Fast resampling for sequential Monte Carlo with millions of particles

L. Martino[⋆], V. Elvira[◇]

[⋆] Universitá degli Studi di Catania, Italy.

[◇] University of Edinburgh, United Kingdom (UK).

Abstract

Particle filtering (PFs) and, more generally, sequential Monte Carlo (SMC) methods are essential tools for Bayesian inference. Over the years, many SMC variants have been proposed, yet their core always relies on importance sampling following by a resampling step. While resampling is crucial to mitigate particle degeneracy and to maintain a stable approximation of the posterior distribution, it often represents a significant computational bottleneck. In this work, we present a novel, fast, resampling procedure that provides significant computational gains in demanding (often high-dimensional) scenarios where a large number of particles is required, and the small effective sample size (ESS) is small. The effectiveness of the proposed approach is demonstrated through a series of numerical experiments showing remarkable performance. In addition, a theoretical analysis and related code implementation are provided.

Keywords: Bayesian inference; Sequential Monte Carlo; Particle Filtering; Resampling.

1 Introduction

Particle filtering also known as sequential Monte Carlo (SMC), has become very popular methodologies in signal processing, statistics, and machine learning [6, 8, 22, 23, 10]. These methods play a central role in Bayesian inference, particularly for nonlinear and non-Gaussian state-space models, where analytical solutions are intractable. More recently, they have also held an important place in approximate Bayesian computation (ABC) and other static inference problems. SMC methods have been adopted in various fields, including finance, geophysical systems, wireless communications, control, navigation and tracking, and robotics, to name a few [12, 14, 17].

Many types of particle filters (PFs) or SMC methods have been introduced in the literature. However, the core of all of them is formed by and *importance sampling (IS) plus resampling* (IS+R) technique (i.e., IS followed by a resampling step), which consists of three main operations [35, 24]. The importance sampling (IS) part includes the first two operations: 1) particle propagation/generation and 2) weight computation. The last operation is 3) the resampling step.

The resampling procedure replaces one the weighted particles at one iteration with another cloud of (equally weighted) particles, according to the normalized importance weights associated to the first set of particles. The resampling is essential for PFs and/or SMC methods: without this step, a PF quickly produces a degenerate set of particles, i.e., a set in which a few particles (often a unique one) dominate the rest of the particles with their weights [6, 8, 22]. This means that the obtained estimates will be inaccurate and/or have extremely large variances. With the application of the resampling, such deteriorations are prevented, which is why it is highly important to SMC methods. Hence, resampling has been extensively researched and, as a consequence, various resampling schemes have been proposed. Some examples are multinomial resampling, hereafter denoted as standard resampling (SR), stratified resampling, systematic resampling, and residual resampling [7, 20, 19]. However, all these resampling schemes differ essentially for *the variance* introduced in the overall SMC algorithm [2, 16].

Therefore, the resampling stepis essential to mitigate particle degeneracy and ensure a stable approximation of the posterior distribution. However, resampling often represents a significant computational bottleneck for PFs and SMC schemes. The first reason is that resampling needs normalized weights and for this scope we need a weight summation that cannot be completely parallelized (without approximations) [1, 3, 32]. The second reason is that the need of cumulative sums and random sampling, which can become costly as the number of particles increases. As a consequence, resampling frequently dominates the overall computational time, especially in large-scale or real-time applications, thereby motivating the development of more efficient and adaptive resampling strategies [5]. For instance, the resampling step is often applied only at certain iterations, when effective sample size (ESS) is small [30, 11, 28]. More recent contributions can be found in [4, 13, 27, 36].

In this work, we introduce a novel resampling procedure that is substantially faster when a large number of particles is used and when the ESS is small. Both circumstances are commonly encountered in practical real-world (high-dimensional) scenarios. The idea is inspired by the information theory concept of minimal codeword length in source coding. Two groups of particles are created: one groups includes $M < N$ samples with the biggest weights, and a second group contains the remaining $N-M$ samples. Then, according to a suitable probability, the resampling is applied much more times within the first (smaller) group. The proposed procedure is particularly efficient when the first group contains an high probability mass, which is often the case in

challenging scenarios. The novel scheme can be applied in combination with any resampling approach, such as multinomial, stratified, systematic, or residual, to name a few [20, 19].
We prove the unbiasedness condition of the method. Furthermore, we describe the feasible working zones where the method is useful. We also discuss the optimal choice of M providing a suitable approximation. Several numerical experiments demonstrate the computational benefits of the proposed resampling scheme, especially for large numbers of particles. Remarkable results have been obtained reducing more than 60% the computational time within a particle filter, in which resampling is applied in approximately half of the iterations. For reproducibility, related Matlab code is available at `http://www.lucamartino.altervista.org/FAST_RESAMPLING_public_code.zip`.
The rest of work is structured as follows. Section 2 contains some background material and preliminaries. Section 3 describes the proposed scheme. Section 4 is devoted to the theoretical analysis of the proposed method. The numerical experiments are given in Section 5. Some final conclusions are provided in Section 6.

2 Preliminaries

2.1 Bayesian inference

In many real world applications, the goal is to infer a variable of interest given a set of data [22, 34]. Let us denote the parameter of interest (static or dynamic) by $\boldsymbol{\theta} = [\theta_1, ..., \theta_{d_\theta}]^\top \in \mathcal{X} \subseteq \mathbb{R}^{d_\theta}$, and let $\mathbf{y} \in \mathbb{R}^{d_y}$ be the observed data. In a Bayesian analysis, all the statistical information is contained in the posterior distribution, which is given by

$$\bar{\pi}(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\mathbf{y}) = \frac{\ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})}{Z(\mathbf{y})}, \tag{1}$$

where $\ell(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood function, $g(\boldsymbol{\theta})$ is the prior pdf, and $Z(\mathbf{y})$ is the Bayesian model evidence (a.k.a. marginal likelihood). The marginal likelihood $Z(\mathbf{y})$ is important for model selection purposes [26]. Generally, $Z(\mathbf{y})$ is unknown, so we are able to evaluate the unnormalized target function, $\pi(\boldsymbol{\theta}) = \ell(\mathbf{y}|\boldsymbol{\theta})g(\boldsymbol{\theta})$. The analytical computation of integrals involving the posterior density $\bar{\pi}(\boldsymbol{\theta}) \propto \pi(\boldsymbol{\theta})$ is often unfeasible, hence numerical approximations are needed. Thus, the goal is often to approximate integrals of the form

$$I = \int_{\mathcal{X}} f(\boldsymbol{\theta})\bar{\pi}(\boldsymbol{\theta})d\boldsymbol{\theta} = \frac{1}{Z}\int_{\mathcal{X}} f(\boldsymbol{\theta})\pi(\boldsymbol{\theta})d\boldsymbol{\theta}, \tag{2}$$

where $f(\boldsymbol{\theta})$ is some integrable function, and

$$Z = \int_{\mathcal{X}} \pi(\boldsymbol{\theta})d\boldsymbol{\theta}. \tag{3}$$

In the literature, random sampling or deterministic quadratures are often used [34, 31, 33]. In this work, we focus on the so-called importance sampling (IS) approach.

2.2 Importance sampling 'plus' resampling (IS+R)

Let us consider a normalized proposal density $q(\boldsymbol{\theta})$.[1] The importance sampling (IS) method consists of drawing N samples, $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$, from $q(\boldsymbol{\theta})$ (also called particles), and then assign to each sample the following unnormalized weights

$$w_n = \frac{\pi(\boldsymbol{\theta}_n)}{q(\boldsymbol{\theta}_n)}, \quad n = 1, \ldots, N. \tag{4}$$

An unbiased estimator of the marginal likelihood Z is given by the arithmetic mean of these unnormalized weights [22, 34], i.e.,

$$\widehat{Z} = \frac{1}{N}\sum_{n=1}^{N} w_n, \quad \text{and } \bar{w}_n = \frac{w_n}{\sum_{i=1}^{N} w_i} = \frac{w_n}{N\widehat{Z}},$$

are the normalized weights, with $n = 1, \ldots, N$. The self-normalized IS estimator of I in Eq. (2) is given by

$$\widehat{I} = \sum_{n=1}^{N} \bar{w}_n f(\boldsymbol{\theta}_n).$$

More generally, regardless of the specific function $f(\boldsymbol{\theta})$, with IS, we obtain a particle approximation of the measure of $\bar{\pi}$, i.e.,

$$\widehat{\pi}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \bar{w}_n \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_n), \tag{5}$$

where $\delta(\boldsymbol{\theta})$ is a delta function. It is important to remark that with this particle approximation, we can approximate several quantities related to the posterior $\bar{\pi}(\boldsymbol{\theta})$, such as any moments and/or credible intervals (not just a specific integral). The quality of this particle approximation is related to the discrepancy between the proposal $q(\boldsymbol{\theta})$ and the posterior $\bar{\pi}(\boldsymbol{\theta})$ [25].

Remark 1. IS+R is the core of the well-known particle filtering techniques, a.k.a., sequential Monte Carlo algorithms.

[1] We assume that $q(\boldsymbol{\theta}) > 0$ for all $\boldsymbol{\theta}$ where $\bar{\pi}(\boldsymbol{\theta}) > 0$, and $q(\boldsymbol{\theta})$ has heavier tails than $\bar{\pi}(\boldsymbol{\theta})$.

