Statistica Sinica **35** (2025), 67-89
doi:https://doi.org/10.5705/ss.202022.0120

STATE SPACE EMULATION AND ANNEALED SEQUENTIAL MONTE CARLO FOR HIGH DIMENSIONAL OPTIMIZATION

Chencheng Cai* and Rong Chen

Washington State University and Rutgers University

Abstract: Many high-dimensional optimization problems can be reformulated as finding the optimal path under an equivalent state-space model setting. Here, we present a general emulation strategy for developing a state-space model with a likelihood function (or posterior distribution) that shares the same general landscape as that of the original objective function. Then, the solution of the optimization problem is the same as the optimal state path that maximizes the likelihood function. To find such an optimal path, we adapt a simulated annealing approach by inserting a temperature control into the emulated dynamic system, and propose a novel annealed sequential Monte Carlo (SMC) method that effectively generates Monte Carlo sample paths based on samples obtained previously on a higher temperature scale. Compared with the vanilla simulated annealing implementation, the annealed SMC is an iterative algorithm for state-space model optimization that generates state paths directly from the equilibrium distributions using a decreasing sequence of temperatures and sequential importance sampling, which does not require burn-in or mixing iterations to ensure a quasi-equilibrium condition. Lastly, we demonstrate the proposed method by presenting several emulation examples and the corresponding simulation results.

Key words and phrases: Emulation, optimization, sequential Monte Carlo, simulated annealing, state space model.

1. Introduction

High-dimensional global optimization algorithms have been widely investigated since the advent of high-dimensional complex data. For example, the gradient descent algorithm and its variations (Bertsekas, 1997) require that the objective function be convex or uni-modal to ensure that the found local optimal is global. Recent research in machine learning involves many nonconvex optimization problems (Anandkumar et al., 2014; Arora et al., 2012; Netrapalli et al., 2014; Agarwal et al., 2014). However, many of these problems remain NP-hard, and theory is only available for their convex relaxations (Jain and Kar, 2017). Deterministic optimization algorithms (Hooke and Jeeves, 1961; Nelder and Mead, 1965; Land and Doig, 1960) may result in an exhaustive search,

*Corresponding author. E-mail: chencheng.cai@wsu.edu

which is computationally expensive in a high-dimensional space. Stochastic optimization algorithms use Monte Carlo simulations to explore the parameter space in a stochastic and often more efficient way (Kiefer and Wolfowitz, 1952; Kirkpatrick, Gelatt and Vecchi, 1983; Mei, Montanari and Nguyen, 2018).

In this article, we propose an emulation approach that reformulates a high-dimensional optimization problem as one of finding the most likely state path in a state-space model. State-space models describe the behavior of a usually high-dimensional random variable as a form of dynamic evolution, with wide applications in mathematics, physics, and many other fields. Many high-dimensional optimization problems can be transformed to finding the optimal state path under an equivalent state-space model with a likelihood function that shares the same general landscape as that of the objective function of the original optimization problem. Specifically, for a high-dimensional optimization problem with the objective function $f(x)$, we construct an emulated state-space model with a likelihood function that is proportional to a Boltzmann-like distribution $\exp\{-\kappa f(x)\}$, where $\kappa > 0$ is the inverted temperature.

Several existing heuristic approaches use the emulation idea. Cai, Tsay and Chen (2009) transform a regression variable selection problem with many predictors into an optimization problem over the high-dimensional binary space $\{0,1\}^p$. The latter problem can be further converted to a most likely path problem in a state-space model with binary-valued states indicating the variable selection, even though the predictors have no chronological order in nature. Kolm and Ritter (2015) reformulate a portfolio optimization problem as a state-space model by mapping the utility function to the log-likelihood function. The utility function is then optimized by finding the most likely path in the corresponding state-space model by applying the Viterbi algorithm (Viterbi, 1967) over Monte Carlo samples. Similarly, Irie and West (2016) relate the multi-period portfolio optimization problem to a log-likelihood of a mixture of linear Gaussian dynamic systems, and propose an algorithm based on the Kalman filter (Kalman, 1960) and EM algorithm (Dempster, Laird and Rubin, 1977) to find the most likely path. Iglesias, Law and Stuart (2013) and Zhang, Song and Liang (2021) reformulate inversion problems as state-space models by segmenting the observations into a sequence, and then optimizing the hidden path using a Kalman filter and an ensemble Kalman filter.

The aforementioned studies map high-dimensional optimizations to problems under state-space model settings. However, finding the most likely path analytically and numerically remains challenging. For example, the approach in Cai, Tsay and Chen (2009) is difficult to generalize to continuous spaces. In addition, the Viterbi algorithm used in Kolm and Ritter (2015) requires the dynamic system to be Markovian and nonsingular, and needs a large sample size, in general, to achieve high accuracy. The combination of the Kalman filter and the EM algorithm proposed in Irie and West (2016) works only when the underlying distribution can be well represented by the mixture of Gaussian distributions.

In this paper, we propose a new sequential Monte Carlo (SMC) simulated annealing approach, called the "annealed SMC", to find the most likely path in a state-space model. The SMC algorithm is one of a class of Monte Carlo methods that draws samples from state-space model systems in a sequential fashion. With the sequential importance sampling and resampling (SISR) scheme, an SMC is extremely powerful in terms of sampling from complex dynamic systems, especially for state-space models (Gordon, Salmond and Smith, 1993; Kitagawa, 1996; Kong, Liu and Wong, 1994; Liu and Chen, 1995, 1998; Pitt and Shephard, 1999; Chen, Wang and Liu, 2000; Doucet, de Freitas and Gordon, 2001). Recall that the likelihood function of the emulated state-space model is designed to be proportional to $\exp\{-\kappa f(x)\}$, where κ is the inverted temperature. To mimic the (physical) annealing procedure in a non-interactive, non-quantum thermodynamic system (Kirkpatrick, Gelatt and Vecchi, 1983), we choose a sequence of decreasing temperatures $\kappa_0 < \kappa_1 < \cdots < \kappa_K$, which corresponds to a sequence of emulated state-space models.

We start by drawing sample paths from the base emulated state-space model at a high base temperature κ_0. Although samples from a low temperature (large κ) system are close to the optimal sample path, because the distribution is sharp at a low temperature, drawing from such a distribution directly is usually difficult. Using the annealed SMC, we can obtain samples of a low temperature system based on samples obtained at a higher temperature. Eventually, all the SMC sample paths converge to the most likely one. The sequence of temperatures $\kappa_0 < \kappa_1 < \cdots < \kappa_K$ provides a slow-changing path from the base emulated state-space model at κ_0, which is easy to sample from, but not very useful for optimization, to the target emulated state-space model at κ_K, which is difficult to sample from but provides solutions to the optimization problem.

This study makes two main contributions to the literature. First, we reformulate the problem as an emulated space-space model, and then we propose an annealed SMC algorithm to find the solution. Two examples are provided, in which the emulated state-space models are natural, simple, and illustrative. Two additional examples are provided in the Supplementary Material to demonstrate the flexibility of the proposed method in solving existing optimization problems, with some new applications.

The rest of the paper is organized as follows. Section 2 briefly reviews state-space models and introduces the principles of state-space emulation. Two illustrative emulation examples are provided in Section 2.3. Section 3 introduces the framework of the annealed SMC, designed to find the most likely path. Simulation results corresponding to the two examples in Section 2.3 are presented in Section 4. Section 5 concludes the paper.

2. State-Space Model and State-Space Emulation

2.1. State-space model

State-space models describe the mechanism of sequential observations $\boldsymbol{y}_T = (y_1, \ldots, y_T)$ using a sequence of latent variables $\boldsymbol{x}_T = (x_1, \ldots, x_T)$. The latent variables $\boldsymbol{x}_T$ are assumed to follow a discrete-time stochastic process governed by the state equations

$$p(x_t \mid \boldsymbol{x}_{t-1}) = p_t(x_t \mid \boldsymbol{x}_{t-1}), \tag{2.1}$$

for $t = 2, \ldots, T$, and x_1 follows its marginal distribution $p_1(x_1)$. When the distribution of x_t conditioned on $\boldsymbol{x}_{t-1}$ does not depend on $\boldsymbol{x}_{t-2}$, such that $p(x_t \mid \boldsymbol{x}_{t-1}) = p(x_t \mid x_{t-1})$, the system is Markovian. The observations $\boldsymbol{y}_T$ are generated independently, conditioned on the latent variables, using the observational equations

$$p(y_t \mid x_t) = g_t(y_t \mid x_t), \tag{2.2}$$

for $t = 1, \ldots, T$. In inference problems, the formulae of the state equations $p_t(\cdot)$ and the observation equations $g_t(\cdot)$ are usually known, except for a set of unknown parameters of interest θ. Here, we assume $p_t(\cdot)$ and $g_t(\cdot)$ are completely known, and we infer the latent states $\boldsymbol{x}_T$. Estimating $\boldsymbol{x}_T$ from the observations $\boldsymbol{y}_T$ under the likelihood principle is known as the most likely path (MLP) problem in hidden Markov models.

