J Glob Optim (2009) 43:329–356
DOI 10.1007/s10898-008-9292-z

The interacting-particle algorithm with dynamic heating and cooling

Orcun Molvalioglu · Zelda B. Zabinsky · Wolf Kohn

Received: 3 April 2007 / Accepted: 4 March 2008 / Published online: 21 March 2008

Abstract We consider an interacting-particle algorithm which is population-based like genetic algorithms and also has a temperature parameter analogous to simulated annealing. The temperature parameter of the interacting-particle algorithm has to cool down to zero in order to achieve convergence towards global optima. The way this temperature parameter is tuned affects the performance of the search process and we implement a meta-control methodology that adapts the temperature to the observed state of the samplings. The main idea is to solve an optimal control problem where the heating/cooling rate of the temperature parameter is the control variable. The criterion of the optimal control problem consists of user defined performance measures for the probability density function of the particles' locations including expected objective function value of the particles and the spread of the particles' locations. Our numerical results indicate that with this control methodology the temperature fluctuates (both heating and cooling) during the progress of the algorithm to meet our performance measures. In addition our numerical comparison of the meta-control methodology with classical cooling schedules demonstrate the benefits in employing the meta-control methodology.

Keywords Interacting-particle algorithm · Meta-control · Optimal control · Global optimization · Simulated annealing · Cooling schedule

1 Introduction

In stochastic global optimization algorithms such as Simulated Annealing (SA) [8,24] and Genetic Algorithms (GA) [22], the samplings are controlled by a set of parameters such as the temperature parameter, the cross-over/mutation rate and the selection rate. The random search process governing the samplings is the main determinant of the overall efficiency and effectiveness of an algorithm. Thus the way the sampling parameters are tuned is a crucial

O. Molvalioglu · Z. B. Zabinsky (✉) · W. Kohn
Department of Industrial Engineering, University of Washington, Seattle, WA 98195-2650, USA
e-mail: zelda@u.washington.edu

component of the overall performance. While good parameter settings can be determined prior to a run by trial and error and past experiences with a particular optimization problem; we consider the possibility of adapting the parameters to the state of the samplings during the progress of the algorithm.

We study this possibility for the temperature parameter of an interacting-particle algorithm. The interacting-particle algorithm of this paper appears in different studies [5,13,16] in similar forms and it has analogies to both SA and GA. Akin to SA, the temperature parameter of the interacting-particle algorithm has to cool down to zero. Instead of fixing a cooling schedule prior to a run, we implement a meta-control methodology that dynamically determines the changes in the temperature parameter by adapting it to the state of the sampling process. The methodology controls the samplings of the interacting-particle algorithm with the temperature parameter to make the algorithm satisfy user-defined performance criteria. The meta-control methodology for prescribing parameters of a deterministic optimization algorithm was introduced by Kohn et al. [9]. Extending the idea to stochastic global optimization algorithms, in [14] we develop the theoretical aspects of a meta-control methodology for the interacting-particle algorithm. This paper discusses the practical implementation of the concept and reports results from numerical experiments.

The idea of adaptively setting the temperature parameter for SA has been previously studied in [3,7,23]. Kolonko and Tran study weak convergence conditions for SA with an adaptive temperature schedule that also allows heating [10]. Shen et al. [20] derives an analytical cooling schedule for SA that adapts to the observed function values. Munakata and Nakamura [17] theoretically solve an optimal control problem to investigate optimal cooling schedules for SA that maximize the probability of sampling the global optima. Their optimal control problem requires knowledge of the whole state space and they obtain practical cooling schedules for small scale traveling salesperson problems.

Our meta-control methodology treats the interacting-particle algorithm as a dynamic system and utilizes an optimal control problem where the fractional change in the temperature parameter is the control variable. The interacting-particle algorithm moves N-particles inside the feasible set according to random samplings of an N-particle exploration mechanism which we implement with the Hit-and-Run kernel [21] and an N-particle selection mechanism which is controlled by the temperature parameter. The state of the control problem reflects the probability density function (p.d.f.) of the particles' locations and the state dynamics are characterized by the samplings of the N-particle mechanisms. The criteria of the optimal control problem includes the user-defined performance measures for the interacting-particle algorithm and they relate to the p.d.f. of the particles. Our performance criteria includes the expected objective function value of the particles, the spread of the particles, and the algorithm running time. The control problem assesses the evolution of the p.d.f. of particles under different temperature parameters and provides a closed-loop feedback on the temperature parameter to optimize our performance measures. In [14] we represent the generic form of this control problem and do not investigate solution techniques. In this study we develop a solution methodology for the optimal control problem and apply it in practice.

In Sect. 2 we state the interacting-particle algorithm and give an overview of the meta-control methodology. In Sects. 3 and 4 we discuss the details of our control methodology. We numerically investigate the behavior of our meta-control methodology under different parameter settings in Sect. 5. In Sect. 6 we numerically compare the meta-controlled interacting-particle algorithm with the interacting-particle algorithm with classic cooling schedules and also the SA algorithm with classic cooling schedules. Finally we conclude the paper in Sect. 7. Appendix A outlines the parameter estimation step in our meta-control methodology. Appendix B provides our least squares approach for estimating parameters of the control

problem. Appendices C and D include the detailed results of our numerical experiments from Sect. 5.

2 Interacting particle algorithm with meta-control of temperature

We address the following optimization problem with continuous decision variables

$$\text{minimize } f(x), \text{ subject to } x \in \Omega \subset \mathbb{R}^d$$

where we require $f(x)$ to be lower and upper bounded on Ω, $-\infty < f(x) < +\infty$ for all $x \in \Omega$, but not necessarily differentiable. The feasible set Ω is assumed to be compact and nonempty. We solve this problem with an interacting-particle algorithm which is population-based like genetic algorithms and also has analogies to simulated annealing. In this study, the interacting-particle algorithm is described in a form similar to the algorithm appearing in [14] but with minor differences due to our current focus on implementation.

Interacting-particle algorithm:
Initialization: Sample N points $y_1^i, i = 1, \ldots, N$ on Ω according to the initial p.d.f ϕ_1 which is uniform on Ω. Set the temperature parameter γ_0 to a positive value.

For $\tau = 1, 2, \ldots$

N-Particle Exploration: Move the particles to intermediate locations $\hat{y}_\tau^i \in \Omega$ with the p.d.f. $E(y_\tau^i, \hat{y}_\tau^i), i = 1, \ldots, N$ which is defined by the Markov kernel E.
Temperature Parameter Update: Update the temperature parameter, $\gamma_\tau = (1 + \epsilon_\tau)\gamma_{\tau-1}$ where $\epsilon_\tau = K(\mathfrak{F}_\tau)$ and $K(\mathfrak{F}_\tau)$ is a function that depends on the information vector $\mathfrak{F}_\tau$ generated by the interacting-particle algorithm up to time τ. The vector $\mathfrak{F}_\tau$ includes historical particle locations, function values and temperature values.
N-Particle Selection: Move the particles $i = 1, \ldots, N$ to the intermediate locations $\hat{y}_\tau^j$ of particles $j = 1, \ldots, N$ and set $y_{\tau+1}^i \leftarrow \hat{y}_\tau^j$ with probability $s_j(\gamma_\tau)$ where $s_j(\gamma_\tau) = \left(\frac{G_{\gamma_\tau}(\hat{y}_\tau^j)}{\sum_{k=1}^N G_{\gamma_\tau}(\hat{y}_\tau^k)}\right)$ and $G_{\gamma_\tau}(y) = \exp(-f(y)/\gamma_\tau)$.

In the initialization phase we determine the starting locations for N particles using the p.d.f ϕ_1. We use the uniform density on Ω for our initial samplings, i.e. $\phi_1(x) = vol(\Omega)^{-1}, \forall x \in \Omega$. Given the locations of N particles, the interacting-particle algorithm moves the particles inside the feasible set according to the N-particle exploration mechanism. The N-particle exploration mechanism is independent of the objective function $f(x)$. At this step given the location of the particle i, y_τ^i, we move the particle to an intermediate location $\hat{y}_\tau^i$ using a Markov kernel E which is called the exploration kernel. We use the standard Hit-and-Run kernel [21] on Ω. Hit-and-Run is a symmetric Markov kernel and its stationary measure is the uniform distribution on Ω [11].

Following the N-particle exploration mechanism we have the N-particle selection mechanism which is dependent on the objective function. Given the set of intermediate locations $\hat{y}_\tau^i, i = 1, \ldots, N$ and the temperature parameter γ_τ, this mechanism moves particle i to the location $\hat{y}_\tau^j, j = 1, \ldots, N$ with probability $s_j(\gamma_\tau)$ where

$$s_j(\gamma_\tau) = \left(\frac{G_{\gamma_\tau}(\hat{y}_\tau^j)}{\sum_{k=1}^N G_{\gamma_\tau}(\hat{y}_\tau^k)}\right)$$

and $G_{\gamma_\tau}(y) = \exp(-f(y)/\gamma_\tau)$. This probability law favors intermediate locations with lower objective function values and the temperature parameter controls the selectiveness among the intermediate points. As $\gamma_\tau \to 0$ the particles become more concentrated around the global optima. After the N-particle selection mechanism there may be multiple particles at the same locations and starting from these new locations we start the next iteration's N-particle exploration mechanism.

The interacting-particle algorithm has similarities to a genetic algorithm studied by Cerf [5] for discrete optimization problems. The selection criteria in the N-particle selection mechanism appears in a more generic form in [5] and the N-particle exploration phase is analogous to a mutation step. However, the interacting-particle algorithm does not have the typical cross-over step of a genetic algorithm. Cerf proves that with conditions on the mutation kernel and the population size which depends on the landscape of the objective function, the mean objective function value of the population asymptotically converges to the global optima. Moral and Miclo [16] study the convergence properties of the interacting-particle algorithm with infinite number of particles and logarithmic cooling schedule on discrete optimization problems. The analysis again takes into account properties of the objective function. Moral [15] analyzes the asymptotic behavior of a similar interacting-particle algorithm in $\mathbb{R}^d$ with infinite number of particles. With certain conditions on the cooling schedule, exploration kernel and the objective function he proves asymptotic convergence of the mean objective function value of the particles to global optima.

In order to make the particles converge towards the global optima the temperature parameter is changed iteratively according to the following rule

$$\gamma_\tau = (1 + \epsilon_\tau)\gamma_{\tau-1} \tag{1}$$

where the variable $\epsilon_\tau > -1$ is the fractional change in the temperature. A negative ϵ_τ implies cooling while a positive ϵ_τ means heating. We determine the value of this variable using the feedback function $K(\mathfrak{F}_\tau)$, which is a function of the information vector $\mathfrak{F}_\tau$. This information vector of past history includes iteration number, all the previously observed particle locations, corresponding function values and temperature values, i.e.

$$\mathfrak{F}_\tau = (y_1, \hat{y}_1, f(y_1), f(\hat{y}_1), \ldots, y_\tau, \hat{y}_\tau, f(y_\tau), f(\hat{y}_\tau), \gamma_0, \ldots, \gamma_{\tau-1}, \tau)$$

where $y_\tau = \left(y_\tau^1, \ldots, y_\tau^N\right)$ and $\hat{y}_\tau = \left(y_\tau^1, \ldots, y_\tau^N\right)$ are vectors in the product space Ω^N representing the locations of the N particles. We also refer to the function values at the locations with vectors $f(y_\tau) = \left(f(y_\tau^1), \ldots, f(y_\tau^N)\right)$ and $f(\hat{y}_\tau) = \left(f(\hat{y}_\tau^1), \ldots, f(\hat{y}_\tau^N)\right)$. Note that the function $K(\mathfrak{F}_\tau)$ defines a very generic temperature schedule. The cooling schedules which depend on iteration count (e.g. geometric, logarithmic, exponential cooling schedules) can be expressed in terms of the function $K(\mathfrak{F}_\tau)$, as well as a more explicit function.

The idea that the temperature schedule can utilize the information supplied by the interacting-particle algorithm appears in [13], however that study does not define this function explicitly. In [14] we develop the theory of a meta-control methodology for defining the function $K(\mathfrak{F}_\tau)$ that adapts to the information vector $\mathfrak{F}_\tau$. The meta-control methodology regards the interacting-particle algorithm as a discrete dynamic system that evolves iteratively with the N-particle mechanisms. The main aspect of the methodology is to design a closed-loop feedback on the temperature to make samplings satisfy user-defined performance measures. The feedback is specified by the solution of an optimal control problem where the fractional change in the temperature is the control variable. The state of the control problem depends on the p.d.f. of the locations of the particles and the state dynamics are characterized by the N-particle mechanisms.

In this paper we use the theoretical ideas from [14] to develop a practical procedure for the feedback function $K(\mathfrak{F}_\tau)$. The procedure has two steps: the *parameter estimation* step which uses the information vector $\mathfrak{F}_\tau$ to estimate the parameters of the optimal control problem; and the *control problem* which provides the optimal fractional change in the temperature value, ϵ_τ. The ideal control problem characterizes the evolution of the p.d.f. of the particles; however it is very difficult to solve. We develop an approximation to the p.d.f. of the particles and use an approximation method from [12] to obtain an auxiliary control problem which is easier to solve. We postpone the statement of the control problem until Sect. 4. We use the information $\mathfrak{F}_\tau$ to update the input parameters $\Theta(\tau)$ for the auxiliary optimal control problem. The meta-control methodology considers a finite time window into the future, using the concepts of receding-horizon control [2]. The control problem predicts the evolution of the p.d.f. of the particles for T_c future iterations. We reserve τ to refer to the iterations in the interacting-particle algorithm and use t to refer to the future iterations in the optimal control problem. The control problem is solved once per iteration of the interacting-particle algorithm to determine the optimal fractional change in the temperature value for the current and subsequent T_c iterations that optimize the user-defined criteria for the p.d.f. of the particles.

