ROYAL SOCIETY
OPEN SCIENCE

rsos.royalsocietypublishing.org

Research

CrossMark
click for updates

Cite this article: Sarkar S, Roy D, Vasu RM. 2015 A global optimization paradigm based on change of measures. *R. Soc. open sci.* **2**: 150123. http://dx.doi.org/10.1098/rsos.150123

Received: 24 March 2015
Accepted: 2 June 2015

Subject Category:
Mathematics

Subject Areas:
computer modelling and simulation

Keywords:
local and global extremizations, martingale problem, random perturbations, particle swarm optimization, differential evolution, covariance matrix adaptation evolution strategy

Author for correspondence:
Debasish Roy
e-mail: royd@civil.iisc.ernet.in

Electronic supplementary material is available at http://dx.doi.org/10.1098/rsos.150123 or via http://rsos.royalsocietypublishing.org.

THE ROYAL SOCIETY
PUBLISHING

A global optimization paradigm based on change of measures

Saikat Sarkar[1], Debasish Roy[1] and Ram Mohan Vasu[2]

[1]Computational Mechanics Laboratory, Department of Civil Engineering, and
[2]Department of Instrumentation and Applied Physics Indian Institute of Science,
Bangalore 560012, India

A global optimization framework, COMBEO (**C**hange **O**f **M**easure **B**ased **E**volutionary **O**ptimization), is proposed. An important aspect in the development is a set of derivative-free additive directional terms, obtainable through a change of measures *en route* to the imposition of any stipulated conditions aimed at driving the realized design variables (particles) to the global optimum. The generalized setting offered by the new approach also enables several basic ideas, used with other global search methods such as the particle swarm or the differential evolution, to be rationally incorporated in the proposed set-up via a change of measures. The global search may be further aided by imparting to the directional update terms additional layers of random perturbations such as 'scrambling' and 'selection'. Depending on the precise choice of the optimality conditions and the extent of random perturbation, the search can be readily rendered either greedy or more exploratory. As numerically demonstrated, the new proposal appears to provide for a more rational, more accurate and, in some cases, a faster alternative to many available evolutionary optimization schemes.

1. Introduction

Potential applications of global optimization are ubiquitous across a large spectrum of problems in science and engineering, from very-large-scale integration circuit design, to optimization of transportation routes to determination of protein structures. An essential aspect of the solution to any such problem is the extremization of an objective functional, possibly subject to a prescribed set of constraints. The solution is deemed to have been achieved if the global extremum of the objective functional is reached. For many practical applications, the objective functional could be non-convex, non-separable and even non-smooth. This precludes gradient-based local optimization approaches to treat these problems, an aspect that has driven extensive research in devising global optimization strategies, many of which employ stochastic (i.e. random evolutionary) search of heuristic or meta-heuristic origin [1]. Indeed, finding the global optimum within a search space of the parameters or design variables is much more challenging than arriving at a local extremum. In the absence of generic directional information (equivalent to the Gateaux or directional derivative in the search for a local extremum of a sufficiently smooth cost functional), a direct search seems to be the only option left. Some of the notable schemes in this genre include variants of the genetic algorithm (GA) [2], differential evolution (DE) [3], particle swarm optimization (PSO) [4], ant colony optimization [5] and covariance matrix adaptation evolution strategy (CMA-ES) [6]. In a search for the global extremum, stochastic schemes typically score over their gradient-based deterministic counterparts [7,8], even for cases involving sufficiently smooth cost functionals with well-defined directional derivatives. Nevertheless, thanks to the proper directional information guiding the update, gradient-based methods possess the benefit of faster convergence to the nearest extremum, as long as the objective functional is sufficiently smooth. To the authors' knowledge, the existing evolutionary global optimization schemes do not offer the benefit of well-founded directional information. Most evolutionary schemes depend on a random (diffusion-type) scatter applied to the available candidate solutions or particles and some criteria for selecting the new particles. Unfortunately, a 'greedy' approach for the global optimum (e.g. selecting particles of higher 'fitness' with higher probability), though tempting from the perspective of faster convergence, faces the hazard of getting trapped in local extrema. As a fix to a premature and possibly wrong convergence, many evolutionarily global search schemes adopt safeguards. Despite the wide acceptability of a few evolutionary methods of the heuristic/meta-heuristic origin [9], the underlying justification is often based on sociological or biological metaphors [1–6,10] that are hardly founded on a sound probabilistic basis even though a random search forms a key ingredient of the algorithm. In this context, some interesting comments on the facile usage of natural metaphors in the development of many global optimization algorithms, without a serious engagement with scientific ratiocination, may be found in [10]. Be that as it may, the popular adoption of these schemes is not only owing to the algorithmic simplicity, but mainly because of their effectiveness in treating many non-deterministic polynomial-time hard optimization problems, which may be contrasted with a relatively poorer performance of some of the well-grounded stochastic schemes, e.g. simulated annealing [11], stochastic tunnelling [12], etc. Whether a slow convergence to the global optimum, not infrequently encountered with some of the popular evolutionary schemes, could be fixed by reorienting them with an alternative stochastic framework, however, remains a moot point, which is addressed in this work to an extent. Another related question is regarding the precise number of parameters in the algorithm that the end-user must tune for an appropriate *exploration–exploitation trade-off* [13,14], a phrase used to indicate the relative balance of random scatter of the particles vis-a-vis their selection based on the 'fitness'. Admittedly, the notion of random evolution, built upon Monte Carlo (MC) sampling of the particles, efficiently explores the search space, though at the cost of possibly slower convergence [15] in contrast to a gradient-based method. For better exploration, many such schemes, e.g. the GA, adopt steps like 'crossover', 'mutation', 'selection', etc. While 'crossover' and 'mutation' impart variations in the particles, by the 'selection' step each particle is assigned a 'weight' or fitness value (a measure of importance) determined as the ratio of the realized value of the objective functional to the available maximum of the same. In the selection step, the fitness values, used to update the particles via selection of the best-fit individuals for subsequent crossover and mutation, may be considered functionally analogous to the derivatives used in gradient-based approaches. Analogy may also be drawn between the fitness values and the likelihood ratios commonly employed in nonlinear stochastic filtering, e.g. the family of sequential importance sampling filters [16]. Even though a weight-based approach reduces a measure of misfit between the available best and the rest within a finite ensemble of particles, they bring with them the curse of degeneracy, wherein all particles but one tend to receive zero weights as the iterations progress [17]. This problem, also referred to as 'particle collapse' in the stochastic filtering parlance, can be arrested, in principle, by exponentially increasing the ensemble size (number of particles), a requirement that can hardly be met in practice [18]. Evolutionary schemes that replace the weight-based multiplicative approach by adopting additive updates for the particles, often succeed in eliminating particle degeneracy. Evolutionary schemes like DE, PSO, etc., that are known to perform better than GA, use such additive particle update strategies without adversely affecting the 'craziness' or randomness in the evolution. On reaching the optimum, the additive correction shrinks to zero. Unfortunately, none of these methods obtain the additive term, which may be looked upon as a stochastic equivalent to the derivative-based directional information, in a rigorous or optimal manner—a fact that is perhaps responsible for a painstakingly large number of functional evaluations in some cases.

In framing a rigorous stochastic basis leading to directional updates of the additive type, several possible schemes are suggested in this work. First, it is shown that the problem of optimization may be generically posed as a *martingale problem* [19,20], which must however be randomly perturbed to facilitate the global search. For instance, one may perform a greedy search for an extremum of an objective functional by solving a martingale problem, which is in the form of an integro-differential equation herein referred to as the extremal equation. In line with the Freidlin–Wentzell theory [21], the martingale problem may further be perturbed randomly so as to enable the greedy scheme to converge to the global extremum as the strength of the perturbation vanishes asymptotically. The solution through this approach, which is viewed as a stochastic process parametered via the iterations, depends crucially on an error or 'innovation' function and is deemed to have been realized when the innovation function, interpreted as a stochastic process, is driven to a zero-mean noise or martingale [19]. The martingale structure of the innovation essentially implies that the mean of the computed cost functional in future iterations remains constant and invariant to zero-mean random perturbations of the argument vector (the design variables). The argument vector should then correspond to an extremum. In the evolutionary random set-up, both the cost functional and its argument vector are treated as stochastic diffusion processes even though the original extremization problem is deterministically posed. Realizing a zero-mean martingale structure for the innovation requires a change of measures that in turn leads to additive gain-type updates on the particles. The gain coefficient matrix, which is a replacement for and generalization over the Frechet derivative of a smooth cost functional, provides an efficacious directional search without requiring the functional to be differentiable. Additionally, in order to facilitate the global search, a general random perturbation strategy (using steps such as 'scrambling', 'relaxation' and 'selection') is adopted to insure against a possible trapping of particles in local extrema.

An important feature of the present approach is the flexibility with which the innovation process may be designed in order to meet a set of conflicting demands *en route* the detection of the global extremum. The efficiency of the global search basically relies upon the ability of the algorithm to explore the search space while preserving some directionality that helps to quickly resolve the nearest extremum. The development of the proposed set-up, referred to as COMBEO (**C**hange **O**f **M**easure **B**ased **E**volutionary **O**ptimization), recognizes the near impossibility of a specific optimization scheme performing uniformly well across a large class of problems. In this context, one may refer to variants of the famous 'no free lunch theorem', where it is proved that the average performance of any two algorithms across all possible problems is identical [22]. Accommodation of this fact had earlier led to a proposal of an evolutionary scheme that simultaneously ran different optimization methods for a given problem with some communications built among the updates by the different methods [23]. A major related contribution of this work is to bring the basic ideas for global search used in a few well-known optimization schemes under a single unified framework propped up by a sound probabilistic basis. In a way explained later, the ideas (or their possible generalizations) behind some of the existing optimization methods may sometimes be readily incorporated in the present setting by just tweaking the innovation process.

The rest of the paper is organized as follows. In §2, the problem of finding an extremum of an objective functional is posed as a martingale problem represented through an integro-differential equation, whose solution only realizes a local optimum. The integro-differential equation is discretized and weakly solved within an MC setting so as to circumvent its inherent circularity. Section 3 discusses several random perturbation schemes to arrive at the global optimum efficiently without getting stuck in local traps. By combining these tools in different ways, a few pseudo-codes are presented in §4. In §5, the performance of COMBEO is compared with DE, PSO and CMA-ES in the context of several benchmark problems. In the same section, the new optimization approach is also applied for the quantitative recovery of boundary layer density variation in a high-speed flow given the light travel time data from an experiment. Finally, the conclusions of this study are drawn in §6.

2. Search for an extremum through a martingale problem

Here, it is demonstrated that functional extremization, subject to a generic set of equality constraints, can be posed as a martingale problem upon a proper characterization of the design variables within a probabilistic setting. However, before elaborating on this, a few fundamental features of the new evolutionary approach are listed below:

(i) at a given iterate, the solution is a random variable (taking values in the search space);

(ii) hence the solution process along the iteration axis may be considered a stochastic process and the iterations must be so designed that the optimal solution is asymptotically approached by the mean (i.e. the first moment) of the solution process; and

(iii) since upon convergence the mean should be iteration-invariant, random fluctuations about the converged mean must assume the structure of a zero-mean stochastic process (along the iteration axis), thereby allowing us to identify the fluctuations as a noise process, e.g. a Brownian motion, or, more generally, a zero-mean martingale.

In posing the global optimization problem within a stochastic framework, a complete probability space $(\Omega, \mathcal{F}, P)$ [19] is considered, within which the solution to the optimization problem must exist as an $\mathcal{F}$-measurable random variable. Here Ω, known as the population set (sample space), necessarily includes all possible candidate solutions from which a finite set of randomly chosen realizations or particles is evolved along the iteration axis τ. The introduction of τ as a positive monotonically increasing function on $\mathbb{R}$ is required to qualify the evolution of the solution as a continuously parametered stochastic process. A necessary aspect of the evolution is a random initial scatter imparted to the particles so as to search the sample space for solutions that extremize the cost functional while satisfying the posed constraints, if any. Since the extremal values, global or otherwise, of the objective functional may not be known *a priori*, the particles are updated iteratively (i.e. along τ) by conditioning on the evolution history of a so-called extremal cost process based on the available particles. The extremal cost process is defined as a stochastic process such that, for any $\xi \in [0, \tau]$, its mean is given by the available best objective functional based on its evaluations across the particles at the iteration step denoted by ξ. Now denoting by $\mathcal{N}_\tau := \{\mathcal{G}_\xi\}_{\xi \in [0,\tau]}$ the filtration (i.e. statistical information pertaining to the evolution history) containing the increasing family of sub σ-algebras generated by the extremal cost process, the aim is to determine the update by conditioning the guessed or predicted solution on $\mathcal{N}_\tau$. The guessed solution could be a Brownian motion or a random walk along τ. In case there are equality constraints, $\mathcal{N}_\tau$ also contains history for the best realized constraint until τ. For a multivariate multimodal nonlinear objective functional $f(\mathbf{x}) : \mathbb{R}^n \mapsto \mathbb{R}$ in $\mathbf{x} = \{x^j\}_{j=1}^n \in \mathbb{R}^n$, a point $\mathbf{x}^*$ needs to be found such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$, $\forall\ \mathbf{x} \in \mathbb{R}^n$. Since the design variable $\mathbf{x}$ is evolved in τ as a stochastic process, it is parametrized as $\mathbf{x}_\tau := \mathbf{x}(\tau)$. It may be noted that there may not be any inherent physical dynamics in its evolution, and in such cases $\mathbf{x}_\tau$ may be evolved as a zero-mean Brownian motion or random walk in Ω (even though, as will be shown later, this step is not absolutely necessary). Since only a finite number of iterations can be performed in practice, τ is discretized as $\tau_0 < \tau_1 < \cdots < \tau_M$ so that $\mathbf{x}_\tau$ is evolved over every τ increment. Thus, for the ith iteration with $\tau \in (\tau_{i-1}, \tau_i]$, $\mathbf{x}_\tau$ may be thought of as being governed by the following stochastic differential equation (SDE):

$$\mathrm{d}\mathbf{x}_\tau = \mathrm{d}\boldsymbol{\xi}_\tau. \tag{2.1}$$

Here $\boldsymbol{\xi}_\tau$ is a zero-mean vector Brownian process with the covariance matrix $\mathbf{g}\mathbf{g}^{\mathrm{T}} \in \mathbb{R}^{n\times n}$, where $\mathbf{g} \in \mathbb{R}^{n\times n}$ is the noise intensity matrix with its (j,k)th element denoted as g^{jk}. The discrete τ-marching map for equation (2.1) has the form:

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \Delta\boldsymbol{\xi}_i, \quad \Delta\boldsymbol{\xi}_i := \boldsymbol{\xi}_i - \boldsymbol{\xi}_{i-1}, \tag{2.2}$$

where $(\cdot)_i$ stands for $(\cdot)_{\tau_i}$. Let the extremal cost process, generating the filtration $\mathcal{N}_\tau$, be denoted as $\widehat{f}_\tau$. Within a strictly deterministic setting, a smooth functional $f(\mathbf{x})$ tends to become stationary, in the sense of a vanishing first variation, as $\mathbf{x}$ approaches an extremal value $\mathbf{x}^*$. In the stochastic setting, a counterpart of this scenario could be that any future conditioning of the process $\mathbf{x}_\tau$ on $\mathcal{N}_\xi$, which is the filtration generated by $\widehat{f}_s$, $s < \xi$, until a past iteration (i.e. $\xi < \tau$), identically yields the random variable $\mathbf{x}_\xi$ itself, i.e. $E(\mathbf{x}_\tau | \mathcal{N}_\xi) = \mathbf{x}_\xi$, where E represents the expectation operator. Via the Markov structure of $\mathbf{x}_\tau$ one may equivalently postulate that a necessary condition for the extremization of $f_\tau := f(\mathbf{x}_\tau)$ is to require that $E(\mathbf{x}_\tau | \widehat{f}_\xi) = \mathbf{x}_\xi$. This characterization renders the process $\mathbf{x}_\tau$ a martingale with respect to the extremal cost filtration $\mathcal{N}_\tau$, i.e. once extremized, any future conditional mean of $\mathbf{x}_\tau$ on the cost filtration remains iteration invariant. See [24] for an introductory treatise on the theory of martingales. An equivalent way of stating this condition is based on an error or 'innovation' process defined as $\widehat{f}_\tau - f_\tau$. An extremization would require updating $\mathbf{x}_\tau$ to statistically match the extremal process $\mathbf{x}_\tau^*$ so that the innovation $\widehat{f}_\tau - f(\mathbf{x}_\tau^*)$ is driven to a zero mean martingale, e.g. a zero-mean Brownian motion or, more generally, a stochastic integral in Ito's sense (or, in a τ-discrete setting, a zero-mean random walk). This enables one to pose the determination of an extremum as a martingale problem, originally conceptualized by Stroock & Varadhan [20] to provide an improved setting for solutions to SDEs beyond

Ito's theory. If the update scheme is effective, one anticipates $\mathbf{x}_\tau \to \mathbf{x}_\tau^*$ asymptotically, i.e. as $\tau \to \infty$. In other words, $E(\lim_{\tau\to\infty} E(\mathbf{x}_\tau|\mathcal{N}_\tau)) = E(\mathbf{x}_\tau^*) = \mathbf{x}^*$ and, as the norm of the noise intensity associated with $\mathbf{x}_\tau$ approaches zero, $\mathbf{x}_\tau^* \to \mathbf{x}^*$. However, since strictly zero noise intensity is infeasible within the stochastic set-up, the solution in a deterministic setting may be thought of as a degenerate version (with a Dirac measure) of that obtained through the stochastic scheme.