The state equations provide the prior information on $\boldsymbol{x}_T$:

$$\pi(\boldsymbol{x}_T) \propto p_1(x_1) \prod_{t=2}^{T} p_t(x_t \mid \boldsymbol{x}_{t-1}), \tag{2.3}$$

and the observation equations serve as the likelihood functions:

$$p(\boldsymbol{y}_T \mid \boldsymbol{x}_T) = \prod_{t=1}^{T} g_t(y_t \mid x_t). \tag{2.4}$$

A maximum-a-posterior (MAP) estimator can be obtained by maximizing the posterior function in (2.5):

$$\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T) \propto p_1(x_1) g_1(y_1 \mid x_1) \prod_{t=2}^{T} p_t(x_t \mid \boldsymbol{x}_{t-1}) g_t(y_t \mid x_t). \tag{2.5}$$

When both $p_t(\cdot)$ and $g_t(\cdot)$ are Gaussian, the maximum of (2.5) can be obtained easily using a Kalman filter and smoother (Kalman, 1960). In general cases, when the analytic solution to optimize (2.5) is infeasible, the MAP estimator can be obtained by drawing sample paths $\{(x_1^{(i)}, \ldots, x_T^{(i)})\}_{i=1,\ldots,n}$ from the posterior distribution (2.5). We discuss estimating the most likely path using Monte Carlo methods in Section 3.

2.2. State-space emulation

We propose a state-space emulation approach for solving high-dimensional optimization problems. The approach constructs a state-space model so that the original optimization problem is equivalent to finding the most likely state path under the state-space model.

Let $f : \mathcal{X}^d \to \mathbb{R}$ be the objective function to be minimized and $\xi : \mathbb{R} \to [0, +\infty)$ be a monotone decreasing function. Then, minimizing $f(x)$ is equivalent to maximizing $\phi(x) := \xi(f(x))$, such that

$$\operatorname*{argmin}_{x \in \mathcal{X}^d} f(x) = \operatorname*{argmax}_{x \in \mathcal{X}^d} \phi(x).$$

Furthermore, if there exists a state-space model with a posterior function (2.5) that is proportional to $\phi(x)$ such that $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T) \propto \phi(\boldsymbol{x}_T) = \xi(f(\boldsymbol{x}_T))$, with artificially designed state equations $\{p_t(\cdot)\}_{t=1,\dots,T}$, observation equations $\{g_t(\cdot)\}_{t=1,\dots,T}$, and $T = d$, we call the state-space model an "emulated" state-space model. The observations $\boldsymbol{y}_T$ can either be observations from the original optimization problem (e.g., the observed points in the smoothing spline problem in Section 2.3.1), or can be designed artificially. Note that it is always possible to rewrite any joint distribution function $\phi(\boldsymbol{x}_T)$ in the form of (2.3) as $\phi(\boldsymbol{x}_T) = \phi(x_1, \dots, x_T) = \phi_1(x_1) \prod_{t=2}^{T} \phi_t(x_t \mid \boldsymbol{x}_{t-1})$, where $\phi_t(x_t \mid \boldsymbol{x}_{t-1}) = \int_{\mathcal{X}^{T-t}} \phi(\boldsymbol{x}_T) dx_{t+1} \cdots d_T / \int_{\mathcal{X}^{T-t+1}} \phi(\boldsymbol{x}_T) dx_t \cdots d_T$ and $\phi_1(x_1) = \int_{\mathcal{X}^{t-1}} \phi(\boldsymbol{x}_T) dx_2 \cdots dx_T$. However, a series of conditional distributions is difficult to sample from and to evaluate.

However, in certain problems, including our examples shown later, it is possible to reformulate the conditional distribution as $\phi_t(x_t \mid \boldsymbol{x}_{t-1}) = p_t(x_t \mid \boldsymbol{x}_{t-1}) g_t(y_t \mid x_t)$, in which it is easy to generate a sample from $p_t(x_t \mid \boldsymbol{x}_{t-1})$, and it is easy to evaluate $g_t(y_t \mid x_t)$, for some designed y_t. In general, objective functions with local dependence between parameters can be easily emulated by Markovian state-space models, as in our examples of smoothing splines, trend filtering, and the optimal trading path. Objective functions with more complex interactions between the parameters usually lead to non-Markovian emulated state-space models, which need more careful designs. The lasso regression in the Supplementary Material is one such case.

Minimizing the objective function is then equivalent to finding the most likely path for the emulated state-space model. The emulated state and observation equations provide guidance for further SMC implementation, even though they are artificial.

A common choice for $\xi(\cdot)$ is the Boltzmann distribution function

$$\xi(s) = e^{-\kappa s}, \tag{2.6}$$

where κ is a positive constant that relates to the temperature in statistical physics.

In statistics, the Boltzmann function in (2.6) links the least squares method to the maximum likelihood approach with independent and identically distributed (i.i.d.) Gaussian noise. With this choice of $\xi(\cdot)$, the system has a physical interpretation: The objective function $f(\cdot)$ is regarded as the possible energy levels in a non-quantum thermodynamic system. Assuming no interactions, the number of particles at the energy $f(x)$ follows the Boltzmann distribution under thermodynamic equilibrium. The integrability of $\phi(x)$ ensures the existence of the canonical partition function, such that this physical canonical system is valid. The minimization of $f(\cdot)$ is now equivalent to finding the base energy level, which inspires the use of simulated annealing of this thermodynamic system; see Section 3 for further discussion.

2.3. Examples

2.3.1. Cubic smoothing spline

Consider a nonparametric regression model $y_t = m(x_t) + \epsilon_t$ with equally spaced x_t. Without loss of generality, let $x_t = t$ and treat them as time. The cubic smoothing spline method (Green and Silverman, 1993) estimates a continuous function $m(t)$ by minimizing

$$L(\boldsymbol{y}_T) = \sum_{t=1}^{T}\{y_t - m(t)\}^2 + \lambda \int \{m''(t)\}^2\, dt. \tag{2.7}$$

The first term in (2.7) is the total squared tracking errors at the observation times, and the second term is the penalty term on the smoothness of the latent function $m(\cdot)$, where λ controls the regularization strength. Given values of $m(1), \ldots, m(T)$, the minimizer of the second term is a natural cubic spline that interpolates $m(1), \ldots, m(T)$ (see Green and Silverman, 1993). Hence, the solution that minimizes (2.7) is a natural cubic spline, which is second-order continuously differentiable and is a cubic polynomial in all intervals $[t, t+1]$, for $t = 1, \ldots, T-1$, and is linear outside $[1, T]$.

Define the derivatives of $m(t)$ at each observation at time t as

$$a_t = m(t), \quad b_t = m'(t), \quad c_t = \frac{m''(t)}{2}, \quad d_t = \lim_{s \to t_-} \frac{m'''(s)}{6}.$$

The natural cubic spline solution to (2.7) is equivalent to an emulated state-space model on $x_t = (a_t, b_t, c_t)$ with a vector autoregressive state equation

$$\begin{bmatrix} a_t \\ b_t \\ c_t \end{bmatrix} = \begin{bmatrix} 1 & 1 & \sqrt{3}/3 \\ 0 & 1 & \sqrt{3}-1 \\ 0 & 0 & -(2-\sqrt{3}) \end{bmatrix} \begin{bmatrix} a_{t-1} \\ b_{t-1} \\ c_{t-1} \end{bmatrix} + \begin{bmatrix} 1/3 \\ 1 \\ 1 \end{bmatrix} \eta_t, \tag{2.8}$$

with $\eta_t \sim \mathcal{N}(0, \sigma_b^2)$ and $\sigma_b^2 = 3(2-\sqrt{3})/(4\lambda\kappa)$. The corresponding observation equation is $y_t = a_t + \epsilon_t$, with $\varepsilon_t \sim \mathcal{N}(0, \sigma_y^2)$, $\sigma_y^2 = 1/(2\kappa)$, and the initial values $a_1 \sim \mathcal{N}(y_1, \sigma_y^2)$, $b_1 \sim 1$, and $c_1 = 0$. The derivation is postponed to the Supplementary Material.

