suppose that the input-output function is given by some mapping $C : x \mapsto C(x)$, and we set

$$A := \{x \,:\, C(x) \in \mathscr{O}\}$$

In this notation, the uncertainty propagation models presented above coincide with the ones discussed in section 1.5.4.3; that is, we have that

$$\text{Proba}\,(O \in \mathscr{O}) = \text{Proba}\,(I \in A)\,\lambda(A) := \sum_{x\in A} \lambda(x)$$

and

$$\text{Proba}\,(I = x \mid O \in \mathscr{O}) = \mu_A(x) := \frac{1}{\mathscr{Z}_A}\; \mathbb{1}_A(x)\; \lambda(x)$$

with the normalizing constant $\mathscr{Z}_A = \lambda(A)$.

In engineering literature, the multilevel genetic type splitting particle algorithms presented in section 1.5.4.3 are also called subset-simulation models. The central idea is to express a rather small failure probability as the product of not so small conditional probabilities:

$$\text{Proba}\,(I \in A) = \prod_{p=0}^{m} \text{Proba}(I \in A_{p+1} | I \in A_p) \tag{1.92}$$

The intermediate levels A_p are judiciously chosen failure regions s.t.

$$A_0 \supset A_1 \supset \ldots \supset A_m = A$$

In this interpretation, the computation of the small failure probability $\text{Proba}(I \in A)$ is now reduced to the computation of larger conditional probabilities $\text{Proba}(I \in A_{p+1} | I \in A_p)$. In the marine engineering problem discussed above, these failure regions are characterized by the choice of an increasing sequence of critical threshold values h_p; that is, if we set

$$I = (I^1, \ldots, I^d) = (x^1, \ldots, x^d) \mapsto O = C(I) = (C^1(I), \ldots, C^{d'}(I)) = (C^1(x), \ldots, C^{d'}(x))$$

then we have that

$$h_p \uparrow \quad \Rightarrow \quad A_p := \left\{ x = (x^1, \ldots, x^d) \mid \sup_{1 \le t \le d'} |C^t(x)| \ge h_p \right\} \downarrow$$

1.5.6.2 An Universal Particle Algorithm Based on Multilevel Splitting

In this section we introduce a generic particle simulation algorithm based on a multilevel splitting mechanism. A simple stochastic particle algorithm consists in propagating a population of N individuals representing potential solutions at each iteration. Hereafter we start by providing the main structure and pseudo-code of a generic multilevel splitting particle algorithm, depicted in Algorithm 1.2.

The algorithm is initialized with N random configurations chosen according to some distribution law ν_0 (in this case we consider it as uniform i.i.d. sampling) in the A_0 set. The algorithm considers a critical level to be reached, in order for the rare event to take place, denoted in the pseudo-code by c_n.

In the following p_1^N stands for the proportion of individuals that succeed to reach the level A_1, these individuals in A_1 being further selected for the next step of the algorithm. As we consider N as a fixed value, the notation will be simplified to p_1 instead of p_1^N. The rejected configurations are then randomly redistributed among the ones that passed into A_1 such that the number of individuals in the population remains constant, N. The following step consists on diversifying and enriching the population of solutions selected in A_1 during the first step. This is performed by applying the transition or perturbation operator F, leading to new candidate samples, while leaving the measure μ_A^1 invariant. In its most simple variants the transition operator can be seen as a mutation operator.

Each individual independently explores the space defined by A_1 following a local Markovian transition that leaves the measure μ_{A_1} invariant. As for the previous step, we denote by p_2 the proportion of individuals having succeeded to pass to the level A_2. We select afterwards the configurations having succeeded to pass at the second level A_2. The rejected configurations are again redistributed randomly among the previously selected ones. Each individual explores afterwards, in an independent manner, the space A_2 following a local Markovian transition that leaves the measure μ_{A_2} invariant. The process is reiterated until the last level n is reached.

[1] Note: h is the equivalent of h_m when $A = A_m$.

Algorithm 1.2. Multilevel splitting particle algorithm

$A_0 := E$, $h_i :=$ threshold levels, $j := 0$

Sample N particles, $(\xi_0^i)_{1\leq i\leq N}$ from a given distribution μ_0 in A_0

$\{A_i = \left\{ x = (x^1,\ldots,x^d) \mid \sup_{1\leq t\leq d'} |C^t(x)| \geq h_i \right\}, i \leq m+1$ s.t. $h_{m+1} = h$ the fixed critical level.[1]$\}$

while $\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{A_{m+1}}(\xi_j^i) = 0$ **do**

$\widehat{p}_{j+1}(A_{j+1} \mid A_j) := \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{A_{j+1}}(\xi_j^i)$

$\xi_j(A_{j+1}) := \{\xi_j^l \mid 1 \leq l \leq N \, s.t. \, \xi_j^l \in A_{j+1}\}$

Selection

for $i = 1$ to N **do**

$\widehat{\xi}_j^i := \begin{cases} \xi_j^i, \; if \; \xi_j^i \in A_{j+1} \\ sample \; randomly \; in \; the \; set \; \xi_j(A_{j+1}), \; otherwise. \end{cases}$

end for

Transition

for $i = 1$ to N **do**

$\xi_{j+1}^i := \begin{cases} F(\widehat{\xi}_j^i), \; if \; F(\widehat{\xi}_j^i) \in A_{j+1} \\ \widehat{\xi}_j^i, \qquad otherwise. \end{cases}$

end for

$j := j+1$

end while

For the multilevel splitting, the result of this simulation can be explained as follows. When N is increased, the population of solutions obtained at each k^{th} iteration is distributed according to the law of the variable X restricted to the set A_k. This is equivalent to saying, that at each iteration k, the variables $I_k^{(i,N)}$, $1 \leq i \leq N$, simulated on the A_k set are distributed approximately as a sequence of random variables $\overline{I}_k^{(i,N)}$, $1 \leq i \leq N$, independent and having the same law μ_{A_k}. This approximation can be further detailed in several forms. For example, we can have:

$$\left\| \text{Law}\left(I_k^{(1,N)},\ldots,I_k^{(q,N)}\right) - \text{Law}\left(\overline{I}_k^{(1,N)},\ldots,\overline{I}_k^{(q,N)}\right) \right\|_{tv} \leq \frac{q}{N}\, c(k)$$

for all $q \leq N$ and for a finite constant $c(k) < \infty$ which can be specified according to the parameters of the model. In the preceding equation, $\|P-Q\|_{tv}$ denotes the total variation distance between two probability measures P and Q. For any bounded function f and for any $\varepsilon > 0$, the following exponential error probabilities are verified:

$$\mathbb{P}\left(\left| \frac{1}{N}\sum_{i=1}^{N} f\left(I_k^{(i,N)}\right) - \mathbb{E}\left(f(I) \mid I \in A_k\right) \right| \geq \varepsilon \right) \leq c_1(k)\, e^{-\frac{N\varepsilon^2}{c_2(k)}}$$

Algorithm 1.3. Particle algorithm using acceptance-rejection

$A_0 := E$
$\hat{A} := \{A_1, A_2, \ldots, A_n\}$ {Constants specifying the fixed levels $(A_k)_{1\leq k\leq n}$ }
Initialization
ξ_0:= sample N particles, $(\xi_0^i)_{1\leq i\leq N}$ randomly of law η_0.
for $k = 1$ to n **do**
{For each of the intermediate levels $(A_k)_{1\leq k\leq n}$ }

Selection
for $i = 1$ to N **do**
{For each particle}

$$\widehat{\xi}_{k-1}^i := \begin{cases} \xi_{k-1}^i, \ if\ \xi_{k-1}^i \in A_k \\ \tilde{\xi}_{k-1}^i, \ random\ variable\ of\ law \ \displaystyle\sum_{i=1}^{N} \frac{\mathbb{1}_{A_k}(\xi_{k-1}^i)}{\sum_{j=1}^{N} \mathbb{1}_{A_k}(\xi_{k-1}^j)}\ \delta_{\xi_{k-1}^i} \ otherwise. \end{cases}$$

end for

Transition
for $i = 1$ to N **do**
{For each particle}

$$\xi_k^i := \begin{cases} F(\widehat{\xi}_{k-1}^i), \ if\ F(\widehat{\xi}_{k-1}^i) \in A_k \\ \widehat{\xi}_{k-1}^i, \quad otherwise \end{cases}$$

{F designates the perturbation operator generating new candidate solutions and ξ_k^i of law $M_k(\widehat{\xi}_{k-1}^i, \cdot\ )$ with a Markovian transition M_k leaving the measure μ_{A_k} invariant.}
end for
end for

for a couple of finite constants $c_1(k), c_2(k) < \infty$ they may be specified depending on the model parameters.