Note that a ready extension of this approach is possible with multi-objective functions, which would merely require building up the innovation as a vector-valued process. Similarly, the approach may also be adapted for treating equality constraints, wherein zero-valued constraint functions, perturbed by zero-mean noise processes, may be used to construct the innovation processes. The easy adaptability of the current set-up for multiple objective functions or constraints may be viewed as an advantage over most existing evolutionary optimization schemes, where a single cost functional needs to be constructed based on all the constraints, a feature that may possibly engender instability for a class of large dimensional problems.

Driving $\mathbf{x}_\tau$ to an $\mathcal{N}_\tau$-martingale, or forcing the innovation process to a zero-mean martingale, may be readily achieved through a change of measures. In evolutionary algorithms, e.g. the GA, the accompanying change of measures requires the assignment of weights or fitness values to the current set of particles and a subsequent selection of those with higher fitness, e.g. by rejection sampling. For a better exploration of the sample space, the GA uses steps like 'crossover' and 'mutation'. These steps are biologically inspired safeguards against an intrinsic limitation of the weight-based update, which tends to diminish all the weights but one to zero, thereby leaving the realized particles nearly identical after a few iterations. This problem is referred to as that of weight collapse or particle impoverishment and precipitates premature convergence to a wrong solution. As a way out of this degeneracy in the realized population set (i.e. the ensemble), the present scheme employs a purely additive update, derived following a Girsanov change of measures $P \to Q$ (see [19] for a detailed exposition on the Girsanov transformation), which ensures that the innovation process, originally described under measure P, becomes a zero-mean martingale under the new measure Q. A basic tool in effecting this change of measure is the Radon–Nikodym derivative $\Lambda_\tau := \mathrm{d}P/\mathrm{d}Q$ (assuming absolute continuity of Q w.r.t. P and vice versa), a scalar valued random variable also called the likelihood ratio (or the fitness or the weight) that multiplicatively updates the design process $\mathbf{x}_\tau$ as $\mathbf{x}_\tau\Lambda_\tau$. In order to bypass the degeneracy problem noted above, one may obtain an additive update by expanding $\mathbf{x}_\tau\Lambda_\tau$ using Ito's formula [19]. Within an MC set-up, an immediate consequence of the additive update is that particles with lower weights are never eliminated, but rather corrected to have more fitness by being driven closer to an extremum. Drawing analogy with Taylor's expansion of a smooth function(al) that obtains its first-order term based on Newton's directional derivative, the present version of the additive correction may be interpreted as a non-Newton directional term that drives the innovation process to a zero-mean martingale and a precise form of this term is derived later in this section. In general, the innovation process may be a vector. One such case occurs when, by way of addressing possible numerical instability, a single cost functional is split into several so that each of the correspondent innovations is driven to a zero-mean martingale. This in turn ensures that the innovation corresponding to the original cost functional is also driven to a zero-mean martingale. For a simpler exposition of the basic ideas, the method below is presented using a single cost functional only. The formulation can be trivially extended for vector innovation processes of any finite dimension.

Within a τ-discrete framework with $\tau \in (\tau_{i-1}, \tau_i]$, $\mathbf{x}_\tau$ is evolved following equation (2.1). An innovation constraint that may be satisfied for an accelerated (greedy) search in a neighbourhood of an extremum is given by

$$\widehat{f}_\tau - f(\mathbf{x}_\tau) = \Delta\eta_\tau, \tag{2.3}$$

where $\widehat{f}_\tau \in \mathbb{R}$ is the extremal cost process, f the objective functional and $\Delta\eta_\tau = \eta_\tau - \eta_{i-1} \in \mathbb{R}$ a P-Brownian increment representing the diffusive fluctuations. Deriving the subsequent integro-differential equation for the search is best accomplished by recasting equation (2.3) as an SDE. Towards this, an $\mathcal{N}_\tau$-measureable process $\breve{f}_\tau := \breve{f}(\tau)$ may be constructed to arrive at the following incremental form:

$$\Delta\breve{f}_\tau := \widehat{f}_\tau\Delta\tau = f(\mathbf{x}_\tau)\Delta\tau + \Delta\eta_\tau\Delta\tau; \Delta\tau = \tau - \tau_{i-1}. \tag{2.4}$$

Here $\Delta\tau_i := \tau_i - \tau_{i-1}$ is taken as a 'small' increment. Since the τ-axis is fictitious, $\Delta\tau$ is replaced by $\Delta\tau_i = \tau_i - \tau_{i-1}$, so that equation (2.4) is modified to

$$\Delta\breve{f}_\tau = f(\mathbf{x}_\tau)\Delta\tau + \Delta\eta_\tau\Delta\tau_i, \tag{2.5}$$

which is essentially correspondent to the SDE:

$$\mathrm{d}\breve{f}_\tau = f(\mathbf{x}_\tau)\,\mathrm{d}\tau + \Delta\tau_i\,\mathrm{d}\eta_\tau. \tag{2.6}$$

Note that the replacement of $\Delta\tau$ by $\Delta\tau_i$ merely modifies the intensity of the noise process $\Delta\eta_\tau$ in equation (2.3). The form of SDE (2.6) has the desirable feature that the fictitious diffusion coefficient $\Delta\tau_i$ is an order 'smaller' relative to the drift coefficient $f(\mathbf{x}_\tau)$. However, since η_τ is not standard Brownian, it is more convenient to rewrite equation (2.6) as

$$\mathrm{d}\breve{f}_\tau = f(\mathbf{x}_\tau)\,\mathrm{d}\tau + \rho_\tau\,\mathrm{d}W_\tau, \tag{2.7}$$

where W_τ is standard P-Brownian and ρ_τ a more general form of (scalar-valued) noise intensity that may be made a function of τ. For multi-objective optimization problems (where $\breve{f}_\tau$ is a vector stochastic process), ρ_τ would be the intensity matrix and the expressions below are so written that they are valid for both scalar and vector cases. Equation (2.7) may be rewritten as

$$\mathrm{d}\tilde{f}_\tau := \rho_\tau^{-1}\,\mathrm{d}\breve{f}_\tau = h(\mathbf{x}_\tau)\,\mathrm{d}\tau + \mathrm{d}W_\tau, \tag{2.8}$$

where $h(\mathbf{x}_\tau) := \rho_\tau^{-1} f(\mathbf{x}_\tau)$. While it is possible, and even desirable, to replace the Brownian noise term above by one whose quadratic variation is zero (e.g. a Poisson martingale), such a modification is not central to the basic idea and will be considered in future extensions of the work. SDE (2.7) is assumed to satisfy the standard conditions [19] so as to ensure the existence of a weak solution. The $\mathcal{N}_\tau$-measurable locally optimal (extremal) solution may now be identified with the conditional mean $E(\mathbf{x}_\tau|\mathcal{N}_\tau)$. Considering a new measure Q under which $\mathbf{x}_\tau$ from equation (2.1) satisfies equation (2.7), the conditional mean may be expressed via the generalized Bayes' formula as

$$\pi_\tau(\mathbf{x}) := E(\mathbf{x}_\tau|\mathcal{N}_\tau) = \frac{E_Q(\mathbf{x}_\tau \Lambda_\tau|\mathcal{N}_\tau)}{E_Q(\Lambda_\tau|\mathcal{N}_\tau)}, \tag{2.9}$$

where the expectation $E_Q(\cdot)$ is taken with respect to the new measure Q and Λ_τ is the scalar fitness given by

$$\Lambda_\tau = \exp\left[\int_{\tau_{i-1}}^{\tau} h_s\,\mathrm{d}\tilde{f}_s - \frac{1}{2}\int_{\tau_{i-1}}^{\tau} h_s^{\mathrm{T}} h_s\,\mathrm{d}s\right]. \tag{2.10}$$

As shown in appendix A (a theorem and its corollary) using Ito's expansions of $\mathbf{x}_\tau\Lambda_\tau$ and $(\Lambda_\tau)^{-1}$, the incrementally additive updates to arrive at an extremum must follow from the differential equation:

$$\mathrm{d}\pi_\tau(\mathbf{x}) = (\pi_\tau(\mathbf{x}f) - \pi_\tau(\mathbf{x})\pi_\tau(f))(\rho_\tau\rho_\tau^{\mathrm{T}})^{-1}(\mathrm{d}\breve{f}_\tau - \pi_\tau(f)\,\mathrm{d}\tau) \tag{2.11}$$

and

$$\pi_\tau(\mathbf{x}) = \pi_{i-1}(\mathbf{x}) + \int_{\tau_{i-1}}^{\tau} (\pi_s(\mathbf{x}f) - \pi_s(\mathbf{x})\pi_s(f))(\rho_s\rho_s^{\mathrm{T}})^{-1}(\mathrm{d}\breve{f}_s - \pi_s(f)\,\mathrm{d}s). \tag{2.12}$$

Note that the directionality of a search process provided by the second (integral) term on the right-hand side (r.h.s.) of the equation above may also be gauged from the fact the integrand in this term can be interpreted as a Malliavin-type derivative using Clarke–Ocone theorem [25]. But the appearance of the unknown term $\pi_\tau(f)$ on the r.h.s. prevents equations (2.11) or (2.12) to be qualified as an SDE in $\pi_\tau(\mathbf{x})$. Indeed, a necessarily nonlinear dependence of f on $\mathbf{x}_\tau$ and the consequent non-Gaussianity of $\pi_\tau(f)$ would prevent writing the latter in terms of $\pi_\tau(\mathbf{x})$. This results in the so-called closure problem in solving for $\pi_\tau(\mathbf{x})$.

While a direct solution of equations (2.11) or (2.12) yields an extremum (i.e. a local extremum only) in principle, exact/analytical solutions are infeasible owing to the circularity inherent in the closure problem. This has a parallel in nonlinear stochastic filtering, wherein the Kushner–Stratonovich equation [26] (an equivalent of equation (2.11)) also suffers from a similar circularity problem. Motivated by the MC filters used to solve nonlinear filtering problems [27,28], an MC scheme may be developed for a numerical treatment of equations (2.11) or (2.12). A two-stage strategy, viz. prediction and update, may apparently be considered, even though, as we will soon see, the prediction step could be entirely eliminated in the final scheme. The prediction step, as in most evolutionary optimization schemes, aims at an initial random exploration based on equation (2.1). Consider the ith iteration, i.e. $\tau \in (\tau_{i-1}, \tau_i]$ and let N denote the ensemble size so that one realizes N predicted particles or MC candidates, $\{\mathbf{x}_\tau^{(j)}\}_{j=1}^N$ that must be updated via equation (2.12). For an MC-based numerical solution to equation (2.12), a sample-averaged form of the equation is first written as

$$\pi_\tau^N(\mathbf{x}) = \pi_{i-1}^N(\mathbf{x}) + \int_{\tau_{i-1}}^{\tau} (\pi_s^N(\mathbf{x}f) - \pi_s^N(\mathbf{x})\pi_s^N(f))(\rho_s\rho_s^{\mathrm{T}})^{-1}(\mathrm{d}\breve{f}_s - \pi_s^N(f)\,\mathrm{d}s). \tag{2.13}$$

$\pi^N(.) = (1/N)\sum_{j=1}^N (.)^{(j)}$ is the ensemble-averaged approximation to the conditional mean $\pi(.)$. A particle-wise representation of equation (2.13) is given by

$$\mathbf{X}_\tau = \mathbf{X}_{i-1} + \frac{1}{N}\int_{\tau_{i-1}}^{\tau} \{\mathbf{X}_s\mathbf{F}_s{}^{\mathrm{T}} - \hat{\mathbf{X}}_s\hat{\mathbf{F}}_s^{\mathrm{T}}\}(\rho_s\rho_s^{\mathrm{T}})^{-1}\{d\,\breve{\mathbf{F}}_s - \mathbf{F}_s\,\mathrm{d}s\}, \tag{2.14}$$

where $\mathbf{X}_\tau := [\mathbf{x}_\tau^{(1)}, \mathbf{x}_\tau^{(2)}, \ldots \mathbf{x}_\tau^{(N)}]$, $\mathbf{F}_\tau := [f_\tau^{(1)}, f_\tau^{(2)}, \ldots, f_\tau^{(N)}]$, $\hat{\mathbf{X}}_\tau := \pi_\tau^N(\mathbf{x})\mathbf{r} \in \mathbb{R}^{n\times N}$, $\hat{\mathbf{F}}_\tau := \pi_\tau^N(f)\mathbf{r} \in \mathbb{R}^N$ and $\mathrm{d}\,\breve{\mathbf{F}}_\tau := \mathrm{d}\breve{f}_\tau\mathbf{r} \in \mathbb{R}^N$. $\mathbf{r} = \{1, 1, \ldots, 1\} \in \mathbb{R}^N$ is an N-dimensional row vector with entries 1. Note that the second term on the r.h.s. of equation (2.14) is the additive update/correction term. For solving equation (2.14), the MC approximation to equation (2.12), a τ-discrete numerical scheme is required. Such a scheme would typically involve the following two steps.

2.1. Prediction

The predicted particle set $\tilde{\mathbf{X}}_\tau = [\tilde{\mathbf{x}}_\tau^{(1)}, \ldots, \tilde{\mathbf{x}}_\tau^{(N)}]$ at τ is generated using an Euler–Maruyama (EM) discretized map

$$\tilde{\mathbf{X}}_\tau = \mathbf{X}_{i-1} + \Delta\boldsymbol{\Psi}_\tau, \tag{2.15}$$

where $\Delta\boldsymbol{\Psi}_\tau := [\Delta\boldsymbol{\xi}_\tau^{(1)}, \ldots, \Delta\boldsymbol{\xi}_\tau^{(N)}]$, $\Delta\boldsymbol{\xi}_\tau^{(1)} = \boldsymbol{\xi}_\tau^{(1)} - \boldsymbol{\xi}_{i-1}^{(1)}$, etc.