2.3.2. Optimal trading path

In asset portfolio management, the optimal trading path problem is a class of optimization problems that typically maximize certain utility functions of the trading path (Markowitz, 1959). Kolm and Ritter (2015) and Irie and West (2016) proposed reformulating such problems as an emulated state-space model. Specifically, let $\boldsymbol{x}_T = (x_0, \ldots, x_T)$ be a trading path in which x_t represents the position held at time t. Kolm and Ritter (2015) propose maximizing the following utility function:

$$u(\boldsymbol{x}_T) = -\sum_{t=1}^{T} c_t(x_t - x_{t-1}) - \sum_{t=0}^{T} h_t(y_t - x_t), \tag{2.9}$$

where $(y_0, \ldots, y_T)$ is a predetermined optimal trading path in an ideal world without trading costs, typically obtained by maximizing the risk-adjusted expected return under the Markowitz mean-variance theory (Markowitz, 1959). Kolm and Ritter (2015) provide a construction of $(y_0, \ldots, y_T)$ based on the term structure of the underlying asset's *alpha* (the excess expected return relative to the market). Let $c_t(\cdot)$ represent the transaction cost, which is often assumed to be a quadratic function of the absolute position change $|x_t - x_{t-1}|$. Without loss of generality, we parametrize it as

$$c_t(|x_t - x_{t-1}|) = \frac{1}{2\sigma_x^2}\left(|x_t - x_{t-1}|^2 + 2\alpha|x_t - x_{t-1}| + \alpha^2\right),$$

where α is a nonnegative constant related to the volatility and liquidity of the asset (Kyle and Obizhaeva, 2011). Let $h_t(\cdot)$ be the utility loss due to the departure of the realized path from the ideal path. We use the squared loss $h_t(y_t - x_t) = (y_t - x_t)^2/(2\sigma_y^2)$. Then, the objective function is

$$e^{-\kappa u(\boldsymbol{x}_T)} \propto \prod_{t=1}^{T} \exp\left\{-\frac{\kappa(|x_t - x_{t-1}| + \alpha)^2}{2\sigma_x^2}\right\} \prod_{t=1}^{T} \exp\left\{-\frac{\kappa(y_t - x_t)^2}{2\sigma_y^2}\right\}.$$

Taking the position constraint $x_0 = x_T$ into consideration, as discussed in Cai, Chen and Lin (2018), an emulated state-space model can therefore be constructed as

$$p_t(x_t \mid x_{t-1}) \propto \exp\left\{-\frac{\kappa(|x_t - x_{t-1}| + \alpha)^2}{2\sigma_x^2}\right\}, \tag{2.10}$$

$$g_t(y_t \mid x_t) \propto \exp\left\{-\frac{\kappa(y_t - x_t)^2}{2\sigma_y^2}\right\}. \tag{2.11}$$

With the state equation (2.10) and the observation equation (2.11), the state-space model has a likelihood function proportional to $\exp\{-\kappa u(\boldsymbol{x}_T)\}$.

3. Annealed SMC

3.1. SMC

The SMC method is a class of sampling methods designed for state-space models. It uses the sequential nature of state-space models, and draws samples incrementally using sequential importance sampling and resampling (SISR) schemes. A typical SMC approach is demonstrated in Algorithm 1.

Algorithm 1 Sequential Monte Carlo (SMC) Algorithm.

- Draw $x_1^{(i)}$ from $p_1(x_1)$ and set weight $w_0^{(i)} = 1$ for $i = 1, \ldots, n$.
- For time $t = 2, \cdots, T$:
  - Propagation: For $i = 1, \cdots, n$,
    - Draw $x_t^{(i)}$ from $q_t(x_t \mid \boldsymbol{x}_{t-1}^{(i)})$ and set $\boldsymbol{x}_t^{(i)} = (x_{t-1}^{(i)}, x_t^{(i)})$.
    - Update weights by setting

$$w_t^{(i)} \leftarrow w_{t-1}^{(i)} \cdot \frac{p_t(x_t^{(i)} \mid \boldsymbol{x}_{t-1}^{(i)}) g_t(y_t \mid \boldsymbol{x}_t^{(i)})}{q_t(x_t^{(i)} \mid \boldsymbol{x}_{t-1}^{(i)})}.$$

  - Resampling (optional):
    - Assign a priority score $\beta_t^{(i)}$ to each sample $x_{0:t}^{(i)}$.
    - Draw samples $\{J_1, \ldots, J_n\}$ from the set $\{1, \ldots, n\}$ with replacement, with probabilities proportional to $\{\beta_t^{(i)}\}_{i=1,\ldots,n}$.
    - Let $\boldsymbol{x}_t^{*(i)} = \boldsymbol{x}_t^{(J_i)}$ and $w_t^{*(i)} = w_t^{(J_i)} / \beta_t^{(J_i)}$.
    - Set $\{(\boldsymbol{x}_t^{(i)}, w_t^{(i)})\}_{i=1,\ldots,n} \leftarrow \{(\boldsymbol{x}_t^{*(i)}, w_t^{*(i)})\}_{i=1,\ldots,n}$.
- Return the weighted sample set $\{(\boldsymbol{x}_T^{(i)}, w_T^{(i)})\}_{i=1,\ldots,n}$.

The function $q_t(\cdot)$ in the propagation step in Algorithm 1 is the proposal distribution. As discussed in Lin, Chen and Liu (2013), the "perfect" choice for the proposal is the conditional distribution with the full information set, such that $q_t(x_t \mid \boldsymbol{x}_{t-1}) = p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_T)$. However, in most cases, it is not possible to evaluate or sample from this conditional probability at time t. The priority score β_t is the weight used in the resampling step, and quantifies the sampler's preference over different sample paths. The most common choice of β_t is $\beta_t^{(i)} \propto w_t^{(i)}$. Variations of the SMC algorithm choose different proposal distributions and different priority scores. The Bayesian particle filter (Gordon, Salmond and Smith, 1993) sets $q_t(x_t \mid \boldsymbol{x}_{t-1}) = p_t(x_t \mid \boldsymbol{x}_{t-1})$. It works well when the observations y_T are relatively noisy compared with the state equation part. With accurate observations, the independent particle filter (Lin et al., 2005) uses $q_t(x_t \mid \boldsymbol{x}_{t-1}) \propto g_t(y_t \mid x_t)$. As an important (with a certain additional cost) compromise over the Bayesian particle filter and the independent particle filter, Kong, Liu and Wong (1994) and Liu and Chen (1998) suggest adopting $q_t(x_t \mid \boldsymbol{x}_{t-1}) \propto p_t(x_t \mid \boldsymbol{x}_{t-1})g_t(y_t \mid x_t)$ to reduce the variance. Other SMC methods focus on finding more appropriate priority scores in resampling, with the help of future information. The auxiliary particle filter (Pitt and Shephard, 1999) conducts resampling with the priority score $\beta_t^{(i)} = w_t^{(i)} p(y_{t+1} \mid x_t)$. The delayed sampling method (Chen, Wang and Liu, 2000; Lin, Chen and Liu, 2013) looks ahead Δ steps, and uses $\beta_t^{(i)} = w_t^{(i)} p(y_{t+1}, \ldots, y_{t+\Delta} \mid x_t)$.

In emulations for the optimizations, we are more interested in generating samples in the high probability density region of $\pi(\boldsymbol{x}_T)$. Hence our problem is essentially a smoothing problem. Briers, Doucet and Maskell (2010) proposed using a generalization of the two-filter smoothing formula to sample approximately from the joint distribution $\pi(\boldsymbol{x}_T)$. Additional local Markov Chain Monte Carlo (MCMC) moves can be adopted to mitigate degeneracy (Gilks and Berzuini, 2001). Many other SMC smoothing algorithm implementations reduce the potential degeneracy in samples; see, for example, Godsill, Doucet and West (2004); Del Moral, Doucet and Singh (2010); Briers, Doucet and Maskell (2010); Guarniero, Johansen and Lee (2017).