Furthermore, the product of the proportions of success $\displaystyle\prod_{l=1}^{k} p_l^N$ is an unbiased estimator of the probability that the variable I is in A_k. Under certain regularity assumptions ([27]), the following convergence result can be proved (holds):

$$\mathbb{E}\left(\prod_{l=1}^{k} p_l^N\right) = \mathbb{P}(I \in A_k) \quad \text{and} \quad \mathbb{E}\left(\left[\prod_{l=1}^{k} p_l^N - \mathbb{P}(I \in A_k)\right]^2\right) \leq \frac{c}{N}\, k\, \mathbb{P}(I \in A_k)^2,$$

for a finite universal constant $c < \infty$. Other convergence results, including estimates of error probability exponentially small are described in the book [38] and also in the articles [52, 41, 57].

1.5.6.3 Variants of Multilevel Splitting Simulation

In the description of the preceding algorithm, it is worth mentioning that the parameters (N, A_n) must be chosen judiciously so that at least one solution among the N ones is in A_n, otherwise the algorithm stops at the nth iteration. For this to be fulfilled we propose choosing a first level disparity $(A_n - A_{n-1})$ sufficiently low. Either case, when the number of proposed solutions N' increases, we can prove that all the levels are reachable and that the algorithm converges towards the desired solutions. It is also possible to chose the levels A_n adaptively online according to the proposed candidate samples. For example, one option consists in choosing the first level A_1 such that a given proportion of the current solutions (e.g. 80 percent) of the solutions proposed in A_0 reach the next level. These adaptive splitting algorithms can be recast in terms of sequential Monte Carlo models with adaptive resampling procedures. For a detailed discussion on these models with precise reference pointers we refer the reader to [41]. To our knowledge these adaptive resampling techniques were first introduced as an heuristic scheme in [36] (remark 1, section 2.1), see also [37]. These adaptive criteria for the choice of the levels were also discussed in three recent studies [26, 28, 27]. For a detailed theoretical analysis of these models, including central limit theorems and exponential cumulative ratios, we refer the reader to [41].

There are also other variants allowing the exploration of the search space according to these new data and solving this stopping problem. These techniques are more complex to describe and they will be detailed in the follow up. The main idea behind this is to create new candidate solutions until configurations in A_1 are reached. This step can also be catastrophic if the level A_1 is badly chosen.

We start by differentiating the two types of multilevel splitting considered in the following, according to the mechanism used in establishing the different splitting levels. The levels can be either fixed *a priori* or established adaptively at each iteration step by the threshold passed by a percentage of the sampled solutions. Also, two types of selection mechanisms are considered: the uniform selection and the acceptance-rejection selection.

1.5.6.4 Case Study

We illustrate these rather abstract models with a marine engineering problem we recently analyzed with Z. Guede from the French marine research institute (IFREMER). In this situation, we want to assess the reliability of an offshore structure, both at the design stage, as to validate the design choice, and in service for maintenance and inspection planning. The goal is to check whether the structure is able to withstand the loads from its environment for its entire planned lifetime, defined according to a physical criterion with respect to the structural response. The structural response is computed by a hydrodynamic numerical code with strong physical and geometrical non-linearity that lead to a complex failure region geometry. In this context, the input parameters are of different natures, some of them representing

Algorithm 1.4. Fixed levels particle algorithm

$A_0 := E$, $h_i :=$ threshold levels, $j := 0$

Sample N particles, $(\xi_0^i)_{1\leq i\leq N}$ uniformly i.i.d. in A_0

$\{A_i = \left\{ x = (x^1,\ldots,x^d) \mid \sup_{1\leq t\leq d'} |C^t(x)| \geq h_i \right\}, i \leq m+1$ s.t. $h_{m+1} = h$ the fixed critical level$\}$

while $\frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{A_{m+1}}(\xi_j^i) = 0$ **do**

$\widehat{p}_{j+1}(A_{j+1} \mid A_j) := \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{A_{j+1}}(\xi_j^i)$

$\xi_j(A_{j+1}) := \{\xi_j^l \mid 1 \leq l \leq N\, s.\, t.\, \xi_j^l \in A_{j+1}\}$

for $i = 1$ to N **do**

$\widehat{\xi}_j^i := \begin{cases} \xi_j^i, \; if \; \xi_j^i \in A_{j+1} \\ \widetilde{\xi}_j^i \text{ a randomly chosen particle in the set } A_{j+1}, \; otherwise. \end{cases}$

end for

for $i = 1$ to N **do**

$\xi_{j+1}^i := \begin{cases} F(\widehat{\xi}_j^i), \; if \; F(\widehat{\xi}_j^i) \in A_{j+1} \\ \widehat{\xi}_j^i, \qquad otherwise. \end{cases}$

end for

$j := j+1$

end while

the spectral properties of wave mixtures, while the other ones represent temperature variations, waves periods and their direction. The outputs of the numerical code are represented by a function like d *prime*-dimensional vector $O = (O^1,\ldots,O^{d'})$ denoting the forces that act on the offshore structure surface at different time periods. In this situation, we are given some critical threshold value h for which a random event of interest is given by the fact that the forces acting on the offshore structure get above this maximal value.

In order to test the performance and the validity of the approaches in a general context, we employ as testbed the estimation of the evolution of a variable that follows a known law. The choice for χ^2 is due, among others, to its resemblance with the quadratic nature appearing in the real-life experiment proposed by IFREMER.

Let $U_1, U_2, ..., U_k$ be k independent random variables following the same standard normal law, then the U variable is defined such that

$$\mathbb{P}(\max_{1\leq i\leq n} X_i^2 \geq c) \tag{1.93}$$

$$\mathbb{P}(\max_{1\leq i\leq n} X_i^2 \geq c) = 1 - (F_{X_1^2}^n(c)) \tag{1.94}$$

Algorithm 1.5. Adaptive particle algorithm

$A_0 := E$, $h := constant$ {Critical level}, $k_0 := 0$, $p := 1$
$\xi_0 := (\xi_0^i)_{1\leq i\leq N}$ {N independent particles of law η_0}

while $k_{p-1} \leq c$ **do**

$$k_p := \inf\left\{k > k_{p-1} \ : \ \eta_{p-1}(\mathbb{1}_{A_k}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{A_k}(\xi_{p-1}^i) \leq \alpha_p, \alpha_p \in]0,1[\right\}$$

$$\widehat{p}_p(A_{k_p}|A_{k_{p-1}}) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{A_{k_p}}(\xi_{p-1}^i)$$

Selection
Select N particles $(\widehat{\xi}_{p-1}^i)_{1\leq i\leq N}$ from the A_{k_p} set
{by using either an accept/reject technique or an uniform sampling}

Transition
for $i = 1$ to N **do**
ξ_p^i constructed by successively applying n_{p+1} Markovian elementary transitions M_{k_p} (of invariant measure η_{k_p}).
end for

$p := p+1$
end while

$$U = \max_{1\leq i\leq k} U_i^2 \tag{1.95}$$

follows a law χ^2 with one degree of freedom. For experimental purposes, we consider a number of $k = 2048$ random variables, as this number of degrees of freedom is considered also relevant for the practical IFREMER application.

$$\begin{aligned}\mathbb{P}\left(\max_{1\leq i\leq k} U_i^2 \geq c\right) &= 1 - \mathbb{P}(\max_{1\leq i\leq k} U_i^2 < c) \\ &= 1 - F_{U_1^2}(c)^n = 1 - \left(1 - 2\mathbb{P}\left(U_1 \geq \sqrt{c}\right)\right)^n \end{aligned} \tag{1.96}$$