2.3 Standard resampling (SR) schemes

All classical resampling schemes are distinct procedures sharing the same goal: drawing N samples within the set $\{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N\}$ according to probability mass $\bar{w}_n$, $n = 1, ..., N$, i.e.,

$$\bar{\boldsymbol{\theta}}_n \sim \widehat{\pi}(\boldsymbol{\theta}) = \sum_{i=1}^{N} \bar{w}_i \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_i). \tag{6}$$

The output of a resampling scheme is a new set of N particles, $\{\bar{\boldsymbol{\theta}}_1, ..., \bar{\boldsymbol{\theta}}_N\}$, where $\bar{\boldsymbol{\theta}}_n \in \{\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_N\}$, for all n. In practice, particles with higher importance are often replicated several times, while those with lower importance are often discarded. Namely, we modify the weighted particle approximation $\widehat{\pi}$ to a particle approximation $\widehat{\pi}_{\texttt{res}}$ by eliminating particles having lower importance weights and by multiplying particles having higher importance weights, i.e.,

$$\widehat{\pi}_{\texttt{res}}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}_n) = \sum_{r=1}^{N} \frac{N_r}{N} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_r), \tag{7}$$

where $N_r \geq 0$ is a non-negative integer number, that is the number of copies of the particle $\boldsymbol{\theta}_r$ in the new set of resampled particles $\{\bar{\boldsymbol{\theta}}_1, ..., \bar{\boldsymbol{\theta}}_N\}$. Hereafter, we denote with $R \leq N$ the total count of distinct particles contained in the set $\{\bar{\boldsymbol{\theta}}_1, ..., \bar{\boldsymbol{\theta}}_N\}$.

Remark 2. There are different standard resampling schemes [19, 20], which essentially differ for the variability introduced by their procedures [16]. Except for some special cases, there is no substantial difference in the overall computational time among them. Moreover, the time required for performing the resampling procedure generally is an increasing function of N. All the results presented here regarding computational time are valid for all these different strategies.

Properness. All the resampling schemes, to be unbiased, must satisfy the proper-weighting condition [16, 21, 29]. Namely, the expected number of times, N_m, that the m-th particle is resampled, must be proportional to $\bar{w}_m$, i.e.,

$$\mathbb{E}\left(N_m \mid \bar{w}_m\right) = N \bar{w}_m = N \frac{w_m}{N \widehat{Z}} = \frac{w_m}{\widehat{Z}}. \tag{8}$$

3 Proposed scheme

Let us assume the normalized weights $\bar{w}_n$, $n = 1, ..., N$. We define the normalized weights sorted in decreasing order, i.e.,

$$\bar{w}_{j_1} \geq \bar{w}_{j_2} \geq ... \geq \bar{w}_{j_N}, \tag{9}$$

where $j_i \in \{1, ..., N\}$ with $i = 1, ..., N$. Namely, we have by definition $\bar{w}_{j_1} = \max \bar{w}_n$, and $\bar{w}_{j_N} = \min \bar{w}_n$. Clearly, the order is valid also for the unnormalized weights, $w_{j_1} \geq w_{j_2} \geq .. \geq w_{j_N}$. Hence, we can also order the samples

$$\widetilde{\boldsymbol{\theta}}_1 = \boldsymbol{\theta}_{j_1},\ \widetilde{\boldsymbol{\theta}}_2 = \boldsymbol{\theta}_{j_2}, \ldots,\ \widetilde{\boldsymbol{\theta}}_N = \boldsymbol{\theta}_{j_N}.$$

Then, the user choose a integer value $M < N$, such that the particles are divided into two groups: from $\widetilde{\boldsymbol{\theta}}_1$ to $\widetilde{\boldsymbol{\theta}}_M$ and from $\widetilde{\boldsymbol{\theta}}_{M+1}$ to $\widetilde{\boldsymbol{\theta}}_N$. Moreover, the probability of selecting the first group will be

$$\bar{s}_M = \sum_{k=1}^{M} \bar{w}_{j_k} = \frac{\sum_{k=1}^{M} w_{j_k}}{\sum_{n=1}^{N} w_n}. \tag{10}$$

Remark 3. Note that $\bar{s}_M = \sum_{k=1}^{M} \bar{w}_{j_k}$ depends on M. Moreover, $\bar{s}_M \in (0, 1]$ by definition.

We can obtain the normalized weights within the first group as

$$\bar{\rho}_m = \frac{w_{j_m}}{\sum_{k=1}^{M} w_{j_k}}, \quad m = 1, ..., M, \tag{11}$$

and in the second group as

$$\bar{\gamma}_k = \frac{w_{j_k}}{\sum_{i=M+1}^{N} \bar{w}_{j_i}}, \quad k = M+1, ..., N. \tag{12}$$

The proposed resampling algorithm is then given in Table 1. This procedure is partially grounded in the group importance sampling approach in [29] and other ideas applied for parallelization purpose (see also [3, 32, 27]).
Step-1 in Table 1 is equivalent to resample N times over 2 possible artificial "particles", due to the random selection of the N binary indices $b_n \in \{0, 1\}$. The number of times that we select $b_n = 1$ is denoted by R. Note that Step-2 in Table 1 consists of performing R resampling steps over M particles. Finally, in Step-3 of Table 1, we perform $N - R$ resampling steps over $N - M$ particles.

4 Theoretical analysis

In the first part of this section, we will answer the questions: *does it properly works?* and *in which scenario is it useful?* The first question is answered in Section 4.1. The second question is addressed in Section 4.2. Furthermore, in the last two sections, we discuss the optimal choice of M and the best possible scenario for the proposed FR scheme.

Table 1: **The proposed fast resampling (FR) scheme.**

- Initialization: Set $i = 0$. Choose a value $M < N$ (or set $M = \widehat{M}_2$ in Eq. (32)). Build the group with M particles and compute $\bar{s}_M$.

1. Draw N binary indices $b_n \in \{0, 1\}$, $n = 1, ..., N$, according to the probability mass $\text{Prob}(b_n = 1) = \bar{s}_M$ and $\text{Prob}(b_n = 0) = 1 - \bar{s}_M$. Let $R = \sum_{n=1}^{N} b_n$ be the number of indices n for which $b_n = 1$ and, as a consequence, $N - R$ is the number of indices with $b_n = 0$.

2. Draw R samples $\bar{\boldsymbol{\theta}}_i$, with $i = 1, ..., R$, within the M possible samples $\{\widetilde{\boldsymbol{\theta}}_1, ..., \widetilde{\boldsymbol{\theta}}_M\}$ according to the probability mass

$$\bar{\rho}_m = \frac{w_{j_m}}{\sum_{k=1}^{M} w_{j_k}}, \qquad m = 1, ..., M. \tag{13}$$

3. Draw $N - R$ samples $\bar{\boldsymbol{\theta}}_k$, with $k = R + 1, ..., N$, within the $N - M$ possible samples $\{\widetilde{\boldsymbol{\theta}}_{M+1}, ..., \widetilde{\boldsymbol{\theta}}_N\}$ according to the probability mass

$$\bar{\gamma}_m = \frac{w_{j,}}{\sum_{i=M+1}^{N} \bar{w}_{j_i}}, \qquad m = M + 1, ..., N. \tag{14}$$

- Output: Return all the N resampled particles $\bar{\boldsymbol{\theta}}_i$, for $i = 1, ..., N$ (the total number of resampled particles is always N).

4.1 Proper-weighting condition

Below, we show that proper-weighting condition in Eq (8) is fulfilled by the proposed FR scheme.

Theorem. The proposed scheme satisfy the theoretical requirement in Eq (8), i.e.,

$$\mathbb{E}\left(N_m \mid \bar{w}_m\right) = N\bar{w}_m, \quad \forall m. \tag{15}$$

Proof. First of all, note that $E[R] = N\bar{s}_M$, if $m \leq M$, we have:

$$\mathbb{E}\left(N_{j_m} \mid \bar{w}_{j_m}\right) = E[R]\bar{\rho}_m = N\bar{s}_M\bar{\rho}_m, \tag{16}$$

$$= N\left(\sum_{k=1}^{M}\bar{w}_{j_k}\right)\frac{w_{j_m}}{\sum_{k=1}^{M} w_{j_k}} \tag{17}$$

$$= N\left(\frac{\sum_{k=1}^{M} w_{j_k}}{\sum_{n=1}^{N} w_n}\right)\frac{w_{j_m}}{\sum_{k=1}^{M} w_{j_k}} \tag{18}$$

$$= N\frac{w_{j_m}}{\sum_{n=1}^{N} w_n} = N\bar{w}_{j_m}, \tag{19}$$

for all $m \leq M$. Otherwise, if $m > M$, we have:

$$\mathbb{E}\left(N_{j_m} \mid \bar{w}_{j_m}\right) = N(1-\bar{s}_M)\bar{\gamma}_m, \tag{20}$$

$$= N\left(1-\sum_{k=1}^{M}\bar{w}_{j_k}\right)\frac{w_{j_m}}{\sum_{i=M+1}^{N} w_{j_i}}, \tag{21}$$

$$= N\left(\frac{\sum_{n=1}^{N} w_n - \sum_{k=1}^{M} w_{j_k}}{\sum_{n=1}^{N} w_n}\right)\frac{w_{j_m}}{\sum_{i=M+1}^{N} w_{j_i}}, \tag{22}$$

$$= N\left(\frac{\sum_{i=M+1}^{N} w_{j_i}}{\sum_{n=1}^{N} w_n}\right)\frac{w_{j_m}}{\sum_{i=M+1}^{N} w_{j_i}}, \tag{23}$$

$$= N\frac{w_{j_m}}{\sum_{n=1}^{N} w_n} = N\bar{w}_{j_m}, \tag{24}$$

for all $m > M$. Since $j_m \in \{1, ..., M\}$ are just reordered indices, we have proved Eq. (8) (or Eq. (15)).