3.2. Finding the most likely path

With emulation, finding the optimum of $f(\boldsymbol{x})$ is now equivalent to finding the mode, or the most likely state path (MLP), of $\pi(\boldsymbol{x}_T)$,

$$\boldsymbol{x}_T^* = \underset{\boldsymbol{x}_T \in \mathcal{X}^T}{\operatorname{argmax}}\, \pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T), \tag{3.1}$$

with $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T)$ defined in (2.5) and $\mathcal{X}$ being the common support for all latent variables. By construction, the mode, which is the optimum of $f(\boldsymbol{x})$, does not depend on κ used in (2.6).

In this article, we focus on finding the MLP from Monte Carlo samples. A set of weighted Monte Carlo samples from the distribution $\pi(\boldsymbol{x}_T)$ can be generated using the SMC and its various implementation schemes. Let $\{(\boldsymbol{x}_T^{(i)}, w_T^{(i)})\}_{i=1,\ldots,n}$ be the samples drawn from the emulated state-space model using the SMC algorithm in Algorithm 1. A natural and easy way is to use the empirical MAP path, such that

$$\hat{\boldsymbol{x}}_T^{(map)} = \underset{\boldsymbol{x}_T \in \{\boldsymbol{x}_T^{(i)}\}_{i=1,\ldots,n}}{\operatorname{argmax}} \pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T). \tag{3.2}$$

Although the empirical MAP involves the least computation given the Monte Carlo samples, it usually requires a very large sample size to achieve high accuracy, especially when the dimension T is large.

Note that the MLP is the same under different κ. However, the distribution $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T, \kappa)$ is more flat for small κ (high temperature), and is more concentrated around the MLP for large κ. Hence, the empirical MAP path tends to be more accurate if the Monte Carlo samples are generated from the target distribution with large κ. When κ is sufficiently large, the average sample path is also a good estimate of the MAP. However, it is much more difficult to generate Monte Carlo samples with large κ, because of the tendency to be trapped in a local optimum. Simulated annealing gradually modifies the easily generated samples at a higher temperature to obtain samples from a lower temperature system with more accurate estimates.

3.3. Annealed SMC

We propose a simulated annealing algorithm for the SMC on state-space models. The idea comes from the thermodynamics analogue discussed in the previous section. When the function $\xi(\cdot)$ is chosen to be Boltzmann-like, as in (2.6), the Monte Carlo samples from the emulated state-space model correspond to a random sample set from the non-interacting particles in a thermodynamic equilibrium system, as discussed in Section 2.2.

If the temperature cools to zero sufficiently slowly that the system is approximately in thermodynamic equilibrium for any temperature in between, all particles will condense to the base energy level. The idea of simulated annealing as an analogy of the physical system was proposed and discussed in Kirkpatrick, Gelatt and Vecchi (1983).

To mimic the thermodynamic procedure, we propose the following system to simulate the annealing procedure for the SMC samples. Let $0 < \kappa_0 < \kappa_1 < \cdots < \kappa_K$ be an increasing sequence of inverse temperatures. Suppose at κ_0, a base emulated state-space model is constructed as

$$\pi(\boldsymbol{x}_T; \kappa_0) \propto e^{-\kappa_0 f(\boldsymbol{x}_T)} \propto p_0(x_0) \prod_{t=1}^{T} p_t(x_t \mid \boldsymbol{x}_{t-1}) g_t(y_t \mid x_t). \tag{3.3}$$

At a higher inverse temperature κ_k, an emulated state-space model can be induced from (3.3) such that

$$\pi(\boldsymbol{x}_T; \kappa_k) \propto e^{-\kappa_k f(\boldsymbol{x}_T)} \propto p_0(x_0; \kappa_k) \prod_{t=1}^{T} p_t(x_t \mid \boldsymbol{x}_{t-1}; \kappa_k) g_t(y_t \mid x_t; \kappa_k), \tag{3.4}$$

where $p_t(x_t \mid \boldsymbol{x}_{t-1}; \kappa_k) \propto \{p_t(x_t \mid \boldsymbol{x}_{t-1})\}^{\kappa_k/\kappa_0}$ and $g_t(y_t \mid x_t; \kappa_k) \propto \{g_t(y_t \mid x_t)\}^{\kappa_k/\kappa_0}$ are the corresponding state equations and observation equations, respectively at κ_k. The starting inverse temperature κ_0 is usually chosen to be relatively small, such that the function $\pi(\boldsymbol{x}_T; \kappa_0) \propto e^{-\kappa_0 f(\boldsymbol{x}_T)}$ is relatively flat and is easy to sample from using the SMC. We start with κ_0, and draw $\{(\boldsymbol{x}_{0,T}^{(j)}, w_{0,T}^{(j)})\}_{j=1,\ldots,m}$ from the base emulated state-space model $\pi(\boldsymbol{x}_T;\kappa_0)$. For $k = 1,\ldots,K$, new samples $\{(\boldsymbol{x}_{k,T}^{(j)}, w_{k,T}^{(j)})\}_{j=1,\ldots,m}$ are drawn with respect to the distribution $\pi(\boldsymbol{x}_T;\kappa_k)$, using the samples $\{(\boldsymbol{x}_{k-1,T}^{(j)}, w_{k-1,T}^{(j)})\}_{j=1,\ldots,m}$ obtained at κ_{k-1}. The procedure is depicted in Algorithm 2. The annealed SMC uses the following proposal distribution at temperature κ_k:

$$q_{k,t}(x_t \mid \boldsymbol{x}_{t-1};\kappa_k) \propto \hat{p}_{k,t}(x_t \mid \boldsymbol{x}_{t-1};\kappa_{k-1}), \tag{3.5}$$

where the conditional distribution $\hat{p}_{k,t}(x_t \mid \boldsymbol{x}_{t-1};\kappa_{k-1})$ is an estimate of $\pi_T(x_t \mid \boldsymbol{x}_{t-1};\kappa_{k-1})$, and can be obtained from the Monte Carlo samples $\{(\boldsymbol{x}_{k-1,T}^{(j)}, w_{k-1,T}^{(j)})\}_{j=1,\ldots,m}$ under κ_{k-1}. We discuss how to obtain such an estimate later. Because κ increases slowly, $\pi_T(x_t \mid \boldsymbol{x}_{t-1};\kappa_{k-1})$ and $\pi_T(x_t \mid \boldsymbol{x}_{t-1};\kappa_k)$ are reasonably close. With a sufficiently large terminating κ_K, samples from the target distribution $\pi(\boldsymbol{x}_T;\kappa_K)$ are highly concentrated around the true optimal path $\boldsymbol{x}_T^*$, and hence are useful for inferring the most likely path.

Algorithm 2 Annealed Sequential Monte Carlo Algorithm.

- Draw $\{(\boldsymbol{x}_{0,T}^{(j)}, w_{0,T}^{(j)})\}_{j=1,\ldots,m}$ from $\pi(\boldsymbol{x}_T;\kappa_0)$ with SMC in Algorithm 1, using a set of proposal distributions $q_{1,t}(x_t \mid \boldsymbol{x}_{t-1};\kappa_0)$.
- For $k = 1,\ldots,K$, draw $\{(\boldsymbol{x}_{k,T}^{(j)}, w_{k,T}^{(j)})\}_{j=1,\ldots,m}$ from $\pi(\boldsymbol{x}_T;\kappa_k)$ with SMC in Algorithm 1 using the proposal distribution

  $$q_{k,t}(x_t \mid \boldsymbol{x}_{t-1};\kappa_k) \propto \hat{p}_{k,t}(x_t \mid \boldsymbol{x}_{k,t-1}^{(j)}),$$

  where the right hand side is an estimate of $\pi_T(x_t \mid \boldsymbol{x}_{t-1};\kappa_{k-1})$.
- Estimate the most likely path from $\{(\boldsymbol{x}_{K,T}^{(j)}, w_{K,T}^{(j)})\}_{j=1,\ldots,m}$.