From this we obtain in fact the value of the distribution function of the variable U^n, which is denoted in the follow up by $F_{U_1^2}(c)$, where c represents a real value corresponding to a given critical level. Furthermore, the computation of the distribution function for a given level c and a variable $X = U_i^2$, $1 \leq i \leq k$ can be done by using:

$$F_X(c) = \left(\frac{\gamma\left(\frac{c}{2}, \frac{1}{2}\right)}{\Gamma\left(\frac{1}{2}\right)}\right)^n = \left[\frac{1}{\sqrt{\pi}}\int_0^{\frac{c}{2}} \frac{e^{-t}}{\sqrt{t}} dt\right]^n \tag{1.97}$$

where γ is the lower incomplete gamma function. We can also use the following estimations in the formula (1.96)

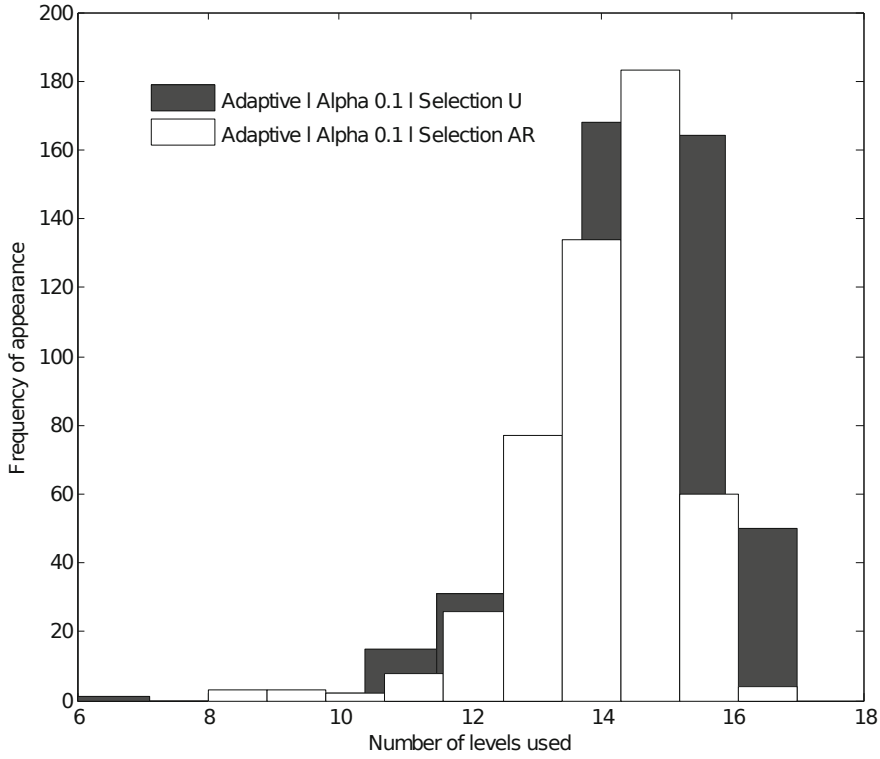


Fig. 1.1 Histogram depicting the number of levels for the adaptive algorithm with α set to 0.1

$$\frac{1}{\sqrt{2\pi}}\,\frac{1}{c+1/c}\,e^{-\frac{c^2}{2}} \leq \mathbb{P}(U_1 \geq c) \leq \frac{1}{\sqrt{2\pi}}\,\frac{1}{c}\,e^{-\frac{c^2}{2}}$$

A proof of these analytic estimations can be found in the book [56].

In order to prepare the calibration for the practical problem and prepare the testing environment we first studied the distribution of the number of levels employed for the adaptive method, as depicted in Figure 1.1. In concordance with the results, the number of chosen levels for the adaptive method was set to 15, this being given by the central tendency of the number of levels for the two adaptive cases.

The next step considered the analysis of the algorithms' evolution according to the theoretical estimate, obtained as described in Equation 1.96. Figure 1.2 depicts on the ordinate axis the distribution of the values obtained for the probability of passing the intermediate levels, while the abscissa represents the values of the system's response, i.e. from 15.0 to 52.2513 (critical level). The evolution of the adaptive algorithms (where the intermediate levels vary among two different executions) is approximated by a least squares method applied on the entire set of obtained values (cloud of points). The approximation in this latter case is done by estimating the average and standard deviation for a normal density. A comparison of the algorithms' evolution is illustrated in Figure 1.2 successively focusing closer to the critical level region. It is thus possible to evaluate the stalling of the algorithm using fixed levels as compared with the curve of the theoretical

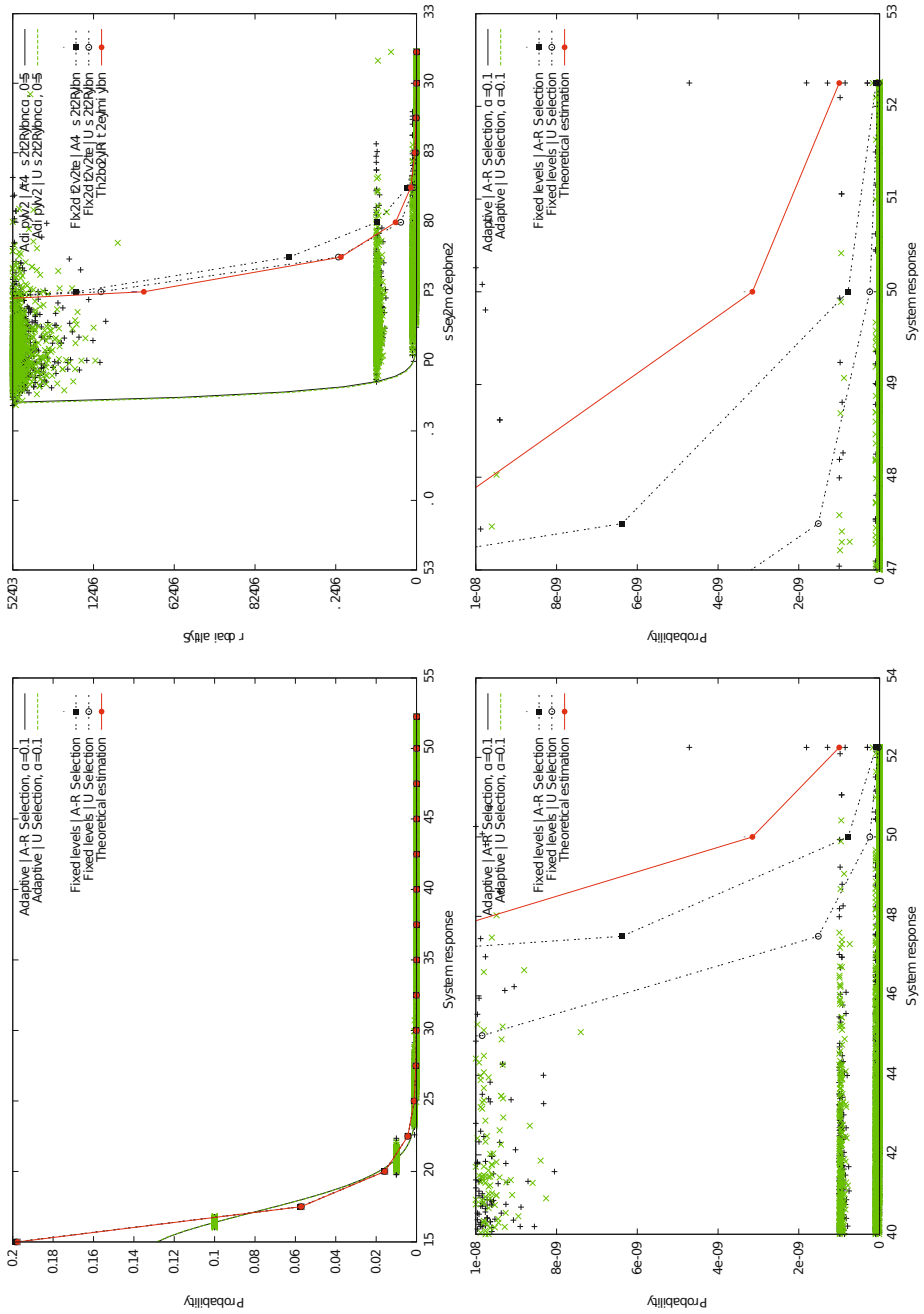


Fig. 1.2 The evolution of the algorithms compared with the theoretical estimate (global view depicted in the left upper corner, followed by successive zooms). Fixed levels algorithm: averaged maximal values per level. Adaptive algorithm: least squares approximation on the set of all the values.