4.2 Feasible working zones

Let us assume that the computational time required for a single resampling step over N particles is $T_{\texttt{tot}} = T_u \cdot \psi(N)$, where T_u is a unit time measure and $\psi(\cdot) : \mathbb{N} \to \mathbb{R}^+$ is a positive increasing function of the number of particles. The simplest case is $\psi(N) = N$ and, in that case, we are assuming linear scaling with respect to N of the computational time required by the chosen resampling procedure. The key point is to compute the averaged number of particles that are resampled in the proposed scheme of Table 1. In any case, we perform at least a resampling with two possible outcomes (with probability $\bar{s}_M$ and $1 - \bar{s}_M$) for deciding in which group we would resample from. If the first group is selected, a resampling over M particles is performed.

If the second group is selected, a resampling over $N - M$ particles is performed. Hence, the total averaged number of particles $N_{\texttt{av}}$ is

$$N_{\texttt{av}} = \phi(\bar{s}_M, M) = 2 + \bar{s}_M M + (1 - \bar{s}_M)(N - M). \tag{25}$$

Recall that $\mathbb{E}[R] = \bar{s}_M$, and $\bar{s}_M$ depends also on M. Note also thats, since $\bar{s}_M$ depends on M, choosing M we have $\bar{s}_M$ and vice-versa; so that we could write $\phi(\bar{s}_M, M) = \phi(M) = \phi(\bar{s}_M)$. Moreover, we desire that the computational time required by the proposed scheme is smaller than $T_{\texttt{tot}} = T_u \cdot \psi(N)$, i.e.,

$$T_u \cdot \psi\Big(\phi(\bar{s}_M, M)\Big) = T_u \cdot \psi\Big(2 + \bar{s}_M M + (1 - \bar{s}_M)(N - M)\Big) + \epsilon < T_u \cdot \psi(N), \tag{26}$$

where $\epsilon > 0$ is the time wasted in all the tasks due to: (a) sort the weights, (b) split properly the particles in two groups and (c) the rest of required code to perform the proposed scheme. The case $\epsilon \to 0$ is theoretically analyzed in Appendix A. These theoretical studies (in an ideal scenario) and the numerical experiments (in real, practical scenario) show that, as N grows, virtually all the pairs $\{\bar{s}_M, M\}$ tends to be feasible.

4.3 Optimal choice of M

In this section, we define the optimal value of M (or $\bar{s}_M$) and discuss its possible approximation. We should minimize the function $\phi(\bar{s}_M, M) : (0, 1] \times \mathbb{N}^+ \to \mathbb{R}$,

$$\phi(\bar{s}_M, M) = 2 + \bar{s}_M M + (1 - \bar{s}_M)(N - M), \tag{27}$$

$$\phi(M) = 2 + \left(\sum_{k=1}^{M} \bar{w}_{j_k}\right) M + \left(1 - \sum_{k=1}^{M} \bar{w}_{j_k}\right)(N - M). \tag{28}$$

As shown in the last expression above recall that, since $\bar{s}_M$ depends on M, choosing M we have $\bar{s}_M$ and vice-versa, so that $\phi(\bar{s}_M, M) = \phi(M) = \phi(\bar{s}_M)$. Hence, for instance, we can write

$$M^* = \arg\min \phi(M) = \arg\min \Big[2 + \bar{s}_M M + (1 - \bar{s}_M)(N - M)\Big]. \tag{29}$$

We denote the corresponding optimal value of $\bar{s}_M$ as $\bar{s}^*_M = \bar{s}_{M^*}$. The analytical optimization is not straightforward, but some general considerations can be done.

Theorem 1. *The optimal value M^* exists and is unique.*

Proof. Observing the function $\phi(M)$ we can see that there are two pieces:

$$\phi(M) = 2 + \underbrace{\bar{s}_M M}_{\texttt{inc}\uparrow} + \underbrace{(1 - \bar{s}_M)(N - M)}_{\texttt{dec}\downarrow}$$

we have an increasing part $f_i(M) = \bar{s}_M M$ and a decreasing part $f_d(M) = (1 - \bar{s}_M)(N - M)$, so that we can write

$$\phi(M) = 2 + f_i(M) + f_d(M).$$

Only for simplify the explanation, let assume that M is a continuous variable and $\bar{s}_M$ is function of this continuous variable M. Computing the derivative $\frac{d\phi}{dM} = \frac{df_i}{dM} + \frac{df_d}{dM}$, we can ensure that exists an M^* such that $\frac{d\phi}{dM}\big|_{M^*} = 0$ since we always have $\frac{df_i}{dM} > 0$ and $\frac{df_d}{dM} < 0$ (and both continuous functions). This is a consequence of the *intermediate value theorem* (a.k.a., *Bolzano's theorem*), i.e., the sum of two continuous increasing and decreasing functions reaches a stationary point, at some point of the domain.
It is not straightforward to compute analytically the optimal value M^*, or $\bar{s}^*_M = \bar{s}_{M^*}$. A numerical search is always possible but, generally, is not fast. Some possible approximations is described below.

A first proxy $\widehat{M}_1$ of M^*. A possible proxy of M^* can be obtained finding the point where the functions $f_i(M) = \bar{s}_M M$ and $f_d(M) = (1 - \bar{s}_M)(N - M)$ are equal, i.e., $f_i(M) = f_d(M)$. Hence,

$$\begin{aligned}
&\bar{s}_M M = (1 - \bar{s}_M)(N - M) \\
&\bar{s}_M M - (1 - \bar{s}_M)(N - M) = 0 \\
&\bar{s}_M M + (\bar{s}_M - 1)N + M - \bar{s}_M M = 0 \\
&(\bar{s}_M - 1)N + M = 0,
\end{aligned} \tag{30}$$

so that

$$\begin{cases} M^* \approx \widehat{M}_1 = \lceil (1 - \widehat{s}_M)N \rceil, \\ \bar{s}^*_M \approx \widehat{s}_M = \dfrac{N - \widehat{M}_1}{N}, \end{cases} \qquad \text{(mainly valid for small } N\text{)}, \tag{31}$$

where we have denoted as $\lceil a \rceil$ the ceiling function of a value a, i.e., the least integer greater than or equal to a. See Figure 1 for two graphical examples.

Remark 4. The approximation $\widehat{M}_1 \approx M^*$ in (31) is usually valid for small N, or for linear decays of the ordered weights $\bar{w}_{j_n}$ in Eq (9). See the numerical simulations in Section 5.2.
However, the computation of $\widehat{M}_1$ is still not straightforward since both equations in (31) must be fulfilled jointly. A more useful alternative is given below.

A second proxy $\widehat{M}_2$ of M^*. We have empirically found that the number of normalized weights bigger or equal to $\frac{1}{N}$, i.e.,

$$\widehat{M}_2 = \text{N-plus} = \#\left\{\bar{w}_n \geq 1/N, \quad \forall n = 1, \ldots, N\right\}, \tag{32}$$

is a good proxy for M^*. We have highlighted above that $\widehat{M}_2$ coincides with an ESS measure introduced in [30] and denoted as N-plus (or N^+). See Section 5.2, for some related numerical results.

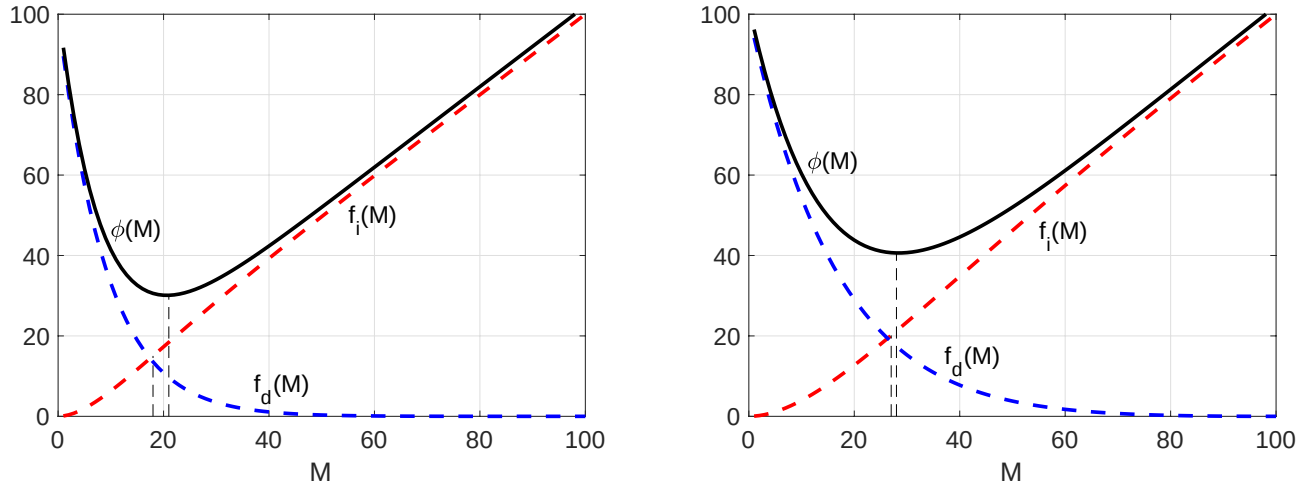


Figure 1: Graphical examples of the functions ϕ, f_i and f_d. The unnormalized weights are chosen as $w_n = \exp(-0.1n)$ in **(a)** and $w_n = \exp(-0.05n)$ in **(b)**. The number of particle is $N = 100$. The first proxy and true optimal values are $\widehat{M}_1 = 18$, $M^* = 21$ in **(a)** and $\widehat{M}_1 = 27$, $M^* = 28$ in **(b)**.

