Hasty Congregational Gradient Descent

Kim L. Blackmore

Robert C. Williamson

Communications Division

Department of Engineering

DSTO

Australian National University

Salisbury, SA.

Canberra, 0200, Australia

Iven M. Y. Mareels

Department of Electrical Engineering

University of Melbourne

Abstract

Stepwise Gradient Descent (SGD) algorithms for online optimization converge to local minima of the relevant cost function. In this paper a globally convergent modification of SGD is proposed, in which several solutions of SGD are run in parallel, together with online estimates of the cost function and its gradient. As each SGD estimate reaches a local minimum of the cost, the fitness of the member is evaluated and the member is immediately restarted unless it is the current best estimate. A number of results concerning the convergence behaviour of the proposed algorithm are derived using results from dynamical systems theory and probability theory.

I. INTRODUCTION

Online optimisation of a cost function is widely used for learning in artificial intelligence, system identification, and signal processing. The backpropagation algorithm for neural network learning is one such algorithm [7], as is the CMA algorithm for blind equalisation [4] and the LMS algorithm for adaptive signal processing [6]. All of these algorithms are Stepwise (or Stochastic) Gradient Descent (SGD) methods—as information is received an estimate solution is updated according to an instantaneous approximation to the true cost function. More recently devised techniques for online optimisation include Genetic Algorithms (GAs)[5]. In GAs a population of solutions is evolved according to operations based on theories of biological genetics. The key idea in GAs is survival of the fittest, whereby the fitness of the various members is evaluated, and the less fit are removed.

In [2], [1] a Congregational Gradient Descent (CGD) algorithm was proposed. The CGD algorithm combines the technique of stepwise gradient descent with ideas from GAs. In contrast to most GAs, our algorithm is sufficiently simple for us to be able to analyse its behaviour theoretically, although as we shall see our analysis is not as complete as one would wish. The essential idea behind the CGD algorithm is that several versions of SGD are run in parallel, and periodically the fittest is selected and the remainder are restarted. In this paper we propose a refined version of the CGD algorithm, called the Hasty CGD (HCGD) algorithm. In the HCGD algorithm the fitness of each member of the congregation is tested as the member reaches a local minimum of the cost, and the member is immediately restarted unless it is the current best estimate. This modification allows the elimination of excess computation in the CGD algorithm, which is caused by different members in the congregation taking different lengths of time to converge to local minima.

It is assumed that the value of the cost function cannot readily be evaluated due to the online nature of the problem. Therefore online estimates of the cost and its gradient are calculated as the members evolve. These estimates are used to evaluate the fitness of members and to identify convergence of members to local minima. The cost estimate appears in the CGD algorithm but the gradient estimate does not. At any time, the HCGD algorithm selects one of the members of the congregation to be the estimate solution. If the cost and gradient estimates are accurate approximations of the true cost and gradient, and the parameters of the HCGD algorithm are appropriately chosen, then the cost of the selected member is always decreasing (except for a small amount of jiggle due to the online nature of the problem). The HCGD algorithm is formally defined and discussed in Section III.

A number of results concerning the convergence behaviour of the HCGD algorithm are derived in Section IV. It is shown that once the selected member enters a small neighbourhood of the global minimiser it remains there. It is then said that the algorithm has converged. A lower bound on the probability that the algorithm has converged at any particular time is calculated, and an upper bound on the expected time until convergence of the algorithm is also calculated. This analysis uses averaging theory to show that the individual members converge to local minima, and that the cost and gradient estimates approximate the true cost and gradient. These results are tied together using the probability of landing in the basin of attraction of the global minimum of the cost function under consideration.

We also provide a comparison between the HCGD and CGD algorithm performance. This turns out to be difficult and not as conclusive as we would like. In doing so we are lead to a result which we prove in section VII concerning the range of possible cost functions that can arise in quadratic cost learning problems which is perhaps of more general interest.

II. NOTATION AND DYNAMICAL SYSTEMS THEORY

In this section the notation used in this paper is defined, a number of definitions from dynamical systems theory are stated, and a result from averaging theory is given. The averaging theory result will be used in Section IV to prove convergence of the HCGD algorithm.