To summarize, the procedure for the function $K(\mathfrak{F}_\tau)$ is,

The function $\mathbf{K}(\mathfrak{F}_\tau)$:
Parameter Estimation: *Use $\mathfrak{F}_\tau$ to estimate the input parameters $\Theta(\tau)$ for the optimal control problem.*
Control Problem: *Solve the control problem with $\Theta(\tau)$ to obtain the optimal fractional change in temperature for $t = 0, \ldots, T_c$ where $t = 0$ denotes the current iteration and $t = T_c$ denotes T_c iterations later.*
Set $\mathbf{K}(\mathfrak{F}_\tau)$: *Set the value of the function $K(\mathfrak{F}_\tau)$ using the optimal control for $t = 0$.*

The control problem is defined in the following sections after we introduce the state variable of this control problem and discuss its expected dynamics.

3 State of the interacting-particle algorithm and its dynamics

The samplings of the interacting-particle algorithm depend on the locations of the particles which change with the N-particle exploration and N-particle selection mechanisms. We take into account the p.d.f. of the locations of the particles and how it evolves with the N-particle mechanisms. We approximate the p.d.f. of the location of the particles using a sequence of orthonormal polynomials as discussed in [14,18]. For the implementation we use the Legendre polynomials defined over a box Σ in $\mathbb{R}^d$, $\Sigma = [l_1, u_1] \times \cdots \times [l_d, u_d]$, where $\Omega \subseteq \Sigma$. The Legendre polynomials define an orthonormal polynomial sequence and form a complete basis of the space of square integrable functions defined over Σ, $L^2(\Sigma)$ [6].

A multivariate Legendre polynomial in $\mathbb{R}^d$ is defined as the multiplication of single variate Legendre polynomials [6]. Let $P_\alpha(x)$ with $x = (x_1, \ldots, x_d)^T$ denote a multivariate Legendre polynomial over the d dimensional box $\Sigma = [l_1, u_1] \times \cdots \times [l_d, u_d]$, then

$$P_\alpha(x) = q_{\upsilon_1}(x_1) q_{\upsilon_2}(x_2) \cdots q_{\upsilon_d}(x_d) \tag{2}$$

where $q_{\upsilon_k}(x_k), k = 1, \ldots, d$ are single variate Legendre polynomials over $[l_k, u_k]$ and $\alpha = (\upsilon_1, \upsilon_2, \ldots, \upsilon_d)$ is the vector indexing the polynomial. For any $k, k = 1, \ldots, d$, the single variate Legendre polynomial $q_{\upsilon_k}(x_k)$ where $x_k \in [l_k, u_k]$ and $\upsilon_k = 0, 1, \ldots$ is said to have

degree υ_k. The first few polynomials, $\upsilon_k = 0, 1, 2$ and 3, are defined as

$$
\begin{aligned}
q_0(x_k) &= 1 \\
q_1(x_k) &= \left(u_k - l_k\right)^{-1}(2x_k - u_k - l_k) \\
q_2(x_k) &= \frac{1}{2}\left(3\left((u_k - l_k)^{-1}(2x_k - u_k - l_k)\right)^2 - 1\right) \\
q_3(x_k) &= \frac{1}{2}\left(5\left((u_k - l_k)^{-1}(2x_k - u_k - l_k)\right)^3 - 3(u_k - l_k)^{-1}(2x_k - u_k - l_k)\right).
\end{aligned}
$$

Traditionally the single variate Legendre polynomial $q_{\upsilon_k}(x_k)$ is defined over $[-1, 1]$, the bounds l_k and u_k appear in the polynomials above in order to scale x_k to a point inside $[-1, 1]$.

Now let $P_\alpha(x), \alpha \in I$ be the sequence of multivariate Legendre polynomials as defined in (2) where I is the index set of the polynomials. We denote the cardinality of the index set I with Q, i.e. Q is the number of multivariate Legendre polynomials, $|I| = Q$. When Q goes to infinity we have a complete orthonormal basis of the square integrable functions over Σ. However we use a finite number of orthonormal polynomials for practical purposes. Given the locations of the particles $y_\tau^1, \ldots, y_\tau^N$, we approximate the p.d.f. of the particle locations with a function $\Psi_\tau(x)$ as follows,

$$\Psi_\tau(x) = \sum_{\alpha \in I} c_\alpha(\tau) P_\alpha(x),$$

where the coefficients $c_\alpha(\tau)$ are functions of y_τ;

$$c_\alpha(\tau) = \frac{1}{N}\sum_{i=1}^{N} P_\alpha(y_\tau^i), \quad \alpha \in I. \tag{3}$$

Similarly we can estimate a p.d.f. for the intermediate particle locations $\hat{y}_\tau^1, \ldots, \hat{y}_\tau^N$. This p.d.f. corresponds to the density of the particles after the N-particle exploration step. In this case we denote the coefficients by $\hat{c}_\alpha$ where

$$\hat{c}_\alpha(\tau) = \frac{1}{N}\sum_{i=1}^{N} P_\alpha(\hat{y}_\tau^i), \quad \alpha \in I. \tag{4}$$

When we impose the constraint that $\int_\Sigma \Psi_\tau(x)dx = 1$ then $\Psi_\tau(x)$ defines a p.d.f. [18]. Because we do not use $\Psi_\tau(x)$ directly for samplings, we do not require the function $\Psi_\tau(x)$ to satisfy such a condition.

The function $\Psi_\tau(x)$ gives us partial information about the p.d.f. of the particles. The current state of the function $\Psi_\tau(x)$ and its evolution due to movement of the particles are directly reflected in the coefficient values $c_\alpha(\tau)$ and their evolution. We represent the coefficients as a vector $c(\tau) = (c_\alpha(\tau))_{\alpha \in I}$ and let the vector $c(\tau)$ characterize the current state of the interacting-particle algorithm. By looking at how the vector $c(\tau)$ evolves we obtain information about the evolution of the p.d.f. of the particles.

We have shown in [14] that the evolution of $c(\tau)$ has stochastic dynamics which involves a deterministic part and a random term whose expectation is zero. The stochastic dynamics leads to a stochastic optimal control problem which is difficult to solve directly. In this study we ignore the randomness in the dynamics by utilizing the concept of certainty equivalence [4] which takes into account the expected values of $c(\tau + t), t = 0, 1, \ldots$ with respect to the information $\mathfrak{F}_\tau$. As we will see the certainty equivalence principle leads to a deterministic optimal control problem.

The evolution of the p.d.f. of the particles' locations and the impact of temperature on this p.d.f. is captured through the evolution of the coefficients $c(\tau)$. By predicting the future values of $c(\tau)$ we can predict the evolution of the p.d.f. of the particles' under different temperature values. Let $c(\tau+t)$ and $\hat{c}(\tau+t)$ be the coefficients at locations $y_{\tau+t}$ and $\hat{y}_{\tau+t}$ at iterations $\tau+t, t=0,1,\ldots$ respectively. Let the vectors $m(t)=(m_\alpha(t))_{\alpha\in I}$ and $\hat{m}(t)=\left(\hat{m}_\alpha(t)\right)_{\alpha\in I}$ be the expectations of $c(\tau+t)$ and $\hat{c}(\tau+t)$ for $t=0,1,\ldots$ with respect to the information vector $\mathfrak{F}_\tau$, i.e. $m(t)=\mathbb{E}(c(\tau+t)|\mathfrak{F}_\tau)$ and $\hat{m}(t)=\mathbb{E}(\hat{c}(\tau+t)|\mathfrak{F}_\tau)$. Note that the information $\mathfrak{F}_\tau$ includes realizations of the latest locations y_τ and $\hat{y}_\tau$, thus we have $m(0)=\mathbb{E}(c(\tau)|\mathfrak{F}_\tau)=c(\tau)$ and $\hat{m}(0)=\mathbb{E}(\hat{c}(\tau)|\mathfrak{F}_\tau)=\hat{c}(\tau)$.

By Theorem 6 of [14] we have the following relationship between the vectors $m(t)$ and $\hat{m}(t)$

$$\hat{m}(t)=Am(t),\quad t=0,1,\ldots \tag{5}$$

where $m(0)=c(\tau)$ and $A=(a_{\alpha\beta})$ is a $Q\times Q$ symmetric matrix with

$$a_{\alpha\beta}=\int_\Omega\int_\Omega E(x,y)P_\alpha(x)P_\beta(y)dxdy,\quad \alpha\in I,\beta\in I. \tag{6}$$

Once the particles move from the intermediate locations $\hat{y}_{\tau+t}$ to $y_{\tau+t+1}$ with the N-particle selection mechanism, the vector $\hat{c}(\tau+t)$ changes to $c(\tau+t+1)$. For notational purposes we denote the temperature value at iteration $\tau+t-1$ with $\gamma(t)$, i.e. $\gamma(t)=\gamma_{\tau+t-1}$. Given $\mathfrak{F}_\tau$ we have the initial value $\gamma(0)=\gamma_{\tau-1}$. We make this shift in the time index of the temperature parameter in order to have a consistent indexing in the dynamics of our control problem. We also move the time subscripts into parenthesis for notational purposes. Then by Theorem 6 of [14], the vectors $m(t+1)$ and $\hat{m}(t)$ satisfy

$$m(t+1)=\frac{B\left(\gamma(t+1)\right)\hat{m}(t)}{p^TB\left(\gamma(t+1)\right)\hat{m}(t)},\quad t=0,1,\ldots \tag{7}$$

where $\hat{m}(0)=\hat{c}(\tau)$, $B(\gamma(t))=\left(b_{\alpha\beta}(\gamma(t+1))\right)$ is a $Q\times Q$ symmetric matrix with

$$b_{\alpha\beta}(\gamma(t+1))=\int_\Omega \exp\left(-f(x)/\gamma(t+1)\right)P_\alpha(x)P_\beta(x)dx,\quad \alpha\in I,\beta\in I \tag{8}$$

and the vector $p\in\mathbb{R}^Q$ is defined as

$$p=\left(\int_\Omega P_\alpha(x)dx\right)_{\alpha\in I}. \tag{9}$$

The denominator of (7) equals the normalizing term $\sum_{k=1}^N G_{\gamma_{\tau+t}}(\hat{y}^k_{\tau+t})$ in the definition of $s_j(\gamma_{\tau+t})$. We use this fact when we are estimating matrix $B(\cdot)$ and the vector p. Now substituting the right hand side of Eq. 5 into 7 and also letting $\gamma(t+1)=(1+\epsilon(t))\gamma(t)$ where $\epsilon(t)$ is the change in the temperature at iteration $\tau+t$, we get the following dynamics for the vector $m(t)$,

$$m(t+1)=\frac{B\left((1+\epsilon(t))\gamma(t)\right)Am(t)}{p^TB\left((1+\epsilon(t))\gamma(t)\right)Am(t)},\quad t=0,1,\ldots \tag{10}$$

with $m(0)=c(\tau)$. Equation 10 as proven in [14] implies that the expectation of the vector $c(\cdot)$ changes nonlinearly with the symmetric matrices A, $B(\cdot)$ and the vector p. The entries of A, $B(\cdot)$ and p are analytically defined by Eqs. 6, 8 and 9. However, instead of using these

expressions we estimate the matrices numerically using the provided information $\mathfrak{F}_\tau$, see Appendix A.

The dynamics for $m(t)$ enable us to predict the changes in the p.d.f. of the particles due to temperature changes as given by $\epsilon(t)$. In the next section we formulate the criteria for the p.d.f. of the particles as functions of $m(t)$ and represent our optimal control problem.

4 Algorithmic criterion and the optimal control problem

At each iteration τ the meta-control methodology determines the function $K(\mathfrak{F}_\tau)$ by solving an optimal control problem where $m(t)$ and the temperature $\gamma(t)$ are state variables and $\epsilon(t)$ is the control variable. This control problem is generically represented in [14] as a discrete stochastic optimal control problem. Here, as we assume certainty equivalence, the problem takes the following deterministic form,

Control Problem 1

$$\min_{\substack{\epsilon(t)\in[\epsilon_{\min},\epsilon_{\max}]\\ t=0,\ldots,T_c}} \Lambda(m(T_c+1),\gamma(T_c+1)) \tag{11}$$

subject to

$$\begin{pmatrix} m(t+1)\\ \gamma(t+1)\end{pmatrix} = \begin{pmatrix} B\left((1+\epsilon(t))\gamma(t)\right) Am(t)\\ p^T B\left((1+\epsilon(t))\gamma(t)\right) Am(t)\\ (1+\epsilon(t))\,\gamma(t)\end{pmatrix}, \quad t=0,\ldots,T_c \tag{12}$$

$$\begin{pmatrix} m(0)\\ \gamma(0)\end{pmatrix} = \begin{pmatrix} c(\tau)\\ \gamma_{\tau-1}\end{pmatrix} \tag{13}$$

where $c(\tau)=\mathbb{E}(c(\tau)|\mathfrak{F}_\tau)$ and $\gamma_{\tau-1}$ are given. Two forms of the criteria Λ are defined in (18) and (19).