4.4 Best scenario

Some additional considerations are given in Appendix B. In this section, we report the final resulting considerations. The main conclusion is that the favorable scenario for FR occurs when the effective sample size (ESS) value is small.

In fact, we can assert that the best (extreme) scenario is when just one particle concentrates all the probability mass, i.e., $\bar{s}_M = \bar{s}_1 = 1$ and $M = 1$ (i.e., minimum possible ESS value). In this case, we can create two groups one with the unique particle ($M = 1$) with $\bar{s}_1 = 1$ and, the second group with the rest of $N-1$ particles, with zero mass $1-\bar{s}_1 = 0$. The computational cost would be $\psi(2+1)T_u = \psi(3)T_u$ (see Eq. (26)). Actually it would be even less, only $\psi(2)T_u$, since the second resampling stage must not be performed because of only one particle is contained in the first group (hence, we have a deterministic choice; recall that the second group will be never chosen since has zero mass).

Remark 5. Note that the resampling steps with particle filters and SMC schemes are required exactly when the ESS is small. Hence, we have the coincidence/agreement between the optimal scenarios for the proposed FR algorithm and the need of resampling steps with SMC methods.

5 Numerical simulations

In this section, we compare the proposed FR and the SR approaches in different scenarios. In Section 5.1, we consider different ESS frameworks, and test the proposed method varying the values of N and M. In a second experiment (Section 5.2). we study the behavior of the proxies $\widehat{M}_1$ and $\widehat{M}_2$. The use within a particle filter is analyzed in Section 5.3. Related Matlab code is available at `http://www.lucamartino.altervista.org/FAST_RESAMPLING_public_code.zip`.

5.1 First experiment

In this section, the weights are constructed as

$$w_n = \exp(-\beta n), \qquad \beta > 0, \tag{33}$$

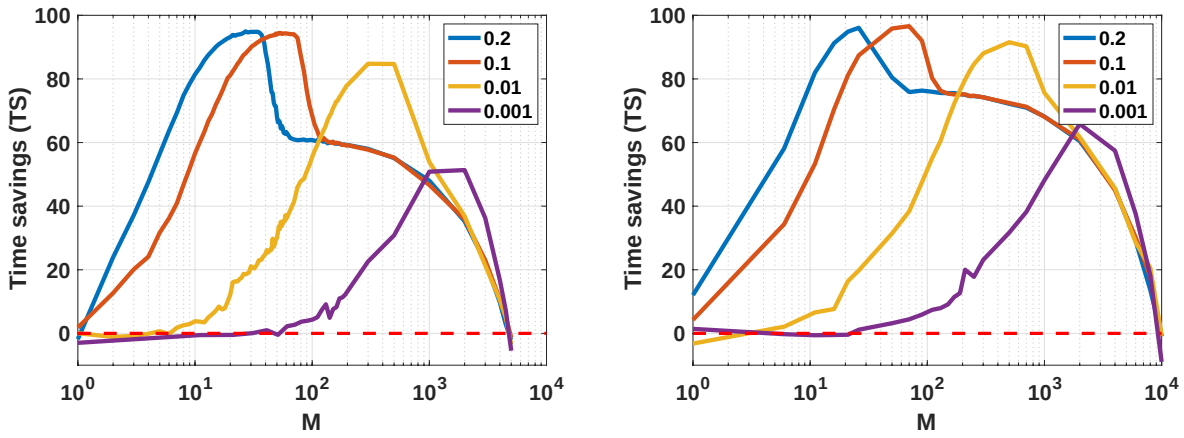
with $n = 1, ..., N$. Then, we compute $\bar{w}_n = \frac{w_n}{\sum_{i=1}^{N} w_i}$. We remark that even the weights are already sorted in decreasing order by construction, we perform the sorting procedure in the code (i.e., we apply the command '`sort`') in order to obtain a fair comparison with the standard resampling (SR) scheme. Note that as $\beta \to 0$ the ESS grows converging to its maximum value, that is N. Otherwise, if $\beta \to \infty$, the ESS decreases converging to its minimum value, that is 1 [30, 11, 28]. We test four values of $\beta\{0.001, 0.01, 0.1, 0.2\}$. We apply the proposed FR and SR and compute the percentage of time savings (TS), i.e.,

$$\mathrm{TS} = \begin{cases} \left(1 - \dfrac{\mathrm{time}(\mathtt{FR})}{\mathrm{time}(\mathtt{SR})}\right) 100. & \text{if} \quad \mathrm{time}(\mathtt{SR}) \geq \mathrm{time}(\mathtt{FR}), \\ -\left(1 - \dfrac{\mathrm{time}(\mathtt{SR})}{\mathrm{time}(\mathtt{FR})}\right) 100, & \text{if} \quad \mathrm{time}(\mathtt{SR}) < \mathrm{time}(\mathtt{FR}). \end{cases} \tag{34}$$

We have that $-100\% \leq \mathrm{TS} \leq 100\%$. Positive values of PTG mean that FR is faster than SR, otherwise negative values mean that SR is faster than FR. Note that a value of $\mathrm{TS} = 100\%$ would mean that FR is extremely faster (virtually instantaneous, since $\mathrm{time}(\mathtt{FR}) = 0$). The results are averaged over 10^3 runs.

Fixing N and vary M. First of all, we fix $N \in \{5000, 10000\}$ and vary the value of M (from $M = 1$ to $M = N - 1$). The results are given in Figure 2, using a log-log-domain. The TS values are virtually always positive, thereby demonstrating the advantage of the proposed scheme.

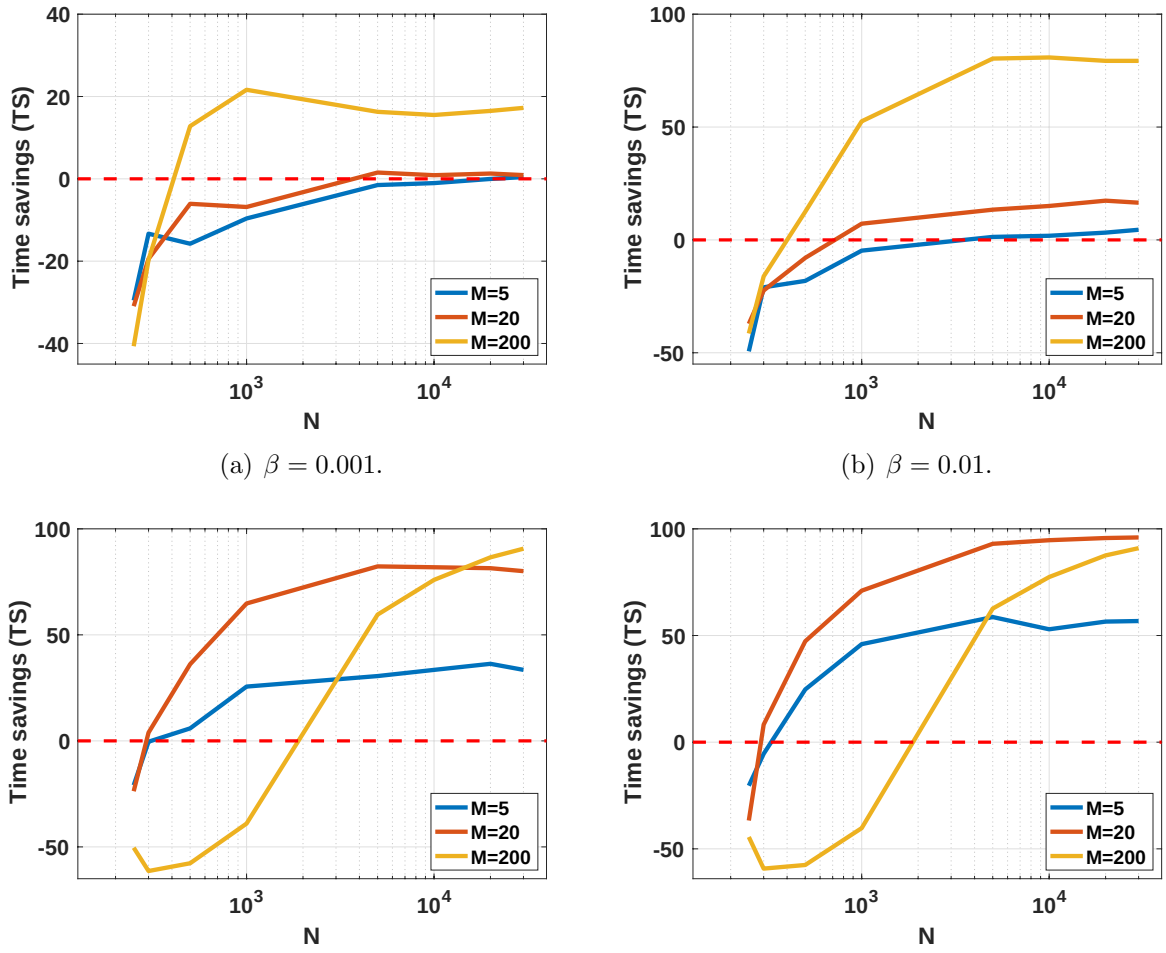
As expected, generally greater values β provide the better results. Again as expected, Figure 2(b) corresponding to a bigger value of $N = 10^4$ depicts curves closer to 100% than Figure 2(a). Moreover, the optimal value of M, corresponding to the peaks of the curves, grows as the ESS increases (i.e., β decreases), as expected. It is important to emphasize that the values around the peaks for $\beta \in \{0.01, 0.1, 0.2\}$ are very close to the maximum possible time saving of 100%.



(a) $N = 5000$. (b) $N = 10000$.