For any $a \in \mathbb{R}^m$, $\|a\|$ denotes the Euclidean norm of a and $B(a, r)$ denotes the closed ball with centre a and radius $r > 0$. For any function $f : A \times X \to \mathbb{R}$, where $A \subset \mathbb{R}^m$ and $X \subset \mathbb{R}^n$, $\frac{\partial f}{\partial a}$ denotes the gradient of f with respect to the first argument. For $a : \mathbb{R} \to \mathbb{R}^n$, $\dot{a}$ denotes the derivative of $a(t)$ with respect to t.

Definition 1: A function $h : \mathbb{R}^+ \to \mathbb{R}$ is called an *order function* if $h(\mu)$ is continuous and sign definite on $(0, \mu_0]$ for some $\mu_0 > 0$, and if $\lim_{\mu \downarrow 0} h(\mu)$ exists.

Let $h(\mu)$ and $l(\mu)$ be order functions. Then the notation $O_\mu(l(\mu))$ and $o_\mu(l(\mu))$ is defined by

$$h(\mu) = O_\mu(l(\mu)) \text{ if } \exists c, \ \mu_1 > 0 \text{ such that } |h(\mu)| \leq c|l(\mu)| \text{ on } (0, \mu_1]$$

$$h(\mu) = o_\mu(l(\mu)) \text{ if } \lim_{\mu \downarrow 0} \frac{h(\mu)}{l(\mu)} = 0.$$

Let $l(\mu)$ be an order function. A property is said to hold for k *on the time scale* $l(\mu)$ if it is true for all k satisfying $0 \leq k \leq Ll(\mu)$, for some constant L.

Definition 2: Consider the initial value problem

$$\dot{a} = F(a(t)) \qquad ; \qquad a(0) = a_0 \tag{1}$$

$t \geq 0$; $a(t) \in \mathbb{R}^m$. Suppose $F(a^*) = 0$ for some $a^* \in \mathbb{R}^m$.

1. The solution $a \equiv a^*$ of the initial value problem (1) is *uniformly asymptotically stable* with basin of attraction $A^* \subset \mathbb{R}^m$ if:
    - it is *stable*:
        for all $\varepsilon > 0$ there exists $\delta > 0$ such that for all $a_0 \in A^*$, $\|a_0 - a^*\| \leq \delta \Rightarrow \|a(t) - a^*\| < \varepsilon$ $\forall t \geq 0$.
    - it is *uniformly attractive* in A^*:
        for all $\delta > 0$ and $\varepsilon > 0$, there exists $\sigma > 0$ such that for all $a_0 \in A^*$, $\|a_0 - a^*\| < \delta \Rightarrow \|a(t) - a^*\| < \varepsilon$ $\forall t \geq \sigma$.
2. The solution $a \equiv a^*$ of the initial value problem (1) is *globally exponentially stable* if:
    for all $a_0 \in \mathbb{R}^m$ and $t \geq 0$, $\|a(t) - a^*\| \leq \|a_0 - a^*\|e^{-t}$.
3. The ODE (1) is *Lagrange stable* if, for all $a_0 \in \mathbb{R}^m$ there exists $\delta \geq 0$ such that $\|a(t)\| \leq \delta$ $\forall t \geq 0$.

The following theorem is a deterministic averaging result that relates the solution of a difference equation depending on a sequence (x_k) of inputs to the corresponding solution of an *averaged* ordinary differential equation. Thus it makes possible the use of results about the critical points of an ODE to characterise the behaviour of the solution of a difference equation.

Theorem 3: Assume that

A1 $A \subset \mathbb{R}^m$, $X \subset \mathbb{R}^n$, X is compact, and $(x_k)_{k \in \mathbb{N}_0}$ is a sequence of points in X;

A2 $H : A \times X \to A$ is bounded and Lipschitz continuous in the first argument (uniformly in the second argument) on a compact domain.