In summary, the annealed SMC provides an iterative procedure for the difficult sampling problem under κ_K by using samples obtained at a higher temperature. On the one hand, the annealed SMC provides a relatively "flat" and easy-to-sample starting distribution $\pi(\boldsymbol{x}_T;\kappa_0)$, and designs a slow-changing path connecting $\pi(\boldsymbol{x}_T;\kappa_0)$ to the desired "sharp" distribution $\pi(\boldsymbol{x}_T;\kappa_K)$. On the other hand, for each iteration $k = 1,\ldots,K$, the annealed SMC adopts an optimal proposal distribution $p(x_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{y}_T;\kappa_{k-1})$ based on the full information set $\boldsymbol{y}_T$, and is usually difficult to evaluate in conventional SMC implementations. In the annealed SMC, the proposal distribution is estimated by using sample paths from the previous iteration. The details of estimating the proposal distribution are discussed in the Supplementary Material.

Our annealing framework falls into the general framework of simulated annealing. The design of temperature sequences $\{\kappa_k\}_{k=0,\ldots,K}$ is known as the "cooling schedule". Kirkpatrick, Gelatt and Vecchi (1983) uses an exponential schedule such that $\kappa_k = \alpha^k\kappa_0$, for some positive number α. A more conservative schedule such that $\kappa_k \propto \log(1+k)$ is suggested by Hajek (1988) and Aarts and Korst (1989) to ensure convergence to a global minimum. Ingber (1989) proposed a fast adaptive cooling schedule that allows the temperature to increase (or κ to decrease) in order to regain the broadness of the samples at a certain point. The specific choice of cooling schedule is beyond the scope of this study. By default, we choose the most aggressive exponential schedule, with a picked value of α for faster convergence, in the example section, and the results are promising.

The conventional simulated annealing algorithm (Kirkpatrick, Gelatt and Vecchi, 1983) is a variation of the MCMC method, which adapts the Metropolis–Hastings algorithm (Metropolis et al., 1953; Hastings, 1970) with an extra temperature control. The convergence of the conventional simulated annealing algorithm is given by Granville, Krivanek and Rasson (1994). In contrast, the annealed SMC does not require a mixing condition, as is usually the case in MCMC algorithms. At each iteration at κ_k, the samples are always properly weighted with respect to the target distribution $\pi(\boldsymbol{x}_T; \kappa_k)$, because of the weight adjustments. The convergence of the SMC is discussed in Crisan and Doucet (2000).

The terminology "annealed SMC" is also used by Ulker, Gunsel and Cemgil (2011) and Wang, Wang and Bouchard-Côté (2019), although differently to how we use it in our method. The method of Ulker, Gunsel and Cemgil (2011) and Wang, Wang and Bouchard-Côté (2019) (henceforth, "SMC annealing") constructs an annealing sequence of intermediate target distributions $\pi_t(\boldsymbol{x})$, indexed by $t = 0, \ldots, T$, with $\pi_0(\boldsymbol{x})$ as the beginning distribution and $\pi_T(\boldsymbol{x})$ as the terminating distribution. The goal the method is to generate a set of samples that follow the terminating distribution by starting from samples that follow a relatively flat beginning distribution. SMC techniques are used when translating samples from the current distribution $\pi_t(\boldsymbol{x})$ to the next $\pi_{t+1}(\boldsymbol{x})$ by adopting an MCMC move as the proposal distribution. Our method also constructs a sequence of annealed target distributions $\pi_k(\boldsymbol{x}_T \mid \kappa_k)$, with the optimization using a Monte Carlo of a (near) degenerated terminating distribution. In our method, within each temperature (κ_k), we use the SMC to sample the high-dimensional $\boldsymbol{x}_T$ under a dynamic system setup. The sequence of SMC proposal distributions within each temperature uses the information contained in the Monte Carlo samples from the previous temperature.

More specifically, there are three major differences between the proposed method and the SMC annealing method. First, the goal of SMC annealing is to draw samples from a target distribution (usually the posterior) that is difficult to sample from directly. The goal of our algorithm is to find the optimum such that the terminating distribution is proportional to the original one, raised to an arbitrarily high power. Second, our method solves the problem when $\boldsymbol{x}$ itself is high dimensional with a dynamic structure, for which the SMC is used to sequentially sample the components of $\boldsymbol{x}$, whereas SMC annealing deals with relatively lower dimensional $\boldsymbol{x}$, without needing SMC sampling. Third, SMC annealing uses the SMC on the sequence of annealing distributions, whereas our method performs as SMC within each annealing temperature, and uses the samples from the previous iteration to construct the internal SMC propagation proposal step in the subsequent temperature.

3.4. Path refinement using the Viterbi algorithm

A more accurate estimate of the mode can be obtained by using the Viterbi algorithm (Viterbi, 1967) on the discrete space consisting of the SMC samples. The Viterbi algorithm is a dynamic programming algorithm originally used to solve the MLP problem in hidden Markov models, where the hidden states are finite. Let $\mathcal{A}_t = \{a_t^{(j)}\}_{j=1,\ldots,m}$ be the grid points for x_t, and $\Omega = \mathcal{A}_1 \times \cdots \times \mathcal{A}_T$ be the Cartesian product of the grid point sets. In state-space models, the Viterbi algorithm searches for the maximum over all possible combinations of the grid points in Ω. Specifically, the MLP obtained by the Viterbi algorithm is

$$\hat{\boldsymbol{x}_T}^{(viterbi)} = \operatorname*{argmax}_{\boldsymbol{x}_T \in \Omega} \pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T). \tag{3.6}$$

The Viterbi algorithm for state-space models based on the grid points $\{a_1^{(j)}\}_{j=1,\ldots,m}$, $\ldots, \{a_T^{(j)}\}_{j=1,\ldots,m}$ is depicted in Algorithm 3.

Algorithm 3 Viterbi Algorithm for Markovian State-Space Models.

- Let $\mathcal{A}_t = \{a_t^{(j)}\}_{j=1,\ldots,m}$ be a set of grid points for x_t for $t = 1, \ldots, T$.
- At time 1, initialize $\ell_0^{(j)} = 0$ and $\hat{\boldsymbol{x}}_1^{(j)} = a_1^{(j)}$ for $j = 1, \ldots, m$.
- At each time $t = 2, \ldots, T$, for $j = 1, \ldots, m$, set

$$\ell_t^{(j)} = \max_{k \in \{1,\ldots,m\}} \ell_{t-1}^{(k)} p_t(a_t^{(j)} \mid \hat{\boldsymbol{x}}_{t-1}^{(k)}) g_t(y_t \mid a_t^{(j)}), \tag{3.7}$$

  and set $\hat{\boldsymbol{x}}_t^{(i)} = (\hat{\boldsymbol{x}}_{t-1}^{(k_j^*)}, a_t^{(j)})$, where j_j^* is the optimal point of (3.7).
- Return $\hat{\boldsymbol{x}}_T^{(j^*)}$, where $j^* = \operatorname{argmax}_{j \in \{1,\ldots,m\}} \ell_T^{(j)}$.

Although the original Viterbi algorithm was designed for discrete state spaces, we adopt it for continuous state spaces by discretizing the state space into a set of selected finite grid points at each time point. The performance depends on the "quality" of the selected grid points (e.g., how densely close to the underlying optimal path) and on the number of grid points used. Here, we use the generated Monte Carlo samples as the discretizing grid points. Because these samples follow the target distribution at a low temperature, they should concentrate in the important regions.

For example, one can set $\mathcal{A}_t = \{x_t^{(i)}\}_{i=1,\ldots,m}$ such that $\Omega = \{x_1^{(i)}\}_{i=1,\ldots,m} \times \cdots \times \{x_T^{(i)}\}_{i=1,\ldots,m}$ is the joint set of all SMC sample points. Running the Viterbi algorithm through these samples improves the result from the Monte Carlo samples, but does not obtain the underlying optimal path in the continuous space. Therefore, we refer to this step as "refinement" rather than "optimization".

One can also add and remove grids points to expand the coverage, with more detail around the more important state paths. For instance, in the lasso regression example in the Supplement Material, a Viterbi refinement helps to shrink the estimate of the zero coefficients to exactly zero.