2.2. Additive update

The update to the predicted particles is through an EM approximation to the integral in equation (2.14). Higher order integration schemes could also be considered, especially for evaluating the correction integral. The discrete update equation is presently given by

$$\mathbf{X}_\tau = \tilde{\mathbf{X}}_\tau + \frac{1}{N}\left\{\tilde{\mathbf{X}}_\tau\tilde{\mathbf{F}}_\tau^{\mathrm{T}} - \tilde{\hat{\mathbf{X}}}_\tau\tilde{\hat{\mathbf{F}}}_\tau^{\mathrm{T}}\right\}(\rho_\tau\rho_\tau^{\mathrm{T}})^{-1}\left\{\Delta\,\breve{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau\Delta\tau\right\}, \tag{2.16}$$

where $\Delta\,\breve{\mathbf{F}}_\tau = \Delta\breve{f}_\tau\mathbf{r}$, $\tilde{\mathbf{F}}_\tau := [\tilde{f}_\tau^{(1)}, \ldots, \tilde{f}_\tau^{(N)}]$ and the predicted solution, $\tilde{\mathbf{x}}_\tau$, is used to compute $\tilde{f}_\tau := f(\tilde{\mathbf{x}}_\tau)$. Moreover, $\tilde{\hat{\mathbf{F}}}_\tau := \pi_\tau^N(\tilde{f})\mathbf{r}$. Equation (2.16) may be recast as

$$\mathbf{X}_\tau = \tilde{\mathbf{X}}_\tau + \frac{1}{N}\left\{(\tilde{\mathbf{X}}_\tau - \tilde{\hat{\mathbf{X}}}_\tau)\tilde{\mathbf{F}}_\tau{}^{\mathrm{T}} + \tilde{\hat{\mathbf{X}}}_\tau(\tilde{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau)^{\mathrm{T}}\right\}(\rho_\tau\rho_\tau^{\mathrm{T}})^{-1}\left\{\Delta\,\breve{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau\Delta\tau\right\}. \tag{2.17}$$

Recall from equation (2.4) that $\Delta\breve{f}_\tau := \widehat{f}_\tau\Delta\tau$ so that equation (2.17) may be rearranged as

$$\mathbf{X}_\tau = \tilde{\mathbf{X}}_\tau + \frac{1}{N}\left\{(\tilde{\mathbf{X}}_\tau - \tilde{\hat{\mathbf{X}}}_\tau)(\tilde{\mathbf{F}}_\tau^{\mathrm{T}}\Delta\tau) + (\tilde{\hat{\mathbf{X}}}_\tau\Delta\tau)(\tilde{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau)^{\mathrm{T}}\right\}(\rho_\tau\rho_\tau^{\mathrm{T}})^{-1}\left\{\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau\right\}, \tag{2.18}$$

where $\widehat{\mathbf{F}}_\tau = \widehat{f}_\tau\mathbf{r} \in \mathbb{R}^N$. When the particles are somewhat away from a local extremum (e.g. during the initial stages of evolution), the (norm of the) correction term is large. Hence the particles traverse more in the search space. In such cases, the innovation process would not behave as a zero-mean martingale as it would have a significant drift component. Since evolutions in this regime may have sharper τ-gradients, it is appropriate to modify the coefficient matrix in equation (2.18) so as to incorporate information on these gradients through previous estimates. Thus, $\tilde{\mathbf{F}}_\tau^{\mathrm{T}}\Delta\tau$ and $\tilde{\hat{\mathbf{X}}}_\tau\Delta\tau$ are replaced, respectively, by the following approximations:

$$\tilde{\mathbf{F}}_\tau^{\mathrm{T}}\Delta\tau \approx (\tilde{\mathbf{F}}_\tau^{\mathrm{T}}\tau - \tilde{\hat{\mathbf{F}}}_{i-1}^{\mathrm{T}}\tau_{i-1} - \Delta\tilde{\hat{\mathbf{F}}}_\tau^{\mathrm{T}}\tau) \tag{2.19}$$

and

$$\tilde{\hat{\mathbf{X}}}_\tau\Delta\tau \approx (\tilde{\hat{\mathbf{X}}}_\tau\tau - \tilde{\hat{\mathbf{X}}}_{i-1}\tau_{i-1}). \tag{2.20}$$

Note that we have used Ito's formula while approximating the r.h.s. in equation (2.19). Using equations (2.19) and (2.20), equation (2.18) may be modified as

$$\mathbf{X}_\tau = \tilde{\mathbf{X}}_\tau + \frac{1}{N}\left\{\begin{matrix} c(\tilde{\mathbf{X}}_\tau - \tilde{\hat{\mathbf{X}}}_\tau)(\tilde{\mathbf{F}}_\tau^{\mathrm{T}}\tau - \tilde{\hat{\mathbf{F}}}_{i-1}^{\mathrm{T}}\tau_{i-1} - \Delta\tilde{\hat{\mathbf{F}}}_\tau^{\mathrm{T}}\tau) \\ +(\tilde{\hat{\mathbf{X}}}_\tau\tau - \tilde{\hat{\mathbf{X}}}_{i-1}\tau_{i-1})(\tilde{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau)^{\mathrm{T}} \end{matrix}\right\}(\rho_\tau\rho_\tau^{\mathrm{T}})^{-1}\{\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau\}. \tag{2.21}$$

Once the converged estimate is obtained, the innovation noise covariance $\rho_\tau\rho_\tau^{\mathrm{T}}$ should satisfy the identity:

$$\begin{aligned}\rho_\tau\rho_\tau^{\mathrm{T}} &\approx \pi_\tau^N((\widehat{f}_\tau - \tilde{f}_\tau)(\widehat{f}_\tau - \tilde{f}_\tau)^{\mathrm{T}}) \\ &= \frac{1}{N-1}((\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau) - (\widehat{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau))((\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau) - (\widehat{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau))^{\mathrm{T}}.\end{aligned} \tag{2.22}$$

Prior to convergence to an extremum, the (norm of the) r.h.s. of the equation above would typically be relatively large. Thus, one could impart higher diffusion to the search in the initial stages by replacing $\rho_\tau\rho_\tau^{\mathrm{T}}$ in equation (2.21) by

$$\alpha\frac{1}{N-1}((\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau) - (\widehat{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau))((\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau) - (\widehat{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau))^{\mathrm{T}} + (1-\alpha)\rho_\tau\rho_\tau^{\mathrm{T}}. \tag{2.23}$$

Here $\alpha \in (0, 1]$. It is typically taken as 0.8 in the numerical illustrations based on *pseudo-codes 1, 2* and *3*. Equation (2.21) thus takes the final form:

$$\begin{aligned}\mathbf{X}_\tau = \tilde{\mathbf{X}}_\tau + \frac{1}{N}&\left\{\begin{matrix} c(\tilde{\mathbf{X}}_\tau - \tilde{\hat{\mathbf{X}}}_\tau)(\tilde{\mathbf{F}}_\tau^{\mathrm{T}}\tau - \tilde{\hat{\mathbf{F}}}_{i-1}^{\mathrm{T}}\tau_{i-1} - \Delta\tilde{\hat{\mathbf{F}}}_\tau^{\mathrm{T}}\tau) \\ +(\tilde{\hat{\mathbf{X}}}_\tau\tau - \tilde{\hat{\mathbf{X}}}_{i-1}\tau_{i-1})(\tilde{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau)^{\mathrm{T}} \end{matrix}\right\} \\ &\times\left\{\begin{matrix} c\alpha\frac{1}{N-1}((\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau) - (\widehat{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau)). \\ ((\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau) - (\widehat{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau))^{\mathrm{T}} + (1-\alpha)\rho_\tau\rho_\tau^{\mathrm{T}} \end{matrix}\right\}^{-1}\{\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau\}.\end{aligned} \tag{2.24}$$

A more concise form of the update equation is

$$\mathbf{X}_\tau = \tilde{\mathbf{X}}_\tau + \tilde{\mathbf{G}}_\tau\{\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau\}, \tag{2.25}$$

where the gain-like update coefficient matrix is given by

$$\begin{aligned}\tilde{\mathbf{G}}_\tau := \frac{1}{N}&\left\{\begin{matrix} c(\tilde{\mathbf{X}}_\tau - \tilde{\hat{\mathbf{X}}}_\tau)(\tilde{\mathbf{F}}_\tau^{\mathrm{T}}\tau - \tilde{\hat{\mathbf{F}}}_{i-1}^{\mathrm{T}}\tau_{i-1} - \Delta\tilde{\hat{\mathbf{F}}}_\tau^{\mathrm{T}}\tau) \\ +(\tilde{\hat{\mathbf{X}}}_\tau\tau - \tilde{\hat{\mathbf{X}}}_{i-1}\tau_{i-1})(\tilde{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau)^{\mathrm{T}} \end{matrix}\right\} \\ \times&\left\{c\alpha\frac{1}{N-1}((\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau) - (\widehat{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau))((\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau) - (\widehat{\mathbf{F}}_\tau - \tilde{\hat{\mathbf{F}}}_\tau))^{\mathrm{T}} + (1-\alpha)\rho_\tau\rho_\tau^{\mathrm{T}}\right\}^{-1}.\end{aligned}$$

In the update strategy of equation (2.25), one may still improve on the search space exploration by multiplying the gain-weighted innovation term by a scalar factor $\beta_\tau < 1$. This is equivalent to increasing the noise covariance and hence allowing the particles to be more diffusive or 'explorative'. Thus, the update equation becomes

$$\mathbf{X}_\tau = \tilde{\mathbf{X}}_\tau + \beta_\tau\tilde{\mathbf{G}}_\tau\{\widehat{\mathbf{F}}_\tau - \tilde{\mathbf{F}}_\tau\}. \tag{2.26}$$

3. Coalescence, scrambling and relaxation: schemes for global search

The possible trapping of particles in local extrema is a major challenge to any global optimization scheme. For $\beta_\tau \geq 1$, the scheme described in equation (2.26) yields a greedy search that may often end up in a local extremum. Although it seems possible to avoid the local traps through the innovation noise, whose intensity could be tuned by the scalar factor β_τ, such an approach may not be quite effective. As a local extremum is approached, the 'strength' (or norm) of the update term would be small and consequently its sensitivity to variations in β_τ would also be poor. This makes the choice of β_τ difficult (e.g. necessitating β_τ to be too small for the search scheme to be efficient) and less effective for the global search. Moreover, smaller β_τ also implies larger diffusion and hence poorer convergence. A more effective way out is a random perturbation applied to the particles so as to force out the ones trapped in the local wells. A general perturbation scheme, combining three basic approaches referred to as ‘coalescence’, ‘scrambling’ and ‘relaxation’, is now described. While the coalescence component is implemented through yet another martingale problem, the scrambling part requires a perturbation kernel. The relaxation component, on the other hand, requires accepting improvements with some probability. The importance of the random perturbation steps, in the present context, may also be gauged from the following fact. For a class of problems considered later, the faster convergence to a local extremum, engendered by the greedy scheme, equation (2.26), may force wrong convergence despite the application of random perturbations in conjunction with the greedy local search. Hence it would be worth exploring alternative search schemes, which completely eliminate scheme (2.26) and rely on the random perturbation steps alone.

Since the aim is to pose ‘coalescence’ as a martingale problem, the associated update should share the same generic structure as equation (2.26), which is recast below for the jth candidate

$$\mathbf{x}_i^{(j)} = \tilde{\mathbf{x}}_i^{(j)} + \tilde{\mathbf{C}}_i^{(j)}, \tag{3.1}$$

where $\tilde{\mathbf{C}}_i^{(j)} := \beta_i \tilde{\mathbf{G}}_i \{\widehat{f}_i - \tilde{f}_i^{(j)}\}$. The basic idea here is to provide to the evolving solution layers of random perturbation, whose inverse intensity may be formally indexed by a positive integer l such that the perturbation vanishes as $l \to \infty$. Within the τ-discrete setting, we start with the prediction $\tilde{\mathbf{x}}_i$ and denote by ${}^{l}\Delta\hat{\mathbf{x}}_i = {}^{l}\mathbf{x}_i - {}^{l}\tilde{\mathbf{x}}_i$ the randomly perturbed, l-indexed increment so that ${}^{\infty}\Delta\hat{\mathbf{x}}_i \to \Delta\mathbf{x}_i$ as the random perturbations vanish asymptotically. During the ith iteration, the perturbed increment is arrived at using two transitional increments, ${}^{l}\mathbf{u}_i$ and ${}^{l}\mathbf{v}_i$ determined by two perturbation operators, say **T1** and **T2**, respectively. While the operator **T1** corresponds to the ‘local search’ and/or ‘coalescence’ operations, **T2** encapsulates the ‘scrambling’ and ‘relaxation’ operations. The transitions may be indicated as ${}^{l}\Delta\mathbf{x}_i \to \lim_{\mathbf{T1}} {}^{l}\mathbf{u}_i \to \lim_{\mathbf{T2}} {}^{l}\mathbf{v}_i \to \lim_{\mathbf{T3}} {}^{l}\Delta\hat{\mathbf{x}}_i$, where **T3** is a selection operator, commonly used with most evolutionary optimization schemes in some form or the other. Here ${}^{l}\Delta\hat{\mathbf{x}}_i$ is the final increment at τ_i, i.e. ${}^{l}\mathbf{x}_i := {}^{l}\tilde{\mathbf{x}}_i + {}^{l}\Delta\hat{\mathbf{x}}_i$, which is input to the next iteration. ${}^{l}\Delta\mathbf{x}_i = {}^{l}\tilde{\mathbf{x}}_i - {}^{l}\mathbf{x}_{i-1}$ denotes the predicted increment. Ideally, one may start the iterations with small l (i.e. high perturbation intensity), which is then gradually increased with progressing iterations. However, in the current implementations of COMBEO, the perturbation intensity is kept uniform all through the iterations. This is possible as, upon convergence to the global optimum, the applied perturbations merely yield zero-mean random fluctuations about the optimum, which are averaged out when the sample expectation operation is performed in the MC simulation. In the view of this and for notational ease, the left superscript l is removed from the variables in the discussion to follow and the perturbed nature of the variables should be clear from the context. The operators are now defined below.

3.1. Operator **T1**: local search and coalescence

The operation for local search has been described in §2 and through equation (3.1). The operation of ‘coalescence’ is now outlined. This perturbation is motivated by the observation that the probability density function (PDF, if it exists) associated with the converged measure $\pi_\tau(.)$ should be unimodal, with its only peak located at the global extremum. Thus, when convergence to the global extremum occurs, all the particles should coalesce at the optimum point, except for a zero-mean noisy scatter around the latter. Ideally, for the sake of optimization accuracy, the noisy scatter should also have a low intensity so as to keep sample fluctuations under control. Once the global optimization scheme converges, the noisy scatter should then behave as a zero-mean martingale as a function of τ and with a unimodal transitional PDF. A zero-mean Brownian motion, which has a Gaussian PDF, is one such martingale. Clearly, such a property does not hold away from the global optimum, where the PDF would be multi-modal with a peak located at every local extremum detected by the algorithm.

Now, the aim is to obtain a scheme (or rather a family of schemes) to force the coalescence of particles. Consider the update of the jth particle $\mathbf{x}_\tau^{(j)}$ such that coalescence of particles can be enforced. The random scatter around $\mathbf{x}_\tau^{(j)}$ could be quantified by $\delta_\tau(j) = \mathbf{x}_\tau^{(\boldsymbol{\sigma}_1(j))} - \mathbf{x}_\tau^{(j)}$, where $\boldsymbol{\sigma}_1(j)$ denotes a random permutation on the indexing set $\{1, N\}\backslash\{j\}$ based on a uniform measure. The goal is then to drive $\delta_\tau(j) = \mathbf{x}_\tau^{(\boldsymbol{\sigma}_1(j))} - \mathbf{x}_\tau^{(j)}$ to a zero-mean vector Brownian increment $\Delta\eta_\tau^c$ with intensity matrix ρ_τ^c (typically assumed to be diagonal with the entries chosen uniformly for all j). In this case, ‘coalescence’ then implies that, in the limit of the noise intensity in $\delta_\tau(j)$ approaching zero, all the particles would tend to coalesce into a single location at the global extremum. Thus, similar to the innovation $\widehat{f}_\tau - f(\mathbf{x}_\tau)$ on the left-hand side of equation (2.3), one treats $\mathbf{x}_\tau^{(\boldsymbol{\sigma}_1(j))} - \mathbf{x}_\tau^{(j)}$ as yet another innovation process so that, upon convergence, the identity $\mathbf{x}_\tau^{(\boldsymbol{\sigma}_1(j))} - \mathbf{x}_\tau^{(j)} = \Delta\eta_\tau^c$ holds almost surely. Here, $\Delta\eta_\tau^c$ is responsible for an additional layer of randomness that gives every particle $\mathbf{x}_\tau^{(j)}$ the structure of a stochastic process. Accordingly, the extremal cost filtration $\mathcal{N}_\tau$ must be suitably expanded/modified to include the sub-filtration generated by $\Delta\eta_s^c$ for $s \le \tau$. If the coalescence innovation, in the form as indicated above, is included within our search process, equation (3.1) must be modified as

$$\mathbf{x}_i^{(j)} = \tilde{\mathbf{x}}_i^{(j)} + \tilde{\mathbf{D}}_i^{(j)} \quad \text{or} \quad \mathbf{u}_i^{(j)} = \tilde{\mathbf{D}}_i^{(j)}, \tag{3.2}$$

where

$$\tilde{\mathbf{D}}_i^{(j)} := \beta_i \tilde{\mathbf{G}}_i \tilde{\mathbf{I}}_i^{(j)}$$
$$\tilde{\mathbf{I}}_i^{(j)} := \begin{Bmatrix} c\widehat{f}_i - \tilde{f}_i^{(j)} \\ \tilde{\mathbf{x}}_i^{(\sigma_1(j))} - \tilde{\mathbf{x}}_i^{(j)} \end{Bmatrix}. \tag{3.3}$$

Recall that over-tildes indicate either the predicted particles or functions evaluated using the predicted particles, as appropriate. With a convenient abuse, the same notation for the gain-like update coefficient matrix $\tilde{\mathbf{G}}_i$ is retained in equation (3.3) (used earlier in the local update equation (2.25)). $\tilde{\mathbf{G}}_i$ may now be computed as

$$\tilde{\mathbf{G}}_i := \frac{1}{N} \begin{Bmatrix} c(\tilde{\mathbf{X}}_i - \hat{\tilde{\mathbf{X}}}_i)(\tilde{\mathbf{F}}_i^{\mathrm{T}} \tau_i - \hat{\tilde{\mathbf{F}}}_{i-1}^{\mathrm{T}} \tau_{i-1} - \Delta \hat{\tilde{\mathbf{F}}}_i^{\mathrm{T}} \tau_i) \\ +(\hat{\tilde{\mathbf{X}}}_i \tau_i - \hat{\tilde{\mathbf{X}}}_{i-1} \tau_{i-1})(\tilde{\mathbf{F}}_i - \hat{\tilde{\mathbf{F}}}_i)^{\mathrm{T}} \end{Bmatrix} \begin{Bmatrix} c\alpha \dfrac{1}{N-1}((\widehat{\mathbf{F}}_i - \tilde{\mathbf{F}}_i) - (\widehat{\mathbf{F}}_i - \hat{\tilde{\mathbf{F}}}_i)) \\ \cdot((\widehat{\mathbf{F}}_i - \tilde{\mathbf{F}}_i) - (\widehat{\mathbf{F}}_i - \hat{\tilde{\mathbf{F}}}_i))^{\mathrm{T}} + (1-\alpha)\gamma_i \gamma_i^{\mathrm{T}} \end{Bmatrix}^{-1} \tag{3.4}$$

and

$$\gamma_i \gamma_i^{\mathrm{T}} := \begin{bmatrix} \rho_i \rho_i^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0} & \rho_i^c (\rho_i^c)^{\mathrm{T}} \end{bmatrix}. \tag{3.5}$$

Here, the jth column of $\tilde{\mathbf{F}}_i$ is given as $\begin{Bmatrix} \widehat{f}_i - \tilde{f}_i^{(j)} \\ \tilde{\mathbf{x}}_i^{\sigma_1(j)} - \tilde{\mathbf{x}}_i^{(j)} \end{Bmatrix}$. Let a be a positive real number and define the nearest integer smaller than a by $\lfloor a \rfloor$. Then the integer valued perturbation parameter for the local extremization cum coalescence step may be identified as $l = \lfloor \| [\gamma_i \gamma_i^{\mathrm{T}}]^{-1} \| \rfloor$.