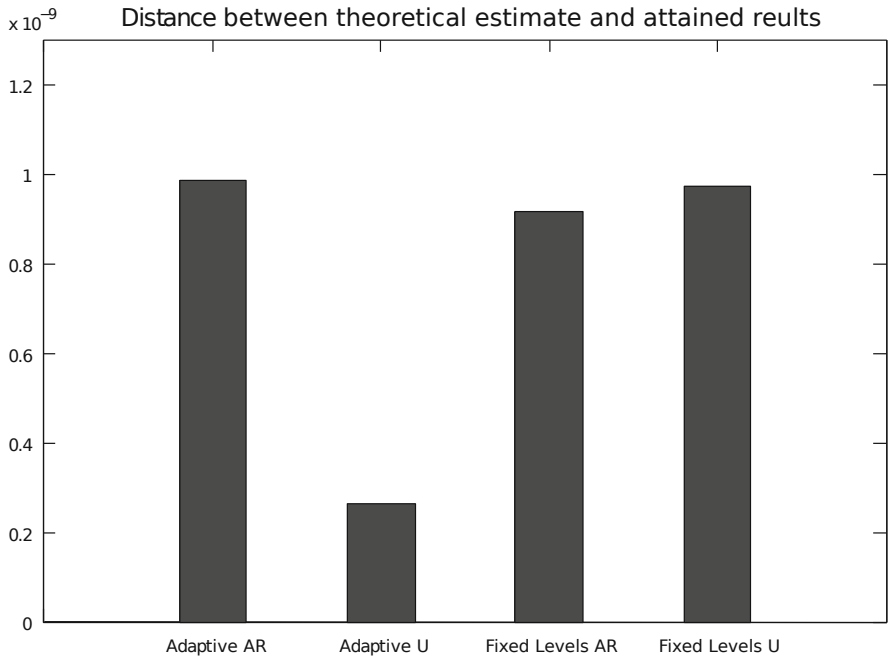


Fig. 1.3 The difference between the final probability attained by the four algorithm variants (averaged over 500 tests) and the theoretical estimate

estimate. Finally, one should note that adaptive variant whose averaged values follow the best the curve of the values estimated by Equation 1.96 is the one using the uniform selection. By considering the simulation of the normal variable described above, for comparative purposes, the adaptive method was compared with the fixed levels variant, by employing two types of selection. The comparison is done by considering the distance between the average of the final probability (obtained on 500 tests per algorithm) and the theoretical estimate of the final probability. As illustrated in Figure 1.3, the best results were obtained with the adaptive variant employing the uniform selection.

1.5.6.5 Multilevel Splitting Simulation

The analysis of rare events arise in various scientific areas including physics, biology, engineering science and financial mathematics. For instance in nuclear physics to study of the performance of a radiation source containment we are interested in computing the probability that a neutron particle emitted by the radiation source escapes from the containment before being absorbed and desintegrated by some obstacle. In biology they may represent an extinction probability of a given population evolution model. In engineering science these rare events are sometimes related to a catastrophic failure such as a buffer excedence in communication networks. Finally in financial mathematics they may represent a ruin process.

The random excursion model is defined in terms of some Markov chain $(X'_n)_{n\geq 0}$ taking values in some finite state space E'. We assume that the chain X'_n starts in some given subset $X_0 \in A \subset E'$ with a given distribution ν_0. We also let (B,C) be a pair of subsets (B,C) such that $A \cap C = \emptyset = B \cap C$. We also assume that the triplet (A,B,C) is chosen so that for any initial state $x \in A$ the chain X'_n hits one of the sets B or C in finite time.

We let T_A be the entrance time of X' into a given subset A; that is, we have

$$T_A = \inf\{n \geq 0 \,:\, X'_n \in A\}$$

One would like to estimate the probability that the chain hits B before C

$$\mathbb{P}(T_{B\cup C} < T_C) = \mathbb{P}(X'_{T_{B\cup C}} \in B) = \mathbb{E}(\mathbb{1}_B(X'_{T_{B\cup C}}))$$

and the law of the random excursion given the fact that it reached B before C

$$\mathrm{Law}(X'_t \,;\, 0 \leq t \leq T_{B\cup C} \mid T_{B\cup C} < T_C) = \mathrm{Law}(X'_t \,;\, 0 \leq t \leq T_{B\cup C} \mid X'_{T_{B\cup C}} \in B)$$

Of course we have implicitly assumed that $\mathbb{P}(T_{B\cup C} < T_C) > 0$ so that the conditional distributions are well defined.

In connection with the previous examples discussed in the early part of this section the rare level set B may represent the outside of the radiation containment, an undesired critical population size or buffer excedance as well as ruin level of a given company. The level set C is usually far from being rare but it corresponds to an almost sure event. For instance in the radiation containment model the set C represents the set of physical obstacles which hopefully absorb the radiation and avoid the particle to come out of the containment. In communication networks models the set C represents a recurrent and well behave buffer size level. In population models C is related to a natural fluctuation size level of the population evolution and in ruin processes it corresponds to a predicted gain or a desired equilibrium level.

During its excursion from A to $(B\cup C)$ the process passes through a decreasing sequence of level sets $\mathscr{B} = (B_n)_{n=0,\ldots,m}$ with

$$A = B_0 \supset B_1 \supset \ldots \supset B_m = B$$

Here again the splitting parameter m and the choice of the level sets $\mathscr{B}$ depends on the problem at hand.

This decomposition reflects the successive levels the stochastic process needs to cross before to enter into the relevant rare event. In other words the increasing levels behave as gateways from which the rare event is more and more likely to happen.

To clarify the presentation we shall slight abuse the notation and we write T_n instead of $T_{B_n\cup C}$ the entrance time of X into $B_n \cup C$. To capture the behavior of X between the different levels we introduce the excursion-valued Markov chain

$$X_n = (T_n, (X'_t \,;\, T_{n-1} \leq t \leq T_n)) \in E = \cup_{p \leq q} (\{q\} \times (E')^{(q-p+1)}) \tag{1.98}$$

By a direct inspection we see that the random sequence of level-crossing times $(T_n)_{0 \leq n \leq m}$ is increasing and whenever $T_n < T_C$ the second component of X_n represents the excursion of the process X' between the successive levels B_{n-1} and B_n so that T_n can be alternatively be defined by the inductive formulae

$$T_n = \inf\{T_{n-1} \leq t \,:\, X'_t \in B_n \cup C\}$$

Under our assumptions we also observe that these entrance times are finite and

$$(T_{B \cup C} < T_C) = (T_m < T_C) = (T_1 < T_C, \ldots, T_m < T_C)$$

One simple way to check whether or not a random path has succeeded to reach the desired n-th level is to consider the potential functions G_n on E defined for each $n \in \{0, \ldots, m\}$ and $x = (x_q)_{p \leq q \leq r} \in (E')^{(r-p+1)}$ by

$$G_n(t,x) = \mathbb{1}_{B_n}(x_r) \tag{1.99}$$

In this notation we have for each $n \leq m$

$$(T_n < T_C) = (T_1 < T_C, \ldots, T_n < T_C) = (G_1(X_1) = 1 \ldots, G_n(X_n) = 1)$$

and

$$(X_0, \ldots, X_n)$$

$$= ((0, X'_0), (T_1, (X'_t \,;\, 0 \leq t \leq T_1)), \ldots, (T_n, (X'_t \,;\, T_{n-1} \leq t \leq T_n)))$$

In we write $[X'_t \,;\, 0 \leq t \leq T_n]$ instead of $(X_0, \ldots, X_n)$ the sequence of excursions of X' between the levels, then for any $n \leq m$ and any function f_n on the product space E_n we have

$$\mathbb{E}_{\nu_0}\left(f_n(X_0, \ldots, X_n) \prod_{p=1}^{n} G_p(X_p) \right) = \mathbb{E}_{\nu_0}\left(f_n([X'_t \,;\, 0 \leq t \leq T_n])\, \mathbb{1}_{T_n < T_C} \right)$$

We denote by $\mathbb{P}_n$ the law of the excursion-valued Markov chain from the origin $p = 0$, up to the time $p = n$

$$\mathbb{P}_n(x_0, \ldots, x_n) = \mathbb{P}\,(X_0 = x_0, \ldots, X_n = x_n)$$

If we set

$$\mathbb{Q}_n(x_0, \ldots, x_n) = \frac{1}{\mathscr{Z}_n} \left\{ \prod_{0 \leq p < n} G_p(x_p) \right\} \mathbb{P}_n(x_0, \ldots, x_n)$$

with the unit potential function $G_0 = 1$ then we have

$$\mathbb{Q}_n = \text{Law}\left(\left[X'_t\,;\,0 \le t \le T_n\right] \mid T_{n-1} < T_C\right) \quad \text{and} \quad \mathscr{Z}_n = \mathbb{P}\left(T_{n-1} < T_C\right)$$

Once again, these measures have exactly the same form as the one presented in (1.33). The corresponding particle approximations are often referred as multilevel splitting particle methods or sequential Monte Carlo samplers in the literature on rare event simulation (see for instance [25, 26, 50, 114], and references therein).