At iteration τ this optimal control problem considers the evolution of $m(t)$ and the temperature $\gamma(t)$ for the next T_c iterations and determines $\epsilon(0),\ \ldots,\epsilon(T_c)$ that optimizes the objective functional given by $\Lambda(\cdot)$. We assume the control variable belongs to the closed bounded set $[\epsilon_{\min},\epsilon_{\max}]$ where $\epsilon_{\min}<0$ and $\epsilon_{\max}\geq 0$.

In order to define the objective functional $\Lambda(\cdot)$ we express the user-defined criteria for the samplings as a function of $m(t)$. The first criterion and the most important in optimization is the expectation of the objective function. Let f_τ and $\hat{f}_\tau$ be the average objective function value of the particles when they are located at y_τ and $\hat{y}_\tau$ respectively,

$$f_\tau=\frac{1}{N}\sum_{i=1}^{N} f(y_\tau^i),\quad \hat{f}_\tau=\frac{1}{N}\sum_{i=1}^{N} f(\hat{y}_\tau^i). \tag{14}$$

As we have shown in [14] the vectors $c(\tau)$ and $\hat{c}(\tau)$ estimated using (3) and (4) relate to f_τ and $\hat{f}_\tau$ as follows

$$f_\tau=F^T c(\tau),\quad \hat{f}_\tau=F^T\hat{c}(\tau) \tag{15}$$

where $F=(F_\alpha)_{\alpha\in I}\in\mathbb{R}^Q$ with

$$F_\alpha=\int_\Omega f(x)P_\alpha(x)dx,\quad \alpha\in I. \tag{16}$$

Instead of calculating the vector F from the integral in Eq. 16 we estimate it using our past observations, see (33). Equation 15 establishes a linear relationship between the vectors $c(\tau)$, $\hat{c}(\tau)$ and the value f_τ. Given F and the expected future state of the interacting-algorithm $m(t)$, we can predict the average objective function value of the particles in the subsequent iterations, i.e. $f_{\tau+t} = F^T m(t)$.

The second criteria relates to the risk involved with the current p.d.f. of the particles' locations. By common sense when the locations of the particles have a wider spread over the feasible set, the particles can explore a wider region. However when the particles are concentrated at a single point; the p.d.f. of the particle locations has most of its mass around that point; the immediate exploration region is narrower and this can lead to getting stuck in a local optima and missing the global optima. The spread of the particles over Ω is maximum when they are uniformly distributed on Ω and we consider the uniform distribution as a risk averse distribution. Our second criteria measures the square deviations of the current p.d.f. of the particles from the uniform distribution and it has the following form

$$\sum_{\alpha \in I} (m_\alpha(t) - c_\alpha^{unif})^2 = \left(m(t) - c^{unif}\right)^T I_Q \left(m(t) - c^{unif}\right), \tag{17}$$

where c_α^{unif} is estimated by Eq. 3 with the initial particle locations $\{y_1^i\}$ which are uniformly sampled and I_Q is the $Q \times Q$ identity matrix. Unless the objective function $f(x)$ is constant over Ω there is a trade off between minimizing the expected function value and minimizing the square deviations from the uniform distribution. As the p.d.f. puts more of its mass towards the global optima the deviation from the uniform distribution increases.

When we have multiple criteria for the interacting-particle algorithm one approach is to take the weighted linear combination of all the criteria. Considering the expected objective function value of the particles and the spread of the particles we can construct the following functional which is quadratic in terms of $m(t)$,

$$\Lambda\left(m(t), \gamma(t)\right) = \omega_1 F^T m(t) + \omega_2 \left(m(t) - c^{unif}\right)^T I_Q \left(m(t) - c^{unif}\right) \tag{18}$$

where $\omega_i, i = 1, 2$ ($\omega_i > 0$ and $\sum \omega_i = 1$) represents the weight we give to each criteria.

We also have another objective functional alternative which is motivated by Sharpe's ratio [19] which is used in finance for evaluating investments. It is a reward to risk (variability) ratio to specify the performance of a random return. In our case we interpret the return from the iteration τ of the interacting-particle algorithm as the difference between the average objective function value f_τ and the average objective function value at the beginning of the algorithm, f_1. Because the initialization p.d.f. is uniform density on Ω, f_1 corresponds to the expectation of $f(x)$ with respect to uniform sampling. On the other hand the risk is the square deviations from the uniform distribution as it is represented in Eq. 17. With these interpretations we get the following criteria

$$\Lambda\left(m(t), \gamma(t)\right) = \frac{F^T m(t) - f_1}{\omega_3 \left(m(t) - c^{unif}\right)^T I_Q \left(m(t) - c^{unif}\right)} \tag{19}$$

where $\omega_3 \in (0, 1]$ is a parameter that shows what percentage of the risk we consider.

We would like to solve Control Problem 1 with $\Lambda(\cdot)$ as in (18) and (19) which is a difficult task; the control variable $\epsilon(t)$ is implicit inside the matrix $B(\cdot)$ whose terms are given by Eq. 8. However, we can use a perturbation method to develop a discrete quadratic control problem that approximates the solution [12].

Assume that at iteration τ we have the optimal solution to the previous iteration's optimal control problem, i.e. $\epsilon^*(0), \ldots, \epsilon^*(T_c)$. Using the previous solution we define a nominal solution to the current problem as follows,

$$\bar{\epsilon}(0) = \epsilon^*(1), \quad \bar{\epsilon}(1) = \epsilon^*(2), \ \ldots, \ \bar{\epsilon}(T_c - 1) = \epsilon^*(T_c), \quad \bar{\epsilon}(T_c) = \epsilon^*(T_c).$$

The known feasible solution $\bar{\epsilon}(t)$ implies the nominal state trajectories $\bar{m}(t)$ and $\bar{\gamma}(t)$ such that

$$\begin{pmatrix} \bar{m}(t+1) \\ \bar{\gamma}(t+1) \end{pmatrix} = \begin{pmatrix} B\left((1+\bar{\epsilon}(t))\bar{\gamma}(t)\right) A\bar{m}(t) \\ p^T B\left((1+\bar{\epsilon}(t))\bar{\gamma}(t)\right) A\bar{m}(t) \\ (1+\bar{\epsilon}(t))\,\bar{\gamma}(t) \end{pmatrix}, \quad t = 0, \ldots, T_c$$
$$\begin{pmatrix} \bar{m}(0) \\ \bar{\gamma}(0) \end{pmatrix} = \begin{pmatrix} \mathbb{E}(c(\tau)|\mathfrak{F}_\tau) \\ \gamma_{\tau-1} \end{pmatrix} \tag{20}$$

Next we introduce the perturbations $\delta\epsilon_+(t) \geq 0$ and $\delta\epsilon_-(t) \geq 0$ around the nominal solution $\bar{\epsilon}(t)$. The variable $\delta\epsilon_+(t)$ indicates how much we increase $\bar{\epsilon}(t)$ and $\delta\epsilon_-(t)$ indicates how much we decrease it. The difference $\delta\epsilon_+(t) - \delta\epsilon_-(t)$ gives us how much we perturbed the trajectory $\bar{\epsilon}(t)$. We use two separate perturbations, for heating and cooling, because the change affects the matrix $B(\cdot)$ asymmetrically which is important when we are numerically estimating the derivatives of $B(\cdot)$ in Appendix A.

When we perturb $\bar{\epsilon}(t)$ with $\delta\epsilon_+(t)$ and $\delta\epsilon_-(t)$, we also create perturbations of the states $m(t)$ and $\gamma(t)$ around the nominal trajectories $\bar{m}(t)$ and $\bar{\gamma}(t)$. We let $\delta m(t)$ be the perturbation around $\bar{m}(t)$, and $\delta\gamma_+(t)$ be the positive perturbation around $\bar{\gamma}(t)$ due to $\delta\epsilon_+(t)$, and $\delta\gamma_-(t)$ be the negative perturbation around $\bar{\gamma}(t)$ due to $\delta\epsilon_-(t)$. Our goal is to obtain the perturbations $\delta\epsilon_+^*(t)$ and $\delta\epsilon_-^*(t)$ such that the new solution defined as $\epsilon^*(t) = \bar{\epsilon}(t) + \delta\epsilon_+^*(t) - \delta\epsilon_-^*(t)$, $t = 0, \ldots, T_c$ improves on the objective functionals. With the perturbations $\delta\epsilon_+^*(t)$ and $\delta\epsilon_-^*(t)$, the states deviate from $\bar{m}(T_c+1)$ and $\bar{\gamma}(T_c+1)$ by $\delta m^*(T_c+1)$ and $\delta\gamma^*(T_c+1) = \delta\gamma_+^*(T_c+1) - \delta\gamma_-^*(T_c+1)$, and we would like these perturbations to decrease the performance criteria at the terminal time T_c, i.e.

$$\begin{aligned} &\Lambda\left(\bar{m}(T_c+1) + \delta m^*(T_c+1), \bar{\gamma}(T_c+1) + \delta\gamma^*(T_c+1)\right) \\ &\quad \leq \Lambda\left(\bar{m}(T_c+1), \bar{\gamma}(T_c+1)\right). \end{aligned}$$

To obtain the perturbation $\delta\epsilon_+^*(t) - \delta\epsilon_-^*(t)$ that gives $\delta m^*(T_c+1)$ and $\delta\gamma^*(T_c+1)$ and optimizes the reduction in the objective functional $\Lambda(\cdot)$, we utilize an auxiliary control problem where $\delta\epsilon_+(t)$ and $\delta\epsilon_-(t)$ are control variables and $\delta m(t)$, $\delta\gamma_+(t)$, $\delta\gamma_-(t)$ are the state variables.

To derive the dynamics of the auxiliary control problem we take the first order Taylor expansion of the dynamics in Eq. 12 around $\bar{m}(t)$, $\bar{\gamma}(t)$ and $\bar{\epsilon}(t)$ with respect to variables $m(t)$, $\gamma(t)$ and $\epsilon(t)$, however we separate the right and left derivatives for $\gamma(t)$. For ease in notation we refer to the dynamics of $m(t)$ and $\gamma(t)$ in Eq. 12 with the functions h_1 and h_2 respectively i.e. $h_1 = h_1(m(t), \gamma(t), \epsilon(t)) = m(t+1)$ and $h_2 = h_2(\gamma(t), \epsilon(t)) = \gamma(t+1)$. Note that the function h_1 has three arguments $m(t)$, $\gamma(t)$ and $\epsilon(t)$, and h_2 has two arguments $\gamma(t)$, $\epsilon(t)$. With this notation we get

$$\begin{pmatrix} \bar{m}(t+1) \\ \bar{\gamma}(t+1) \end{pmatrix} + \begin{pmatrix} \delta m(t+1) \\ \delta\gamma_+(t+1) - \delta\gamma_-(t+1) \end{pmatrix}$$
$$= \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} + \begin{pmatrix} \frac{\partial h_1}{\partial m} & \frac{\partial h_1}{\partial \gamma_+} & -\frac{\partial h_1}{\partial \gamma_-} \\ 0 & \frac{\partial h_2}{\partial \gamma_+} & -\frac{\partial h_2}{\partial \gamma_-} \end{pmatrix} \begin{pmatrix} \delta m(t) \\ \delta\gamma_+(t) \\ \delta\gamma_-(t) \end{pmatrix}$$
$$+ \begin{pmatrix} \frac{\partial h_1}{\partial \epsilon_+} & -\frac{\partial h_1}{\partial \epsilon_-} \\ \frac{\partial h_2}{\partial \epsilon_+} & -\frac{\partial h_2}{\partial \epsilon_-} \end{pmatrix} \begin{pmatrix} \delta\epsilon_+(t) \\ \delta\epsilon_-(t) \end{pmatrix}, \quad t = 0, \ldots, T_c$$

where the variables $\delta\gamma_+(t) \geq 0$ and $\delta\gamma_-(t) \geq 0$ represent the deviations from $\bar{\gamma}(t)$ in the positive and negative directions respectively and the functions h_1 and h_2 and their partial derivatives are evaluated at $\bar{m}(t), \bar{\gamma}(t), \bar{\epsilon}(t)$. Similarly $\delta\epsilon_+(t) \geq 0$ and $\delta\epsilon_-(t) \geq 0$ stand for the amount of positive and negative deviations from $\bar{\epsilon}(t)$. We associate $\delta\gamma_+(t)$ with $\delta\epsilon_+(t)$ and $\delta\gamma_-(t)$ with $\delta\epsilon_-(t)$, as an increase in $\bar{\epsilon}(t)$ means a positive deviation from $\bar{\gamma}(t)$, while a decrease in $\bar{\epsilon}(t)$ implies a negative deviation from $\bar{\gamma}(t)$.