A real strength of the coalescence step is in the non-unique choice of the innovation vector—a feature that enables one to design powerful global search schemes. Indeed, one may also borrow some of the basic concepts from different existing global optimization schemes and adapt them within the present coalescence framework. For example, the personal and global information used in the PSO may be incorporated here by constructing the innovation as given below:

$$\tilde{\mathbf{I}}_i^{(j)} = \begin{Bmatrix} {}^p\tilde{\mathbf{x}}_i^{(j)} - \tilde{\mathbf{x}}_i^{(j)} \\ {}^g\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i^{(j)} \end{Bmatrix}. \tag{3.6}$$

Here ${}^p\tilde{\mathbf{x}}_i^{(j)}$ is the personal best location corresponding to the jth particle along its evolution history until τ_i. ${}^g\tilde{\mathbf{x}}_i$ is the available best location among all the particles in the population (ensemble) until τ_i, i.e. during the same evolution history. In contrast to the PSO, the gain-type coefficient matrix here is founded on a sound probabilistic basis and hence its matrix structure enables iteration-dependent differential weighting of the scalar components of the innovation vector, thereby yielding faster convergence without losing the exploratory efficacy of the global search. Indeed, it is well recognized that the use of three parameters (e.g. the cognitive and social learning factors and the inertia weight) in the original PSO may lead to solutions that sensitively depend upon the choice of these parameters. Even though it is possible to incorporate within the current set-up, the basic ideas behind some of the augmented PSO schemes, which attempt at removing some of the shortfalls of the original PSO [29], such a detailed exercise is kept outside the ambit of the current work. While we do not provide a proof here, the convergence and uniqueness of the iterative increment through this step, for a non-decreasing sequence of l converging to a limit point l^* which may be large, could be shown based on the seminal work of Stroock & Varadhan [20] on martingale problems.

3.2. Operator **T2**: scrambling and relaxation

Similar to **T1**, the operator **T2** also corresponds to random perturbations of the particles in the population. Identifying the gain-like coefficient matrix in the update equation (3.1) as a derivative-free stochastic counterpart to the Frechet derivative, the term $\tilde{\mathbf{C}}_i^{(j)}$ may be considered the equivalent of a directional derivative term responsible for updating the jth particle. Consequently, around any extremum, the $L^2(P)$ norm $\|\tilde{\mathbf{C}}_i^{(j)}\|$ is likely to be small. This has the effect of rendering further updates of the *j*th particle small, leading to a possible stalling of the optimization scheme. In order to move out of these local traps, a possible way is to swap the gain-weighted directional information for the *j*th particle with that of another randomly selected one. This form of random perturbation at τ_i may be implemented by replacing the update equation, e.g. equation (3.2), by any one of the following two perturbed equations:

$$\mathbf{x}_i^{(j)} = \tilde{\mathbf{x}}_i^{(j)} + \tilde{\mathbf{D}}_i^{\boldsymbol{\sigma}_2(j)} \tag{3.7}$$

or

$$\mathbf{x}_i^{(j)} = \tilde{\mathbf{x}}_i^{\boldsymbol{\sigma}_2(j)} + \tilde{\mathbf{D}}_i^{(j)}, \tag{3.8}$$

where $\tilde{\mathbf{D}}_i^{(j)}$ is update vector originally computed for the *j*th particle via equation (3.2) and $\boldsymbol{\sigma}_2$ a random permutation on the integer set $\{1, N\}$. Out of the two perturbed equations above to implement scrambling, equation (3.8) is presently adopted. Formally, this perturbation may be described by a probability kernel p_l on $[1, N] \times [1, N]$ such that: $\sum_{i\in[1,N]} p_l(i,j) = 1 \forall\ j \in [1, N]$. Clearly, as $l \to \infty$ (or as $\tau \to \infty$, i.e. as convergence to the global optimum takes place), the matrix $p_l(i,j)$ should ideally approach the identity matrix, i.e. $p_l(i,j) \to \delta_{ij}$, where δ_{ij} is the Kronecker delta.

By borrowing a basic idea from the DE and probably at the cost of a somewhat slower convergence for a class of problems, a more effective modification of the above scrambling strategy may be contemplated with a view to ensuring that the particles explore even more in the search space. Noting that every particle is an *n*-dimensional vector, one may execute the scrambling operation separately (*n* times) for individual scalar components of the particles instead of swapping the update terms for the particles as a whole. This may further be followed up by a so-called relaxation strategy, wherein the resulting modification is accepted with some probability $\gamma < 1$. This improvement, enabling element crossovers across different particles, allows the particles to assume larger variations and prompt them to explore more. The resulting update equation is then given by

$$x_i^{m,(j)} = \tilde{x}_i^{m,\boldsymbol{\sigma}_2(j)} + \tilde{D}_i^{m,(j)}, \quad m \in [1, n], \tag{3.9}$$

where, for instance, $x_i^{m,(j)}$ denotes the updated *m*th scalar component of the *j*th particle at τ_i. Since, upon convergence (and possibly owing to the coalescence step, if applied), all particles crowd around the peak of a unimodal PDF with progressing iterations, directional scrambling with relaxation across such converged particles should not, in any way, affect the numerical accuracy of the estimated global extremum. Hence, in practical implementations of our scheme, $p_l(i,j)$ need not strictly approach the identity matrix for large *l*.

3.3. Operator **T3**: selection

Use of diffusion-based random perturbations during exploration might sometimes result in 'bad' candidates. This may necessitate a selection step wherein candidates for the next iteration (say, the *i*th iteration) are chosen based on some selection criteria quantified by a selection or fitness function $g(\boldsymbol{\upsilon}|\mathbf{x}_{i-1})$, $\boldsymbol{\upsilon} \in \Omega$. A general construction of the function, which is a Markov transition kernel on Ω and is conditioned on the ensemble of particles at the last iteration, should satisfy the following properties:

(i) $g(\boldsymbol{\upsilon} = \mathbf{x}_i^{(j)} | \mathbf{x}_{i-1} = \mathbf{x}_{i-1}^{(k)}) = 0$ if $j \neq k; \forall\ j, k \in [1, N]$,

(ii) $g(\boldsymbol{\upsilon} = \mathbf{x}_i^{(j)} | \mathbf{x}_{i-1} = \mathbf{x}_{i-1}^{(j)}, f(\mathbf{x}_i^{(j)}) \geq f(\mathbf{x}_{i-1}^{(j)})) = \varsigma \forall\ j \in [1, N]$, where $\varsigma \in (0, 1]$, and

(iii) $g(\boldsymbol{\upsilon} = \mathbf{x}_{i-1}^{(j)} | \mathbf{x}_{i-1} = \mathbf{x}_{i-1}^{(j)}, f(\mathbf{x}_i^{(j)}) < f(\mathbf{x}_{i-1}^{(j)})) = \varsigma$.

The updated *j*th particle appearing in the above clauses is computed using equation (3.7) (or equation (3.8)), which may incorporate different combinations of the three operations, e.g. local extremization, coalescence and scrambling-cum-relaxation. For the operator **T3**, the integer-valued inverse-perturbation parameter *l* may be identified with $l = \lfloor 1/(1 - \varsigma) \rfloor$. In the current numerical implementations of COMBEO as described in §4, ς=1 is consistently adopted. This corresponds to *l* being infinity across all iterations and implies that the selection procedure is deterministic.

This work has primarily been aimed at the proposal for a class of new evolutionary optimization schemes and a verification of its performance purely through the numerical route. Accordingly, a detailed convergence analysis of the asymptotic dynamics, based on a combination of the martingale theory of Stroock–Varadhan and the random perturbation theory of Freidlin–Wentzell, will be considered in a separate study.

4. Algorithm development

The local search and random perturbation approaches, as described in §§2 and 3, merely provide a set of general tools whose different combinations could lead to different evolutionary schemes for optimization. For instance, incorporation of the innovation term in equation (2.3) yields a greedy algorithm that, despite possibly faster convergence, is likely to be a poor performer in the global search. On the other hand, to facilitate a more exhaustive global search at the cost of a substantively slower convergence, only a PSO-type innovation as in equation (3.6) could be adopted. For better clarity and a more objective assessment of the strengths and demerits of different search tools, three pseudo-codes are presented in this section. A more ambitious alternative would have been to combine all the presented ideas in a single pseudo-code that could automatically offer a right mix of convergence speed with exploratory efficiency depending on the nature of the problem at hand. This exercise however needs a non-trivial extension of the current work and hence constitutes an interesting future problem at this stage. *pseudo-code 1* uses the innovation term of equation (2.3) for a greedy local search while employing scrambling of particles as a whole (not element wise) for the global search. Importantly, it may be noted that while the prediction step of equation (2.1) appears to be helpful in exploration, this step is not practically useful with our schemes, especially as it does not exploit any directional information and as more efficient tools for explorations have already been laid out. Hence, given a random initial scatter to the particles provided at the beginning of the iterations, the prediction step has been completely eliminated from all subsequent iterations. This would typically mean that the number of evaluations of the objective functional would be reduced by half. Note that, as there is no prediction step, the over-tildes in the notations of the variables are henceforth removed.

COMBEO *pseudo-code 1*

1. Discretize the τ-axis, say $[\tau_{\min}, \tau_{\max}]$, using a partition $\{\tau_0 = \tau_{\min}, \tau_1, \ldots, \tau_M = \tau_{\max}\}$ such that $\tau_0 < \cdots < \tau_M$ and $\tau_i - \tau_{i-1} = \Delta\tau_i$ $(= \Delta\tau := 1/M$ if a uniform step-size is chosen for $i = 0, \ldots, M-1)$. Assign $\tau_0 = 1$ and adopt $\Delta\tau$ small (approx. 10^{-7}). Choose an ensemble size N.
2. From the domain of definition randomly generate, following a uniform distribution, the ensemble of initial particles (the initial population) $\{\mathbf{x}_0^{(j)}\}_{j=1}^{N}$ for the solution vector. For each discrete τ_i, $i = 1, \ldots, M-1$, execute the following steps.
3. *Additive update*

   Choose $\alpha \in (0, 1)$; a typical prescription could be $\alpha \approx 0.8$, even though the method generally performs well for other values in the interval indicated. Update each particle as

   $$\mathbf{x}_i^{(j)} = \mathbf{x}_{i-1}^{\boldsymbol{\sigma}_2(j)} + \mathbf{D}_i^{(j)}, \quad j = 1, \ldots, N,$$

   where $\mathbf{D}_i^{(j)}$ is the ith correction vector. $\boldsymbol{\sigma}_2(j)$ is the jth random permutation based on a uniformly distributed probability measure over the integer set$\{1, \ldots, N\}$. For convenience, the expression for $\tilde{\mathbf{D}}_i^{(j)}$ is reproduced below:

   $$\mathbf{D}_i^{(j)} := \beta_i \mathbf{G}_i \mathbf{I}_i^{(j)}, \quad j = 1, \ldots, N$$

   $$\mathbf{I}_i^{(j)} := \begin{Bmatrix} \widehat{f}_{i-1} - f_{i-1}^{(j)} \\ \mathbf{x}_{i-1}^{\boldsymbol{\sigma}_1(j)} - \mathbf{x}_{i-1}^{(j)} \end{Bmatrix},$$

   $\beta_i = \hat{\beta}_i \zeta_i$ ($\hat{\beta}_i$ is a scalar constant and ζ_i a uniform random number between 0 and 1). $\boldsymbol{\sigma}_1(j)$ is defined as the jth candidate from another (independent) random permutation of the integer set $\{1, \ldots, N\}$. Finally,

   $$\mathbf{G}_i := \frac{1}{N} \begin{Bmatrix} c(\mathbf{X}_{i-1} - \hat{\mathbf{X}}_{i-1})({\mathbf{F}_{i-1}}^{\mathrm{T}} \tau_i - \hat{\mathbf{F}}_{i-1}^{\mathrm{T}} \tau_{i-1} - \Delta\hat{\mathbf{F}}_{i-1}^{\mathrm{T}} \tau_i) \\ +(\hat{\mathbf{X}}_{i-1}\tau_i - \hat{\mathbf{X}}_{i-1}\tau_{i-1})(\mathbf{F}_{i-1} - \hat{\mathbf{F}}_{i-1})^{\mathrm{T}} \end{Bmatrix} \times \begin{Bmatrix} c\alpha \frac{1}{N-1}((\widehat{\mathbf{F}}_{i-1} - \mathbf{F}_{i-1}) - (\widehat{\mathbf{F}}_{i-1} - \hat{\mathbf{F}}_{i-1})) \\ ((\widehat{\mathbf{F}}_{i-1} - \mathbf{F}_{i-1}) - (\widehat{\mathbf{F}}_{i-1} - \hat{\mathbf{F}}_{i-1}))^{\mathrm{T}} + (1-\alpha)\boldsymbol{\gamma}_i \boldsymbol{\gamma}_i^{\mathrm{T}} \end{Bmatrix}^{-1}.$$

4. If $f(\mathbf{x}_i^{(j)}) \geq f(\mathbf{x}_{i-1}^{(j)})$, $j = 1, \ldots, N$, then retain $\mathbf{x}_i^{(j)}$ as the updated particle; else set $\mathbf{x}_i^{(j)} = \mathbf{x}_{i-1}^{(j)}$.
5. If $i < M$, go to step 3 with $i = i + 1$, else terminate the algorithm and report $(1/N)\sum_{j=1}^{N} \mathbf{x}_i^{(j)}$ as an estimate for the global optimum.

In the next pseudo-code, the DE-like scalar element-wise scrambling scheme for particles is implemented and the resulting modification accepted with some relaxation criteria. The greedy local search based on the innovation term $\widehat{f}_{i-1} - f_{i-1}^{(j)}$ is not included in this pseudo-code as it might negatively bear on the global exploration. It may be possible to modify the pseudo-code to augment the innovation vector with additional innovation terms like $\widehat{f}_{i-1} - f_{i-1}^{(j)}$ after properly weighing the different terms. However, such a variation is presently not attempted.

COMBEO *pseudo-code 2*

1. Replicate steps 1 and 2 from *pseudo-code 1*.
2. *Additive update*

   Choose $\alpha \in (0, 1)$ as in *pseudo-code 1* and update each particle as described below. Draw a random number r uniformly from the set $\{1, \ldots, n\}$. As in *pseudo-code 1*, define $\sigma_2(j)$ as the jth candidate from a uniformly random permutation over the integer set $\{1, \ldots, N\}$.