A3 The function

$$H^{av}(a) = \lim_{L \to \infty} \frac{1}{L} \sum_{k=0}^{L-1} H(a, x_k)$$

exists uniformly for all $a \in \mathbb{R}^m$. That is, for any $L \in \mathbb{N}$, $\delta(\mu) = o_\mu(1)$, where

$$\delta(\mu) := \sup_{k_0 \in \mathbb{N}_0} \sup_{a \in A} \sup_{k \in \left[0, \frac{L}{\mu}\right)} \mu \left\| \sum_{l=k_0}^{k_0+k-1} (H(a, x_l) - H^{av}(a)) \right\|;$$

A4 $\beta(\mu) = o_\mu(1)$;

A5 For each $k \in \mathbb{N}_0$, $h_k : A \times X \to A$ is bounded and Lipschitz continuous in the first argument (uniformly in the second argument and in k) on a compact domain.

Let a_k and $a_{av}(t)$ be defined according to the following equations for all $k_0, k \in \mathbb{N}_0$ and $k, t \geq k_0$:

$$a_{k+1} = a_k - \mu H(a_k, x_k) - \mu \beta(\mu) h_k(a_k, x_k); \quad a_{k_0} = a_0 \in A \tag{2}$$

$$\dot{a}_{av} = -\mu H^{av}(a_{av}(t)) \quad ; \quad a_{av}(k_0) = a_0 \tag{3}$$

Then there exists a constant μ_0 and an $o_\mu(1)$ function $l(\mu)$ such that for all $\mu \leq \mu_0$,

1. $\|a_k - a_{av}(k)\| \leq l(\mu)$ for k on the time scale $\frac{1}{\mu}$.
2. If there exists a globally exponentially stable critical point of (3) then $\|a_k - a_{av}(k)\| \leq l(\mu)$ for all $k \geq k_0$.

3. If there exists a uniformly asymptotically stable critical point $a^c \in$ interior(A) of (3), with basin of attraction $A^c \subset A$, then for any compact set $B^c \subset A^c$ there exists a function $K(\mu) : \mathbb{R}^+ \to \mathbb{R}$ such that if $a_0 \in B^c$ then then $\|a_k - a_{av}(k)\| \leq l(\mu)$ for all $k_0 \leq k \leq K(\mu)$.

The proof of results *1* and *3* of Theorem 3 appear in [2], and the proof of *2* can be derived using the same techniques. The constant μ_0 and the functions $l(\mu)$, $K(\mu)$ depend on the particular sequence (x_k), the distribution of the local minima, and the boundaries of the basins of attraction of the critical point of (1). They are independent of k_0 because of the uniform convergence to the average update H^{av} that is required in Assumption *A3*.

III. THE HASTY CONGREGATIONAL GRADIENT DESCENT ALGORITHM

Consider the problem of locating the global minimum of some cost function $J : A \to \mathbb{R}$, where $A \subset \mathbb{R}^m$. The cost function is not known explicitly, but rather it is the average of a known function $\phi : A \times X \to \mathbb{R}$ over a known sequence (x_k) of points in X. That is,

$$J(a) = \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} \phi(a, x_k). \tag{4}$$

We say that $\phi(a, x_k)$ is the instantaneous cost at time k, evaluated at the *parameter* $a \in A$ and the input point $x_k \in X \subset \mathbb{R}^n$. The inputs are received sequentially, and it is desired to have a parameter estimate which is updated as each input is received. Often ϕ is given by some loss function, such as $\phi(a, x_k) = (y_k - f(a, x_k))^2$, where y_k is a sequence of desired values of the examples x_k and $f : A \times X \to \mathbb{R}$ is a parametrized family of functions.

Stepwise gradient descent of J is achieved by updating estimate parameters a_k according to

$$a_{k+1} = a_k - \mu \left. \frac{\partial \phi}{\partial a} \right|_{(a_k, x_k)} \qquad ; \qquad a_{k_0} = a_0 \tag{5}$$

From Theorem 3, it can be shown that the estimate parameters generated by this recursion will converge to a neighbourhood of the global minimiser of J provided that the initial parameter estimate is in a certain region of parameter space. As $\mu \to 0$, this region approaches the basin of attraction of the global minimiser in the associated averaged ODE. However, if there are non-global local minima of J, some choices of the initial estimate will cause the estimate parameters to converge to a local minimiser. In general the basin of attraction of the global minimum is not known, so SGD cannot be guaranteed to find the global minimum.