From (20) we know that the first term on the left hand side of the above expression equals the first term on the right hand side, thus we can cancel these terms. Next we rearrange the matrices to separate the states $\delta\gamma_+(t)$ and $\delta\gamma_-(t)$, obtaining the following dynamics for $\delta m(t)$, $\delta\gamma_+(t)$ and $\delta\gamma_-(t)$,

$$\begin{pmatrix} \delta m(t+1) \\ \delta\gamma_+(t+1) \\ \delta\gamma_-(t+1) \end{pmatrix} = D_1(t) \begin{pmatrix} \delta m(t) \\ \delta\gamma_+(t) \\ \delta\gamma_-(t) \end{pmatrix} + D_2(t) \begin{pmatrix} \delta\epsilon_+(t) \\ \delta\epsilon_-(t) \end{pmatrix}, \quad t = 0, \ldots, T_c \tag{21}$$

where

$$D_1(t) = \begin{pmatrix} \frac{\partial h_1}{\partial m} & \frac{\partial h_1}{\partial \gamma_+} & -\frac{\partial h_1}{\partial \gamma_-} \\ 0 & \frac{\partial h_2}{\partial \gamma_+} & 0 \\ 0 & 0 & \frac{\partial h_2}{\partial \gamma_-} \end{pmatrix}, \quad D_2(t) = \begin{pmatrix} \frac{\partial h_1}{\partial \epsilon_+} & -\frac{\partial h_1}{\partial \epsilon_-} \\ \frac{\partial h_2}{\partial \epsilon_+} & 0 \\ 0 & \frac{\partial h_2}{\partial \epsilon_-} \end{pmatrix} \tag{22}$$

and partial derivatives are evaluated at $\bar{m}(t), \bar{\gamma}(t)$ and $\bar{\epsilon}(t)$. The function h_1 depends on the parameters $\gamma(t)$ and $\epsilon(t)$ through the matrix $B((1+\epsilon(t))\gamma(t))$ whose entries are given by (8). Thus, the partial derivatives of h_1 with respect to $\gamma(t)$ and $\epsilon(t)$ requires the partial derivatives of the matrix $B((1+\epsilon(t))\gamma(t))$ with respect to these variables. Let the matrices $\Delta_{\gamma_+}B$, $\Delta_{\gamma_-}B$, $\Delta_{\epsilon_+}B$ and $\Delta_{\epsilon_-}B$ denote the partial left and right derivatives of the entries of the matrix $B((1+\epsilon(t))\gamma(t))$ with respect to γ and ϵ evaluated at $\bar{\epsilon}(t)$ and $\bar{\gamma}(t)$. For simplicity in notation also denote the matrix $B((1+\bar{\epsilon}(t))\bar{\gamma}(t))$ with B, then the entries of the matrices $D_1(t)$ and $D_2(t)$ are explicitly defined as follows,

$$\frac{\partial h_1}{\partial m} = \frac{B\ A}{p^T BA\bar{m}(t)} - \frac{BAp^T BA}{\left(p^T BA\bar{m}(t)\right)^2}$$
$$\frac{\partial h_1}{\partial \gamma_+} = \frac{\Delta_{\gamma_+} BA\bar{m}(t)}{p^T BA\bar{m}(t)} - \frac{BA\bar{m}(t)p^T \Delta_{\gamma_+} BA\bar{m}(t)}{\left(p^T BA\bar{m}(t)\right)^2}$$

$$\frac{\partial h_1}{\partial \gamma_-} = \frac{\Delta_{\gamma_-} BA\bar{m}(t)}{p^T BA\bar{m}(t)} - \frac{BA\bar{m}(t)p^T \Delta_{\gamma_-} BA\bar{m}(t)}{\left(p^T BA\bar{m}(t)\right)^2}$$

$$\frac{\partial h_2}{\partial \gamma_+} = 1 + \bar{\epsilon}(t), \quad \frac{\partial h_2}{\partial \gamma_-} = 1 + \bar{\epsilon}(t)$$

$$\frac{\partial h_1}{\partial \epsilon_+} = \frac{\Delta_{\epsilon_+} BA\bar{m}(t)}{p^T BA\bar{m}(t)} - \frac{BA\bar{m}(t)p^T \Delta_{\epsilon_+} BA\bar{m}(t)}{\left(p^T BA\bar{m}(t)\right)^2}$$

$$\frac{\partial h_1}{\partial \epsilon_-} = \frac{\Delta_{\epsilon_-} BA\bar{m}(t)}{p^T BA\bar{m}(t)} - \frac{BA\bar{m}(t)p^T \Delta_{\epsilon_-} BA\bar{m}(t)}{\left(p^T BA\bar{m}(t)\right)^2}$$

$$\frac{\partial h_2}{\partial \epsilon_+} = \bar{\gamma}(t), \quad \frac{\partial h_2}{\partial \epsilon_-} = \bar{\gamma}(t).$$

By (21) we have the dynamics of the states $\delta m(t)$, $\delta\gamma_+(t)$ and $\delta_-\gamma(t)$. The initial conditions of these states are

$$\begin{pmatrix} \delta m(0) \\ \delta\gamma_+(0) \\ \delta\gamma_-(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{23}$$

Due to the perturbations the value of the objective functional at $T_c + 1$ change and we can approximate this change by a second order Taylor expansion of $\Lambda(m(t), \gamma(t))$ at $\bar{m}(T_c + 1)$ and $\bar{\gamma}(T_c + 1)$. Let $\delta\gamma(T_c + 1) = \delta\gamma_+(T_c + 1) - \delta\gamma_-(T_c + 1)$ then

$$\begin{aligned} &\Lambda\left(\bar{m}(T_c+1) + \delta m(T_c+1), \bar{\gamma}(T_c+1) + \delta\gamma(T_c+1)\right) \\ &\quad -\Lambda\left(\bar{m}(T_c+1), \bar{\gamma}(T_c+1)\right) \\ &\quad = \begin{pmatrix} \delta m(T_c+1) \\ \delta\gamma_+(T_c+1) \\ \delta\gamma_-(T_c+1) \end{pmatrix}^T C_1 \begin{pmatrix} \delta m(T_c+1) \\ \delta\gamma_+(T_c+1) \\ \delta\gamma_-(T_c+1) \end{pmatrix} + C_2 \begin{pmatrix} \delta m(T_c+1) \\ \delta\gamma_+(T_c+1) \\ \delta\gamma_-(T_c+1) \end{pmatrix} \end{aligned} \tag{24}$$

where

$$C_1 = \frac{1}{2}\begin{pmatrix} \frac{\partial^2\Lambda}{\partial m^2} & \frac{\partial^2\Lambda}{\partial m \partial\gamma_+} & -\frac{\partial^2\Lambda}{\partial m \partial\gamma_-} \\ \frac{\partial^2\Lambda}{\partial\gamma_+ \partial m} & \frac{\partial^2\Lambda}{\partial\gamma_+^2} & -\frac{\partial^2\Lambda}{\partial\gamma_+ \partial\gamma_-} \\ -\frac{\partial^2\Lambda}{\partial\gamma_- \partial m} & -\frac{\partial^2\Lambda}{\partial\gamma_- \partial\gamma_+} & \frac{\partial^2\Lambda}{\partial\gamma_-^2} \end{pmatrix}, \quad C_2 = \begin{pmatrix} \frac{\partial\Lambda}{\partial m} \\ \frac{\partial\Lambda}{\partial\gamma_+} \\ -\frac{\partial\Lambda}{\partial\gamma_-} \end{pmatrix}^T. \tag{25}$$

All the partial derivatives of $\Lambda(m(t), \gamma(t))$ in the definition of C_1 and C_2 are evaluated at $\bar{m}(T_c + 1)$ and $\bar{\gamma}(T_c + 1)$ which we did not indicate above for ease in notation.

We would like $\delta\epsilon_+(t)$ and $\delta\epsilon_-(t)$ to minimize the left hand side of expression (24); we want the optimal reduction in the terminal cost. For this we can solve an optimal control problem where $\delta\epsilon_+(t)$ and $\delta\epsilon_-(t)$ are the control variables and the state variables are $\delta m(t)$, $\delta\gamma_+(t)$ and $\delta\gamma_-(t)$. These states follow the dynamics in Eqs. 21 and 23. The objective to optimize is the reduction in the terminal cost $\Lambda(\cdot)$ which is approximated in (24). Because in the original problem $\epsilon(t)$ is box constrained, i.e. $\epsilon(t) \in [\epsilon_{\min}, \epsilon_{\max}]$, the perturbations $\delta\epsilon_+(t)$ and $\delta\epsilon_-(t)$ should belong to the set $[0, \delta\epsilon^+_{\max}(t)]$ and $[0, \delta\epsilon^-_{\max}(t)]$, such that

$$\delta\epsilon^+_{\max}(t) = \epsilon_{\max} - \epsilon(t), \quad \delta\epsilon^-_{\max}(t) = \epsilon(t) - \epsilon_{\min}. \tag{26}$$

This derivation leads to a discrete quadratic optimal control problem which has the following form,

Control Problem 2

$$\min_{\substack{0\le\delta\epsilon_+(t)\le\delta\epsilon^+_{\max}(t)\\ 0\le\delta\epsilon_-(t)\le\delta\epsilon^-_{\max}(t)\\ t=0,\ldots,T_c}} \begin{pmatrix}\delta m(T_c+1)\\ \delta\gamma_+(T_c+1)\\ \delta\gamma_-(T_c+1)\end{pmatrix}^T C_1 \begin{pmatrix}\delta m(T_c+1)\\ \delta\gamma_+(T_c+1)\\ \delta\gamma_-(T_c+1)\end{pmatrix} + C_2 \begin{pmatrix}\delta m(T_c+1)\\ \delta\gamma_+(T_c+1)\\ \delta\gamma_-(T_c+1)\end{pmatrix} \tag{27}$$

subject to

$$\begin{pmatrix}\delta m(t+1)\\ \delta\gamma_+(t+1)\\ \delta\gamma_-(t+1)\end{pmatrix} = D_1(t)\begin{pmatrix}\delta m(t)\\ \delta\gamma_+(t)\\ \delta\gamma_-(t)\end{pmatrix} + D_2(t)\begin{pmatrix}\delta\epsilon_+(t)\\ \delta\epsilon_-(t)\end{pmatrix}, \quad t=0,\ldots,T_c \tag{28}$$

$$\begin{pmatrix}\delta m(0)\\ \delta\gamma_+(0)\\ \delta\gamma_-(0)\end{pmatrix} = \begin{pmatrix}0\\0\\0\end{pmatrix} \tag{29}$$

where $D_1(t)$ and $D_2(t)$ are defined by (22), C_1 and C_2 are defined in (25), and $\delta\epsilon^+_{\max}(t)$ and $\delta\epsilon^-_{\max}(t)$ are given in (26).

The matrices $D_1(t)$, $D_2(t)$, C_1 and C_2 have to be estimated to solve this control problem. The notation $\Theta(\tau)$ used in Sect. 2 represents all these matrices, i.e. $\Theta(\tau) = \{C_1, C_2, D_1(t), D_2(t)$ for $t = 0,\ldots,T_c\}$ and the *parameter estimation* step in the definition of the function $K(\mathfrak{F}_\tau)$ is responsible for this task. To estimate these matrices it is necessary to calculate the matrices A, B, $\Delta_{\gamma_+}B$, $\Delta_{\gamma_-}B$, $\Delta_{\epsilon_+}B$, $\Delta_{\epsilon_-}B$ and the vectors p and F. We utilize the new information $\mathfrak{F}_\tau$ and use a least squares approach, see Appendix A.

Once we have $D_1(t)$, $D_2(t)$, C_1 and C_2, we solve the control problem to obtain $\delta\epsilon^*_+(0)$, $\delta\epsilon^*_-(0),\ldots,\delta\epsilon^*_+(T_c)$, $\delta\epsilon^*_-(T_c)$. In order to solve the control problem we structure it as a standard constrained quadratic optimization problem. Using the optimal solution we set $\epsilon^*(t) = \bar\epsilon(t) + \delta\epsilon^*_+(t) - \delta\epsilon^*_-(t)$ for $t = 0,\ldots,T_c$. In the interacting-particle algorithm, we set

$$\epsilon_\tau = K(\mathfrak{F}_\tau) = \epsilon^*(0) = \bar\epsilon(0) + \delta\epsilon^*_+(0) - \delta\epsilon^*_-(0)$$

and implement the N-particle selection mechanism with the new temperature $\gamma_\tau = (1+\epsilon_\tau)\gamma_{\tau-1}$.

In the following section we implement the function $K(\mathfrak{F}_\tau)$ defined with our meta-control methodology on test problems.

5 Numerical results

We have implemented the meta-control methodology and applied it to several test problems which are defined in $\mathbb{R}^d$. We have chosen a set of test problems from [1] which include the hardest ones and the ones defined in 10 and 20 dimensions. For defining the multi-variate Legendre polynomials $P_\alpha(x)$, we use up to 5th order single variate Legendre polynomials per dimension, i.e. $0 \le \upsilon_k \le 5$ for $k = 1,\ldots,d$. We solve Control Problem 2 with the Matlab quadratic programming function $quadprog(\cdot)$, and the parameters are estimated using the Matlab least squares function $lsqr(\cdot)$.

Initially we would like to illustrate the behavior of the meta-control methodology for different weights $(\omega_1,\omega_2,\omega_3)$ in the control objectives. For the following results we set the horizon of the control problem to $T_c = 10$, thus the meta-control methodology assesses the state of the samplings for the subsequent 10 iterations. The initial temperature is set to

$\gamma_0 = 2$ for all problems. The bounds for the control variable $\epsilon(t)$ are set to $\epsilon_{\min} = -0.05$ and $\epsilon_{\max} = 0.05$.