   Initialize: $\mathbf{x}_i^{(j)} = \mathbf{x}_{i-1}^{\sigma_2(j)}, j = 1, \ldots, N$.

   If $\zeta_i < c_i$ ($0 < c_i \leq 1$ is a scalar chosen by the end-user and ζ_i an independent uniform random variable in $[0, 1]$), then

   $$x_i^{r,(j)} = x_{i-1}^{r,\sigma_2(j)} + D_i^{r,(j)}, \quad j = 1, \ldots, N,$$

   with $\mathbf{D}_i^{(j)}$ as in *pseudo-code 1* and $\mathbf{I}_i^{(j)} := \{\mathbf{x}_{i-1}^{\sigma_1(j)} - \mathbf{x}_{i-1}^{(j)}\}$, where $\sigma_1(j)$ is a uniformly random permutation on the index set $\{1, N\} \backslash \{j\}$;

   else

   $$x_i^{m,(j)} = x_{i-1}^{m,\sigma_2(j)}.$$

   Set $r \to r + 1$. Run the loop on r till it reaches n.
3. Replicate steps 4 and 5 from *pseudo-code 1*.

By using the available personal and global best information as in the PSO, one may propose yet another global optimization pseudo-code, as suggested earlier while discussing the coalescence step. Thus, the search here is based purely on a martingale problem without requiring a random perturbation, especially the scrambling step. This is an interesting proposal in the sense that the search for the global optimum is attempted by just constructing an innovation vector and driving the same to a zero-mean martingale. In the previous two algorithms (*pseudo-codes 1* and *2*), the update terms were not constructed using past information beyond one step. In *pseudo-code 3* presented below, the update term is however based on the evolution history until the last iteration in order to explore the search space more exhaustively. Without a loss of generality, the specific algorithm given below is for a global maximization problem.

COMBEO *pseudo-code 3*

1. Same as in *pseudo-code 1*.
2. Generate the initial population $\{\mathbf{x}_0^{(j)}\}_{j=1}^{N}$.

   Initialize: $\vartheta_{\max} = 1$ and $\vartheta_{\min} = 0.1$.

   Initialize the local and global best particles ${}^{p}\mathbf{x}_0^{(j)} = \mathbf{x}_0^{(j)}$, $j = 1, \ldots, N$ and ${}^{g}\mathbf{x}_0 = \arg\max\{f(\mathbf{x}_0^{(1)}), \ldots, f(\mathbf{x}_0^{(N)})\}$. Initialize the correction term $\mathbf{D}_0^{(j)} = \mathbf{0}$, $j = 1, \ldots, N$. For each discrete τ_i, $i = 1, \ldots, M - 1$, execute the following steps.
3. *Additive update*

   Choose $\alpha \in (0, 1)$.

   Compute $\vartheta_i = \vartheta_{\max} - (\vartheta_{\max} - \vartheta_{\min})(i/M)$.

   Update each particle as:

   $$\mathbf{x}_i^{(j)} = \mathbf{x}_{i-1}^{(j)} + \mathbf{D}_i^{(j)}, \quad j = 1, \ldots, N.$$

   $\mathbf{D}_i^{(j)}$ is the jth update vector. The expression for $\mathbf{D}_i^{(j)}$ is also reproduced below:

   $$\mathbf{D}_i^{(j)} := \vartheta_i \mathbf{D}_{i-1}^{(j)} + \beta_i \mathbf{G}_i \mathbf{I}_i^{(j)}, \quad j = 1, \ldots, N$$

   $$\tilde{\mathbf{I}}_i^{(j)} = \begin{Bmatrix} \tilde{\mathbf{x}}_i^{p(j)} - \tilde{\mathbf{x}}_i^{(j)} \\ \tilde{\mathbf{x}}_i^{g} - \tilde{\mathbf{x}}_i^{(j)} \end{Bmatrix}.$$

   $\beta_i = \hat{\beta}_i \zeta_i$ ($\hat{\beta}_i$ is a scalar constant and ζ_i a uniform random number in $[0, 1]$).

4. Update the personal and global information as:
   if $f(\mathbf{x}_i^{(j)}) \geq f(\mathbf{x}_{i-1}^{(j)})$, $j = 1, \ldots, N$, then retain ${}^{p}\mathbf{x}_i^{(j)} = \mathbf{x}_i^{(j)}$ as the updated particle;
   if $\max\{f(\mathbf{x}_i^{(1)}), \ldots, f(\mathbf{x}_i^{(N)})\} \geq f({}^{g}\mathbf{x}_{i-1})$ then ${}^{g}\mathbf{x}_i = \arg\max\{f(\mathbf{x}_i^{(1)}), \ldots, f(\mathbf{x}_i^{(N)})\}$;
   else ${}^{g}\mathbf{x}_i = {}^{g}\mathbf{x}_{i-1}$.

Before concluding this section, a word of caution regarding the claims made on a relatively superior performance of the present approach vis-a-vis a few others, e.g. the DE and the PSO, should be in order. Optimization schemes like the DE or the PSO have been extensively improved over the last 15 years or so and it would be wrong to qualify the current method (as captured through the pseudo-codes above) as being the most competent. Nevertheless, the inherent flexibility of our proposal should permit the incorporation of the basic concepts behind some of the recent improvements in the DE or the PSO [30–33], thereby rendering the latter even more effective. Such details however need to be worked out and do not form part of this study.

5. Numerical illustrations

5.1. Benchmark problems

While the efficiency of a global optimization strategy is largely dependent on its ability to search all the promising search regions, the complexity level of a given optimization problem could exponentially increase with increasing number of design variables. As an example, even though Rosenbrock's function is unimodal in two dimensions ($n = 2$), it is highly multimodal in still higher dimensional search spaces. Hence, a global optimization scheme, which works successfully for lower dimensional problems, may very well fail as the dimension of the problem increases. Moreover, the level of difficulty in finding the global optimum may also be dependent on the specific characteristics of the objective function, which goes to explain the varied degree of difficulty in solving different optimization problems of the same dimension. If the problem is separable or partially separable, i.e. if the objective function can be additively split in terms of component functions, each of which is expressible in terms of just one element or a small subset of elements of the design variable vector $\mathbf{x}$, the original problem may actually be decoupled into a set of sub-problems. Each sub-problem, involving one scalar variable or a small set of scalar variables, may be solved separately, e.g. by a simple line search for one scalar variable. Depending on the degree of complexity, optimization problems could be categorized as separable, m-non-separable and non-separable [34]. In between the two extreme cases, i.e. separable and non-separable, are the m-non-separable functionals (m being the maximum number of scalar design variables appearing in the descriptions of the component functions) that correspond to partially separable problems. The nomenclature *separability* bears a similar meaning as *epistasis* in biology. In this work, in testing the effectiveness of the proposed schemes to some extent, the following problems have been considered.

I. Separable functions:
   (a) $F1$: shifted elliptic function
   (b) $F2$: shifted Rastrigin's function
   (c) $F3$: shifted Ackley's function

II. Single-group m-non-separable functions:
   (d) $F4$: single-group shifted and m-rotated elliptic function
   (e) $F5$: single-group shifted and m-rotated Rastrigin's function
   (f) $F6$: single-group shifted m-dimensional Schwefel's Problem 1.2
   (g) $F7$: single-group shifted m-dimensional Rosenbrock's function

III. $n/2m$-group m-non-separable functions:
   (h) $F8$: $n/2m$-group shifted and m-rotated Rastrigin's function
   (i) $F9$: $n/2m$-group shifted and m-rotated Schwefel's Problem 1.2

IV. n/m-group m-non-separable functions:
   (j) $F10$: n/m-group shifted and m-rotated Schwefel's Problem 1.2

V. Non-separable functions:
   (k) $F11$: shifted Schwefel's Problem 1.2

The above functions are constructed using the following basic functions.

$B1$. The sphere function: $F_{\text{sphere}}(\mathbf{x}) = \sum_{j=1}^{n} {(x^j)}^2$

$B2$. The elliptic function: $F_{\text{elliptic}}(\mathbf{x}) = \sum_{j=1}^{n} [{(10^6)}^{(j-1)/(n-1)} {(x^j)}^2]$

$B3$. The rotated elliptic function: $F_{\text{rot\_elliptic}}(\mathbf{x}) = F_{\text{elliptic}}(\mathbf{z})$, $\mathbf{z} = \mathbf{x} * \mathbf{M}$ ($\mathbf{M}$ being an orthogonal matrix)

$B4$. Schwefel's Problem 1.2: $F_{\text{Schwefel}}(\mathbf{x}) = \sum_{k=1}^{n} {(\sum_{j=1}^{k} x^j)}^2$

$B5$. Rosenbrock's function: $F_{\text{Rosenbrock}}(\mathbf{x}) = \sum_{j=1}^{n-1} [100{({(x^j)}^2 - x^{j+1})}^2 + {(x^j - 1)}^2]$

$B6$. Rastrigin's function: $F_{\text{Rastrigin}}(\mathbf{x}) = \sum_{j=1}^{n} [{(x^j)}^2 - 10\cos(2\pi x^j) + 10]$

$B7$. Rotated Rastrigin's function: $F_{\text{rot\_Rastrigin}}(\mathbf{x}) = F_{\text{Rastrigin}}(\mathbf{z})$, $\mathbf{z} = \mathbf{x} * \mathbf{M}$ ($\mathbf{M}$ being a orthogonal matrix)

$B8$. Ackley's function: $F_{\text{Ackley}}(\mathbf{x}) = -20\exp\left(-0.2(1/n)\sqrt{\sum_{j=1}^{n} {(x^j)}^2}\right) - \exp((1/n)\sum_{j=1}^{n} \cos(2\pi x^j)) + 20 + \exp(1)$

$B9$. Rotated Ackley's function: $F_{\text{rot\_Ackley}}(\mathbf{x}) = F_{\text{Ackley}}(\mathbf{z})$, $\mathbf{z} = \mathbf{x} * \mathbf{M}$

Explicit expressions for the functions $F1$–$F11$ are given below. Let $\mathbf{z} = \mathbf{x} - \mathbf{o}$. $\mathbf{o} \in \mathbb{R}^n$ denote a shifted global optimum and σ_n the random permutation of the vector $\{1, \ldots, n\}$.

(a) $F1(\mathbf{x}) = F_{\text{elliptic}}(\mathbf{z})$
(b) $F2(\mathbf{x}) = F_{\text{Rastrigin}}(\mathbf{z})$
(c) $F3(\mathbf{x}) = F_{\text{Ackley}}(\mathbf{z})$
(d) $F4(\mathbf{x}) = F_{\text{rot\_elliptic}}(\mathbf{z}(\sigma_n(1:m))) \times 10^6 + F_{\text{elliptic}}(\mathbf{z}(\sigma_n(m+1:n)))$
(e) $F5(\mathbf{x}) = F_{\text{rot\_Rastrigin}}(\mathbf{z}(\sigma_n(1:m))) \times 10^6 + F_{\text{Rastrigin}}(\mathbf{z}(\sigma_n(m+1:n)))$
(f) $F6(\mathbf{x}) = F_{\text{Schwefel}}(\mathbf{z}(\sigma_n(1:m))) \times 10^6 + F_{\text{sphere}}(\mathbf{z}(\sigma_n(m+1:n)))$
(g) $F7(\mathbf{x}) = F_{\text{Rosenbrock}}(\mathbf{z}(\sigma_n(1:m))) \times 10^6 + F_{\text{sphere}}(\mathbf{z}(\sigma_n(m+1:n)))$
(h) $F8(\mathbf{x}) = \sum_{k=1}^{n/2m} \left[\begin{array}{c} F_{\text{rot\_Rastrigin}}(\mathbf{z}(\sigma_n((k-1)m+km))) \times 10^6 \\ +F_{\text{Rastrigin}}\left(\mathbf{z}\left(\sigma_n\left(\frac{n}{2}+1:n\right)\right)\right) \end{array} \right]$
(i) $F9(\mathbf{x}) = \sum_{k=1}^{n/2m} \left[\begin{array}{c} F_{\text{Schwefel}}(\mathbf{z}(\sigma_n((k-1)m+km))) \times 10^6 \\ +F_{\text{sphere}}\left(\mathbf{z}\left(\sigma_n\left(\frac{n}{2}+1:n\right)\right)\right) \end{array} \right]$
(j) $F10(\mathbf{x}) = \sum_{k=1}^{n/m} F_{\text{Schwefel}}(\mathbf{z}(\sigma_n((k-1)m+km)))$
(k) $F11(\mathbf{x}) = F_{\text{Schwefel}}(\mathbf{z})$.

In solving optimization problems involving functions $F1$–$F11$, the performance of *pseudo-code 2*, as compared with the parent DE, is given in table 1. The population set in both the cases consists of 2000 particles. The scalar parameter c_i, which is similar to the acceptance criteria given by the crossover constant (CR) in DE [3] is taken as 0.1. In implementing DE, CR is also taken as 0.1. For both the schemes, the programme is terminated if the number of iterations exceeds the maximum allowed threshold, which is presently given by it_max $= 1 \times 10^5$. In all the tables, ε denotes an error norm $\leq 10^{-5}$. By the way of investigating the consistent reproducibility of the results reported herein, the proposed optimization scheme (*pseudo-code* 2) and the DE are run five times for $F3$. It is observed that in all the cases *pseudo-code 2* provides the optimum in roughly the same number of iterations (same range) mentioned in table 2, whereas the DE fails to converge in all the runs. In the literature, the above benchmark problems are typically considered to be 'difficult' to solve by most of the existing evolutionary optimization schemes [34].

Indeed, for these problems, the PSO, *pseudo-codes 1* and *3* could not converge to the global optimum within the presently allowed number of iterations. However, in order to demonstrate that *pseudo-codes 1 and 3* could indeed be very accurate with faster convergence for many global optimization problems, these codes are tested with benchmark global optimization problems $B1$–$B9$. While these problems are also challenging in their own right and often used in the literature to assess the performance of optimization algorithms, they do pose a comparatively lesser degree of complexity in reaching the global optimum vis-a-vis $F1$–$F11$. The results, reported in table 2, are also compared against the PSO under a similar environment. In all the cases, the population set is taken to consist of 50 particles. As anticipated, *pseudo-code 3* and especially *pseudo-code 1* show conspicuously faster convergence than the PSO while maintaining the same or higher level of accuracy. In any case, there is clearly scope for improving these two pseudo-codes (*1* and *3*) so that they can be rendered as competitive as

Table 1. Comparative performance of *pseudo-code 2* and DE against objective functions *F*1–*F*11 ($n = 40$). (OF, objective functional; NI, number of iterations; MEN, mean error norm.)

	pseudo-code 2		DE	
OF	NI	MEN	NI	MEN
*F*1	1.69×10^3	ε	2.42×10^4	ε
*F*2	4.68×10^3	ε	2.16×10^4	ε
*F*3	3.73×10^3	ε	$it_{\max}$	20
*F*4	1.34×10^4	ε	$it_{\max}$	1.2×10^4
*F*5	3×10^3	ε	2.37×10^4	ε
*F*6	2.51×10^3	ε	2.12×10^4	ε
*F*7	1.11×10^4	ε	2.81×10^4	ε
*F*8	4.15×10^4	ε	5.84×10^4	ε
*F*9	2.29×10^3	ε	3.62×10^4	ε
*F*10	8.84×10^3	ε	it_max	575
*F*11	it_max	7.7×10^{-4}	it_max	5.78×10^3

Table 2. Comparative performance of *pseudo-code 3*, *pseudo-code 1* and PSO against objective functions *B*1–*B*9 ($n = 40$). (OF, objective functional; NI, number of iterations; MEN, mean error norm.)

	pseudo-code 3 (*pseudo-code 1*)		PSO	
OF	NI	MEN	NI	MEN
*B*1	146 (37)	ε	330	ε
*B*2	172 (55)	ε	353	ε
*B*3	212 (64)	ε	630	ε
*B*4	190 (49)	ε	456	ε
*B*5	154 (35)	ε	280	ε
*B*6	155 (42)	ε	353	ε
*B*7	172 (46)	ε	330	ε
*B*8	195 (61)	ε	485	ε
*B*9	201 (59)	ε	440	ε

pseudo-code 2. In exploring modification over the present algorithms so as to render them more robust and applicable to higher dimensional optimization problems, one possible way, for instance, would be *intelligent* augmentation of multiple adaptive strategies [35]. Moreover, since there are no specific formulae for selecting the parameter values in the present algorithms, parameter control mechanisms within the martingale problem set-up may also have to be suitably devised [35].