Acknowledgements. The second and third author acknowledge that their work has been supported by the National Research Fund, Luxembourg, and cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND).

References

1. Ackley, D., Littman, M.: A case for lamarckian evolution. Artifical Life III: SFI studies in the sciences of complexity XVII, 3–10 (1993)
2. Alba, E., Luque, G.: Performance of Distributed GAs on DNA Fragment Assembly. In: Parallel Evolutionary Computations, pp. 97–116. Springer (2006)
3. Aldous, D., Vazirani, U.: Go with the winners algorithms. In: Proc. 35th Symp. Foundations of Computer Sci., pp. 492–501 (1994)
4. Anderson, D.P., Cobb, J., Korpela, E., Lebofsky, M., Werthimer, D.: SETI@home: an experiment in public-resource computing. Commun. ACM 45(11), 56–61 (2002)
5. Ashlock, D.A.: Evolutionary computation for modeling and optimization. Springer (2006)
6. Assaraf, R., Caffarel, M., Khelif, A.: Diffusion Monte Carlo methods with a fixed number of walkers. Phys. Rev. E 61, 4566–4575 (2000)
7. Bäck, T., Hoffmeister, F., Schwefel, H.P.: A survey of evolution strategies. In: Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 2–9. Morgan Kaufmann (1991)
8. Bäck, T., Fogel, D.B., Michalewicz, Z.: Handbook of Evolutionary Computation. IOP Publishing Ltd., Bristol (1997)
9. Bäck, T., Hammel, U., Schwefel, H.P.: Evolutionary computation: comments on the history and current state. IEEE Trans. Evolutionary Computation 1(1), 3–17 (1997)
10. Barricelli, N.A.: Esempi numerici di processi di evoluzione. Methodos, 45–68 (1954)
11. Barricelli, N.A.: Symbiogenetic evolution processes realized by artificial methods. Methodos 9(35-36), 143–182 (1957)
12. Battiti, R., Brunato, M., Mascia, F.: Reactive Search and Intelligent Optimization. In: Operations Research/Computer Science Interfaces. Springer (2008) doi:10.1007/978-0-387-09624-7
13. Baum, E.B.: Towards practical 'neural' computation for combinatorial optimization problems. In: AIP Conference Proceedings 151 on Neural Networks for Computing, pp. 53–58. American Institute of Physics Inc., Woodbury (1987), http://dl.acm.org/citation.cfm?id=24140.24150
14. Belew, R.K., Booker, L.B. (eds.): Proceedings of the 4th International Conference on Genetic Algorithms. Morgan Kaufmann, San Diego (1991)
15. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm intelligence: from natural to artificial systems. Oxford University Press, Inc., New York (1999)

16. Bremermann, H.J., Rogson, M., Salaff, S.: Global Properties of Evolution Processes. In: Pattee, H.H., Edlsack, E.A., Fein, L., Callahan, A.B. (eds.) Natural Automata and Useful Simulations, pp. 3–41. Spartan Books, Washington, DC (1966)
17. Broyden, C.G.: The Convergence of a Class of Double-rank Minimization Algorithms: 2. The New Algorithm. IMA Journal of Applied Mathematics 6(3), 222–231 (1970), abstract/6/3/222, doi:10.1093/imamat/6.3.222
18. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Ozcan, E., Woodward, J.R.: A classification of hyper-heuristic approaches. Handbook of Metaheuristics 146, 1–21 (2010), http://www.springerlink.com/index/XXM7126130381913.pdf
19. Campillo, F., Rossi, V.: Convolution particle filtering for parameter estimation in general state-space models. In: Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, USA (2006)
20. Campillo, F., Rossi, V.: Convolution filter based methods for parameter estimation in general state-space models. IEEE Transactions on Aerospace and Electronic Systems 45(3), 1063–1071 (2009)
21. Cantu-Paz, E.: A survey of parallel genetic algorithms. Calculateurs Paralleles Reseaux et Systems Repartis 10(2), 141–171 (1998), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=879173
22. Carpenter, J., Clifford, P., Fearnhead, P.: An improved particle filter for non-linear problems. IEE Proceedings F 146, 2–7 (1999)
23. Carvalho, H., Del Moral, P., Monin, A., Salut, G.: Optimal Non-linear Filtering in GPS/INS Integration. IEEE-Trans. on Aerospace and Electronic Systems 33(3), 835–850 (1997)
24. Cerf, R.: Asymptotic convergence of genetic algorithms. Adv. Appl. Probab. 30, 521–550 (1998)
25. Cérou, F., Del Moral, P., LeGland, F., Lezaud, P.: Limit Theorems for multilevel splitting algorithms in the simulation of rare events (preliminary version). In: Kuhl, M.E., Steiger, N.M., Armstrong, F.B., Joines, J.A. (eds.) Proceedings of the 2005 Winter Simulation Conference (2005)
26. Cérou, F., Del Moral, P., LeGland, F., Lezaud, P.: ALEA Lat. Am. J. Probab. Math. Stat. 1, 181–203 (2006)
27. Cérou, F., Del Moral, P., Guyader, A.: A non asymptotic variance theorem for unnormalized Feynman-Kac particle models. Technical Report HAL-INRIA RR-6716 (2008), Annales de l'Institut H. Poincaré, Série: Probabilités(B) 47(3) (2011)
28. Cérou, F., Del Moral, P., Furon, T., Guyader, A.: Rare event simulation for a static distribution. Research Report RR-6792, INRIA (2009)
29. Chopin, N.: A sequential particle filter method for static models. Biometrika 89, 539–552 (2002)
30. Coello Coello, C.: List of references on evolutionary multiobjective optimization, http://www.lania.mx/~ccoello/EMOObib.html
31. Coello Coello, C., Van Veldhuizen, D., Lamont, G.: Evolutionary Algorithms for Solving Multi-Objective Problems. In: Genetic Algorithms and Evolutionary Computation, vol. 5. Kluwer Academic Publishers, Boston (2002)
32. Cole, N., Desell, T., Lombraña González, D., Fernández de Vega, F., Magdon-Ismail, M., Newberg, H., Szymanski, B., Varela, C.: Evolutionary Algorithms on Volunteer Computing Platforms: The MilkyWay@Home Project. In: de Vega, F.F., Cantú-Paz, E. (eds.) Parallel and Distributed Computational Intelligence. SCI, vol. 269, pp. 63–90. Springer, Heidelberg (2010)
33. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed Optimization by Ant Colonies. In: European Conference on Artificial Life, pp. 134–142 (1991)