In Figs. 3 and 6 we use the normalized mean function values of the particles, defined as

$$g_\tau = \frac{f_\tau - f^*}{f_w - f^*} = \frac{1}{N}\sum_{i=1}^{N} \frac{f(y_\tau^i) - f^*}{f_w - f^*}. \tag{30}$$

where f_w is the worst function value observed during a run and $f^* = \min_{x\in\Omega} f(x)$ is the minimum value of the objective function over Ω. We normalize the mean function value of the particles to make the results on different functions comparable and consistent. With this normalization g_τ is scaled between zero and one. During the course of the algorithm, as the particles concentrate around the global optima, we expect $g_\tau \to 0$. We also report the incumbent normalized mean function values of the particles in the tables of Appendices C and D denoted by g_τ^* where

$$g_\tau^* = \min_{i=1,\ldots,\tau} g_i. \tag{31}$$

We first apply the methodology on the centered 20-dimensional sinusoidal problem which is stated in [1]. We chose the centered 20-dimensional sinusoidal problem as it was also used in our previous study of the interacting-particle algorithm in [13] and it is also used by other studies for SA and GA in [1] and [20]. We use the objective criterion in (18) and set ω_1 and ω_2 to different values. In addition we set the number of particles N to 1,000. This number of particles is chosen arbitrarily and we have chosen a large number of particles to reduce the noise in the observations of mean function values and coefficients of the parameters. In this section we are investigating the affects of ω_1, ω_2 and ω_3 on the algorithm rather than the number of particles. For each parameter setting we have implemented 50 independent runs. We have stopped a run when the normalized function value g_τ is less than the threshold 0.0005, i.e. $g_\tau \leq 0.0005$, or when the maximum number of 200 iterations is reached. This implies that the maximum number of function evaluations is 200,000 and the control problem is solved a maximum of 200 times. If a run stops at an iteration $s < 200$ due to the former stopping rule we assume $g_\tau = g_s, \epsilon_\tau = 0$ and $\gamma_\tau = \gamma_s, \epsilon_\tau = 0$ and $\gamma_\tau = \gamma_s$ for the remaining iterations $\tau > s$.

When $\omega_1 = 1$ and $\omega_2 = 0$ the control problem only considers the expectation of the objective function and the temperature cools down very fast as shown in Fig. 1. The graph shows the progress of the temperature value averaged over 50 independent runs. As we increase ω_2 to 0.5 and 0.75 we get the other results shown in Fig. 1. The results for $(\omega_1, \omega_2) = (0.75, 0.25)$ are omitted from Figs. 1, 2, 3 and 4 as they resemble the results for the case $(\omega_1, \omega_2) = (1, 0)$. When $\omega_1 = 1$ and $\omega_1 = 0.75$ the main objective is to optimize the mean function value of the particles and the temperature cools down to zero quickly. In these cases the runs terminate prior to the 200 iterations limit as g_τ becomes less than 0.0005. However, we plot the graphs up to 200 iterations by assuming the temperature stays constant after the runs stop. On the other hand observe that when $\omega_1 = 0.5$ ($\omega_2 = 0.5$) and $\omega_1 = 0.25$ ($\omega_2 = 0.75$) the temperature never cools down to zero. The risk of cooling dominates the benefit of a low expected objection function value and the temperature stabilizes around 0.6. In these cases the runs terminate due to the maximum iteration limit.

In Fig. 2 we show the average ϵ_τ as given by our meta-control methodology. One can see that for $\omega_1 = 1$, ϵ_τ goes to its lower bound which is -0.05. In this case all of the 50 runs stop between 100 and 150 iterations. As ω_1 decreases, ω_2 increases and ϵ_τ starts to fluctuate and we begin to see positive values of ϵ_τ which implies heating in the temperature. When $\omega_2 = 0.5$ and $\omega_2 = 0.75$ we observe that the runs do not stop before 200 iterations.

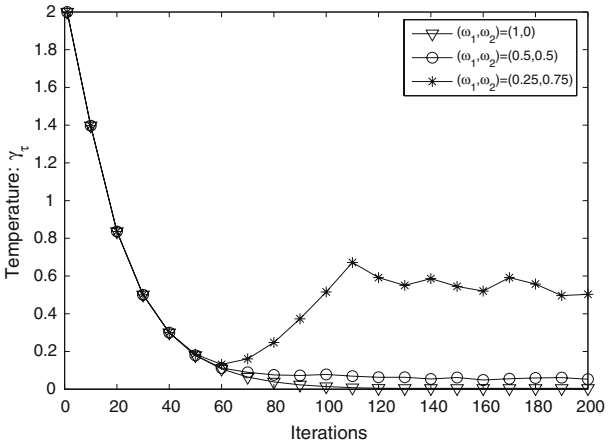

Fig. 1 Average temperature γ_τ over 50 independent runs for the 20-dimensional centered sinusoidal problem using the control objective in (18) with $(\omega_1, \omega_2) = (1, 0)$, $(0.5, 0.5)$ and $(0.25, 0.75)$

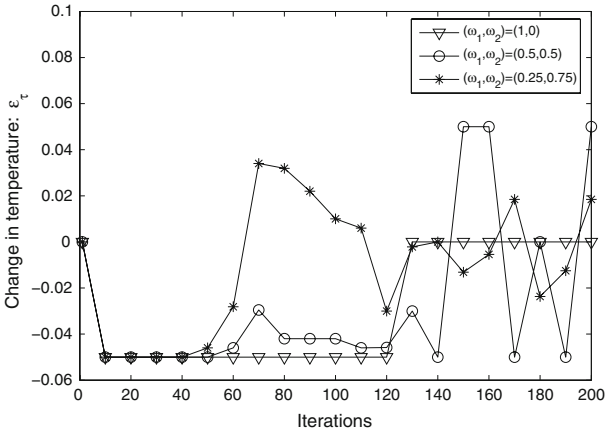

Fig. 2 Average fractional change in temperature ϵ_τ over 50 independent runs for the 20-dimensional centered sinusoidal problem using the control objective in (18) with $(\omega_1, \omega_2) = (1, 0)$, $(0.5, 0.5)$ and $(0.25, 0.75)$

In Figs. 3 and 4 we depict the progress of normalized mean function values and risk averaged over 50 runs with different ω_1 and ω_2. The tradeoff between mean function value and the risk is evident from these figures. Note that in the last parameter setting when $\omega_1 = 0.25$ and $\omega_2 = 0.75$ the particles do not converge towards the global optima because of the high weight we give to the risk.

In order to demonstrate the behavior of the interacting-particle algorithm further, we apply the methodology with objective criterion in (19) to the shifted sinusoidal problem which also appears in [1,13,20]. The centered sinusoidal problem was relatively easy to solve for the interacting-particle algorithm and it is not very sensitive to the parameter settings so in order to make the problem harder this time we shift the location of the global optima by $\pi/3$ and we increase the number of particles to $N = 2{,}000$. The number of particles is again set arbitrarily and our goal here is not to investigate the affects of N to our results. Once more we do 50 independent runs on the shifted sinusoidal problem and we stop each run when g_τ is less than the threshold 0.0005 or when 200 iterations limit is reached.

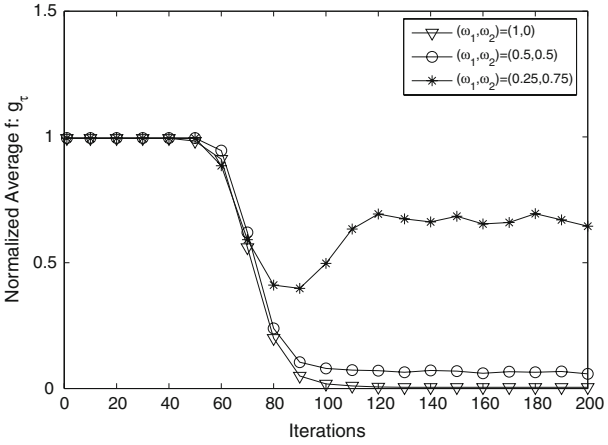

Fig. 3 Average normalized mean function value g_τ over 50 independent runs for the 20-dimensional centered sinusoidal problem using the objective in (18) with $(\omega_1, \omega_2) = (1, 0)$, $(0.5, 0.5)$ and $(0.25, 0.75)$

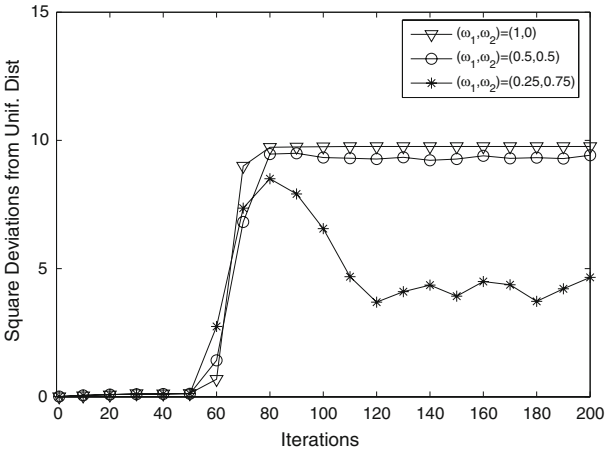

Fig. 4 Average square deviations from uniform distribution on Ω (risk) over 50 independent runs for the 20-dimensional centered sinusoidal problem using the objective in (18) with $(\omega_1, \omega_2) = (1, 0)$, $(0.5, 0.5)$ and $(0.25, 0.75)$

In this case we get more interesting results. As we increase ω_3 from 0.25 towards 1 the temperature starts to heat and cool more often as shown in Fig. 5. We omit the results for $\omega_3 = 0.75$ from this figure as they are similar to $\omega_3 = 1$. For better comparison we visualize the normalized mean function values of the particles in Fig. 6. In all parameter cases not all of the 50 runs were successful. However, we observe an improvement in the final mean objective function value due to increasing the risk factor. This case illustrates the importance of adding the risk in our criteria. When we only consider the mean objective function value of the particles, the cooling may be too fast. By considering the spread of the particles more, we are able to achieve better function values.

We extend our analysis to a larger problem set with 18 test problems from [1] where we chose the hardest problems or the ones in 10 and 20 dimensions. For each parameter setting and each problem we do 50 independent runs with the same stopping criteria as above, i.e. we stop when $g_\tau \leq 0.0005$ or 200 iterations is reached. Once again if a run stops at iteration $s < 200$ we set $g_\tau = g_s$ for the remaining iterations $\tau > s$. For these tests we report the incumbent g_τ values after 25, 100, 150 and 200 iterations as defined in (31). Again, we average g_τ^* over 50 independent runs. Note that the interacting-particle algorithm does N function evaluations for each iteration, so the number of function evaluations after τ iterations equals

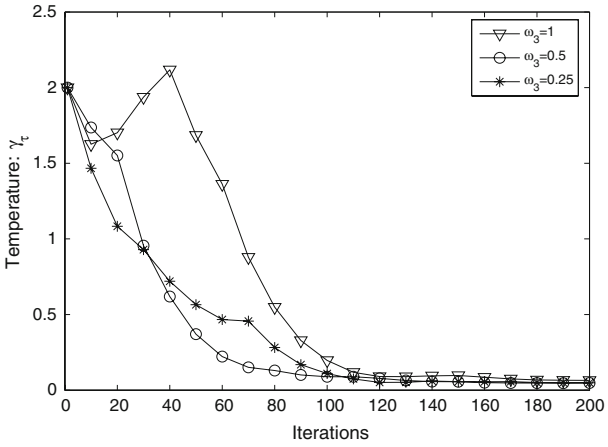

Fig. 5 Average temperature over 50 independent runs for the 20-dimensional shifted sinusoidal problem using the objective in (19) with $\omega_3 = 1, 0.5$ and 0.25

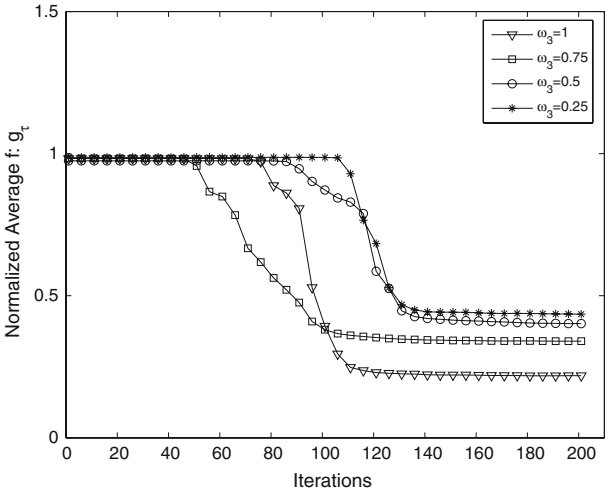

Fig. 6 Average normalized mean function value g_τ over 50 independent runs for the 20-dimensional shifted sinusoidal problem using the objective in (19) with $\omega_3 = 1, 0.75, 0.5$ and 0.25

τN. The number of particles N is chosen arbitrarily by trial and error and it is higher for difficult problems.

Appendix C gives the results of our numerical test with objective criterion (18). For some problems we observe the behavior illustrated in Fig. 3 and increasing ω_2 results in not converging or slower convergence towards the global optima. However for some of the test problems increasing ω_2 leads to better outcomes.

For 5 and 10 dimensional Michalewicks problems we obtain an improvement in the final average objective function of the particle by increasing ω_2 from 0 to 0.75. For these test problems we obtained the best outcome when $\omega_1 = 0.75$ and $\omega_2 = 0.25$. We observe analogous results for 10 and 20 dimensional shifted sinusoidal problems. The 10 dimensional Shekel's foxholes problem shows the most sensitivity to parameters of the control problem. For this problem we observe significant improvement in the final objective function of the particles by increasing ω_2.