In order to gain further understanding on the performance of COMBEO, *pseudo-code* 2 is tested against yet another set of benchmark problems with the cost functionals, herein denoted as $F(I)1$–24, as considered in (http://coco.gforge.inria.fr/doku.php). In evolutionary algorithms, tuning of some of the parameters could be a tricky issue [36,37]. Since the performance of a scheme may crucially depend upon the choice of such parameters [36,37], which are often left to be selected at the user-end, a robust scheme should have the number of such tunable parameters kept minimum. Note that only one tuning parameter (CR) is introduced in the proposed scheme (*pseudo-code 2*). A detailed comparison of the performance of CMA-ES vis-a-vis COMBEO with varying CR values (e.g. 0.1, 0.9 and a number randomly drawn from

Table 3. Comparative performance of *pseudo-code 2* (with varying CR) and CMAES against objective functions $F(I)1$–$F(I)14$ ($n = 20$).

$F(I)$ index	method	*EF(I)E*	error 1	error 2	error 3	error 4
$F(I)1$	*pseudo-code 2*(CR = 0.1)	32 909	8.70×10^{-9}	8.70×10^{-9}	8.70×10^{-9}	8.70×10^{-9}
$F(I)1$	*pseudo-code 2*(CR = rand)	50 909	7.98×10^{-9}	7.98×10^{-9}	7.98×10^{-9}	7.98×10^{-9}
$F(I)1$	*pseudo-code 2*(CR = 0.9)	1 000 000	4.95×10^{-5}	3.24×10^{-5}	5.98×10^{-5}	4.44×10^{-5}
$F(I)1$	CMA-ES	4537	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
$F(I)2$	*pseudo-code 2*(CR = 0.1)	46 818	8.31×10^{-9}	8.31×10^{-9}	8.31×10^{-9}	8.31×10^{-9}
$F(I)2$	*pseudo-code 2*(CR = rand)	70 455	6.67×10^{-9}	6.67×10^{-9}	6.67×10^{-9}	6.67×10^{-9}
$F(I)2$	*pseudo-code 2*(CR = 0.9)	1 000 000	4.15×10^{-1}	3.30×10^{-1}	6.39×10^{-1}	4.09×10^{-1}
$F(I)2$	CMA-ES	21 147	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
$F(I)3$	*pseudo-code 2*(CR = 0.1)	72 909	9.50×10^{-9}	9.50×10^{-9}	9.50×10^{-9}	9.50×10^{-9}
$F(I)3$	*pseudo-code 2*(CR = rand)	1 000 000	5.68×10^{0}	5.32×10^{0}	9.32×10^{0}	5.32×10^{0}
$F(I)3$	*pseudo-code 2*(CR = 0.9)	1 000 000	9.49×10^{1}	8.92×10^{1}	1.07×10^{2}	9.71×10^{1}
$F(I)3$	CMA-ES	7822	2.33×10^{1}	1.99×10^{1}	2.59×10^{1}	2.39×10^{1}
$F(I)4$	*pseudo-code 2*(CR = 0.1)	79 727	6.87×10^{-9}	6.87×10^{-9}	6.87×10^{-9}	6.87×10^{-9}
$F(I)4$	*pseudo-code 2*(CR = rand)	1 000 000	9.32×10^{0}	8.89×10^{0}	1.20×10^{1}	8.89×10^{0}
$F(I)4$	*pseudo-code 2*(CR = 0.9)	1 000 000	1.04×10^{2}	1.01×10^{2}	1.06×10^{2}	1.06×10^{2}
$F(I)4$	CMA-ES	7421	3.62×10^{1}	2.69×10^{1}	5.27×10^{1}	2.89×10^{1}
$F(I)5$	*pseudo-code 2*(CR = 0.1)	4727	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
$F(I)5$	*pseudo-code 2*(CR = rand)	3455	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
$F(I)5$	*pseudo-code 2*(CR = 0.9)	2818	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
$F(I)5$	CMA-ES	340	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
$F(I)6$	*pseudo-code 2*(CR = 0.1)	1 000 000	1.49×10^{-2}	1.42×10^{-2}	2.08×10^{-2}	1.42×10^{-2}
$F(I)6$	*pseudo-code 2*(CR = rand)	1 000 000	2.06×10^{-1}	1.58×10^{-1}	2.65×10^{-1}	1.58×10^{-1}
$F(I)6$	*pseudo-code 2*(CR = 0.9)	1 000 000	5.54×10^{2}	4.62×10^{2}	6.44×10^{2}	5.65×10^{2}
$F(I)6$	CMA-ES	16 332	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
$F(I)7$	*pseudo-code 2*(CR = 0.1)	1 000 000	1.15×10^{0}	1.05×10^{0}	1.50×10^{0}	1.11×10^{0}
$F(I)7$	*pseudo-code 2*(CR = rand)	1 000 000	1.46×10^{0}	1.05×10^{0}	1.82×10^{0}	1.74×10^{0}
$F(I)7$	*pseudo-code 2*(CR = 0.9)	1 000 000	3.97×10^{1}	3.81×10^{1}	4.02×10^{1}	4.02×10^{1}
$F(I)7$	CMA-ES	3126	4.99×10^{0}	4.07×10^{0}	6.84×10^{0}	4.68×10^{0}
$F(I)8$	*pseudo-code 2*(CR = 0.1)	1 000 000	1.20×10^{1}	1.17×10^{1}	1.26×10^{1}	1.17×10^{1}
$F(I)8$	*pseudo-code 2*(CR = rand)	1 000 000	1.52×10^{1}	1.51×10^{1}	1.56×10^{1}	1.51×10^{1}
$F(I)8$	*pseudo-code 2*(CR = 0.9)	1 000 000	1.92×10^{1}	1.88×10^{1}	2.00×10^{1}	1.92×10^{1}
$F(I)8$	CMA-ES	21 876	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
$F(I)9$	*pseudo-code 2*(CR = 0.1)	1 000 000	1.73×10^{1}	1.72×10^{1}	1.73×10^{1}	1.73×10^{1}
$F(I)9$	*pseudo-code 2*(CR = rand)	1 000 000	1.66×10^{1}	1.65×10^{1}	1.67×10^{1}	1.67×10^{1}
$F(I)9$	*pseudo-code 2*(CR = 0.9)	1 000 000	1.83×10^{1}	1.81×10^{1}	1.85×10^{1}	1.84×10^{1}
$F(I)9$	CMA-ES	22 236	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
$F(I)10$	*pseudo-code 2*(CR = 0.1)	1 000 000	2.83×10^{4}	2.65×10^{4}	3.14×10^{4}	2.65×10^{4}
$F(I)10$	*pseudo-code 2*(CR = rand)	1 000 000	1.01×10^{5}	8.36×10^{4}	1.55×10^{5}	8.36×10^{4}

(*Continued*.)

Table 3. (*Continued*.)

F(*I*) index	method	*EF*(*I*)*E*	error 1	error 2	error 3	error 4
F(*I*)10	*pseudo-code 2*(CR = 0.9)	1 000 000	3.79×10^{5}	3.53×10^{5}	6.37×10^{5}	3.53×10^{5}
F(*I*)10	CMA-ES	21 391	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
F(*I*)11	*pseudo-code 2*(CR = 0.1)	1 000 000	6.29×10^{1}	6.29×10^{1}	6.29×10^{1}	6.29×10^{1}
F(*I*)11	*pseudo-code 2*(CR = rand)	1 000 000	1.69×10^{2}	1.44×10^{2}	2.19×10^{2}	1.60×10^{2}
F(*I*)11	*pseudo-code 2*(CR = 0.9)	1 000 000	1.31×10^{2}	1.15×10^{2}	2.23×10^{2}	1.15×10^{2}
F(*I*)11	CMA-ES	17 781	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
F(*I*)12	*pseudo-code 2*(CR = 0.1)	1 000 000	1.13×10^{-1}	2.05×10^{-2}	1.33×10^{-1}	1.33×10^{-1}
F(*I*)12	*pseudo-code 2*(CR = rand)	1 000 000	9.25×10^{-2}	1.52×10^{-2}	3.05×10^{-1}	1.52×10^{-2}
F(*I*)12	*pseudo-code 2*(CR = 0.9)	1 000 000	1.72×10^{3}	1.42×10^{3}	3.27×10^{3}	1.42×10^{3}
F(*I*)12	CMA-ES	36 406	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}
F(*I*)13	*pseudo-code 2*(CR = 0.1)	1 000 000	5.08×10^{-1}	1.60×10^{-1}	1.31×10^{0}	4.58×10^{-1}
F(*I*)13	*pseudo-code 2*(CR = rand)	1 000 000	5.11×10^{-2}	3.30×10^{-2}	8.28×10^{-2}	3.30×10^{-2}
F(*I*)13	*pseudo-code 2*(CR = 0.9)	1 000 000	4.38×10^{1}	4.31×10^{1}	4.38×10^{1}	4.38×10^{1}
F(*I*)13	CMA-ES	8332	2.96×10^{-7}	2.96×10^{-7}	2.96×10^{-7}	2.96×10^{-7}
F(*I*)14	*pseudo-code 2*(CR = 0.1)	1 000 000	2.91×10^{-3}	2.46×10^{-3}	4.67×10^{-3}	2.46×10^{-3}
F(*I*)14	*pseudo-code 2*(CR = rand)	1 000 000	5.81×10^{-3}	4.56×10^{-3}	8.44×10^{-3}	4.56×10^{-3}
F(*I*)14	*pseudo-code 2*(CR = 0.9)	1 000 000	8.22×10^{-2}	8.22×10^{-2}	8.22×10^{-2}	8.22×10^{-2}
F(*I*)14	CMA-ES	36 078	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}	-1.00×10^{-8}

a uniform distribution over 0 and 1) in the context of the new set of benchmark problems as considered in (http://coco.gforge.inria.fr/doku.php) has been undertaken and the results are reported in tables 3 and 4, only for the case of $n = 20$. In order to limit the length of the article, a more detailed comparison for different values of n is provided in the electronic supplementary material. It may be noted that, in order to maintain a fair comparison between COMBEO and CMA-ES, the realized state with the minimum cost functional value in the available ensemble is considered as the solution. In these tables, EFE, error 1, error 2, error 3 and error 4, respectively, denote the average number of function evaluations among independent restarts, mean error, best error, worst error and median error. Tables 3 and 4 and electronic supplementary material, tables SI–SX, also include simulation results via CMA-ES [6]. The source code for CMA-ES is taken from https://www.lri.fr/~hansen/cmaes_inmatlab.html.

As we see from tables 3 and 4 and the ones included in the electronic supplementary material, while COMBEO generally scores in achieving higher accuracy while approaching the global optimum, CMA-ES does have a distinct computational advantage in that it can perform with substantively lower ensemble sizes, N. It would thus be worthwhile to incorporate some of the adaptive features of CMA-ES within COMBEO in a future work.

5.2. An application to quantitative estimation of density variation in high-speed flows through inversion of light travel-time data

Having assessed the proposed global optimization framework through a variety of benchmark problems, we now consider a considerably larger dimensional optimization problem that involves the quantitative estimation of density variation in a high-speed flow obstructed by a blunt-nose aerodynamic vehicle. This example is taken up to demonstrate that, while *pseudo-code 2* often provides an effective search tool for the global extremum of complex objective functionals, a greedier search scheme (as in *pseudo-code 1* or its possible variants) could nevertheless be more appropriate for many practically useful, yet large

Table 4. Comparative performance of *pseudo-code 2* (with varying CR) and CMAES against objective functions *F*(*I*)1–*F*(*I*)14 ($n = 20$).

F(*I*) index	method	*EF(I)E*	error 1	error 2	error 3	error 4
F(*I*)15	*pseudo-code 2*(CR = 0.1)	1 000 000	8.66×10^{1}	8.11×10^{1}	9.36×10^{1}	8.54×10^{1}
F(*I*)15	*pseudo-code 2*(CR = rand)	1 000 000	8.94×10^{1}	8.48×10^{1}	1.05×10^{2}	8.48×10^{1}
F(*I*)15	*pseudo-code 2*(CR = 0.9)	1 000 000	1.22×10^{2}	1.21×10^{2}	1.33×10^{2}	1.21×10^{2}
F(*I*)15	CMA-ES	7657	1.69×10^{1}	1.69×10^{1}	1.69×10^{1}	1.69×10^{1}
F(*I*)16	*pseudo-code 2*(CR = 0.1)	1 000 000	1.40×10^{1}	1.09×10^{1}	1.64×10^{1}	1.41×10^{1}
F(*I*)16	*pseudo-code 2*(CR = rand)	1 000 000	1.55×10^{1}	1.52×10^{1}	1.71×10^{1}	1.52×10^{1}
F(*I*)16	*pseudo-code 2*(CR = 0.9)	1 000 000	1.94×10^{1}	1.83×10^{1}	2.20×10^{1}	1.83×10^{1}
F(*I*)16	CMA-ES	11 041	7.93×10^{-1}	6.31×10^{-1}	2.41×10^{0}	6.31×10^{-1}
F(*I*)17	*pseudo-code 2*(CR = 0.1)	1 000 000	1.18×10^{-2}	8.39×10^{-3}	1.70×10^{-2}	1.20×10^{-2}
F(*I*)17	*pseudo-code 2*(CR = rand)	1 000 000	9.43×10^{-3}	8.35×10^{-3}	1.34×10^{-2}	8.35×10^{-3}
F(*I*)17	*pseudo-code 2*(CR = 0.9)	1 000 000	5.46×10^{-1}	5.03×10^{-1}	6.17×10^{-1}	5.44×10^{-1}
F(*I*)17	CMA-ES	12 505	7.83×10^{-3}	3.43×10^{-3}	1.40×10^{-2}	4.95×10^{-3}
F(*I*)18	*pseudo-code 2*(CR = 0.1)	1 000 000	3.45×10^{-1}	2.93×10^{-1}	4.10×10^{-1}	3.39×10^{-1}
F(*I*)18	*pseudo-code 2*(CR = rand)	1 000 000	1.65×10^{-1}	1.50×10^{-1}	1.90×10^{-1}	1.50×10^{-1}
F(*I*)18	*pseudo-code 2*(CR = 0.9)	1 000 000	3.22×10^{0}	3.18×10^{0}	3.59×10^{0}	3.18×10^{0}
F(*I*)18	CMA-ES	12 964	4.95×10^{-2}	4.12×10^{-2}	7.00×10^{-2}	4.12×10^{-2}
F(*I*)19	*pseudo-code 2*(CR = 0.1)	1 000 000	3.07×10^{0}	2.75×10^{0}	3.30×10^{0}	3.12×10^{0}
F(*I*)19	*pseudo-code 2*(CR = rand)	1 000 000	2.65×10^{0}	2.53×10^{0}	3.89×10^{0}	2.53×10^{0}
F(*I*)19	*pseudo-code 2*(CR = 0.9)	1 000 000	4.29×10^{0}	4.02×10^{0}	4.95×10^{0}	4.02×10^{0}
F(*I*)19	CMA-ES	147 687	1.62×10^{0}	7.59×10^{-1}	4.16×10^{0}	7.59×10^{-1}
F(*I*)20	*pseudo-code 2*(CR = 0.1)	1 000 000	5.07×10^{-2}	4.96×10^{-2}	6.16×10^{-2}	4.96×10^{-2}
F(*I*)20	*pseudo-code 2*(CR = rand)	1 000 000	6.86×10^{-1}	6.43×10^{-1}	6.98×10^{-1}	6.98×10^{-1}
F(*I*)20	*pseudo-code 2*(CR = 0.9)	1 000 000	2.41×10^{0}	2.38×10^{0}	2.44×10^{0}	2.44×10^{0}
F(*I*)20	CMA-ES	5963	1.57×10^{0}	1.39×10^{0}	1.81×10^{0}	1.55×10^{0}
F(*I*)21	*pseudo-code 2*(CR = 0.1)	1 000 000	1.31×10^{-2}	9.34×10^{-4}	1.44×10^{-2}	1.44×10^{-2}
F(*I*)21	*pseudo-code 2*(CR = rand)	87 364	9.12×10^{-9}	9.12×10^{-9}	9.12×10^{-9}	9.12×10^{-9}
F(*I*)21	*pseudo-code 2*(CR = 0.9)	1 000 000	2.07×10^{0}	1.52×10^{0}	2.54×10^{0}	2.51×10^{0}
F(*I*)21	CMA-ES	5113	7.29×10^{0}	6.92×10^{-1}	6.44×10^{1}	1.80×10^{0}
F(*I*)22	*pseudo-code 2*(CR = 0.1)	1 000 000	1.95×10^{0}	1.95×10^{0}	1.95×10^{0}	1.95×10^{0}
F(*I*)22	*pseudo-code 2*(CR = rand)	1 000 000	1.95×10^{0}	1.95×10^{0}	1.95×10^{0}	1.95×10^{0}
F(*I*)22	*pseudo-code 2*(CR = 0.9)	1 000 000	1.96×10^{0}	1.96×10^{0}	1.96×10^{0}	1.96×10^{0}
F(*I*)22	CMA-ES	4753	2.52×10^{0}	6.92×10^{-1}	1.83×10^{1}	6.92×10^{-1}
F(*I*)23	*pseudo-code 2*(CR = 0.1)	1 000 000	1.41×10^{0}	1.12×10^{0}	1.58×10^{0}	1.58×10^{0}
F(*I*)23	*pseudo-code 2*(CR = rand)	1 000 000	1.28×10^{0}	1.16×10^{0}	1.83×10^{0}	1.16×10^{0}
F(*I*)23	*pseudo-code 2*(CR = 0.9)	1 000 000	1.88×10^{0}	1.73×10^{0}	2.34×10^{0}	1.73×10^{0}
F(*I*)23	CMA-ES	3823	2.90×10^{0}	2.64×10^{0}	3.15×10^{0}	3.04×10^{0}
F(*I*)24	*pseudo-code 2*(CR = 0.1)	1 000 000	1.05×10^{2}	1.03×10^{2}	1.12×10^{2}	1.04×10^{2}
F(*I*)24	*pseudo-code 2*(CR = rand)	1 000 000	9.77×10^{1}	9.27×10^{1}	1.12×10^{2}	9.27×10^{1}
F(*I*)24	*pseudo-code 2*(CR = 0.9)	1 000 000	1.30×10^{2}	1.25×10^{2}	1.36×10^{2}	1.30×10^{2}
F(*I*)24	CMA-ES	6175	4.67×10^{1}	3.89×10^{1}	4.96×10^{1}	4.96×10^{1}