The Viterbi algorithm explores all combinations of sample points, and results in a better mode estimation than that of the empirical MAP in (3.2). However, it has limitations in terms of implementation with state-space models. One limitation is that the Viterbi algorithm works only on Markovian state-space models. In addition, it works only with a nonsingular state evolution in which the degrees of freedom is the same as the state variable dimension. Otherwise, the state paths cannot be re-assembled by the Viterbi algorithm. For example, in the cubic spline problem, the state evolution is singular. Although one can reduce the dimension of the state variable to make the evolution nonsingular, the state evolution then becomes non-Markovian. Another limitation is the requirement of the Monte Carlo sample size. The Monte Carlo samples induced by Ω provide a discretization of the support $\mathcal{X}$ for each time t. The accuracy of the Viterbi algorithm depends strongly on the discretization quality, especially when $\mathcal{X}$ is continuous. In general, the denser the Monte Carlo samples are around the true MLP, the more accurate the Viterbi algorithm solution is. As a result, it often requires a large Monte Carlo sample size to generate better discretization and to achieve high accuracy. To reduce the path error $\|\hat{x}_{1:T}^{(viterbi)} - x_{1:T}^*\|$ by half, the Monte Carlo sample size m needs to be doubled, because the discretization size is reduced by half, on average, when the sample size doubles. On the other hand, the computational cost increases quadratically with the sample size m. One way to improve this is to apply the Viterbi algorithm iteratively by shrinking to the high value region of the previous iteration, and regenerating grid points there. However, similar to an iterative grid search, the iterative Viterbi algorithm may yield a suboptimal solution.

4. Simulation Results

In this section, we provide simulated results for the annealed SMC in terms of finding the most likely path for the two emulated state-space models from Section 2.3.

Note that the smoothing spline problem has a closed-form solution. Even in the emulated state-space model setting, the Kalman filter provides the exact solution. It is used for illustration purposes only. On the other hand, the

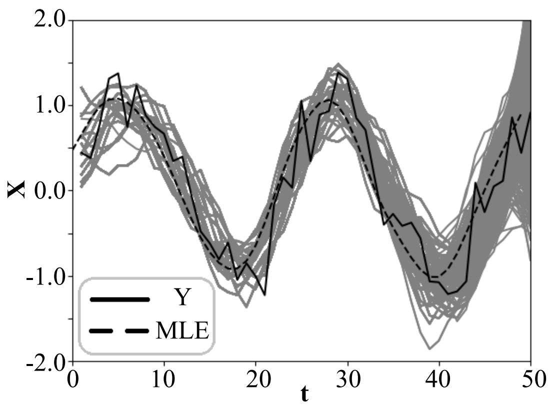

Figure 1. Sample paths at $\kappa_0 = 4$.

optimal trading path problem is not trivial, and is a real application to which the proposed method is ideally suited, especially when nonlinear solvers usually give less accurate solutions.

Two additional examples are provided in the Supplementary Material. We aim to demonstrate the flexibility of the proposed method by solving existing optimization problems with some new applications, though our approach may not yield better performance than that of specially designed optimization algorithms for general problems.

4.1. Cubic smoothing spline

In this simulation study, we consider the cubic smoothing spline problem in Section 2.3.1. The observations are generated by $y_t = \sin\{9(t-1)/100\} + \zeta_t$, for $t = 1, \ldots, 50$, with $\zeta_t \sim \mathcal{N}(0, 1/16)$, and we fix $\lambda = 10$ in the objective function (2.7).

Because the dynamic system is linear and Gaussian, the most likely path is obtained by the Kalman smoother (Kalman, 1960). We use this as the benchmark. We start from the initial inverse temperature $\kappa = \kappa_0 = 4$. Figure 1 demonstrates $m = 1{,}000$ samples (in gray) drawn from the target distribution $\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T; \kappa_0) \propto [\pi(\boldsymbol{x}_T \mid \boldsymbol{y}_T)]^{\kappa_0}$ by the SMC algorithm described in Algorithm 1, along with the observations $\boldsymbol{y}_T$ (the solid line) and the true most likely path (the dashed line).

The proposal distribution $q_t(\cdot)$ used at κ_0 is chosen to be proportional to $p_t(x_t \mid \boldsymbol{x}_{t-1}) g_t(y_t \mid x_t)$. At each time t, η_t is drawn from the proposal distribution $q_t(\eta_t \mid a_{t-1}, b_{t-1}, c_{t-1}, y_t)$, which is Gaussian. Resampling is conducted when the effective sample size (ESS) defined in (4.1) is less than $0.3m$:

$$ESS = \frac{(\sum_{i=1}^{m} w_t^{(i)})^2}{\sum_{i=1}^{m} (w_t^{(i)})^2}. \tag{4.1}$$

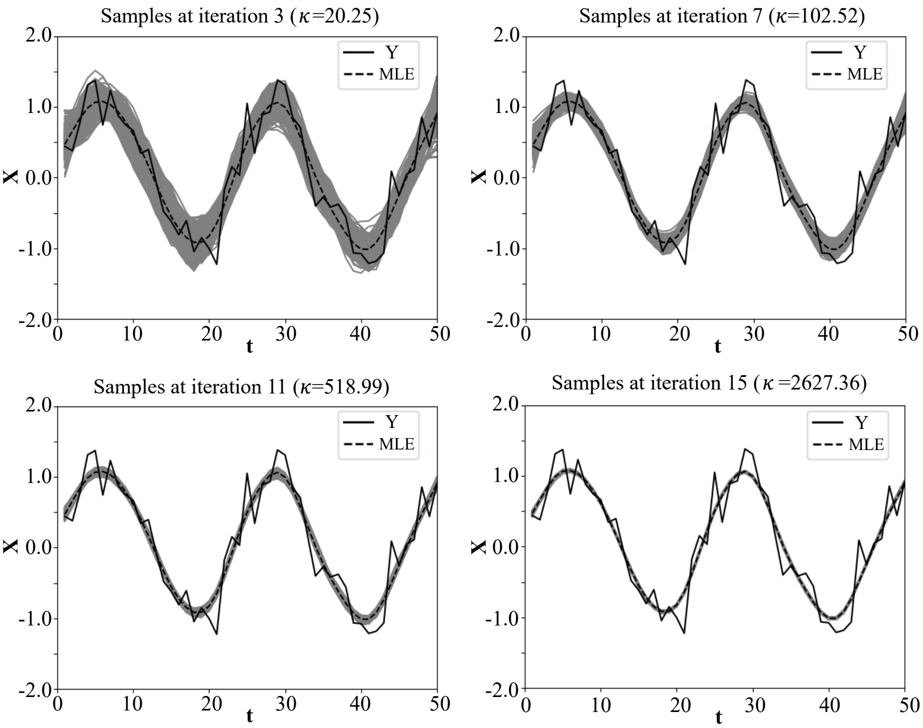

Figure 2. Sample paths at different κ's.

To find the most likely path stochastically and numerically, we apply the annealed SMC approach in Algorithm 2 with a predetermined sequence of inverted temperatures $\kappa_k = 1.5^k \kappa_0$, for $k = 1, \ldots, 16$. The proposal distribution for the annealed SMC is estimated using the parametric approach (see the Supplementary Material). Specifically, because the innovation in the state equation is of one dimension, at κ_k, we need only to generate proposal samples for c_t. To do so, we first fit $\{(c^{(j)}_{k-1,t}, a^{(j)}_{k-1,t-1}, b^{(j)}_{k-1,t-1}, c^{(j)}_{k-1,t-1})\}_{j=1,\ldots,m}$ with a multivariate Gaussian distribution, and then sample from the conditional distribution. To prevent degeneracy, the resampling step is only conducted at the end of each annealed SMC iteration, and after each iteration, one post-MCMC move is conducted to regenerate the sample states. The post-MCMC move uses blocked Gibbs sampling (Jensen, Kjærulff and Kong, 1995), owing to the special structure of the state dynamic. At each iteration of the Gibbs sampling, (x_t, x_{t+1}, x_{t+2}) are updated together.