Appendix D reports the test results with the objective criterion in (19). Our results indicate that increasing ω_3 to 1 has positive effects or no effect for all test problems. For the

Michalewicks problems, shifted sinusoidal problems and the Shekel's foxholes problem we see improvements in the results as we increase ω_3 towards 1. For the other test problems increasing $\omega_3 = 1$ does not affect final convergence towards the global optima.

With the objective criterion (19) we actually get more robust results than with the criterion (18). The advantage of the criterion in (19) is that it adjusts the scales of reward and risk by taking their ratio.

6 Comparison with classical cooling schedules and simulated annealing

In addition to the numerical experiments of Sect. 5 we performed a computational study comparing the interacting-particle algorithm with our meta-control methodology, the interacting-particle algorithm with two classical cooling schedules, and SA with the same two classical cooling schedules. Thus in this section we compare five algorithms to each other on test problems from Sect. 5. To denote the temperature parameter of SA at iteration τ we again use γ_τ. As classical cooling schedules we employ the exponential cooling schedule and logarithmic cooling schedule which also appear in the numerical experiments of [20]. We use the following parameter settings for these schedules

Exponential Cooling Schedule [24]

$$\gamma_\tau = \gamma_0 \mu^\tau$$
$$\gamma_0 = 1 \quad \text{and} \quad \mu = 0.99$$

Logarithmic Cooling Schedule [24]

$$\gamma_\tau = \frac{\gamma_0}{\ln(\tau + 1)}$$
$$\gamma_0 = 1.$$

For the meta-control methodology we use the control criterion in (19) with $\omega_3 = 1$. This control objective and parameter setting are chosen because in Sect. 5 they showed consistently good results for the test problems. We set the initial temperature to $\gamma_0 = 1$ and the maximum and minimum allowed fractional changes in temperature are $\epsilon_{\min} = -0.01$ and $\epsilon_{\max} = 0.01$. With these parameters the temperature update of the meta-control methodology is akin to the exponential cooling schedule but we also allow heating.

For the interacting-particle algorithm with classical cooling schedules and meta-control methodology we set the number of particles $N = 250$. The number of particles in this section is much less than the number of particles of the previous section. In Sect. 5 our aim was to investigate the behavior of the interacting-particle algorithm with different parameter settings and we wanted to minimize the affects of the number of particles in our observations. Thus in Sect. 5 we set N to large values. In the numerical experiments of this section we reduce N to 250 because in [13] this number of particles was used in comparing the interacting-particle algorithm with SA.

In summary we apply five different algorithms instances given as,

1. Interacting-particle algorithm with $N = 250$ and $\epsilon_\tau = K(\mathfrak{F}_\tau)$ calculated by the meta-control methodology using control objective (19) with $\omega_3 = 1$, $\epsilon_{\min} = -0.01$ and $\epsilon_{\max} = 0.01$.
2. Interacting-particle algorithm with $N = 250$ and exponential cooling schedule i.e. $\epsilon_\tau = K(\mathfrak{F}_\tau) = -0.01$.

3. Interacting-particle algorithm with $N = 250$ and logarithmic cooling schedule i.e. $\epsilon_\tau = K(\mathfrak{F}_\tau) = \frac{\ln(\tau)}{\ln(\tau+1)} - 1$.
4. SA with exponential cooling schedule i.e. $\gamma_\tau = 0.99^\tau$.
5. SA with logarithmic cooling schedule i.e. $\gamma_\tau = \frac{1}{\ln(\tau+1)}$.

For this comparison study we chose the 20 dimensional centered and shifted sinusoidal problems, 10 dimensional Michalewicks problem and 10 dimensional Shekel's foxholes problem. The sinusoidal functions are chosen because they also appeared in the previous comparison studies of [13] and [20]. The other two functions Shekel's foxholes and Michalewicks problem are used as they were shown to be the hardest problems of Sect. 5.

We do 50 independent runs for all of the five algorithms on four different problems. We stop each run when the maximum number of 1,000 iterations is reached for the interacting-particle algorithm and when 250,000 iterations is reached for SA. The interacting-particle algorithm uses $N = 250$ function evaluations per iteration while SA uses 1 function evaluation per iteration thus the maximum number of function evaluations for all of the five algorithms equals 250,000. To compare the algorithms we plot the incumbent normalized function value versus the number of function evaluations where we normalize using the global minima f^* of the functions and the worst value observed f_w. Using the normalized function value scales the values between 0 and 1 for all of the four test problems. We average the results over the 50 independent runs for each algorithm and test problem.

In Figs. 7–10 we plot the incumbent normalized function values versus the number of function evaluations of the five algorithms on the four test problems averaged over 50 independent runs. We can see that the interacting-particle algorithm with classical cooling schedules and meta-control methodology performs better than SA in all of the test problems. Because SA is doing one function evaluation per iteration the incumbent objective function value drops quickly however SA is more likely to get stuck in a local optima thus the average incumbent solution over 50 runs does not reach zero in any of the test functions. The average final outcome of the interacting-particle algorithm with classical cooling schedules is better than SA in all of the tests which is also consistent with the findings in [13] where the interacting-particle algorithm was numerically shown to be better than SA. As our numerical results suggest the meta-control methodology clearly brings a benefit to the interacting-particle algorithm. With a fluctuating temperature the particles achieve better function values.

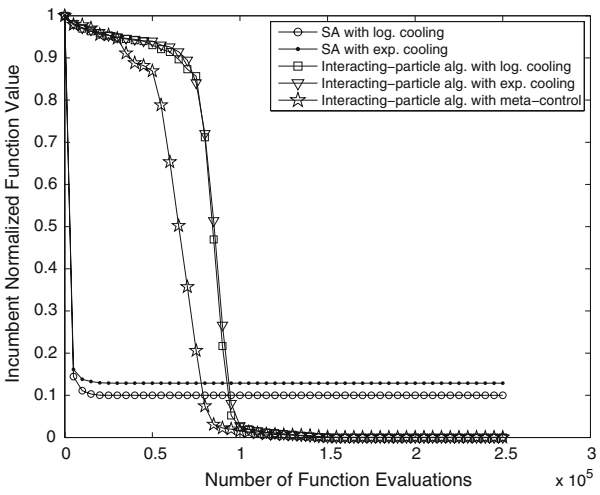

Fig. 7 Average incumbent normalized function value over 50 independent runs for the 20-dimensional centered sinusoidal problem using the five algorithms

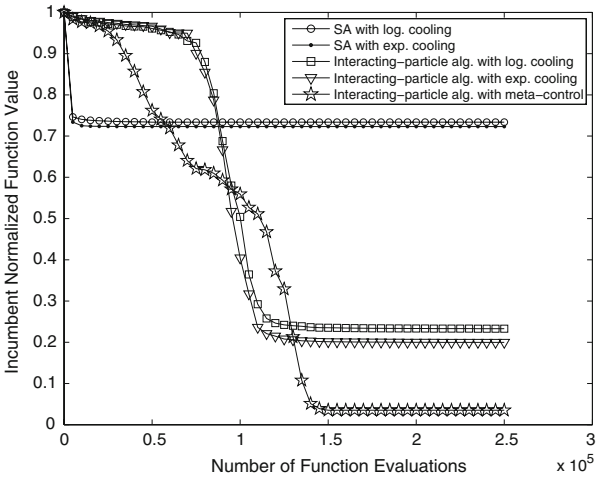

Fig. 8 Average incumbent normalized function value over 50 independent runs for the 20-dimensional centered sinusoidal problem using the five algorithms

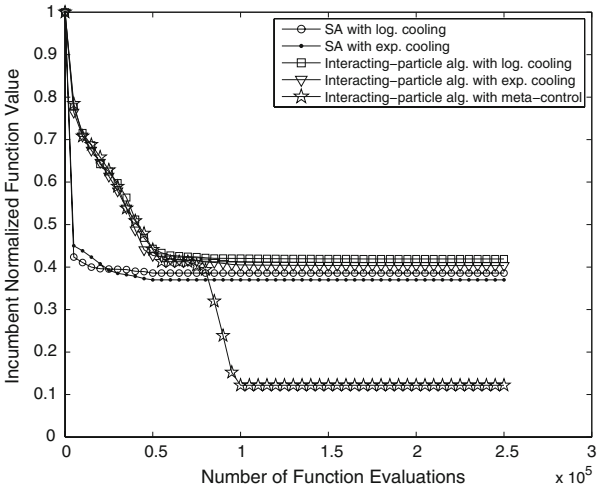

Fig. 9 Average incumbent normalized function value over 50 independent runs for the 10-dimensional Michalewicks problem using the five algorithms

Finally we report the CPU time needed to compute the function $K(\mathfrak{F}_\tau)$ by the meta-control methodology and compare it to the CPU time spend doing $N = 250$ function evaluations. We average the results from 1,000 iterations of the 50 independent runs. The times are for a Pentium 4 processor with 512 RAM.

The CPU time of the meta-control methodology is only affected by the dimension of the problem due to the fact that in higher dimensions we have more polynomials in our orthonormal basis so the number of parameters to estimate for the control problem increases. While the CPU time of the meta-control methodology is much higher than the CPU time of the function evaluations for these four simple functions, the CPU time of the meta-control methodology is independent of the complexity of the objective function because it does not include any function evaluations. Thus for problems whose objective function evaluations take longer CPU times, we expect the extra computation for the meta-control methodology to be overshadowed by the N function evaluations (Table 1).

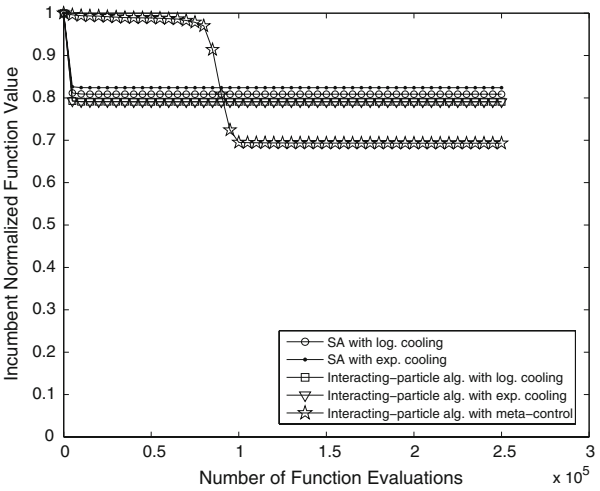

Fig. 10 Average incumbent normalized function value over 50 independent runs for the 10-dimensional Shekel's foxholes problem using the five algorithms

Table 1 Average CPU times for the meta-control methodology and $N = 250$ function evaluations

	Meta-control methodology	250-Function evaluations
Centered sinusoidal	1.943	0.0011
Shifted sinusoidal	1.95	0.0012
Michalewicks	0.734	0.00078
Shekel's foxholes	0.725	0.00015

7 Conclusion and future research

We have implemented a meta-control methodology for determining the temperature parameter of the interacting-particle algorithm. Our meta-control approach regards the interacting-particle algorithm as a discrete dynamical system which evolves with the samplings of the N-particle mechanisms. The p.d.f. of the particle's locations is considered as the state of the system and the evolution of this p.d.f. depends on the temperature parameter through the N-particle selection mechanism. By approximating the p.d.f. with a sequence of orthonormal polynomials we obtained an optimal control problem which has the fractional change in the temperature as a control variable.

In order to solve this control problem we have implemented a perturbation method which gave us a discrete quadratic optimal control problem. We estimated the parameters of the optimal control problem using the observations coming from the interacting-particle algorithm. Solving the control problem provides feedback on how to set the temperature parameter in order to optimize the performance of the samplings for the next T_c iterations.

Our numerical tests were carried out with two choices of objective criterion for the optimal control problem. While some of the problems do not show sensitivity to the weights in the control objective, with more difficult problems we obtained better results by considering the deviations from the uniform distribution in our criteria.

Clearly the performance of the meta-control methodology depends on the objective criterion. However, the methodology is not limited to the two control objectives which we used in our numerical tests. The performance of the methodology can be improved by designing more effective and efficient objective criterion. We leave this issue open for future research.

Another research direction is to consider parameterized exploration kernels and to apply the meta-control methodology to obtain feedback on the parameters of the exploration kernel.

Finally our comparison experiments indicate that there is a benefit in using the meta-control methodology to set the temperature parameter of the interacting-particle algorithm. For all of the four test problems of the comparison experiments we obtained consistently satisfying results for the meta-control methodology.

Acknowledgements This research was supported in part by National Science Foundation under the grant DMI-0244286.