34. Di Chio, C., Brabazon, A., Di Caro, G.A., Drechsler, R., Farooq, M., Grahl, J., Greenfield, G., Prins, C., Romero, J., Squillero, G., Tarantino, E., Tettamanzi, A.G.B., Urquhart, N., Uyar, A.Ş. (eds.): EvoApplications 2011, Part II. LNCS, vol. 6625. Springer, Heidelberg (2011)
35. De Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer (2002), http://books.google.com/books?hl=en&lr=&id=aMFP7p8DtaQC&oi=fnd&pg=PA1&dq=Artificial+immune+systems+a+new+computational+intelligence+approach&ots=zHjlTG5TiP&sig=VKMxGqTe4FhtUai-ET3wdQ2mJ78
36. Del Moral, P.: Non Linear Filtering: Interacting Particle Solution. Markov Processes and Related Fields 2(4), 555–580 (1996)
37. Del Moral, P.: Measure Valued Processes and Interacting Particle Systems. Application to Non Linear Filtering Problems. Annals of Applied Probability 8(2), 438–495 (1998)
38. Del Moral, P.: Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications. Springer, New York (2004)
39. Del Moral, P., Doucet, A.: Particle motions in absorbing medium with hard and soft obstacles. Stochastic Anal. Appl. 22, 1175–1207 (2004)
40. Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. Royal Statist. Soc. B 68, 411–436 (2006)
41. Del Moral, P., Doucet, A., Jasra, A.: On Adaptive Resampling Procedures for Sequential Monte Carlo Methods. Research Report INRIA (HAL-INRIA RR-6700), 46p. (October 2008); In: Bernoulli 18(1), 252–278 (2012)
42. Del Moral, P., Guionnet, A.: On the stability of measure valued processes with applications to filtering. C. R. Acad. Sci. Paris Sér. I Math. 329, 429–434 (1999)
43. Del Moral, P., Guionnet, A.: On the stability of interacting processes with applications to filtering and genetic algorithms. Annales de l'Institut Henri Poincaré 37(2), 155–194 (2001)
44. Del Moral, P., Jacod, J.: Interacting Particle Filtering With Discrete Observations. In: Doucet, A., de Freitas, J.F.G., Gordon, N.J. (eds.) Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science, pp. 43–77. Springer (2001)
45. Del Moral, P., Jacod, J., Protter, P.: The Monte-Carlo Method for filtering with discrete-time observations. Probability Theory and Related Fields 120, 346–368 (2001)
46. Del Moral, P., Jacod, J.: The Monte-Carlo Method for filtering with discrete time observations. Central Limit Theorems. In: Lyons, T.J., Salisbury, T.S. (eds.) The Fields Institute Communications, Numerical Methods and Stochastics. American Mathematical Society (2002)
47. Del Moral, P., Kallel, L., Rowe, J.: Modeling genetic algorithms with interacting particle systems. Revista de Matematica, Teoria y Aplicaciones 8(2) (July 2001)
48. Del Moral, P., Miclo, L.: Asymptotic Results for Genetic Algorithms with Applications to Non Linear Estimation. In: Naudts, B., Kallel, L. (eds.) Proceedings Second EvoNet Summer School on Theoretical Aspects of Evolutionary Computing. Natural Computing. Springer (2000)
49. Del Moral, P., Miclo, L.: On the Stability of Non Linear Semigroup of Feynman-Kac Type. Annales de la Faculté des Sciences de Toulouse 11(2), (2002)
50. Del Moral, P., Lezaud, P.: Branching and interacting particle interpretation of rare event probabilities. In: Blom, H., Lygeros, J. (eds.) Stochastic Hybrid Systems: Theory and Safety Critical Applications. Springer, Heidelberg (2006)

51. Del Moral, P., Miclo, L.: A Moran particle system approximation of Feynman-Kac formulae. Stochastic Processes and their Applications 86, 193–216 (2000)
52. Del Moral, P., Miclo, L.: Branching and interacting particle systems approximations of Feynman-Kac formulae with applications to non linear filtering. In: Azéma, J., Emery, M., Ledoux, M., Yor, M. (eds.) Séminaire de Probabilités XXXIV. Lecture Notes in Mathematics, vol. 1729, pp. 1–145. Springer (2000)
53. Del Moral, P., Miclo, L.: Genealogies and Increasing Propagations of Chaos for Feynman-Kac and Genetic Models. Annals of Applied Probability 11(4), 1166–1198 (2001)
54. Del Moral, P., Miclo, L.: Particle approximations of Lyapunov exponents connected to Schrödinger operators and Feynman-Kac semigroups. ESAIM: Probability and Statistics 7, 171–208 (2003)
55. Del Moral, P., Miclo, L.: Annealed Feynman-Kac models. Comm. Math. Phys. 235, 191–214 (2003)
56. Del Moral, P., Rémillard, B., Rubenthaler, S.: Introduction aux Probabilités. Ellipses Edition (2006)
57. Del Moral, P., Rio, E.: Concentration inequalities for mean field particle models. Technical report HAL-INRIA RR-6901 (2009). Annals of Applied Probability 21(3), 1017–1052 (2011)
58. Del Moral, P., Hu, P., Wu, L.: On the Concentration Properties of Interacting Particle Processes. Foundations and Trends in Machine Learning 3(3-4), 225–389 (2012)
59. Del Moral, P., Rigal, G., Salut, G.: Estimation and nonlinear optimal control: An unified framework for particle solutions LAAS-CNRS, Toulouse, Research Report no. 91137, DRET-DIGILOG- LAAS/CNRS contract (April 1991)
60. Del Moral, P., Rigal, G., Salut, G.: Nonlinear and non Gaussian particle filters applied to inertial platform repositioning. LAAS-CNRS, Toulouse, Research Report no. 92207, STCAN/DIGILOG-LAAS/CNRS Convention STCAN no. A.91.77.013, 94p. (September 1991)
61. Del Moral, P., Rigal, G., Salut, G.: Estimation and nonlinear optimal control: Particle resolution in filtering and estimation. Experimental results. Convention DRET no. 89.34.553.00.470.75.01, Research report no.2, 54p. (January 1992)
62. Del Moral, P., Rigal, G., Salut, G.: Estimation and nonlinear optimal control: Particle resolution in filtering and estimation. Theoretical results Convention DRET no. 89.34.553.00.470.75.01, Research report no.3, 123p. (October 1992)
63. Del Moral, P., Noyer, J.-C., Rigal, G., Salut, G.: Particle filters in radar signal processing: detection, estimation and air targets recognition. LAAS-CNRS, Toulouse, Research Report no. 92495 (December 1992)
64. Del Moral, P., Rigal, G., Salut, G.: Estimation and nonlinear optimal control: Particle resolution in filtering and estimation. Studies on: Filtering, optimal control, and maximum likelihood estimation. Convention DRET no. 89.34.553.00.470.75.01. Research report no.4, 210p. (January 1993)
65. Del Moral, P., Noyer, J.C., Rigal, G., Salut, G.: Traitement non-linéaire du signal par réseau particulaire: Application RADAR. In: Proceedings XIV Colloque GRETSI, Traitement du Signal et des Images, Juan les Pins, France, pp. 399–402 (September 1993)
66. Del Moral, P., Noyer, J.C., Salut, G.: Resolution particulaire et traitement non linéaire du signal: Application radar/sonar. Revue du Traitement du Signal (Septembre 1995)
67. Doucet, A., de Freitas, J.F.G., Gordon, N.J. (eds.): Sequential Monte Carlo Methods in Practice. Springer, New York (2001)

68. Doucet, A., Godsill, S.J., Andrieu, C.: On sequential Monte Carlo sampling methods for Bayesian filtering. Statistics and Computing 10, 197–208 (2000)
69. Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: fifteen years later. In: Crisan, D., Rozovsky, B. (eds.) Handbook of Nonlinear Filtering. Cambridge University Press (2009)
70. Eiben, A.E., Bäck, T.: Empirical investigation of multiparent recombination operators in evolution strategies. Evolutionary Computation 5(3), 347–365 (1997)
71. Eiben, A.E., Hinterding, R., Hinterding, A.E.E.R., Michalewicz, Z.: Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation 3, 124–141 (2000)
72. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer (2003)
73. Eiben, A., Schut, M.: New ways to calibrate evolutionary algorithms. In: Advances in Metaheuristics for Hard Optimization. Natural Computing, pp. 153–177. Springer (2008), http://dblp.uni-trier.de/db/conf/ncs/metaheuristics2008.html#EibenS08
74. Ellouze, M., Gauchi, J.P., Augustin, J.C.: Global sensitivity analysis applied to a contamination assessment model of Listeria monocytogenes in cold smoked salmon at consumption. Risk Anal. 30, 841–852 (2010)
75. Ellouze, M., Gauchi, J.P., Augustin, J.C.: Use of global sensitivity analysis in quantitative microbial risk assessment: Application to the evaluation of a biological time temperature integrator as a quality and safety indicator for cold smoked salmon. In: Food Microbiol. (2010), doi:10.1016/j.fm.2010.05.022
76. Fearnhead, P.: Computational methods for complex stochastic systems: A review of some alternatives to MCMC. Statistics and Computing 18, 151–171 (2008)
77. Fletcher, R., Powell, M.: A rapidly convergent descent method for minimization. Computer Journal 6, 163–168 (1963)
78. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. Computer Journal 7, 149–154 (1964)
79. Fletcher, R.: A new approach to variable metric algorithms. The Computer Journal 13(3), 317–322 (1970), http://comjnl.oxfordjournals.org/cgi/content/abstract/13/3/317, doi:10.1093/comjnl/13.3.317
80. Gauchi, J.P., Vila, J.P., Coroller, L.: New prediction confidence intervals and bands in the nonlinear regression model: Application to the predictive modelling in food. Communications in Statistics, Simulation and Computation 39(2), 322–330 (2009)
81. Gauchi, J.P., Bidot, C., Augustin, J.C., Vila, J.P.: Identification of complex microbiological dynamic system by nonlinear filtering. In: 6th Int. Conference on Predictive Modelling in Foods, Washington DC (2009)
82. Glasserman, P., Heidelberger, P., Shahabuddin, P., Zajic, T.: Multilevel splitting for estimating rare event probabilities. Operations Research 47, 585–600 (1999)
83. Glover, F.: Heuristics for integer programming using surrogate constraints. Decision Sciences 8(1), 156–166 (1977), http://dx.doi.org/10.1111/j.1540-5915.1977.tb01074.x, doi:10.1111/j.1540-5915.1977.tb01074.x
84. Glover, F.: Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. 13(5), 533–549 (1986), http://dx.doi.org/10.1016/0305-05488690048-1, doi:10.1016/0305-0548(86)90048-1
85. Glover, F.: A template for scatter search and path relinking. In: Hao et al. [93], pp. 1–51 (1997)