Figure 2: Percentage of time savings (TS) versus M in a log-log domain: **(a)** with $N = 5000$ and **(b)** with $N = 10000$. In both cases, we have a quite great value of number of particles $N \in \{5000, 10000\}$. In these scenarios, we have a positive TS almost for all values of M.

Fixing M and vary N. Here, we fix $M \in \{5, 20, 200\}$ (testing 3 different values) and change the number of particles N from $N = 250$ to $N = 30000$. We also consider the four different β values, i.e., $\beta \in \{0.001, 0.01, 0.1, 0.2\}$. The time savings (TS) are provided in Figure 3 (using a log-log-domain). As N grows, the TS values become positive for all values of M and β, confirming the benefits of the proposed scheme for great values of N. Moreover, as N grows, all the values of M seems feasible. As expected, smaller values of β (then we have bigger ESS values) requires bigger values of M. As N and β grows better results are provided, obtaining TS values close to 100%, as shown in Figure 3(d). As expected, as β grows (then we have smaller ESS values), the smaller values M work better for almost any N.



(a) $\beta = 0.001$. (b) $\beta = 0.01$.

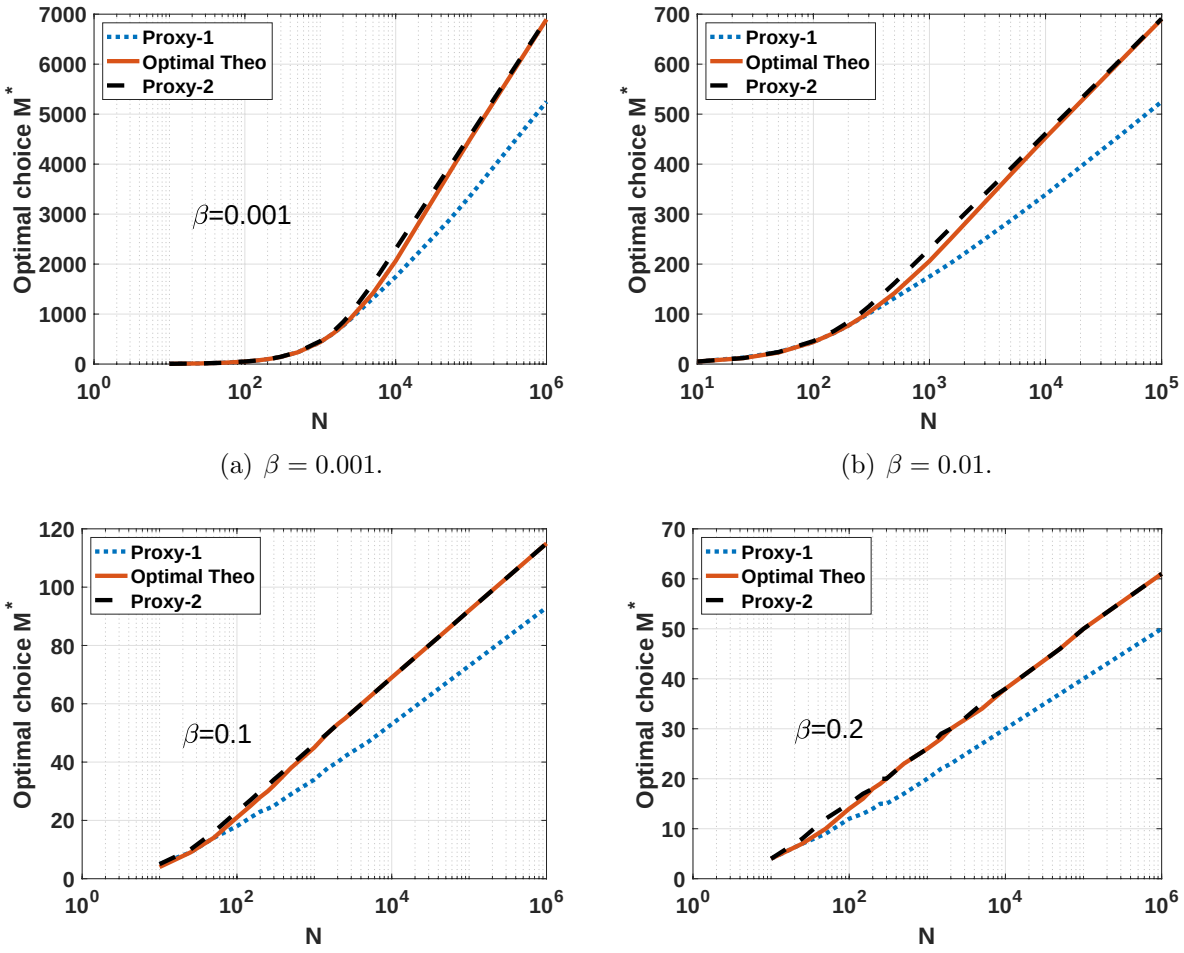
(c) $\beta = 0.1$. (d) $\beta = 0.2$.

Figure 3: Percentage of time savings (TS) versus N in a log-log domain. We test 3 different values of M, i.e., $M \in \{5, 20, 200\}$ (represented by the 3 different curves in each figure). We also consider $\beta \in \{0.001, 0.01, 0.1, 0.2\}$, each value corresponding to a figure.

5.2 Second Experiment: behaviors of $\widehat{M}_1$ and $\widehat{M}_2$

In this section, we consider different construction of the weights,

$$\begin{aligned} \textbf{first type:} \quad & w_n = \exp(-\beta n), \\ \textbf{second type:} \quad & w_n = 1 - \frac{1}{N} n, \\ \textbf{third type:} \quad & w_n = 1/n^{\alpha}, \end{aligned} \tag{35}$$

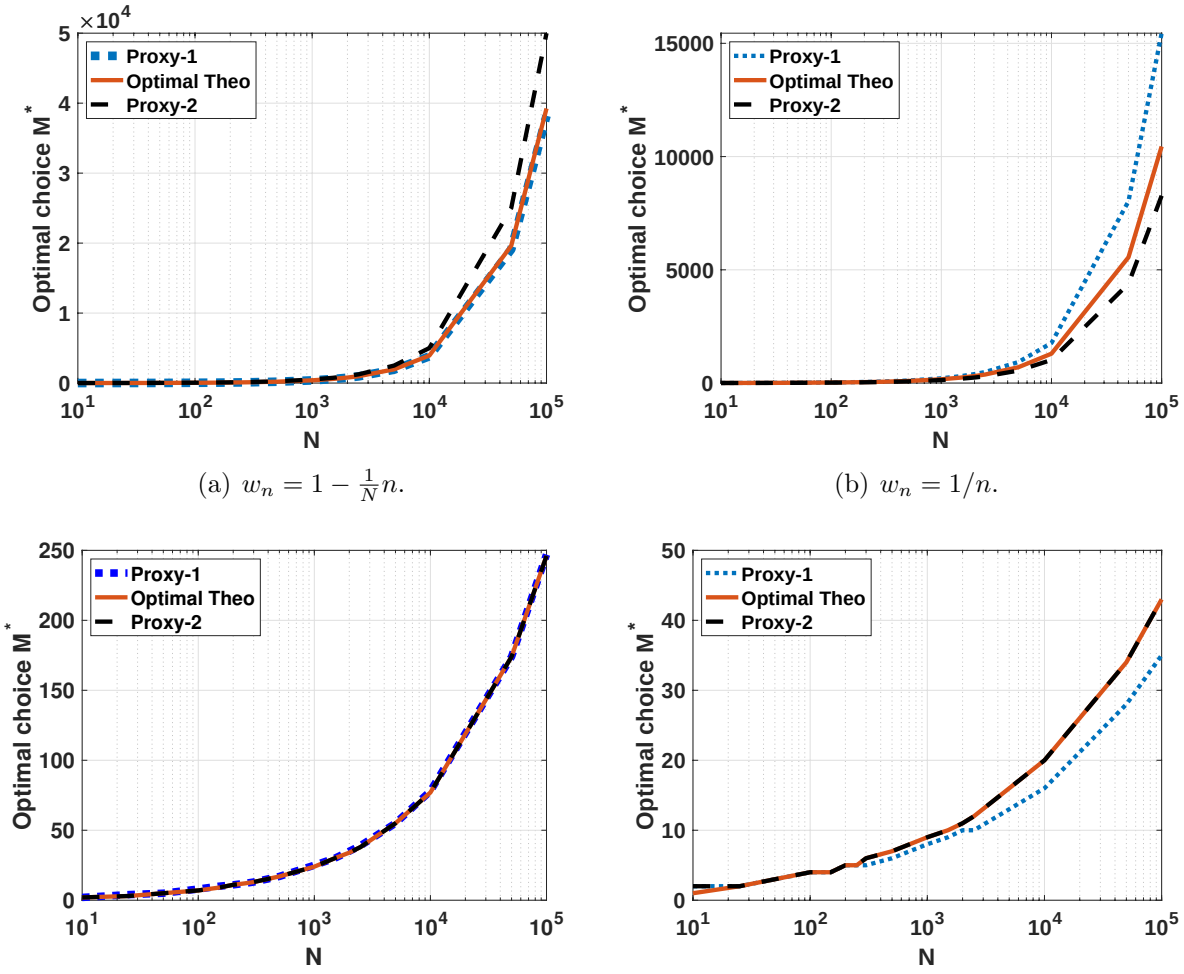


(a) $\beta = 0.001$. (b) $\beta = 0.01$.

(c) $\beta = 0.1$. (d) $\beta = 0.2$.