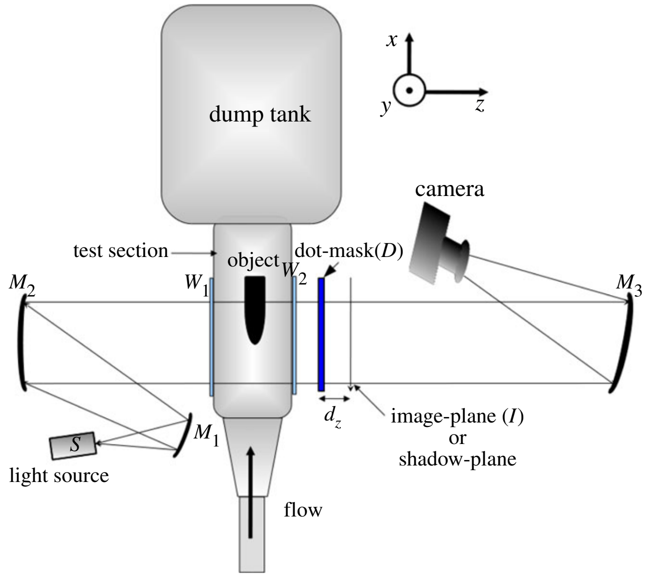

Figure 1. A schematic diagram of the experimental set-up.

dimensional, problems where a scheme like *pseudo-code 2* (or most available global schemes such as the DE or the PSO) might prove computationally prohibitive.

The experimental set-up is shown in figure 1, see [38] for more details. Here, the region of interest (ROI), the flow around the object (in this case, a blunt-nosed missile model), is illuminated by a plane wave. The distorted plane wave trans-illuminates a random dot pattern (RDP). The RDP in effect serves the purpose of the expensive lenslet split Shack–Hartmann sensor. The geometric shadow cast at an adjacent plane is imaged by a high-speed camera. The shadow, in comparison to the original RDP, is space-shifted. Local mean shifts $\boldsymbol{d} = (d_x, d_y)$ and a related quantity, the slopes of the wavefront $\boldsymbol{\theta} = (d_x/d_z, d_y/d_z)$, are estimated by cross-correlating local sub-region of the distorted RDP with the corresponding original and finding the shift of the cross-correlation peak. The estimated local slopes are integrated to find the smooth wavefront Φ [39]. The recovered unwrapped Φ, which carries the distortion owing to its passage through the ROI, is shown in figure 2. The distorted wavefront Φ^i (or equivalently the phase delay) evaluated at the *i*th detector, is related to the refractive index distribution $\varrho(\mathbf{r})$ in the ROI, $\mathbf{r} = (x, y, z)$ being the position vector, through the equation:

$$\Phi^i = \frac{2\pi}{\lambda} \int_0^{L^i} \varrho(\mathbf{r})\, \mathrm{d}s.$$

Here L^i is the length of the *i*th ray and ds is an element from the Fermat's path of the light-ray through $\varrho(\mathbf{r})$, which obeys the Eikonel equation:

$$\frac{\mathrm{d}}{\mathrm{d}s}\left(\varrho \frac{\mathrm{d}\mathbf{r}}{\mathrm{d}s}\right) = \nabla \varrho.$$

Therefore, the equation connecting Φ^i to $\varrho(\mathbf{r})$ is nonlinear. The collection of Φ^i at all the detectors is denoted using the vector $\boldsymbol{\Phi} := \{\Phi^i\}$. The problem of recovery of $\varrho(\mathbf{r})$ from the experimentally measured data, denoted by $\boldsymbol{\Phi}_{\mathrm{exp}}$, obtained from Φ corresponding to the locations of detectors, is posed as an optimization problem which is solved within the proposed framework. The corresponding objective functional may be written as

$$f_{\mathrm{exp}} := \sqrt{(\boldsymbol{\Phi}_{\mathrm{exp}} - \boldsymbol{\Phi})^{\mathrm{T}}(\boldsymbol{\Phi}_{\mathrm{exp}} - \boldsymbol{\Phi})}.$$

However, given the large system dimension ($n = 101^2$) and with the algorithm requiring repetitive numerical inversions for Φ, most available global optimization schemes (e.g. the DE or the PSO) are rendered impracticable using the commonly available computing systems. An acceleration of the computing speed is however possible in the present framework by so modifying the innovation processes as to render the search greedier while retaining some features of the global search, especially the scrambling step. One way could be to start with *pseudo-code 1*. In order to make the algorithm faster, we might replace the innovation used in *pseudo-code 1* by the difference of the objective functional f_{exp}

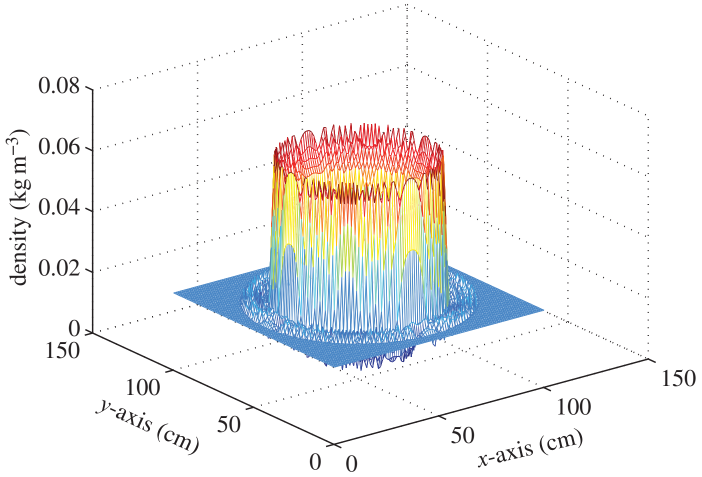

Figure 2. A three-dimensional view of the reconstruction via the proposed framework.

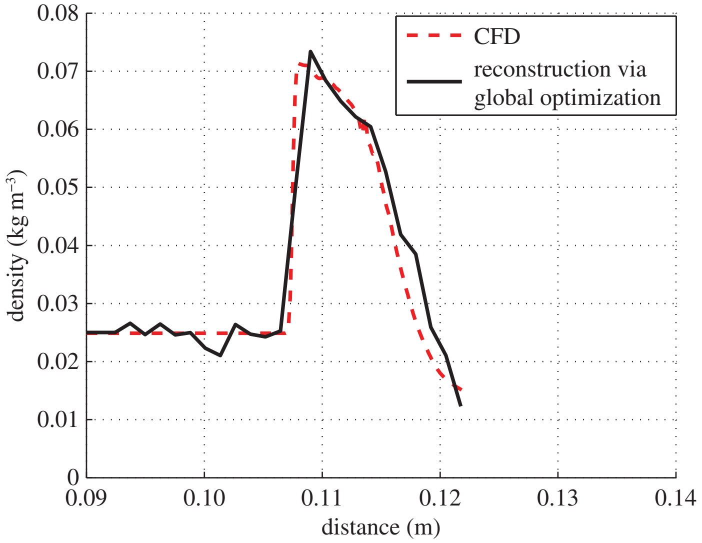

Figure 3. A projection of the reconstruction via the proposed framework.

computed over two succeeding iterations. Unfortunately, this leads to instability in the algorithm owing to the fact that the innovation becomes one-dimensional in contrast to the number of design variables which is of order $O(10^4)$. This may be remedied by splitting the original objective functional into 101^2 individual ones (i.e. one corresponding to each design variable) and thus constructing 101^2 separate scalar innovation processes for the new objective function vector. This enables solving the problem in only a few iterations (as few as just five iterations) with only 30 MC particles. Accordingly, we now redefine $f^2_{\text{exp}} = \sum_{\varsigma=1}^{n} g^{\varsigma}_{\text{exp}}$, where $g^{\varsigma}_{\text{exp}} := (\Phi^{\varsigma}_{\text{exp}} - \Phi^{\varsigma})^2$. The present modification requires the realized innovation $\mathbf{I}_i$ at the ith iteration in *pseudo-code 1* (specifically in step 3) to be given as

$$\mathbf{I}_i := \begin{Bmatrix} g^1_{\text{exp}_{i-1}} \\ \cdots \\ g^n_{\text{exp}_{i-1}} \end{Bmatrix}.$$

It is recalled that this modified algorithm uses only scrambling for effecting the global search. This is an interesting example of the practical advantages in being able to suitably design the innovation process within the present framework. It is worth noting that our efforts at rendering the DE and the PSO greedier by tuning their appropriate algorithmic parameters have failed to make them effective in solving the current problem.

The recovered $\varrho(\mathbf{r})$ is converted into density distribution using the Gladstone–Dale equation. A cross section of the recovered density profile is shown in figures 2 and 3. The above reconstruction is verified (figure 3) by computing the flow density distribution close to the boundary layer by solving Navier–Stoke's equation using the commercially available CFD software FLUENT. The comparison of the density variations near the boundary layer, shown in figure 3, is seen to be quite close.

6. Concluding remarks

Despite the widely reported success and appealing simplicity of the methodological aspects of many existing evolutionary schemes for global optimization, often inspired by biological or social observations, the specific forms of their update scheme in approaching the global extremum are generally not rigorously grounded. While a major aspect of this work is the attempt at finding an underlying logical thread that perhaps relates the basic ideas behind many such update strategies, the notion of a non-uniquely specifiable martingale approach in deriving the update term and its functional similarity with the directional derivative, used in most deterministic optimization schemes, has by far been the most fundamental aspect in this article. Such a directional search, free from the notion of classical derivatives, has indeed been responsible for superior convergence features of the COMBEO family of schemes in contrast to some of the well-known evolutionary methods, e.g. the PSO, the DE and the CMA-ES. An effective global exploration of the search space however requires several random perturbation strategies, of which the coalescence step is treated as part of the martingale problem. It is shown that a variant of the coalescence step may mimic a PSO-like search. A DE-like search, on the other hand, may be organized through a variant of scrambling, which is yet another random perturbation strategy involving directional information swaps. The framework of COMBEO, founded on the theory of stochastic processes, promises to yield highly competitive update schemes in order to meet the conflicting demands of faster convergence and efficacious exploration. As demonstrated via numerical work on a host of higher dimensional benchmark objective functions, the COMBEO schemes show consistently superior performance in comparison to several existing evolutionary methods, prominently the DE and the PSO. The flexibility of the proposed optimization paradigm is also demonstrated by evolving a greedier version of the search scheme and applying the same to the reconstruction of boundary layer density variation in a high-speed flow experiment.

The generality of the framework afforded by COMBEO should motivate further research in improving the specific numerical schemes considered in this work. For instance, the COMBEO algorithms could possibly be rendered computationally more competitive by incorporating the adaptive features of the CMA-ES or some of the many available modifications or augmentations of the DE.

Data accessibility. Parts of the data supporting the claims in the article (i.e. those not provided in the main text) have been included in the electronic supplementary material. The results reported therein have been generated using the resources available in the following links: (i) https://www.lri.fr/~hansen/cmaes_inmatlab.html. (ii) http://coco.gforge.inria.fr/doku.php.

Authors' contributions. The study was conceived of by D.R. and S.S. S.S. and D.R. were also responsible for the development of the reported methodologies as well as the drafting of the manuscript. The numerical results were generated by S.S. R.M.V. was responsible for organizing the experiment reported in §5.2, data collection and the final revision of the manuscript. All authors gave final approval for publication.

Competing interests. We declare we have no competing interests.

Funding. The authors have not received any funding for the work reported in the manuscript.

Acknowledgements. We thank Prof. K. P. J. Reddy, Department of Aerospace Engineering, Indian Institute of Science, for sharing the experimental data with us.

Appendix A. Derivation of the extremal equation

Theorem A.1. *For $\phi \in C_b^2$ the normalized conditional law $\pi_\tau(\phi)$ of the solution process $\boldsymbol{x}_\tau$ satisfies the following extremal equation:*

$$\mathrm{d}\pi_\tau(\phi) = \frac{1}{2}\pi_\tau\left(\sum_{j,k=1}^{n}\sum_{l=1}^{d}\left(\frac{\partial^2\phi}{\partial x^j \partial x^k}\right)g^{jl}g^{kl}\right)\mathrm{d}\tau + (\pi_\tau(\phi h) - \pi_\tau(\phi)\pi_\tau(h))(\mathrm{d}\breve{f}_\tau - \pi_\tau(h)\,\mathrm{d}\tau).$$

Proof. The gain based update equation for a bounded and at least twice continuously differentiable function $\phi_\tau := \phi(\mathbf{x}_\tau)$ of $\mathbf{x}_\tau$ may be arrived at by expanding $\phi(\mathbf{x}_\tau)\Lambda_\tau$, where $\tau \in (\tau_{i-1}, \tau_i]$, using

Ito's formula:

$$\mathrm{d}(\phi_\tau \varLambda_\tau) = \phi_\tau \, \mathrm{d}\varLambda_\tau + \mathrm{d}\phi_\tau \varLambda_\tau + \langle \mathrm{d}\phi_\tau, \, \mathrm{d}\varLambda_\tau \rangle, \tag{A 1}$$

$\langle\cdot\rangle$ denotes the quadratic covariation. A further expansion leads to

$$\mathrm{d}(\phi_\tau \varLambda_\tau) = \phi_\tau \varLambda_\tau h_\tau \, \mathrm{d}\breve{f}_\tau + \varLambda_\tau \phi'^{\mathrm{T}}_\tau \, \mathrm{d}\mathbf{x}_\tau + \tfrac{1}{2} \varLambda_\tau \langle \mathrm{d}\mathbf{x}_\tau, \phi''_\tau \, \mathrm{d}\mathbf{x}_\tau \rangle. \tag{A 2}$$

By explicitly writing out the term $\langle \mathrm{d}\mathbf{x}_\tau, \phi''_\tau \, \mathrm{d}\mathbf{x}_\tau \rangle$, we get

$$\mathrm{d}(\phi_\tau \varLambda_\tau) = \phi_\tau \varLambda_\tau h_\tau \, \mathrm{d}\breve{f}_\tau + \varLambda_\tau {\phi'_\tau}^{\mathrm{T}} \, \mathrm{d}\mathbf{x}_\tau + \frac{1}{2} \sum_{j,k=1}^{n} \sum_{l=1}^{d} \left(\frac{\partial^2 \phi}{\partial x^j \partial x^k} \right)_\tau g^{jl} g^{kl} \, \mathrm{d}\tau. \tag{A 3}$$