86. Glynn, P.W., Ormoneit, D.: Hoeffding's inequality for uniformly ergodic Markov chains. Statist. Probab. Lett. 56(2), 143–146 (2002)
87. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
88. Goldfarb, D.: A family of variable metric updates derived by variational means. Mathematics of Computation 24, 23–26 (1970)
89. Gordon, N.J., Salmond, D., Smith, A.F.M.: A novel approach to state estimation to nonlinear non-Gaussian state estimation. IEE Proceedings F 40, 107–113 (1993)
90. Grassberger, P.: Pruned-enriched Rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000. Phys. Rev. E, 3682–3693 (1997)
91. Hansen, N., Müller, S.D., Koumoutsakos, P.: Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). Evol. Comput. 11(1), 1–18 (2003), http://dx.doi.org/10.1162/106365603321828970, doi:10.1162/106365603321828970
92. Hansen, N., Ostermeier, A., Gawelczyk, A.: On the adaptation of arbitrary normal mutation distributions in evolution strategies: The generating set adaptation. In: Proceedings of the 6th International Conference on Genetic Algorithms, pp. 57–64. Morgan Kaufmann Publishers Inc., San Francisco (1995), http://dl.acm.org/citation.cfm?id=645514.657936
93. Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (eds.): AE 1997. LNCS, vol. 1363. Springer, Heidelberg (1998)
94. Harris, T.E., Kahn, H.: Estimation of particle transmission by random sampling. Natl. Bur. Stand. Appl. Math. Ser. 12, 27–30 (1951)
95. Herrera, F., Lozano, M.: Heuristic Crossovers for Real-Coded Genetic Algorithms Based on Fuzzy Connectives. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 336–345. Springer, Heidelberg (1996), http://www.springerlink.com/content/y42m98n165872533, doi:10.1007/3-540-61723-X_998
96. Herrera, F., Lozano, M., Sánchez, A.M.: A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. Int. J. Intell. Syst. 18(3), 309–338 (2003), http://dx.doi.org/10.1002/int.10091, doi:10.1002/int.10091
97. Herrera, F., Lozano, M., Verdegay, J.: Fuzzy connective based crossover operators to model genetic algorithms population diversity. Tech. Rep. DECSAI-95110. University of Granada, Spain (1995)
98. Herrera, F., Lozano, M., Verdegay, J.: Dynamic and heuristic fuzzy connectives-based crossover operators for controlling the diversity and convergence of real-coded genetic algorithms. Int. J. Intell. Syst. 11, 1013–1041 (1996)
99. Herrera, F., Lozano, M., Verdegay, J.: Fuzzy connectives based crossover operators to model genetic algorithms population diversity. Fuzzy Set. Syst. 92(1), 21–30 (1997), doi:10.1016/S0165-0114(96)00179-0
100. Hestenes, M., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Research NBS 49(6), 409–436 (1952)
101. Hestenes, M.R.: Iterative methods for solving linear equations. Report 52-9, NAML (1951); reprinted in J. Optimiz. Theory App. 11, 323–334 (1973)
102. Hetherington, J.H.: Observations on the Statistical Iteration of Matrices. Phys. Rev. A. 30, 2713–2719 (1984)

103. Hinterding, R., Michalewicz, Z., Eiben, A.E.: Adaptation in Evolutionary Computation: A Survey. In: Proceedings of the 4th IEEE International Conference on Evolutionary Computation, pp. 65–69 (1997), http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=592270
104. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
105. Hooke, R., Jeeves, T.: Direct search solution of numerical and statistical problems. Journal of the ACM 8(2), 212–229 (1961), doi:http://doi.acm.org/10.1145/321062.321069
106. Horn, J.: Multicriteria decision making and evolutionary computation. In: Handbook of Evolutionary Computation, Institute of Physics Publishing, London (1997)
107. Ikonen, E., Del Moral, P., Najim, K.: A genealogical decision tree solution to optimal control problems. In: IFAC Workshop on Advanced Fuzzy/Neural Control, Oulu, Finland, pp. 169–174 (2004)
108. Ikonen, E., Najim, K., Del Moral, P.: Application of genealogical decision trees for open-loop tracking control. In: Proceedings of the16th IFAC World Congress, Prague, Czech (2005)
109. Ingber, L.: Adaptive simulated annealing (asa), global optimization c-code. Tech. rep. Caltech Alumni Association (1993)
110. Ingber, L.: Simulated annealing: Practice versus theory. Math. Comput. Model. 18(11), 29–57 (1993)
111. Ingber, L.: Adaptive simulated annealing (asa): Lessons learned. Control and Cybern. 25, 33–54 (1996)
112. Ingber, L.: Adaptive simulated annealing (asa) and path-integral (pathint) algorithms: Generic tools for complex systems. Tech. rep. Chicago, IL (2001)
113. Ingber, L., Rosen, B.: Genetic algorithms and very fast simulated reannealing: A comparison. Math. Comput. Model. 16(11), 87–100 (1992)
114. Johansen, A.M., Del Moral, P., Doucet, A.: Sequential Monte Carlo Samplers for Rare Events. In: Proceedings of 6th International Workshop on Rare Event Simulation, Bamberg, Germany (2006)
115. Jong, K.A.D.: Evolutionary computation - a unified approach. MIT Press (2006)
116. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995), doi:10.1109/ICNN.1995.488968
117. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science 220(4598), 671–680 (1983), citeseer.ist.psu.edu/kirkpatrick83optimization.html
118. Kitagawa, G.: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J. Comp. Graph. Statist. 5, 1–25 (1996)
119. Kolokoltsov, V.N., Maslov, V.P.: Idempotent analysis and its applications. Mathematics and its Applications, vol. 401. Kluwer Academic Publishers Group, Dordrecht (1997); Translation of Idempotent analysis and its application in optimal control, Russian, Nauka Moscow (1994); translated by Nazaikinskii, V. E. With an appendix by Pierre Del Moral: Maslov Optimization Theory: Optimality Versus Randomness, pp. 243–302
120. Künsch, H.R.: State-space and hidden Markov models. In: Barndorff-Nielsen, O.E., Cox, D.R., Kluppelberg, C. (eds.) Complex Stochastic Systems, pp. 109–173. CRC Press (2001)
121. Lagarias, J., Reeds, J., Wright, M., Wright, P.: Convergence properties of the Nelder-Mead simplex algorithm in low dimensions. SIAM J. Optimiz. 9, 112–147 (1998)