Appendix A

The Control Problem 2 requires the parameters $\Theta(\tau) = \{C_1, C_2, D_1(t), D_2(t) \text{ for } t = 0, \ldots, T_c\}$. The *parameter estimation* step of the procedure for function $K(\mathfrak{F}_\tau)$ uses the historical information $\mathfrak{F}_\tau$ supplied by the interacting-particle algorithm in estimating these matrices. We can summarize the operations done by this step in a pseudo code form as follows,

Parameter estimation:

Step 1: Given y_τ and $\hat{y}_\tau$ estimate the new state of the interacting-particle algorithm $c(\tau)$ and $\hat{c}(\tau)$ using (3) and (4).
Step 2: Using $c(\tau)$ and $\hat{c}(\tau)$ estimate the matrix A.
Step 3: Using $f(y_\tau)$, $f(\hat{y}_\tau)$, $c(\tau)$ and $\hat{c}(\tau)$ estimate the vector F.
Step 4: Set $\bar{m}(0) = c(\tau)$ and $\bar{\gamma}(0) = \gamma_{\tau-1}$.
Step 5: *For* $t = 0$ *to* T_c

- Step 5.1: Let $\bar{\gamma}(t+1) = (1+\bar{\epsilon}(t))\bar{\gamma}(t)$. Using $\hat{y}_\tau$ and $f(\hat{y}_\tau)$ estimate the entries of the matrices B, $\Delta_{\gamma_+}B$, $\Delta_{\gamma_-}B$, $\Delta_{\epsilon_+}B$, $\Delta_{\epsilon_-}B$ and the vector p at $\bar{\gamma}(t+1)$.
- Step 5.2: Estimate $D_1(t)$ and $D_2(t)$ according to Eqs. 22. Calculate $\bar{m}(t+1)$ according to Eq. 20.

Step 6: Using $\bar{m}(T_c+1)$ and $\bar{\gamma}(T_c+1)$ estimate C_1 and C_2 using Eqs. 25. Send $\Theta(\tau) = \{C_1, C_2, D_1(t), D_2(t) \text{ for } t = 0, \ldots, T_c\}$ to the optimal control problem.

In Step 1 we obtain the particle locations y_τ and $\hat{y}_\tau$ and estimate the current states $c(\tau)$ and $\hat{c}(\tau)$ of the interacting-particle algorithm according to the Eq. 3. These states are used to estimate the matrix A in Step 2. The matrix A characterizes the change in the vector $m(t)$ due to N-particle exploration step; it relates the vector $m(t)$ to $\hat{m}(t+1)$ (Eq. 5). Using $c(\tau)$ and $\hat{c}(\tau)$ we estimate a matrix A such that

$$A = \arg\min_W \parallel Wc(\tau) - \hat{c}(\tau) \parallel \tag{32}$$

where $\parallel \cdot \parallel$ denotes the l^2 norm. We formulate this task as a least squares problem (see Appendix B). In fact, A by definition (6) stays constant for all τ. After some iterations we may choose not to update this matrix to reduce the computational effort.

The vector F estimated in Step 3 shows how the average function value of the particles changes with the vector $c(\tau)$, c.f. (15). Given the latest function values of particles at locations y_τ and $\hat{y}_\tau$, $f(y_\tau)$ and $f(\hat{y}_\tau)$ we estimate the average function values f_τ and $\hat{f}_\tau$ using (14). Then we solve a least squares problem and set

$$F = \arg\min_{v \in \mathbb{R}^Q} \|c(\tau)v - f_\tau\| + \|\hat{c}(\tau)v - \hat{f}_\tau\|. \tag{33}$$

The main goal of Step 5 is to estimate $D_1(t)$ and $D_2(t)$ for $t = 0, \ldots, T_c$. This requires estimation of matrices B, $\Delta_{\gamma^+}B$, $\Delta_{\gamma^-}B$, $\Delta_{\epsilon^+}B$, $\Delta_{\epsilon^-}B$ (cf. 22) and the vector p, thus in Step 5.1 these matrices are computed.

The matrix B characterizes the affect of N-particle selection mechanism on $m(t)$, see Eq. 7. In order to estimate the matrix $B\,(\bar{\gamma}(t+1))$ for $\bar{\gamma}(t+1) = (1+\bar{\epsilon}(t))\,\bar{\gamma}(t), t = 0, \ldots, T_c$ we use the latest intermediate particle locations and the corresponding function values at these locations, i.e. $\hat{y}_\tau$ and $f(\hat{y}_\tau)$. For each $t = 0, \ldots T_c$ we do the same calculations outlined below.

Assume the temperature γ_τ equals $\bar{\gamma}(t+1)$ at iteration τ. The N-particle selection mechanism moves the particle i from location $\hat{y}_\tau^i$ to the location $\hat{y}_\tau^j$ with probability $s_j(\bar{\gamma}(t+1))$ and the normalizing term of $s_j(\bar{\gamma}(t+1))$ equals $\sum_{k=1}^N G_{\bar{\gamma}(t+1)}(\hat{y}_{\tau+t}^k)$ which we denote with $Z(\bar{\gamma}(t+1))$. Therefore, if the N-particle mechanism is implemented at temperature $\bar{\gamma}(t+1)$, with probability $s_j(\bar{\gamma}(t+1))$ we expect the location $\hat{y}_\tau^j$ to be selected and the α^{th} orthonormal polynomial takes the value $P_\alpha(\hat{y}_\tau^j)$ at this location. An unbiased estimator of $c_\alpha(\tau+1)$ is

$$\mathbb{E}\,(c_\alpha(\tau+1)|\mathfrak{F}_\tau) = \sum_{j=1}^{N} s_j(\bar{\gamma}(t+1))\,P_\alpha(\hat{y}_\tau^j). \tag{34}$$

Let $\mathbb{E}\,(c(\tau+1))$ be the expected $c(\tau+1)$ if the temperature equals $\hat{\gamma}(t+1)$, as estimated using (34). We know by Eq. 7 that

$$\mathbb{E}\,(c(\tau+1)) = \frac{B(\bar{\gamma}(t+1))\hat{c}(\tau)}{Z(\bar{\gamma}(t+1))}. \tag{35}$$

With known $\mathbb{E}\,(c(\tau+1))$ and $Z(\gamma(t+1))$, we compute the matrix $B(\hat{\gamma}(t+1))$ by minimizing

$$\parallel B(\bar{\gamma}(t+1))\hat{c}(\tau) - Z(\bar{\gamma}(t+1))\mathbb{E}\,(c(\tau+1)) \parallel . \tag{36}$$

The minimization is once again modeled as a least squares problem (Appendix B).

The other matrices $\Delta_{\gamma^+}B$, $\Delta_{\gamma^-}B$, $\Delta_{\epsilon^+}B$ and $\Delta_{\epsilon^-}B$ are the right and left partial derivatives of the entries of matrix B with respect to γ and ϵ. In order to calculate these matrices at temperatures $\bar{\gamma}(t+1) = (1+\bar{\epsilon}(t))\,\bar{\gamma}(t), t = 0, \ldots, T_c$ we do similar operations as in the calculation of the matrix B. Below we only discuss the estimation of $\Delta_\gamma^+ B$ for any t in details; the calculations for the other three matrices involve the analogous operations.

Let $\bar{\gamma}^+(t+1) = (1+\bar{\epsilon}(t))\,(\bar{\gamma}(t)+u_\gamma)$ be the temperature value when $\bar{\gamma}(t)$ is increased by infinitesimal amount $u_\gamma > 0$. Using Eq. 34 we estimate expected vector $c(\tau+1)$ if the temperature equals $\bar{\gamma}^+(t+1)$ at the N-particle selection mechanism. We denote this expectation with $\mathbb{E}\left(c^+(\tau+1)\right)$. In addition we have the normalizing term $Z(\bar{\gamma}^+(t+1)) = \sum_{k=1}^N G_{\bar{\gamma}^+(t+1)}(\hat{y}_{\tau+t}^k)$. We first estimate a matrix $B(\bar{\gamma}^+(t+1))$ by minimizing

$$\parallel B(\bar{\gamma}^+(t+1))\hat{c}(\tau) - Z(\bar{\gamma}^+(t+1))\mathbb{E}\left(c^+(\tau+1)\right) \parallel, \tag{37}$$

which is again a least squares problem. This gives us the matrix $B(\bar{\gamma}^+(t+1))$. To estimate $\Delta_{\gamma^+}B(\bar{\gamma}(t+1))$ we set

$$\Delta_{\gamma^+}B(\bar{\gamma}(t+1)) = \frac{B(\bar{\gamma}^+(t+1)) - B(\bar{\gamma}(t+1))}{u_\gamma} \tag{38}$$

where $B(\bar{\gamma}(t+1))$ comes from expression (36). For the other three matrices we do the calculations in (34), (37) and (38) with temperatures $\bar{\gamma}^-(t+1) = (1+\bar{\epsilon}(t))\,(\bar{\gamma}(t)-u_\gamma)$, $\bar{\gamma}^{\epsilon+}(t+1) = (1+\bar{\epsilon}(t)+u_\epsilon)\,\bar{\gamma}(t)$ and $\bar{\gamma}^{\epsilon-}(t+1) = (1+\bar{\epsilon}(t)-u_\epsilon)\,\bar{\gamma}(t)$.

The vector p is a part of the normalizing terms $Z(\hat{\gamma}(t+1))$ for $t = 1, \ldots, T_c$. Given $\hat{c}(\tau)$ and $B(\bar{\gamma}(t+1))$ from our previous calculations we let p to be

$$p = \arg\min_{v \in \mathbb{R}^Q} \sum_{t=0}^{T_c} \|\hat{c}(\tau)^T B(\bar{\gamma}(t+1))^T v - Z(\bar{\gamma}(t+1))\|.$$

In Step 5.2 we estimate $D_1(t)$ and $D_2(t)$ according to Eqs. 22. Finally in Step 6 we calculate C_1 and C_2 according to Eqs. 25. All the parameters are sent to the control problem to obtain a feedback on the temperature. In the following section we implement the function $K(\mathfrak{F}_\tau)$ defined with our meta-control methodology on test problems.

Appendix B

Suppose numerical values for two sets of Q dimensional vectors are given, $q_s = (q_s^1, \ldots, q_s^Q)^T$ and $p_s = (p_s^1, \ldots, p_s^Q)^T$ for $s = 1, \ldots, z$. We would like to estimate a $Q \times Q$ symmetric matrix W such that it satisfies

$$W q_s = p_s,$$

for all $s = 1, \ldots, z$. We use a least squares approach and consider this problem as a minimization problem; we would like W to minimize

$$\sum_{s=1}^{z} \|W q_s - p_s\|,$$

where $\|\cdot\|$ denotes the l^2 norm. We structure this minimization problem as a standard least squares problem by representing the entries of matrix W as a vector. The decision variable becomes the $w = Q(Q+1)/2$ dimensional vector such that

$$v = (W_{11}, W_{12}, \ldots, W_{1K}, W_{22}, \ldots, W_{2K}, \ldots, W_{KK})^T.$$

The number of variables to estimate is reduced from Q^2 to w because of the symmetry of the matrix W.

From the vectors $q_s, s = 1, \ldots, z$ we construct a $(zQ) \times w$ matrix L such that

$$L = \begin{pmatrix} L_1 \\ L_2 \\ \vdots \\ L_z \end{pmatrix}.$$

where $L_s, s = 1, \ldots, z$ are $Q \times w$ matrices defined as

$$L_s = \begin{pmatrix} q_s^1 & q_s^2 & q_s^3 & \ldots & q_s^Q & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & q_s^1 & 0 & \ldots & 0 & q_s^2 & q_s^3 & \ldots & q_s^Q & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 0 & q_s^1 & \ldots & 0 & 0 & q_s^2 & \ldots & 0 & q_s^3 & \ldots & q_s^Q & 0 & \ldots & 0 \\ \vdots & \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots & & \ddots & & \vdots & & \vdots \\ 0 & 0 & \ldots & \ldots & q_s^1 & 0 & \ldots & \ldots & q_s^2 & 0 & \ldots & q_s^3 & 0 & \ldots & q_s^Q \end{pmatrix} \quad \text{for } s = 1, \ldots, z.$$

Using the vector $p_s, s = 1, \ldots, z$ we construct a vector r where

$$r = \left(p_1^1, \ldots, p_1^Q, p_2^1, \ldots, p_2^Q, \ldots, p_z^1, \ldots, p_z^Q\right)^T.$$

Next we let

$$v^* = \arg\min_{v \in \mathbb{R}^w} \|Lv - r\|,$$

and the vector v^* provides the optimal matrix W.