The *congregational* gradient descent algorithm presented in [2] runs a number of estimates in parallel, instead of the single estimate normally generated by SGD. At the same time, an estimate of the cost function at each of the parameter estimates is calculated. Periodically the cost estimates are compared and the member is restarted, unless it is the current best estimate. Members are restarted according to some continuous probability distribution D_a with compact support $A^0 \subset \mathbb{R}^n$. Let $A^* \subset A^0$ be the basin of attraction of a^* in (9). The probability of initialising a member in A^* is $\sigma := D_a(A^*)$.

The algorithm presented below is a refinement of the CGD algorithm which utilizes an online gradient estimate. Convergence of a parameter estimate to a local minimum is identified by a small value of the gradient estimate, since around any critical point of the cost function there must

HCGD Algorithm
Choose the flatness $\gamma > 0$;
Choose the cost step size $\alpha \in (0, 1)$;
Choose the gradient step size $\beta \in (0, 1)$;
Choose the parameter step size $\mu > 0$;
Choose the transition time $\tau > 0$;
Choose the congregation size $N \geq 2$;
for $n \in \{1, \ldots, N\}$ **do**
  $a_0^n := random(A)$;
  $\Phi_0^n := 0$;
  $\Gamma_0^n := 0$;
od
$\kappa_0 := 0$;
$\hat{n} := 1$;
$k := 0$;
while *(true)* **do**
  if $\kappa_k \geq \tau$ **then**
    $\hat{n} = \min \; \underset{m \in \{1,\ldots,N\}}{\arg\min} \; \Phi_k^m$;
   for $n \in \{1, \ldots, N\}$ **do**
    if $\|\Gamma_k^n\| \leq \gamma$ *and* $n \neq \hat{n}$ **then**
     $a_k^n := random(A)$;
     $\Phi_k^n := 0$;
     $\Gamma_k^n := 0$;
     $\kappa_k := 0$;
    fi
   od
  fi
  for $n \in \{1, \ldots, N\}$ **do**

$$a_{k+1}^n := a_k^n - \mu \left.\frac{\partial \phi}{\partial a}\right|_{(a_k^n, x_k)}; \tag{6}$$

$$\Phi_{k+1}^n := (1 - \alpha)\Phi_k^n + \alpha\phi(a_k^n, x_k); \tag{7}$$

$$\Gamma_{k+1}^n := (1 - \beta)\Gamma_k^n + \beta \left.\frac{\partial \phi}{\partial a}\right|_{(a_k^n, x_k)}; \tag{8}$$

  od
  $\kappa_{k+1} := \kappa_k + 1$;
  $k := k + 1$;
od

be a small region where the gradient of the cost function is small. The algorithm is called *hasty* congregational gradient descent because members are restarted as soon as they converge, whereas in the CGD algorithm an "epoch length" is chosen long enough to ensure that *all* members have converged to their respective minima before comparing the cost estimates. The heuristic motivation for the algorithm is the hope that the use of the online gradient estimate will enable the elimination of excessive computation due to members in the congregation taking different lengths of time to converge.

In the HCGD algorithm randomly chosen initial parameter estimates land in the basin of attraction of some local minimum. It is desired to keep the estimate only if it originated in the basin

of attraction of the global minimum. However the only way to distinguish the global basin of attraction from the basins of nonglobal local minima is to allow the estimates to evolve until they reach a minima, and then compare the costs at each of the minima. Once an estimate has reached a local minimum the cost at the estimate will not improve, so there is no reason to continue further updating the estimate. At this stage the member can be compared with other members, even though the other members are still evolving. If it is the current best estimate (it has the smallest cost), it should be kept until such time as a better estimate appears; otherwise it can be discarded and the member can be restarted, since the other members will continue improving as they continue updating.