122. Langdon, W., Poli, R.: Foundations of Genetic Programming, vol. 5. Springer (2002), http://discovery.ucl.ac.uk/124583/
123. Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer, New York (2001)
124. Martin, O., Otto, S.W., Felten, E.W.: Large-step markov chains for the traveling salesman problem. Complex Systems 5, 299–326 (1991)
125. Melik-Alaverdian, V., Nightingale, M.P.: Quantum Monte Carlo methods in statistical mechanics. Internat. J. of Modern Phys. C. 10, 1409–1418 (1999)
126. Metropolis, N., Ulam, S.: The Monte Carlo Method. Journal of the American Statistical Association 44(247), 335–341 (1949)
127. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E.: Equation of state calculations by fast computing machines. J. Chem. Phys. 21(6), 1087–1092 (1953), http://link.aip.org/link/?JCP/21/1087/1, doi:10.1063/1.1699114
128. Michalewicz, Z.: Genetic algorithms + data structures = evolution programs, 2nd, extended edn. Springer-Verlag New York, Inc., New York (1994)
129. Mitavskiy, B., Rowe, J.: An Extension of Geiringer's Theorem for a Wide Class of Evolutionary Search Algorithms. Evolutionary Computation 14(1), 87–118 (2006)
130. Mitavskiy, B., Rowe, J., Wright, A., Schmitt, L.: Quotients of Markov chains and asymptotic properties of the stationary distribution of the Markov chain associated to an evolutionary algorithm. Genetic Programming and Evolvable Machines 9(2), 109–123 (2008)
131. Mladenović, N.: A variable neighborhood algorithm – a new metaheuristics for combinatorial optimization. In: Abstracts of Papers Presented at Optimization Days, Montreal (1995)
132. Mladenović, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. 24(11), 1097–1100 (1997), http://dx.doi.org/10.1016/S0305-05489700031-2, doi:10.1016/S0305-0548(97)00031-2
133. Moscato, P.: Memetic algorithms: a short introduction. In: Corne, D., Dorigo, M., Glover, F., Dasgupta, D., Moscato, P., Poli, R., Price, K. (eds.) New Ideas in Optimization, pp. 219–234. McGraw-Hill Ltd., UK (1999), http://dl.acm.org/citation.cfm?id=329055.329078
134. Mühlenbein, H., Schlierkamp-Voosen, D.: Analysis of selection, mutation and recombination in genetic algorithms. Evolution and Biocomputation, 142–168 (1995)
135. Nelder, J.A., Mead, R.: A simplex method for function minimization. Comput. J. 7(4), 308–313 (1965).http://comjnl.oxfordjournals.org/cgi/content/abstract/7/4/308, doi:10.1093/comjnl/7.4.308
136. Neri, F., Cotta, C., Moscato, P.: Handbook of Memetic Algorithms. SCI. Springer (2011), http://books.google.lu/books?id=uop6UvKu8q4C
137. Nocedal, J.: Updating quasi-newton matrices with limited storage. Math. Comput. 35(151), 773–782 (1980), http://www.jstor.org/stable/2006193
138. Pelikan, M., Goldberg, D.E., Lobo, F.G.: A survey of optimization by building and using probabilistic models. Comput. Optim. Appl. 21(1), 5–20 (2002), http://dx.doi.org/10.1023/A:1013500812258, doi:10.1023/A:1013500812258
139. Polak, E., Ribière, G.: Note sur la convergence des méthodes de directions conjuguées. Revue Française d'informatique et de Recherche Opérationnelle 16, 35–43 (1969)
140. Powell, M.: On the Convergence of the Variable Metric Algorithm. Journal of the Institute of Mathematics and its Applications 7, 21–36 (1971)
141. Rao, S., Shanta, C.: Numerical Methods: With Program in Basic, Fortan, Pascal & C++. Orient Blackswan (2004)

142. Reynolds, R.G., Sverdlik, W.: Problem solving using cultural algorithms. In: International Conference on Evolutionary Computation, pp. 645–650 (1994)
143. Rosenbluth, M.N., Rosenbluth, A.W.: Monte-Carlo calculations of the average extension of macromolecular chains. J. Chem. Phys. 23, 356–359 (1955)
144. Vila, J.-P., Rossi, V.: Nonlinear filtering in discret time: A particle convolution approach. Biostatistic Group of Montpellier, Technical Report 04-03 (2004), http://vrossi.free.fr/recherche.html
145. Rudolph, G.: Convergence of Evolutionary Algorithms in General Search Spaces. In: International Conference on Evolutionary Computation, pp. 50–54 (1996)
146. Rudolph, G.: Finite Markov Chain Results in Evolutionary Computation: A Tour d'Horizon. Fundam. Inform. 35(1-4), 67–89 (1998)
147. Schmitt, F., Rothlauf, F.: On the Importance of the Second Largest Eigenvalue on the Convergence Rate of Genetic Algorithms. In: Beyer, H., Cantu-Paz, E., Goldberg, D., Parmee, Spector, L., Whitley, D. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2001), pp. 559–564. Morgan Kaufmann Publishers, San Francisco (2001)
148. Schwefel, H.P., Rudolph, G.: Contemporary Evolution Strategies. In: Morán, F., Merelo, J.J., Moreno, A., Chacon, P. (eds.) ECAL 1995. LNCS, vol. 929, pp. 893–907. Springer, Heidelberg (1995)
149. Shanno, D.: Conditioning of quasi-newton methods for function minimization. Math. Comput. 24(111), 647–656 (1970)
150. Shewchuk, J.: An introduction to the conjugate gradient method without the agonizing pain. Tech. rep., Carnegie Mellon University, Pittsburgh, Pittsburgh, PA, USA (1994), http://portal.acm.org/citation.cfm?id=865018
151. Solis, F., Wets, R.B.: Minimization by random search techniques. Math. Oper. Res. 6, 19–30 (1981)
152. Spears, W.M., Jong, K.A.D., Ba, T., Fogel, D.B., Garis, H.D.: An overview of evolutionary computation. Evolutionary Computation 667(1), 442–459 (1993), http://www.springerlink.com/index/Y03055H012777681.pdf
153. Spendley, W., Hext, G., Himsworth, F.: Sequential application of simplex designs in optimisation and evolutionary operation. Technometrics 4(4), 441–461 (1962)
154. Stadler, P.: Towards a theory of landscapes. In: Lopéz-Peña, R., Capovilla, R., García-Pelayo, R., Waelbroeck, H., Zertuche, F. (eds.) Complex Systems and Binary Networks, vol. 461, pp. 77–163. Springer, Berlin (1995)
155. Stadler, P., Flamm, C.: Barrier trees on poset-valued landscapes. Genet. Program. Evol. M. 4(1), 7–20 (2003), http://dblp.uni-trier.de/db/journals/gpem/gpem4.html%5c#StadlerF03
156. Stewart, C.A., Mueller, M.S., Lingwall, M.: Progress Towards Petascale Applications in Biology: Status in 2006. In: Lehner, W., Meyer, N., Streit, A., Stewart, C. (eds.) Euro-Par Workshops 2006. LNCS, vol. 4375, pp. 289–303. Springer, Heidelberg (2007), http://dl.acm.org/citation.cfm?id=1765606.1765638
157. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. of Global Optimization 11(4), 341–359 (1997), http://dx.doi.org/10.1023/A:1008202821328, doi:10.1023/A:1008202821328
158. Surhone, L.M., Tennoe, M.T., Henssonow, S.F.: Leiden Classical. VDM Verlag Dr. Mueller AG & Company Kg (2010)
159. Tantar, E., Dhaenens, C., Figueira, J.R., Talbi, E.G.: A priori landscape analysis in guiding interactive multi-objective metaheuristics. In: IEEE Congress on Evolutionary Computation, pp. 4104–4111 (2008)

160. Tantar, E., Schuetze, O., Figueira, J.R., Coello, C.A.C., Talbi, E.G.: Computing and selecting epsilon-efficient solutions of 0,1-knapsack problems. In: Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems. Lecture Notes in Econom. and Math. Systems, vol. 634, pp. 379–387 (2010)
161. Tsang, E., Voudouris, C.: Fast local search and guided local search and their application to British Telecom's workforce scheduling problem. Oper. Res. Lett. 20(3), 119–127 (1997), http://www.sciencedirect.com/science/article/pii/S0167637796000429, doi:10.1016/s0167-6377(96)00042-9
162. Weinberger, E.: Correlated and uncorrelated fitness landscapes and how to tell the difference. Biol. Cybern. 63, 325–336 (1990), http://dx.doi.org/10.1007/BF00202749, doi:10.1007/BF00202749