■

In deriving the above equation, equation (2.1) is made use of. The incremental form in equation (A 3) may be given the following integral representation:

$$\phi_\tau \varLambda_\tau = \phi_{i-1} \varLambda_{i-1} + \int_{\tau_{i-1}}^{\tau} \varLambda_s \phi_s h_s \mathrm{d}\breve{f}_s + \int_{\tau_{i-1}}^{\tau} \varLambda_s \left(\phi'_s \, \mathrm{d}\boldsymbol{\xi}_s + \frac{1}{2} \sum_{j,k=1}^{n} \sum_{l=1}^{d} \left(\frac{\partial^2 \phi}{\partial x^j \partial x^k} \right)_s g^{jl} g^{kl} \right) \mathrm{d}s. \tag{A 4}$$

Taking conditional expectation with respect to $\mathcal{N}_\tau$ under Q, we get

$$\begin{aligned} E_Q(\phi_\tau \varLambda_\tau | \mathcal{N}_\tau) = E_Q(\phi_{i-1} \varLambda_{i-1} | \mathcal{N}_\tau) &+ E_Q \left(\left(\int_{\tau_{i-1}}^{\tau} \varLambda_s \phi_s h_s \, \mathrm{d}\breve{f_s} \right) \middle| \mathcal{N}_\tau \right) + E_Q \left(\left(\int_{\tau_{i-1}}^{\tau} \varLambda_s \phi'_s \, \mathrm{d}\boldsymbol{\xi}_s \right) \middle| \mathcal{N}_\tau \right) \\ &+ \frac{1}{2} E_Q \left(\left(\int_{\tau_{i-1}}^{\tau} \sum_{j,k=1}^{n} \sum_{l=1}^{d} \left(\frac{\partial^2 \phi}{\partial x^j \partial x^k} \right)_s g^{jl} g^{kl} \, \mathrm{d}s \right) \middle| \mathcal{N}_\tau \right). \end{aligned} \tag{A 5}$$

Using Fubini's theorem,

$$\begin{aligned} E_Q(\phi_\tau \varLambda_\tau | \mathcal{N}_\tau) = E_Q(\phi_{i-1} \varLambda_{i-1} | \mathcal{N}_\tau) &+ \int_{\tau_{i-1}}^{\tau} E_Q(\varLambda_s \phi_s h_s | \mathcal{N}_s) \, \mathrm{d}\breve{f}_s \\ &+ \int_{\tau_{i-1}}^{\tau} E_Q(\varLambda_s \phi'_s | \mathcal{N}_s) \, \mathrm{d}\boldsymbol{\xi}_s + \frac{1}{2} \int_{\tau_{i-1}}^{\tau} E_Q \left(\sum_{j,k=1}^{n} \sum_{l=1}^{d} \left(\frac{\partial^2 \phi}{\partial x^j \partial x^k} \right)_s g^{jl} g^{kl} \middle| \mathcal{N}_s \right) \mathrm{d}s. \end{aligned} \tag{A 6}$$

Noting that $\int_{\tau_{i-1}}^{\tau} E_Q((\varLambda_s \phi'_s) | \mathcal{N}_s) \, \mathrm{d}\boldsymbol{\xi}_s = 0$ and for notational convenience denoting the unnormalized conditional expectation operator, $E_Q((\centerdot)_\tau \varLambda_\tau | \mathcal{N}_s)$ as $\varTheta_\tau(\centerdot)$ we arrive at the following equation:

$$\varTheta_\tau(\phi) = \varTheta_{i-1}(\phi) + \int_{\tau_{i-1}}^{\tau} \varTheta_s(\phi h) \, \mathrm{d}\breve{f}_s + \frac{1}{2} \int_{\tau_{i-1}}^{\tau} \varTheta_s \left(\sum_{j,k=1}^{n} \sum_{l=1}^{d} \left(\frac{\partial^2 \phi}{\partial x^j \partial x^k} \right)_s g^{jl} g^{kl} \right) \mathrm{d}s. \tag{A 7}$$

An incremental representation of equation (A 7) may be given as

$$\mathrm{d}\varTheta_\tau(\phi) = \varTheta_\tau(\phi h) \, \mathrm{d}\breve{f}_\tau + \frac{1}{2} \varTheta_\tau \left(\sum_{j,k=1}^{n} \sum_{l=1}^{d} \left(\frac{\partial^2 \phi}{\partial x^j \partial x^k} \right) g^{jl} g^{kl} \right) \mathrm{d}\tau. \tag{A 8}$$

In order to obtain the normalized conditional law, i.e. $\pi_\tau(\phi) = \varTheta_\tau(\phi)/\varTheta_\tau(1)$, it is expanded using Ito's formula as given below:

$$\mathrm{d}\pi_\tau(\phi) = \frac{\mathrm{d}\varTheta_\tau(\phi)}{\varTheta_\tau(1)} + \varTheta_\tau(\phi) \, \mathrm{d} \left(\frac{1}{\varTheta_\tau(1)} \right) + \left\langle \mathrm{d}\varTheta_\tau(\phi), \, \mathrm{d} \left(\frac{1}{\varTheta_\tau(1)} \right) \right\rangle, \tag{A 9}$$

where $\mathrm{d}(1/\varTheta_\tau(1))$ may be expanded as

$$\mathrm{d} \left(\frac{1}{\varTheta_\tau(1)} \right) = -\frac{1}{\varTheta_\tau^2(1)} \, \mathrm{d}\varTheta_\tau(1) + \frac{1}{\varTheta_\tau^3(1)} \langle \mathrm{d}\varTheta_\tau(1), \, \mathrm{d}\varTheta_\tau(1) \rangle. \tag{A 10}$$

Putting $\phi = 1$ in equation (A 8), we get an Ito expansion for $d\varTheta_\tau(1)$, which is given below:

$$\mathrm{d}\varTheta_\tau(1) = \varTheta_\tau(h) \, \mathrm{d}\breve{f}_\tau. \tag{A 11}$$

Using equation (A 11) in equation (A 10),

$$\mathrm{d} \left(\frac{1}{\varTheta_\tau(1)} \right) = -\frac{\pi_\tau(h)}{\varTheta_\tau(1)} \, \mathrm{d}\tilde{f}_\tau + \frac{\pi_\tau^2(h)}{\varTheta_\tau(1)} \, \mathrm{d}\tau. \tag{A 12}$$

Using equation (A 8) and (A 12) in equation (A 9), we get

$$\mathrm{d}\pi_\tau(\phi) = \pi_\tau(\phi h)\,\mathrm{d}\breve{f}_\tau + \frac{1}{2}\pi_\tau\left(\sum_{j,k=1}^{n}\sum_{l=1}^{d}\left(\frac{\partial^2\phi}{\partial x^j\partial x^k}\right)g^{jl}g^{kl}\right)\mathrm{d}\tau$$

$$+ (-\pi_\tau(\phi)\pi_\tau(h)\,\mathrm{d}\breve{f}_\tau + \pi_\tau(\phi)\pi_\tau^2(h)\,\mathrm{d}\tau) - \pi_\tau(\phi h)\pi_\tau(h)\,\mathrm{d}\tau. \qquad \text{(A 13)}$$

Thus, we arrive at the extremal equation for the evolution of the normalized conditional estimate $\pi_\tau(\phi)$ given as

$$\mathrm{d}\pi_\tau(\phi) = \frac{1}{2}\pi_\tau\left(\sum_{j,k=1}^{n}\sum_{l=1}^{d}\left(\frac{\partial^2\phi}{\partial x^j\partial x^k}\right)g^{jl}g^{kl}\right)\mathrm{d}\tau + (\pi_\tau(\phi h) - \pi_\tau(\phi)\pi_\tau(h))(\mathrm{d}\breve{f}_\tau - \pi_\tau(h)\,\mathrm{d}\tau). \qquad \text{(A 14)}$$

Remark A.2. Equation (A 7) finds its equivalence with the Zakai equation well known in stochastic filtering.

Remark A.3. Equation (A 14) has its parallel in the Kushner–Stratonovich equation, which is again well known in stochastic filtering.

Theorem A.4. *When* $\phi(x) = x$ equation (A 14) *may be rewritten as*

$$\mathrm{d}\pi_\tau(x) = (\pi_\tau(xf) - \pi_\tau(x)\pi_\tau(f))(\rho_\tau\rho_\tau^{\mathrm{T}})^{-1}(\mathrm{d}\breve{f}_\tau - \pi_\tau(f)\,\mathrm{d}\tau).$$

Proof. Since we are typically interested in the evolution of the conditional estimate of $\mathbf{x}_\tau$, i.e. ϕ is the identity function, equation (A 14) may be simplified as

$$\mathrm{d}\pi_\tau(\mathbf{x}) = (\pi_\tau(\mathbf{x}h) - \pi_\tau(\mathbf{x})\pi_\tau(h))(\mathrm{d}\breve{f}_\tau - \pi_\tau(h)\,\mathrm{d}\tau). \qquad \text{(A 15)}$$

Replacing $h(\mathbf{x}_\tau)$ by $\rho_\tau^{-1}f(\mathbf{x}_\tau)$ in equation (A 15), we get

$$\mathrm{d}\pi_\tau(\mathbf{x}) = (\pi_\tau(\mathbf{x}f) - \pi_\tau(\mathbf{x})\pi_\tau(f))(\rho_\tau\rho_\tau^{\mathrm{T}})^{-1}(\mathrm{d}\breve{f}_\tau - \pi_\tau(f)\,\mathrm{d}\tau). \qquad \text{(A 16)}$$

■

References

1. Holland JH. 1975 *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan. (Internal report).
2. Goldberg DE. 1989 *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.
3. Storn R, Price K. 1997 Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optimiz.* **11**, 341–359. (doi:10.1023/A:1008202821328)
4. Kennedy J, Eberhart R. 1995 Particle swarm optimization. *Proc. IEEE Int. Conf. Neural Networks* **4**, 1942–1948.
5. Dorigo M, Birattari M. 2010 Ant colony optimization. In *Encyclopedia of machine learning* (eds C Sammut, GI Webb), pp. 36–39. New York, NY: Springer, Science and Business Media.
6. Hansen N. 2006 The CMA evolution strategy: a comparing review. In *Towards a new evolutionary computation* (eds JA Lozano, P Larrañaga, I Inza, E Bengoetxea), pp. 75–102. Berlin, Heidelberg, Germany: Springer.
7. Fletcher R. 1987 *Practical methods of optimization*. New York, NY: John Wiley.
8. Chong EKP, Zak SH. 2013 *An introduction to optimization*, vol. 76. New York, NY: John Wiley and Sons.
9. Glover F, Kochenberger GA (eds) 2003 *Handbook of metaheuristics*. Dordrecht, The Netherlands: Kluwer Academic Publisher.
10. Sörensen K. 2015 Metaheuristics: the metaphor exposed. *Int. Trans. Operat. Res.* **22**, 3–18. (doi:10.1111/itor.12001)
11. VanLaarhoven RJ, Aarts EH. 1987 *Simulated annealing*, pp. 7–15. Dordrecht, The Netherlands: Springer.
12. Wenzel W, Hamacher K. 1999 Stochastic tunneling approach for global minimization of complex potential energy landscapes. *Phys. Rev. Lett.* **82**, 3003–3007. (doi:10.1103/PhysRevLett.82.3003)
13. Enrique A, Dorronsoro B. 2005 The exploration/exploitation tradeoff in dynamic cellular genetic algorithms. *Evol. Comput. IEEE Trans.* **9**, 126–142. (doi:10.1109/TEVC.2005.843751)
14. Eiben AE, Schippers CA. 1998 On evolutionary exploration and exploitation. *Fundam. Inform.* **35**, 35–50.
15. Renders JM, Flasse SP. 1996 Hybrid methods using genetic algorithms for global optimization. *IEEE Trans. Syst. Man Cybern. B* **26**, 243–258. (doi:10.1109/3477.485836)
16. Gordon NJ, Salmond DJ, Smith AFM. 1993 Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F* (*Radar and Signal Processing*) **140**, 107–113. (doi:10.1049/ip-f-2.1993.0015)
17. Arulampalam S, Maskell N, Gordon N, Clapp T. 2002 A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **50**, 174–188. (doi:10.1109/78.978374)
18. Snyder C, Bengtsson T, Bickel P, Anderson J. 2008 Obstacles to high-dimensional particle filtering. *Mon. Weather Rev.* **136**, 4629–4640. (doi:10.1175/2008MWR2529.1)
19. Oksendal BK. 2003 *Stochastic differential equations: an introduction with applications*, 6th edn. New York, NY: Springer.
20. Stroock DW, Varadhan SR. 1972 On the support of diffusion processes with applications to the strong maximum principle. In *Proc. Sixth Berkeley Symposium on Mathematical Statistics and Probability, University of California, Berkeley, California, USA, 1970–1971*, vol. 3, pp. 333–359.
21. Freidlin MI, Szücs J, Wentzell AD. 2012 *Random perturbations of dynamical systems*, vol. 260. Berlin, Germany: Springer.
22. Wolpert DH, William GM. 1997 No free lunch theorems for optimization. *Evol. Comput. IEEE Trans.* **1**, 67–82. (doi:10.1109/4235.585893)
23. Vrugt JA, Robinson BA. 2007 Improved evolutionary optimization from genetically adaptive multimethod search. *Proc. Natl Acad. Sci. USA* **104**, 708–711. (doi:10.1073/pnas.0610471104)
24. Klebaner FC. 2005 *Introduction to stochastic calculus with applications*, vol. 57. London, UK: Imperial College Press.
25. Nualart D. 2006 *The Malliavin Calculus and related topics*. Berlin, Germany: Springer.
26. Kushner HJ. 1967 Dynamical equations of optimal nonlinear filtering. *J. Diff. Equ.* **3**, 179–190. (doi:10.1016/0022-0396(67)90023-X)

27. Sarkar S, Chowdhury SR, Venugopal M, Vasu RM, Roy D. 2014 A Kushner Stratonovich Monte Carlo filter applied to nonlinear dynamical system identification. *Physica D: Nonlinear Phenomena* **270**, 46–59. (doi:10.1016/j.physd.2013.12.007)
28. Sarkar S, Roy D. Submitted. An ensemble Kushner–Stratonovich (EnKS) nonlinear filter: additive particle updates in non-iterative and iterative forms. Preprint (http://arxiv.org/abs/1402.1253)
29. Sun J, Zhao J, Wu X, Fang W, Cai Y, Xu W. 2010 Parameter estimation for chaotic systems with a drift particle swarm optimization method. *Phys. Lett. A* **374**, 2816–2822. (doi:10.1016/j.physleta.2010.04.071)
30. Neri F, Tirronen V. 2010 Recent advances in differential evolution: a survey and experimental analysis. *Artif. Intell. Rev.* **33**, 61–106. (doi:10.1007/s10462-009-9137-2)
31. Das S, Suganthan PN. 2011 Differential evolution: a survey of the state-of-the-art. *Evol. Comput. IEEE Trans.* **15**, 4–13. (doi:10.1109/TEVC.2010.2059031)
32. Mallipeddi R, Suganthan PN, Pan QK, Tasgetiren MF. 2011 Differential evolution algorithm with ensemble of parameters and mutation strategies. *Appl. Soft Comput.* **11**, 1679–1696. (doi:10.1016/j.asoc.2010.04.024)
33. Li C, Yang S, Nguyen TT. 2012 A self-learning particle swarm optimizer for global optimization problems. *Syst. Man Cybermetics B Cybermetics IEEE Trans.* **42**, 627–646. (doi:10.1109/TSMCB.2011.2171946)
34. Tang K, Yao X, Suganthan PN, MacNish C, Chen YP, Chen CM, Yang Z. 2007 Benchmark functions for the CEC 2008 special session and competition on large scale global optimization. Technical report, Nature inspired computation and applications laboratory, USTC, China, November 2007. See http://nical.ustc.edu.cn/cec08ss.php.
35. Hu M, Wu T, Weir JD. 2013 An adaptive particle swarm optimization with multiple adaptive methods. *IEEE Trans. Evol. Comput.* **17**, 705–720. (doi:10.1109/TEVC.2012.2232931)
36. Bartz-Beielstein T, Chiarandini M, Paquete L, Preuss M (eds). 2010 *Experimental methods for the analysis of optimization algorithms*. Berlin, Germany: Springer.
37. Hoos HH. 2012 Automated algorithm configuration and parameter tuning. In *Autonomous search* (eds Y Hamadi, E Monfroy, F Saubion), pp. 37–71. Berlin, Germany: Springer.
38. Medhi B, Hegde GM, Reddy KPJ, Roy D, Vasu RM. 2014 Quantitative estimation of density variation in high-speed flows through inversion of light travel-time data. *Opt. Eng.* **53**, 124107. (doi:10.1117/1.OE.53.12.124107)
39. Ghiglia DC, Romero LA. 1994 Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods. *J. Opt. Soc. Am. A* **11**, 107–117. (doi:10.1364/JOSAA.11.000107)