Figure 2 shows the sample paths (after the post-MCMC step) at the end of different annealed SMC iterations. When the temperature shrinks to zero as κ increases, the sample paths move to a small neighborhood region around the true most likely path. Figure 3 shows the value of the objective function at the weighted average path of the samples for different numbers of iterations. The

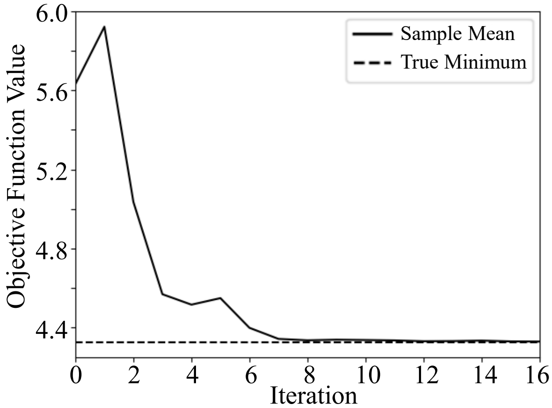

Figure 3. Value of the objective function against the number of iterations.

true optimal value (the objective function value at the optimal path) obtained by the Kalman smoother is plotted as the dashed horizontal line. As the number of iterations increases, the objective function value at the averaged path decreases stochastically, and converges at roughly the seventh iteration.

To compare the computational efficiency, we record the computing time needed for different approaches, as follows. The Kalman smoother takes 2.2 ms, Scipy minimizer takes 129.6 ms and the annealed SMC takes 232.9 ms. The Scipy approach uses the nonlinear optimizer provided by the python package Scipy (Jones, Oliphant and Peterson, 2001), which implements the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm by default. The annealed SMC records the time until convergence (the time when the value of the objective function is not improved by further iteration). The Kalman smoother is the fastest one to find the most likely path for linear Gaussian models, owing to its deterministic nature. The annealed SMC is slower than the nonlinear solver program provided by Scipy, but achieves similar accuracy. Note that this is a simple convex optimization problem in which a straightforward optimization algorithm such as the Scipy performs well. Our estimation approach is more flexible, and this example serves as an illustration of how the algorithm works.

4.2. Optimal trading path

In this simulation, we consider the optimal trading path problem in Section 2.3.2. Following Cai, Chen and Lin (2018), we set $T = 20$, $\sigma_x^2 = 0.25$, $\sigma_y^2 = 1$, and $\alpha = 0.5$. The ideal trading path is given by

$$y_t = 25 \exp\Big(-\frac{t+1}{8}\Big) - 40 \exp\Big(-\frac{t+1}{4}\Big).$$

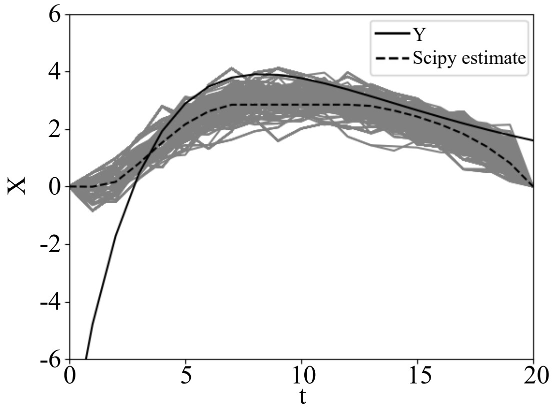

Figure 4. Sample paths at κ_0.

We start from the initial temperature $\kappa = \kappa_0 = 1.0$. The sample paths at κ_0 are drawn using the constrained SMC (Cai, Chen and Lin, 2018), where the resampling step is performed with the priority scores $\beta_t(\boldsymbol{x}_t) \propto \hat{p}(y_{t+1}, \ldots, y_T \mid x_t)$. The priority scores are estimated from a set of backward pilot samples (Cai, Chen and Lin, 2018). In this example, we use $m^* = 300$ backward pilot samples. The resulting $m = 1000$ (forward) sample paths are shown in Figure 4. The observations $y_1, \ldots, y_T$, which represent the ideal optimal trading strategy without trading costs, are plotted as the solid line. An estimated path (dashed line) is provided by the Scipy nonlinear optimization algorithm.

We use the following sequence of inverted temperatures for annealing: $\kappa_k = 2^k \kappa_0$, for $k = 1, \ldots, 20$. The proposal distribution in the annealed SMC is sampled using the parametric approach by approximating the joint distribution of $x_{k-1,t}$ and $x_{k-1,t-1}$ with a bivariate normal distribution. The annealed $m = 1000$ sample paths are resampled at the end of each iteration, and no post-MCMC step is conducted. Samples at several different inverted temperatures are shown in Figure 5. We use the sample average as our estimator for the most likely path. The value of the objective function at the sample average path decreases stochastically, as shown in Figure 6, eventually converging to around the 11th iteration. The optimal objective function value achieved by the annealed SMC is 89.459, whereas that obtained by the Scipy nonlinear optimizer is 89.462. The values of the objective function at the sample paths at the 20th iteration have an average of 89.459 and a standard deviation of 1.09×10^{-5}. The annealed SMC gains some improvement in accuracy at the cost of extra computation. The Scipy nonlinear optimizer takes 78 ms, and the annealed SMC takes 1.820 s for the initial emulated model (including the time for backward sampling) and costs around 2 ms for each subsequent iteration. Sampling from the base emulated model costs much more than in subsequent iterations for two reasons. First, it requires a large sample size for the base model, because of high degeneracy.

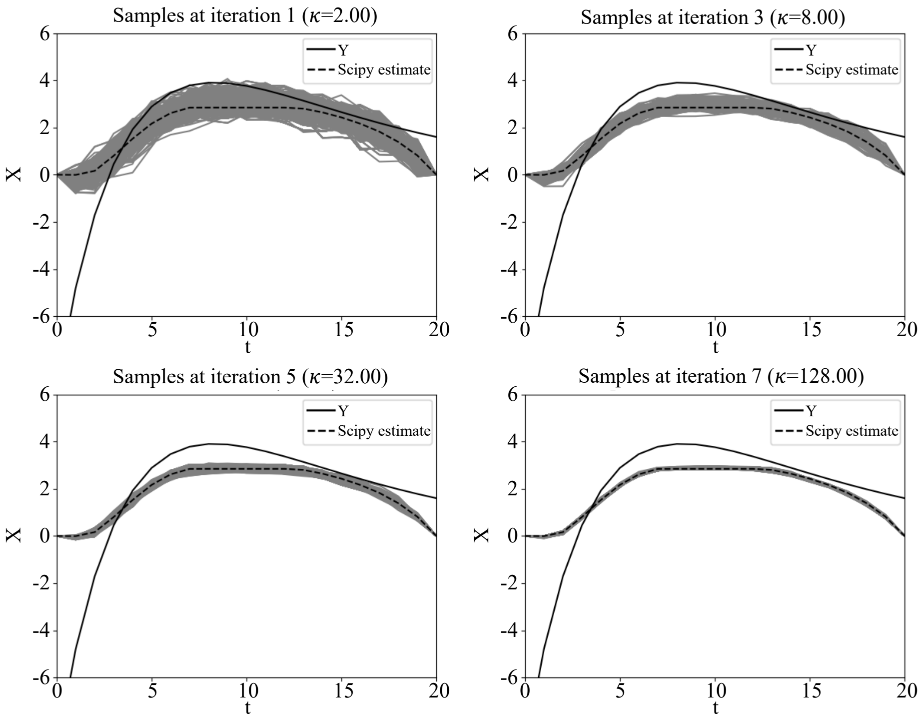

Figure 5. Sample paths at different κ's.

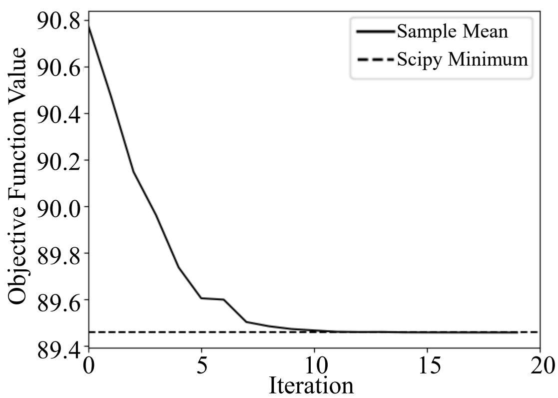

Figure 6. Value of the objective function against the number of iterations.

Second, the end point constraint is imposed and an additional backward pilot run is needed to reduce degeneracy.

5. Conclusion

In this article, we have proposed a general framework for state-space model emulation in high-dimensional optimization problems. The main idea of emulation is to change the goal from optimization to sampling. We have demonstrated that by constructing a proper state-space model, many high-dimensional optimization problems can be reformulated in terms of finding the optimal (most likely) path under the state-space model. In order to reduce the accuracy loss due to the nature of sampling, we propose the annealing steps with an extremely sharp terminating distribution, where the samples, though random, are highly concentrated around the optimum (the most likely path). We demonstrate the procedure of state-space model emulation using two conventional problems in the main content and in two additional problems given in the Supplementary Material and show how they can be solved using the proposed annealed SMC approach.