Figure 4: The curves (log-domain versus N) of the optimal value M^* in continuous red line, the proxy $\widehat{M}_1$ in dotted blue line, and the proxy $\widehat{M}_2$ in dashed red line, for the exponential weight construction with $\beta \in \{0.001, 0.01, 0.1, 0.2\}$. We can note that $\widehat{M}_2$ is an excellent approximation of M^* in all cases.

for $n = 1, ..., N$. Then, we set $\bar{w}_n = \frac{w_n}{\sum_{i=1}^{N} w_i}$ for all cases. We test different β and α, more precisely, $\beta \in \{0.001, 0.01, 0.1, 0.2\}$ and $\alpha \in \{1, 2, 3\}$. We remark again that even the weights are already sorted in decreasing order by construction, we perform the sorting procedure in the code (i.e., we apply the command '`sort`') in order to obtain a fair comparison with the standard resampling (SR) scheme.



(a) $w_n = 1 - \frac{1}{N}n$.

(b) $w_n = 1/n$.

(c) $w_n = 1/n^2$.

(d) $w_n = 1/n^3$.

Figure 5: The curves (log-domain versus N) of the optimal value M^* in continuous red line, the proxy $\widehat{M}_1$ in dotted blue line, and the proxy $\widehat{M}_2$ in dashed red line, for the weight constructions $w_n = 1 - \frac{1}{N}n$ and $w_n = 1/n^\alpha$ with $\alpha \in \{1, 2, 3\}$. The proxy $\widehat{M}_2$ shows some difficulties in the scenarios (a) and (b) for great values of N, but overall it provides a good approximation of M^*, in all cases. For instance, $\widehat{M}_2$ provides virtually a perfect match in cases (c) and (d).

The results are provided in Figures 4 and 5. They show the curves versus N of the optimal value M^* (continuous red line), the proxy $\widehat{M}_1$ in Eq. (31) (dotted blue line), and the proxy $\widehat{M}_2$ in Eq. (32) (dashed red line), for the different the weight constructions. The proxy $\widehat{M}_2$ is an excellent

approximation of M^*, for all values of N, in all the scenarios in Figures 4(a)-4(b)-4(c)-4(d) and Figures 5(c)-5(d). Whereas the proxy $\widehat{M}_2$ shows some difficulties in the scenarios Figures 5(a)-5(b) for great values of N, However, overall it provides a good approximation of M^*, for all values of N. The proxy $\widehat{M}_1$ works well only for small values of N and in Figure 5(c) (corresponding to the construction $w_n = 1/n$). For this reason, we suggest the use of $\widehat{M}_2$.

5.3 Particle filtering for a stochastic volatility model

In this example, we test the SR and the proposed FR within a particle filter (a.k.a., sequential Monte Carlo). We consider a stochastic volatility model where the hidden state θ_t follows an auto-regressive process and represents the log-volatility [18] of a time series at time $t \in \mathbb{N}$, i.e.,

$$\begin{cases} \theta_t = \alpha\theta_{t-1} + u_t, \\ y_t = \exp\left(\frac{\theta_t}{2}\right) v_t, \end{cases} \qquad t = 1, \dots, T. \tag{36}$$

where $\alpha = 0.99$, and u_t and v_t are Gaussian independent noises, with zero-mean with variances $\sigma_u^2 = 1$ and $\sigma_v^2 = 0.5$, respectively. Note that u_t is an additive noise, whereas v_t is a multiplicative noise. We implement a standard particle filter (PF) [6, 9, 15] using as propagation equation of the particles exactly the auto-regressive process of the true model, i.e., the next generation of particles $\theta_{i,t}$'s is generated as $\theta_{i,t} \sim \mathcal{N}(\theta|\alpha\theta_{i,t-1}, \sigma_u^2)$, where $i = 1, \dots, N$ is the particle index. The application of the resampling is adaptively decided according to ESS (see below). We set $T = 100$ and simulate a run of the series θ_t and y_t according to the model in Eq. (36), starting from $\theta_0 = 0$. Given, the generated data y_t, we run a particle filter with different values of numbers of particles N.

The resampling is performed adaptively, only a certain iterations. More specifically, the resampling is applied at the iteration t such that

$$\mathrm{ESS}(t) = \frac{1}{\sum_{n=1}^{N} \bar{w}_{n,t}^2} \leq \epsilon N \tag{37}$$

where $\epsilon \in [0, 1]$ is a constant threshold value (with $\epsilon = 0$, no resampling step is performed; with $\epsilon = 1$, the resampling is applied at each iteration). We set $\epsilon = 0.75$. Note that we have used the classical definition of ESS [30, 11, 28]. In one specific run, we can define the rate of resampling steps as res-rate $= \frac{\#\text{ Resampling}}{T}$.

In the proposed FR scheme we employ the proxy $\widehat{M}_2$ as choice for M. Thus, we run 100 independent particle filters using SR and FR, and calculate the averaged, required computational time. More precisely, we also compute the normalized rate of time savings (TS) in Eq. 34. We recall that $-100\% \leq \mathrm{TS} \leq 100\%$ Positive values of TS mean that FR is faster than SR,

otherwise negative values mean that SR is faster than FR. In this experiment, we also compute the percentage of time increase (TI) using SR instead of the proposed FR, i.e.,

$$\mathrm{TI} = \frac{\mathrm{time}(\texttt{SR})}{\mathrm{time}(\texttt{FR})} 100. \tag{38}$$

The results are given in Table 2. The average res-rate was ≈ 0.47, i.e., the resampling step was performed in almost half of the iterations. The reported time values naturally depend on the computational resources (here, a laptop) used to perform the experiment. However, the relevant information are the rates of time savings. We can observe that with FR, we save about 60% of the computational time when using a large number of particles N. For instance, with a million of particles ($N = 10^6$), we spend around 10 hours instead an entire day (more than 23 hours).

Table 2: Time (seconds; in the first row) required for executing the particle filters with different number of particles N. The actual time values clearly depend on the specific machine (laptop) employed for running the experiment. The relevant values are the percentages of time increase using SR in Eq. (38), and time savings (TS) obtained using FR instead SR, defined as in Eq. (34).

N	10^3	$5 \cdot 10^3$	10^4	$5 \cdot 10^4$	10^5	$2 \cdot 10^5$	10^6
time(SR)	0.08 s.	1.47 s.	4.70 s.	87.07 s. 1:27 min.	360.9 s. 06:01 min.	1573.8 s. 26:13 min	83801.2 s. 23:16:41 h.
time(FR)	0.06 s.	0.94 s.	2.44 s.	39.86 s.	149.8 s. 02:29 min.	600.7 s. 10:01 min	34133.2 s. 09:28:53 h.
$\frac{\mathrm{time}(\texttt{SR})}{\mathrm{time}(\texttt{FR})}\%$	133.3%	156.4%	192.6%	218.5%	241.0%	262.0%	245.5%
Time savings (TS)	25%	36.05%	48.09%	54.22%	58.49%	61.83%	59.26%

6 Conclusions

In this work, we propose a novel resampling scheme that outperforms existing techniques in terms of speed when a large number of particles is used. The new fast resampling (FR) method is based on the division of the particles into two groups: a first set with M most weighted particles, and a second group with the rest of $N - M$ particles. The proposed algorithm is highly efficient when the first group contains only a small number of particles M but capturing a significant probability mass. This occurs when the effective sample size (ESS) is small, which is precisely the situation in which resampling becomes necessary in particle filtering/SMC methods. The novel scheme is compatible with all the resampling approaches, including multinomial, stratified, systematic,

and residual, to name a few. We have theoretically analyzed the proposed resampling scheme by verifying (a) the validity of the proper-weighting condition, (b) characterizing the feasible working zones, and (c) determining the optimal choice of M^*. We have also found a good approximation of M^*, denoted as $\widehat{M}_2$. Thus, the resulting method is fully automatic, as the choice of M is determined directly by the algorithm.
The behavior of the proposed FR scheme have been widely analyzed in different numerical scenarios. The numerical results confirm that FR is convenient for large values of N and small ESS values. Remarkable results have been obtained within a particle filter in which resampling is applied in approximately half of the iterations (instead 10 hours instead of approximately 1 day of computation). Clearly, the more resampling steps are performed, the greater the advantages provided by the proposed FR procedure. The corresponding (non-optimized) Matlab code is also made available for reproducibility at `http://www.lucamartino.altervista.org/FAST_RESAMPLING_public_code.zip`.

References

[1] S.A. Alam and O. Gustafsson. Improved particle filter resampling architectures. *J. Sign. Process. Syst.*, 92:555–568, 2020.

[2] M. Bolić, P. M. Djurić, and S. Hong. Resampling algorithms for particle filters: A computational complexity perspective. *EURASIP Journal on Advances in Signal Processing*, 2004(15):2267–2277, November 2004.

[3] M. Bolić, P. M. Djurić, and S. Hong. Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions Signal Processing*, 53(7):2442–2450, July 2005.

[4] N. Chopin, S. S. Singh, T. Soto, and M. Vihola. On resampling schemes for particle filters with weakly informative observations. *arXiv:2203.10037*, 2022.

[5] P. Del Moral, A. Doucet, A. Jasra, et al. On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, 18(1):252–278, 2012.

[6] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez. Particle filtering. *IEEE Signal Processing Magazine*, 20(5):19–38, September 2003.

[7] R. Douc, O. Cappé, and E. Moulines. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4^{th} International Symposium on Image and Signal Processing and Analysis*, pages 64–69, September 2005.

[8] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.

[9] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York (USA), 2001.

[10] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo Sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 2000.

[11] V. Elvira, L. Martino, and C. P. Robert. Rethinking the effective sample size. *International Statistical Review*, 90(3):525–550, 2022.