Appendix C

Average test results for meta-control methodology using the objective criterion in (18), with differing ω_1 and ω_2

Test function (dimension)	(ω_1, ω_2)	N	Avg g^*_{25}	Avg g^*_{100}	Avg g^*_{200}
Centered sinusodial (10)	(1, 0)	1,000	0.6144	0.00653	0.00479
	(0.75, 0.25)	1,000	0.636613	0.00472	0.004583
	(0.5, 0.5)	1,000	0.65007	0.00551	0.004717
	(0.25, 0.75)	1,000	0.67329	0.67329	0.69349
Shifted sinusodial (10)	(1, 0)	1,250	0.44755	0.00472	0.00472
	(0.75, 0.25)	1,250	0.40755	0.00485	0.004765
	(0.5, 0.5)	1,250	0.500963	0.21401	0.213623
	(0.25, 0.75)	1,250	0.47938	0.47938	0.47938
Michalewicks (5)	(1, 0)	2,000	0.423315	0.04379	0.04108
	(0.75, 0.25)	2,000	0.41548	0.0076	0.00629
	(0.5, 0.5)	2,000	0.36463	0.02518	0.018273
	(0.25, 0.75)	2,000	0.379493	0.02715	0.024427
Michalewicks (10)	(1, 0)	5,000	0.39831	0.30245	0.301675
	(0.75, 0.25)	5,000	0.36228	0.21312	0.21083
	(0.5, 0.5)	5,000	0.429697	0.4297	0.61291
	(0.25, 0.75)	5,000	0.47938	0.47938	0.656757
Levy and Montolva (10)	(1, 0)	1,000	0.027285	0.00467	0.004665
	(0.75, 0.25)	1,000	0.02717	0.00477	0.00477
	(0.5, 0.5)	1,000	0.02335	0.02112	0.01842
	(0.25, 0.75)	1,000	0.067647	0.06765	0.067647
Rosenbrock (10)	(1, 0)	1,000	0.00402	0.00402	0.00402
	(0.75, 0.25)	1,000	0.00447	0.00447	0.00447
	(0.5, 0.5)	1,000	0.004473	0.00447	0.004473
	(0.25, 0.75)	1,000	0.00416	0.00416	0.00416
Griewank (10)	(1, 0)	1,000	0.179315	0.05081	0.044015
	(0.75, 0.25)	1,000	0.013197	0.00444	0.00444
	(0.5, 0.5)	1,000	0.012367	0.00418	0.004183
	(0.25, 0.75)	1,000	0.012713	0.0042	0.004203
Pavianni (10)	(1, 0)	1,000	0.006285	0.00488	0.00488
	(0.75, 0.25)	1,000	0.0064	0.00469	0.004693
	(0.5, 0.5)	1,000	0.00524	0.00476	0.00476
	(0.25, 0.75)	1,000	0.014187	0.01419	0.014187
Shekel's foxholes (10)	(1, 0)	5,000	0.990673	0.86153	0.861218
	(0.75, 0.25)	5,000	0.990668	0.77497	0.774755
	(0.5, 0.5)	5,000	0.990946	0.69423	0.694485
	(0.25, 0.75)	5,000	0.990287	0.58976	0.588094
Schewel (10)	(1, 0)	1,000	0.25803	0.05802	0.055175
	(0.75, 0.25)	1,000	0.273747	0.06531	0.06257
	(0.5, 0.5)	1,000	0.259227	0.05038	0.045507
	(0.25, 0.75)	1,000	0.449793	0.44979	0.449793

Test function (dimension)	(ω_1, ω_2)	N	Avg g^*_{25}	Avg g^*_{100}	Avg g^*_{200}
Salomon (10)	(1, 0)	1,000	0.074583	0.00634	0.006243
	(0.75, 0.25)	1,000	0.073377	0.00545	0.00422
	(0.5, 0.5)	1,000	0.076573	0.00738	0.005993
	(0.25, 0.75)	1,000	0.273747	0.00653	0.006257
Centered sinusodial (20)	(1, 0)	2,000	0.994671	0.01685	0.004649
	(0.75, 0.25)	2,000	0.993314	0.01728	0.004786
	(0.5, 0.5)	2,000	0.995575	0.07968	0.058229
	(0.25, 0.75)	2,000	0.994828	0.45331	0.645031
Shifted sinusodial (20)	(1, 0)	2,500	0.97723	0.51412	0.4959
	(0.75, 0.25)	2,500	0.988501	0.72092	0.836949
	(0.5, 0.5)	2,500	0.959941	0.74705	0.747045
	(0.25, 0.75)	2,500	0.989539	0.7579	0.757895
Ackley (20)	(1, 0)	2,000	0.451175	0.07252	0.07005
	(0.75, 0.25)	2,000	0.430515	0.04083	0.034306
	(0.5, 0.5)	2,000	0.420946	0.04686	0.039612
	(0.25, 0.75)	2,000	0.426518	0.04953	0.038898
Levy and Montolva (20)	(1, 0)	2,000	0.022215	0.00469	0.004685
	(0.75, 0.25)	2,000	0.026747	0.00482	0.004816
	(0.5, 0.5)	2,000	0.026723	0.00554	0.004862
	(0.25, 0.75)	2,000	0.079014	0.04194	0.041937
Rosenbrock (20)	(1, 0)	2,000	0.00478	0.00478	0.00478
	(0.75, 0.25)	2,000	0.004584	0.00458	0.004584
	(0.5, 0.5)	2,000	0.004336	0.00434	0.004336
	(0.25, 0.75)	2,000	0.004294	0.00429	0.004294
Griewank (20)	(1, 0)	2,000	0.01055	0.00497	0.004975
	(0.75, 0.25)	2,000	0.010011	0.00495	0.004954
	(0.5, 0.5)	2,000	0.009455	0.00494	0.004935
	(0.25, 0.75)	2,000	0.010034	0.00495	0.004952
Salomon (20)	(1, 0)	2,000	0.102595	0.01345	0.010395
	(0.75, 0.25)	2,000	0.09511	0.0132	0.008655
	(0.5, 0.5)	2,000	0.11357	0.01303	0.01005
	(0.25, 0.75)	2,000	0.118845	0.01268	0.011115

Appendix D

Average test results for meta-control methodology using the objective criterion in (19), with differing ω_3

Test function (dimension)	(ω_3)	N	Avg g^*_{25}	Avg g^*_{100}	Avg g^*_{200}
Centered sinusoidal (10)	(0.25)	1,000	0.588157	0.005107	0.004243
	(0.5)	1,000	0.67146	0.006913	0.004687
	(0.75)	1,000	0.681357	0.00497	0.004853
	(1)	1,000	0.748223	0.00834	0.004723
Shifted sinusoidal (10)	(0.25)	1,250	0.39391	0.005293	0.004783
	(0.5)	1,250	0.437413	0.005323	0.00448
	(0.75)	1,250	0.432497	0.006087	0.004383
	(1)	1,250	0.40904	0.005197	0.004853
Michalewicks (5)	(0.25)	2,000	0.62922	0.032723	0.028007
	(0.5)	2,000	0.424303	0.032127	0.028237
	(0.75)	2,000	0.351807	0.013707	0.009533
	(1)	2,000	0.377197	0.022827	0.017397
Michalewicks (10)	(0.25)	5,000	0.410933	0.294833	0.29416
	(0.5)	5,000	0.37869	0.28454	0.283073

Test function (dimension)	(ω_3)	N	Avg g^*_{25}	Avg g^*_{100}	Avg g^*_{200}
	(0.75)	5,000	0.401313	0.286197	0.276813
	(1)	5,000	0.39489	0.10933	0.10661
Levy and Montolvo (10)	(0.25)	1,000	0.028517	0.004683	0.004683
	(0.5)	1,000	0.023823	0.004713	0.004713
	(0.75)	1,000	0.029903	0.00487	0.00487
	(1)	1,000	0.030783	0.00495	0.00495
Rosenbrock (10)	(0.25)	1,000	0.004263	0.004263	0.004263
	(0.5)	1,000	0.004447	0.004447	0.004447
	(0.75)	1,000	0.00425	0.00425	0.00425
	(1)	1,000	0.00441	0.00441	0.00441
Griewank	(0.25)	1,000	0.013457	0.00462	0.00462
	(0.5)	1,000	0.01328	0.004313	0.004313
	(0.75)	1,000	0.01347	0.004673	0.004673
	(1)	1,000	0.0126	0.00469	0.00469
Pavianni (10)	(0.25)	1,000	0.00648	0.004773	0.004773
	(0.5)	1,000	0.006053	0.00483	0.00483
	(0.75)	1,000	0.006463	0.004887	0.004887
	(1)	1,000	0.006963	0.004803	0.004803
Shekel's foxholes (10)	(0.25)	5,000	0.990903	0.825623	0.824247
	(0.5)	5,000	0.991333	0.865898	0.829973
	(0.75)	5,000	0.991673	0.895638	0.807884
	(1)	5,000	0.991318	0.781973	0.78015
Schewel (10)	(0.25)	1,000	0.26681	0.078167	0.075293
	(0.5)	1,000	0.234563	0.063987	0.061277
	(0.75)	1,000	0.265193	0.066923	0.06415
	(1)	1,000	0.253187	0.041027	0.0382
Salomon (10)	(0.25)	1,000	0.074697	0.00642	0.00631
	(0.5)	1,000	0.07433	0.007557	0.006107
	(0.75)	1,000	0.072503	0.006323	0.005157
	(1)	1,000	0.07095	0.00627	0.006147
Centered sinusoidal (20)	(0.25)	2,000	0.995217	0.0762	0.00485
	(0.5)	2,000	0.997077	0.021083	0.004663
	(0.75)	2,000	0.979835	0.02951	0.00437
	(1)	2,000	0.997185	0.05108	0.00483
Shifted sinusoidal (20)	(0.25)	2,500	0.980479	0.579473	0.438487
	(0.5)	2,500	0.978902	0.567423	0.409863
	(0.75)	2,500	0.981065	0.741271	0.369507
	(1)	2,500	0.982633	0.392905	0.220915
Ackley (20)	(0.25)	2,000	0.417147	0.05927	0.052067
	(0.5)	2,000	0.4145	0.051427	0.046753
	(0.75)	2,000	0.384613	0.043467	0.03939
	(1)	2,000	0.381483	0.031627	0.02629
Levy and Montolvo (20)	(0.25)	2,000	0.026053	0.004903	0.004903
	(0.5)	2,000	0.023623	0.00483	0.00483
	(0.75)	2,000	0.025987	0.00486	0.00486
	(1)	2,000	0.07095	0.00427	0.004147
Rosenbrock (20)	(0.25)	2,000	0.004113	0.004113	0.004113
	(0.5)	2,000	0.004597	0.004597	0.004597
	(0.75)	2,000	0.00413	0.00413	0.00413
	(1)	2,000	0.00451	0.00451	0.00451
Griewank (20)	(0.25)	2,000	0.008713	0.00493	0.00493
	(0.5)	2,000	0.009863	0.00493	0.00493
	(0.75)	2,000	0.00957	0.004957	0.004957
	(1)	2,000	0.009583	0.004973	0.004973
Salomon (20)	(0.25)	2,000	0.11487	0.013147	0.01019
	(0.5)	2,000	0.102073	0.012607	0.010587
	(0.75)	2,000	0.104337	0.013667	0.010607
	(1)	2,000	0.10762	0.01164	0.008723

References

1. Ali, M.M., Khompatraporn, C., Zabinsky, Z.B.: A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J. Glob. Optim. **31**(4), 631–672 (2005)
2. Allgöwer, F., Zheng, A. (eds.): Nonlinear model predictive control. Progress in Systems and Control Theory, vol. 26. Birkhäuser Verlag, Basel (2000)
3. Azizi, N., Zolfaghari, S.: Adaptive temperature control for simulated annealing: a comparative study. Comput. Oper. Res. **31**, 2439–2451 (2004)
4. Bertsekas, D.P.: Dynamic Programming and Optimal Control, vol. 1. Athena Scientific, Belmont (1995)
5. Cerf, R.: A new genetic algorithm. Ann. Appl. Probab. **6**(3), 778–817 (1996)
6. Dunkl, C.F., Xu, Y.: Orthogonal Polynomials of Several Variables. Cambridge University Press, Cambridge (2001)
7. Ingber, L.: Adaptive simulated annealing (asa): lessons learned. Control Cybern. **25**, 22–54 (1996)
8. Kirkpatrick, S., Gelatt, C.D.J., Vecchi, M.P.: Optimisation by simulated annealing. Science **220**, 671–680 (1983)
9. Kohn, W., Zabinsky, Z.B., Brayman, V.: Optimization of algorithmic parameters using a meta-control approach. J. Glob. Optim. **34**(2), 293–316 (2006)
10. Kolonko, M., Tran, M.T.: Convergence of simulated annealing with feedback temperature schedules. Probab. Eng. Inform. Sci. **11**, 279–304 (1997)
11. Lovász, L.: Hit-and-run mixes fast. Math. Program. **86**, 443–461 (1999)
12. Mitter, S.K.: Successive approximation methods for the solution of optimal control problems. Automatica **3**, 135–149 (1966)
13. Molvalioglu, O., Zabinsky, Z.B., Kohn, W.: Multi-particle simulated annealing. In: Törn, A., Žilinskas, J. (eds.) Models and Algorithms for Global Optimization. Optimization and its applications, vol. 4. Springer (2006)
14. Molvalioglu, O., Zabinsky, Z.B., Kohn, W.: Meta-control of an Interacting-particle Algorithm for Global Optimization. Technical report, University of Washington (2007)
15. Moral, P.D.: Feynman-Kac Formulae: Genological and Interacting Particle Systems with Applications. Springer-Verlag, New York (2004)
16. Moral, P.D., Miclo, L.: Dynamiques recuites de type Feynman-Kac: résultats précis et conjectures (French). ESAIM: Probab. Stat. **10**, 76–140 (2006)
17. Munakata, T., Nakamura, Y.: Temperature control for simulated annealing. Phys. Rev. E **64**(4), 46–127 (2001)
18. Scott, D.: Multivariate Density Estimation: Theory, Practice, and Visualization. Wiley, New York (1992)
19. Sharpe, F.W.: The sharpe ratio. J. Portfolio Manage. **21**, 49–59 (1994)
20. Shen, Y., Kiatsupaibul, S., Zabinsky, Z.B., Smith, R.L.: An analytically derived cooling schedule for simulated annealing. J. Glob. Optim. **38**, 333–365 (2007)
21. Smith, R.L.: Efficient Monte Carlo procedures for generating points uniformly distributed over bounded region. Oper. Res. **32**, 1296–1308 (1984)
22. Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. IEEE Comp. **27**(6), 17–26 (1994)
23. Triki, E., Collette, Y., Siarry, P.: A theoretical study on the behavior of simulated annealing leading to a new cooling schedule. Eur. J. Oper. Res. **166**(1), 77–92 (2005)
24. Zabinsky, Z.B.: Stochastic Adaptive Search for Global Optimization. Kluwer Academic Publishers, Boston (2003)