The proposed annealed SMC approach shares some properties with traditional simulated annealing methods. Both can optimize a wide range of objective functions, including nonconvex functions and multi-modal functions, and both often require a heavier computation cost than the simpler standard optimization algorithms, such as the gradient descent algorithms. However, the annealed SMC approach for state-space models differs from the traditional simulated annealing methods with an MCMC for stochastic optimization in the following ways. First, emulating an optimization problem as a state-space model is advantageous when the problem is high dimensional, and when the system is inherently dynamic (such as the trading path problem or the ℓ_1 trend filtering problem) or when the parameters to be estimated inherently play similar roles in the problem (such as the parameters in the regression problem). Second, the SMC as an alternative to the MCMC has certain advantages in many fixed-dimensional problems, such as those in which the "dependence" between the parameters in the emulated target distribution is local and (locally) very strong. In such problems, the MCMC encounters slow mixing difficulties, whereas the SMC naturally takes advantage of such properties. Third, given any temperature, the SMC samples target the equilibrium distribution, whereas the MCMC samples often move toward the target distribution gradually. Hence, the annealed SMC may tolerate a faster cooling schedule. Fourth, the inherited parallel structure of the SMC allows for faster computation, and enables better adaption to multi-modal problems.

The state-space model emulation and the annealed SMC provide an alternative way to solve high-dimensional optimization problems. Of course, the approach may not be suitable for all problems, owing to its high computational cost and its requirement of certain structures. Nevertheless, the proposed approach is a useful high-dimensional optimization method for a wide range of complex problems that more traditional methods struggle to solve. Although the examples presented here do not demonstrate a significant improvement of the state-space emulation approach over the traditional one, they effectively show how to implement, and how to use it for other problems.

Supplementary Material

The online Supplementary Material contains technical details related to the annealed SMC algorithm, and two additional emulation examples with simulation results.

Acknowledgments

The authors thank the editor, associate editor, and two anonymous referees for their helpful comments and suggestions. Chen's research was supported, in part, by National Science Foundation grants DMS-1737857, DMS-2052949, DMS-2027855, CCF-1934924, and IIS-1741390.

References

Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing.* John Wiley & Sons, Inc.

Agarwal, A., Anandkumar, A., Jain, P., Netrapalli, P. and Tandon, R. (2014). Learning sparsely used overcomplete dictionaries. In *Proceedings of the 27th Conference on Learning Theory* **PMLR 35**, 123–137.

Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M. and Telgarsky, M. (2014). Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research* **15**, 2773–2832.

Arora, S., Ge, R., Kannan, R. and Moitra, A. (2012). Computing a nonnegative matrix factorization–provably. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing*, 145–162.

Bertsekas, D. P. (1997). Nonlinear programming. *Journal of the Operational Research Society* **48**, 334–334.

Briers, M., Doucet, A. and Maskell, S. (2010). Smoothing algorithms for state-space models. *Annals of the Institute Statistical Mathematics* **62**, 61–89.

Cai, A., Tsay, R. S. and Chen, R. (2009). Variable selection in linear regression with many predictors. *Journal of Computational and Graphical Statistics* **18**, 573–591.

Cai, C., Chen, R. and Lin, M. (2018). Resampling strategy in sequential Monte Carlo for constrained sampling problems. *arXiv:1706.02348.*

Chen, R., Wang, X. and Liu, J. S. (2000). Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering. *IEEE Transactions on Information Theory* **46**, 2079–2094.

Crisan, D. and Doucet, A. (2000). Convergence of sequential Monte Carlo methods. Technical report. University of Cambridge, Cambridge. Web: `https://www.stats.ox.ac.uk/~doucet/crisain_doucet_convergenceofSMC2000.pdf`.

Del Moral, P., Doucet, A. and Singh, S. (2010). Forward smoothing using sequential Monte Carlo. *arXiv:1012.5390.*

Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **38**, 1–22.

Doucet, A., de Freitas, N. and Gordon, N. (2001). An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo Methods in Practice*, 3–14. Springer.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **63**, 127–146.

Godsill, S. J., Doucet, A. and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association* **99**, 156–168.

Gordon, N. J., Salmond, D. J. and Smith, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)* **140**, 107–113.

Granville, V., Krivanek, M. and Rasson, J. (1994). Simulated annealing: A proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16**, 652–656.

Green, P. J. and Silverman, B. W. (1993). *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. CRC Press.

Guarniero, P., Johansen, A. M. and Lee, A. (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association* **112**, 1636–1647.

Hajek, B. (1988). Cooling schedules for optimal annealing. *Mathematics of Operations Research* **13**, 311–329.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57**, 97–109.

Hooke, R. and Jeeves, T. A. (1961). "Direct search" solution of numerical and statistical problems. *Journal of the ACM* **8**, 212–229.

Iglesias, M. A., Law, K. J. H. and Stuart, A. M. (2013). Ensemble Kalman methods for inverse problems. *Inverse Problems* **29**, 045001.

Ingber, L. (1989). Very fast simulated re-annealing. *Mathematical and Computer Modelling* **12**, 967–973.

Irie, K. and West, M. (2016). Bayesian emulation for optimization in multi-step portfolio decisions. *arXiv:1607.01631.*

Jain, P. and Kar, P. (2017). Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning* **10**, 142–336.

Jensen, C. S., Kjærulff, U. and Kong, A. (1995). Blocking gibbs sampling in very large probabilistic expert systems. *International Journal of Human-Computer Studies* **42**, 647–666.

Jones, E., Oliphant, T. and Peterson, P. (2001). SciPy: Open source scientific tools for Python.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* **82**, 35–45.

Kiefer, J. and Wolfowitz, J. (1952). Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics* **23**, 462–466.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983). Optimization by simulated annealing. *Science* **220**, 671–680.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics* **5**, 1–25.

Kolm, P. N. and Ritter, G. (2015). Multiperiod portfolio selection and Bayesian dynamic models. *Risk* **28**, 50–54.

Kong, A., Liu, J. S. and Wong, W. H. (1994). Sequential imputations and Bayesian missing data problems. *Journal of the American Statistical Association* **89**, 278–288.

Kyle, A. and Obizhaeva, A. (2011). Market microstructure invariants: Theory and implications of calibration. Web: `https://ssrn.com/abstract=1978932`.

Land, A. H. and Doig, A. G. (1960). An automatic method of solving discrete programming problems. *Econometrica* **28**, 497–520.

Lin, M., Chen, R. and Liu, J. S. (2013). Lookahead strategies for sequential Monte Carlo. *Statistical Science* **28**, 69–94.

Lin, M. T., Zhang, J. L., Cheng, Q. and Chen, R. (2005). Independent particle filters. *Journal of the American Statistical Association* **100**, 1412–1421.

Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association* **90**, 567–576.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American statistical association* **93**, 1032–1044.

Markowitz, H. M. (1959). *Portfolio Selection: Efficient Diversification of Investments.* John Wiley & Sons, New York; Chapman & Hall, London.

Mei, S., Montanari, A. and Nguyen, P.-M. (2018). A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences* **115**, E7665–E7671.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* **21**, 1087–1092.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal* **7**, 308–313.

Netrapalli, P., Niranjan, U., Sanghavi, S., Anandkumar, A. and Jain, P. (2014). Non-convex robust PCA. In *Advances in Neural Information Processing Systems* **27**, 1107–1115.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association* **94**, 590–599.

Ulker, Y., Gunsel, B. and Cemgil, A. T. (2011). Annealed SMC samplers for nonparametric Bayesian mixture models. *IEEE Signal Processing Letters* **18**, 3–6.

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory* **13**, 260–269.

Wang, L., Wang, S. and Bouchard-Côté, A. (2019). An annealed sequential Monte Carlo method for Bayesian phylogenetics. *Systematic Biology* **69**, 155–183.

Zhang, P., Song, Q. and Liang, F. (2021). A langevinized ensemble Kalman filter for large-scale static and dynamic learning. *arXiv:2105.05363*.

(Received May 2022; accepted April 2023)