[12] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99:10143–10162, 1994.

[13] A. Gandy and F. D. H. Lau. The chopthin algorithm for resampling. *IEEE Transaction on Signal Processing*, 64(16):4273–4281, 2016.

[14] T. Ghirmai, M. F. Bugallo, J. Míguez, and P. M. Djurić. A sequential Monte Carlo method for adaptive blind timing estimation and data detection. *IEEE Transactions Signal Processing*, 53(8):2855–2865, August 2005.

[15] N. Gordon, D. Salmond, and A. F. M. Smith. Novel approach to nonlinear and non-Gaussian Bayesian state estimation. *IEE Proceedings-F Radar and Signal Processing*, 140:107–113, 1993.

[16] J. D. Hol, T. B. Schon, and F. Gustafsson. On resampling algorithms for particle filters. In *IEEE Nonlinear Statistical Signal Processing Workshop*, pages 79–82, 2006.

[17] C. Hue, J.-P. LeCadré, and P. Pérez. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Transactions on Signal Processing*, 50(2):309–325, 2002.

[18] E. Jacquier, N. G. Polson, and P. E. Rossi. Bayesian analysis of stochastic volatility models. *Journal of Business and Economic Statistics*, 12(4):371–389, October 1994.

[19] C. Kuptametee and N. Aunsri. A review of resampling techniques in particle filtering framework. *Measurement*, 193:110836, 2022.

[20] T. Li, M. Bolic, and P. M. Djuric. Resampling methods for particle filtering: Classification, implementation, and strategies. *IEEE Signal Processing Magazine*, 32(3):70–86, 2015.

[21] D. P. Liu, Q. T. Zhang, and Q. Chen. Structures and performance of noncoherent receivers for unitary space-time modulation on correlated fast-fading channels. *IEEE Transactions Vwhicular Technology*, 53(4):1116–1125, July 2004.

[22] J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, 2004.

[23] J. S. Liu and R. Chen. Sequential Monte Carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93(443):1032–1044, September 1998.

[24] J. S. Liu, R. Chen, and T. Logvinenko. A theoretical framework for sequential importance sampling with resampling. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, chapter 11, pages 225–246. Springer, 2001.

[25] F. Llorente and L. Martino. Optimality in importance sampling: a gentle survey. *arXiv:2502.07396*, 2025.

[26] F. Llorente, L. Martino, D. Delgado, and J. López-Santiago. Marginal likelihood computation for model selection and hypothesis testing: An extensive review. *SIAM Review*, 65(1):3–58, 2023.

[27] L Martino and V. Elvira. Compressed Monte Carlo with application in particle filtering. *Information Sciences*, 553:331–352, 2021.

[28] L. Martino and V. Elvira. Effective sample size approximations as entropy measures. *Computational Statistics*, pages 1–32, 2025.

[29] L. Martino, V. Elvira, and G. Camps-Valls. Group importance sampling for particle filtering and MCMC. *Digital Signal Processing*, 82:133–151, 2018.

[30] L. Martino, V. Elvira, and F. Louzada. Effective sample size for importance sampling based on discrepancy measures. *Signal Processing*, 131:386–401, 2017.

[31] L. Martino, D. Luengo, and J. Mguez. *Independent Random Sampling Methods*. Springer Publishing Company, Incorporated, 1st edition, 2018.

[32] J. Míguez. Analysis of parallelizable resampling algorithms for particle filtering. *Signal Processing*, 87(12):3155–3174, 2007.

[33] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. Society for Industrial Mathematics, 1992.

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

[35] D. B. Rubin. A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: the sir algorithm. *Journal of the American Statistical Association*, 82:543–546, 1987.

[36] N. Zhou, L. Lau, R. Bai, and T. Moore. A genetic optimization resampling based particle filtering algorithm for indoor target tracking. *Remote Sensing*, 13(1), 2021.

A Feasible zones with $\epsilon \approx 0$

Assuming that the wasted time ϵ (in additional operations) in Eq. (26) negligible, i.e., $\epsilon \approx 0$, and recalling the $\psi(\cdot)$ is monotonic increasing (and hence invertible), we have

$$T_u \cdot \psi\left(2 + \bar{s}_M M + (1 - \bar{s}_M)(N - M)\right) < T_u \cdot \psi(N), \tag{39}$$

$$\psi\left(2 + \bar{s}_M M + (1 - \bar{s}_M)(N - M)\right) < \psi(N), \tag{40}$$

$$\underbrace{2 + \bar{s}_M M + (1 - \bar{s}_M)(N - M)}_{\phi(\bar{s}_M, M)} < N. \tag{41}$$

Then, after some algebra, we obtain

$$\begin{aligned}
&2 + \bar{s}_M M + (1 - \bar{s}_M)(N - M) < N,\\
&2 + N - \bar{s}_M N - M + 2\bar{s}_M M < N,\\
&2 - \bar{s}_M N - M + 2\bar{s}_M M < 0,\\
&\bar{s}_M(2M - N) < M - 2
\end{aligned}$$

$$\Rightarrow \text{if } M > N/2 \text{ then } \bar{s}_M < \bar{u} = \frac{M-2}{2M-N}, \tag{42}$$

$$\Rightarrow \text{if } M < N/2 \text{ then } \bar{s}_M > \bar{u} = \frac{M-2}{2M-N}. \tag{43}$$

Recalling that $0 < \bar{s}_M < 1$ and $\bar{s}_M$ depends on M (with $M < N$), the inequalities above define the *feasible zones* of pairs $\{\bar{s}_M, M\}$ with the assumption $\epsilon \approx 0$, such that we have an improvement in terms of the computational time using the proposed scheme. Interesting considerations are given below.

Property 1. Fixing N, $\bar{u} = \frac{M-2}{2M-N}$ is always a decreasing function of M, with a vertical asymptote at $M = N/2$. For $M < N/2$, it starts with the value $\bar{u} = \frac{1}{N-2}$ at $M = 1$, and tends to $-\infty$ at $M \to \frac{N}{2}^-$. Whereas, for $M > N/2$, the function $\bar{u} = \frac{M-2}{2M-N}$ decreases from $+\infty$ at $M \to \frac{N}{2}^+$ until $1/2$ for $M \to \infty$ (i.e., there is an horizontal asymptote for $M \to \infty$). More precisely,

$$\lim_{M \to \frac{N}{2}^-} \bar{u} = -\infty, \qquad \lim_{M \to \frac{N}{2}^+} \bar{u} = \infty, \quad \text{for all } N. \tag{44}$$

See Figures 6 and 8.

Property 2. Note that, for the decreasing property of $\bar{u}$, we have: for $M < N/2$, the value of $\bar{u}$ at $M = 1$ is the maximum value, $\frac{1}{N-2} > \frac{M-2}{2M-N}$. For $M > N/2$, the value of $\bar{u}$ at $M = N - 1$, i.e.,

$\frac{N-3}{N-2}$, is the minimum value (considering $M = N-1$ the maximum value of M). Hence, we have $\bar{u} = \frac{M-2}{2M-N} > \frac{N-3}{N-2}$. As summary, we have

$$\bar{u} = \frac{M-2}{2M-N} < \frac{1}{N-2}, \quad \text{for } M < N/2, \tag{45}$$

$$\bar{u} = \frac{M-2}{2M-N} > \frac{N-3}{N-2}, \quad \text{for } M > N/2. \tag{46}$$

Property 3. The first threshold value $\bar{u} = \frac{1}{N-2}$ (at $M = 1$) vanishes to zero, as $N \to \infty$. The last value $\bar{u} = \frac{N-3}{N-2}$ (at $M = N-1$) tends to 1, as $N \to \infty$. Since $\bar{s}_M \in (0, 1]$ and considering Eqs. (42)-(43) and Property 2 with Eq. (45)-(46) above, this means that any pair $\{\bar{s}_M, M\}$ tend to be feasible as N grows.

Property 4. Let us consider $N > 4$. At $M = \frac{N}{2}$, the threshold value $\bar{u} = \frac{M-2}{2M-N}$ diverges to $\pm\infty$ as shown in Eq.(44), for all N. Thus we always have $\bar{s}_M > \bar{u} \to -\infty$ at $M = \frac{N}{2}^-$ and $\bar{s}_M < \bar{u} \to +\infty$ at $M = \frac{N}{2}^+$. Then, any pair $\{\bar{s}_M, M = \frac{N}{2}\}$ is always feasible.

With $N = 4$ and $M = \frac{N}{2} = 2$, we have indeterminate form $\bar{u} = \frac{0}{0}$. However, replacing $N = 4$ and $M = \frac{N}{2} = 2$ into $\phi(\bar{s}_M, M)$, we obtain $2 + 2\bar{s}_M + 2(1 - \bar{s}_M) = 4$ that is not smaller than 4, then the pairs $\{\bar{s}_M, M = \frac{N}{2} = 2\}$, for a generic $\bar{s}_M$, are all unfeasible. It can be shown that this statement is also valid for $N < 4$. Finally note that, in this section, we have not considered that the samples are ordered (as in the proposed resampling scheme).

Remark 6. Clearly all these considerations have been done assuming the ideal conditions $\epsilon \approx 0$ in Eq. (26). In a non-ideal and practical scenario, the feasible zones are generally much more restricted. See, for instance, Figures 3 in the numerical experiments. However even in real scenarios, as N grows, all the pairs $\{\bar{s}_M, M\}$ tends to be feasible.

B Considerations regarding the ESS

By treating M and $\bar{s} = \bar{s}_M$ as independent quantities (although this is not the case in practice), we can derive some additional observations. More specifically, in this section, we do two important assumptions:

- We assume that M and $\bar{s} = \bar{s}_M$ as independent quantities. We remove the subindex M in $\bar{s}$ to highlight this premise.
- Moreover, in the observations provided in this section, *we are not considering that the samples are ordered* (in decreasing order) according to normalized weights.

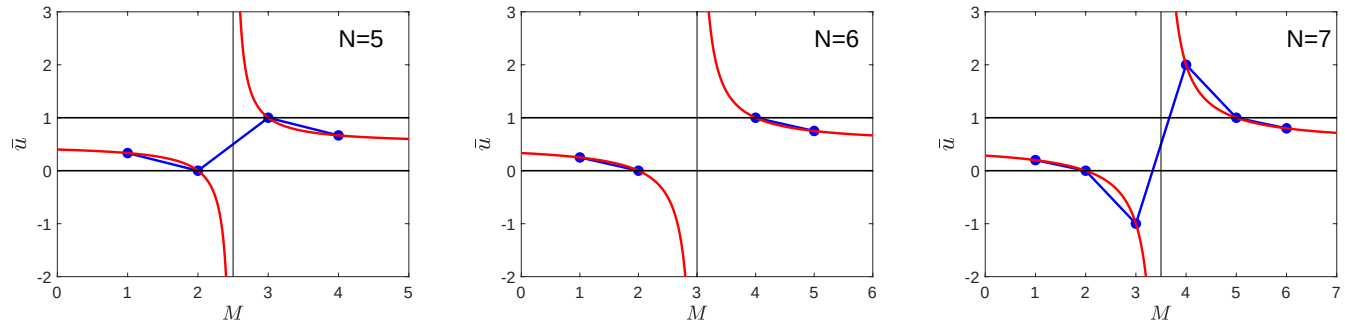


Figure 6: The threshold $\bar{u} = \frac{M-2}{2M-N}$ versus $M = 1, 2, ..., N-1$, for different values of $N \in \{5, 6, 7\}$. The function $\bar{u} = \frac{M-2}{2M-N}$, considering M as a continuous variable, is shown with a red solid line. The blue circles depict the $\bar{u}$-values at the discrete values $M = 1, 2, ..., N-1$ (note that for $N = 6$, at $M = 3$ we have $\bar{u} \to \infty$). Recall that the feasible zones are defined by $\bar{s}_M < \bar{u}$ for $M < N/2$, and $\bar{s}_M > \bar{u}$ for $M > N/2$. Recall also that $\bar{s}_M \in (0, 1]$, hence negative values of $\bar{u}$ for $2M < N$, and values greater than 1, $\bar{u} \geq 1$ for $2M > N$, mean that any value of $\bar{s}_M$ is suitable (see Fig. 7).

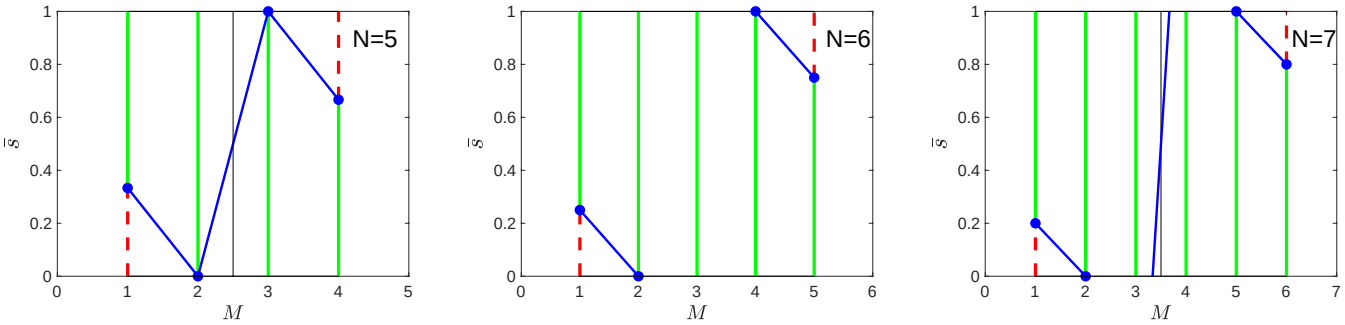


Figure 7: The feasible zones of the pairs $\{\bar{s}_M, M\}$ where $M = 1, 2, ..., N-1$ with $N \in \{5, 6, 7\}$. Recall $\bar{s}_M \in (0, 1]$. The feasible zones are depicted with green segments. The unfeasible zones are depicted with dashed red segments. The threshold values $\bar{u} = \frac{M-2}{2M-N}$, contained in $[0, 1]$, are shown with blue circles, determining the border of the feasible zones. As an example, with $N = 5$, only some values of $\bar{s}_M$ at $M = 1$ and at $M = 4$ are not admissible. As $N \to \infty$, the unfeasible zones tend to disappear (see also Fig. 8).

However, despite this false assumptions, we obtain general results outlined in the remarks below. Fixing $\bar{s}$ or M, we can write the two straight lines,

$$\phi(\bar{s}|M) = \underbrace{(2M - N)}_{\texttt{slope}}\,\bar{s} + \underbrace{2 + N - M}_{\texttt{intercept}}, \tag{47}$$

$$\phi(M|\bar{s}) = \underbrace{(2\bar{s} - 1)}_{\texttt{slope}}\,M + \underbrace{2 + N - \bar{s}N}_{\texttt{intercept}}. \tag{48}$$

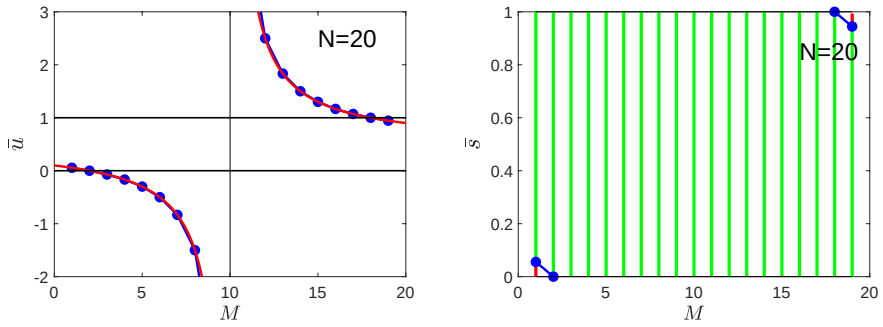


Figure 8: The thresholds $\bar{u}$ and the feasible zones of the pairs $\{\bar{s}_M, M\}$ with $N = 20$. We can observe that virtually there are not unfeasible zones, since *almost all* the values of $\bar{u}$ are $\bar{u} < 0$ for $M < 10$ and $\bar{u} > 1$ for $M > 10$. As $N \to \infty$, the unfeasible zones tend to disappear.

Recall that we desire to minimize ϕ, which represents the required computational time.

With respect to M, the function $\phi(M|\bar{s})$ in Eq. (48) is a straight line with slope coefficient $2\bar{s} - 1$. Hence, If $\bar{s} < 0.5$, the straight line is decreasing, and we would like to have the greatest value of M (to decrease ϕ), that is $M = N - 1$ (thus we would have two groups, with $N - 1$ particles and the other with only sample). If $\bar{s} > 0.5$, the straight line is increasing, we would like to have the smallest value of M to decrease ϕ, that is $M = 1$.

Remark 7. Given the previous observations, we can assert that the best performance (smallest computational time) are given when the maximum probability mass between $\bar{s}$ and $1 - \bar{s}$ are concentrated in just one particle. Hence, this means that the best performance is given when we have a small effective sample size (ESS) [30, 11, 28], possibly the smallest one.

If we set $\bar{s} = 0.5$, then $\phi(M|\bar{s}) = 2 + 0.5N$, does not depend with M (it is constant). Moreover, note that

$$2 + 0.5N < N \implies \quad N > 4.$$

This final observation yields the remark below.

Property 5. For $N > 4$, the pair $\{\bar{s}_M = 0.5, M\}$ is always a feasible pair for any possible M (recall $M < N$). Moreover, the computational time ($\phi(M|\bar{s}) = 2 + 0.5N$) is invariant with respect to the value of M.

With respect to $\bar{s}$, the function $\phi(\bar{s}|M)$ in Eq. (47) is a straight line with slope coefficient $2M - N$. Therefore, if we fix an M such that $M < N/2$, then $\phi(\bar{s}|M)$ is decreasing, and we would like to

have the greatest possible value of $\bar{s}$ (to decrease ϕ), that is $\bar{s} = 1$. Otherwise, If we have a value M such that $M > N/2$, $\phi(\bar{s}|M)$ is increasing, we would like to have the smallest possible value of $\bar{s}$ (to decrease ϕ), that is $\bar{s} = 0$.

Remark 8. Given the previous observations, we can assert that the best performance (smallest computational time) when the group with the smallest number of particles has associated the total mass $\bar{s} = 1$, whereas the group with the greatest number of particles has associated $\bar{s} = 0$. Therefore, this again indicates that favorable scenarios generally occur when the ESS values are small [11, 30].

Note that when $M = N/2$, then again $\phi(\bar{s}|M) = 2 + 0.5N$ does not depend on $\bar{s}$. Since $2 + 0.5N < N$ when $N > 4$, we arrive again to Property 4: for $N > 4$, the pair $\left\{\bar{s}_M, M = \frac{N}{2}\right\}$ is always a feasible pair for any possible $\bar{s}_M$. Now we can also state that the computational time ($\phi(\bar{s}|M) = 2 + 0.5N$) remains unchanged, regardless of the value of $\bar{s}_M$.