The online gradient estimate is denoted $\Gamma_{k,T}^n$. The estimate is stopped if the value of $\|\Gamma_{k,T}^n\|$ falls below some pre-determined flatness parameter γ, which ideally can be chosen so that the cost is only called flat if the estimate is close to a critical point of J. The online cost and gradient estimates only become good estimates of the average cost and gradient after some *transition time* τ. The online estimates are not used for the first τ iterations after a member has been restarted. The transition time plays a converse role to the epoch length in [2] (it is a minimum rather than a maximum), and thus can take on much smaller values.

At time k, the parameter estimate defined by the HCGD algorithm is $a_k^{\hat{n}}$. Initially this is simply the first member in the congregation, but once the first transition time is passed, $\hat{n}$ selects the member in the congregation which has the lowest cost estimate. The selected member may change at every iteration, until such time as a member in the congregation is restarted. When a member is restarted, the transition time recommences, and for the next τ iterations $\hat{n}$ is not updated, so the same member of the congregation is selected for the τ iterations following any restart.

Setting $\hat{n} = \min \arg\min_{m\in\{1,...,N\}} \Phi_k^m$ is necessary in order to ensure that $\hat{a}_k$ is uniquely defined, since it is possible for two members of the congregation to have the same cost estimate. In that case, there is no need to keep both. For the purposes of analysis the possibility can be ignored, since the event that $\Phi_k^m = \Phi_k^n$ exactly occurs with probability zero,

If γ is chosen too large, the cost function J may be deemed to be flat in regions which are not connected to a critical point of J. In this case the HCGD Algorithm will restart members as they pass through the flat region, so that members that are initialised in the basin of attraction of the global minimum may not be allowed to progress all the way to the global minimum. This will have the effect of shrinking the size of the basin of attraction of the global minimum, but will not break the algorithm (unless γ is so large that J is considered flat everywhere). In fact, the "optimal" value of γ (in the sense of minimizing the expected computation required to find the global minimum) may well be large enough for this to occur, since a larger value of γ will mean more frequent random restarts.

It is apparent from the algorithm's structure, that HCGD, like the original CGD algorithm, can easily exploit parallel computing hardware.

IV. CONVERGENCE ANALYSIS

In this section the convergence properties of the HCGD Algorithm will be discussed. Three results concerning the convergence of the algorithm are stated in Theorem 4. The proof of Theorem 4 is quite complicated, so it is divided into three stages. Section V gives a heuristic analysis of the behaviour of the algorithm, by considering an alternate, simplified, version of the algorithm. The

extension to the actual HCGD algorithm appears in the appendix.

In order to analyse the convergence behaviour of the HCGD Algorithm the following general assumptions are made. Assumptions *C1*, *C3* and *C4* are essentially equivalent to Assumptions *A1*, *A2* and *A3* of Theorem 3. Assumption *C2* deals with the random initialisation of members. Assumption *C5* concerns the shape of the average cost function. It may be possible to eliminate the assumption of Lagrange stability in *C5*, as discussed in [2]. Finally, Assumption *C6* defines a linkage between the small parameters α, β, and μ. This enables the use of split time scale averaging in the detailed proof (omitted in this shortened version) since the online estimates of the cost and gradient converge more quickly than the parameter estimates.

Assumptions:.

C1 $A^0 \subset A \subset \mathbb{R}^m$, $X \subset \mathbb{R}^n$, A^0 and X are compact, and $(x_k)_{k \in \mathbb{N}_0}$ is a sequence of points in X.

C2 The probability distribution D_a is continuous and the support of D_a is A^0.

C3 Both $\phi(a, x)$ and $\frac{\partial \phi}{\partial a}$ are bounded and Lipschitz continuous in the first parameter (uniformly in the second) on a compact domain in $\mathbb{R}^m$.

C4 The average J defined in (4) exists, and for any $L \in \mathbb{N}$ (independent of μ),

$$\delta(\mu) = \sup_{k_0 \in \mathbb{N}_0} \sup_{a \in A} \sup_{k \in [0, \frac{L}{\mu})} \mu \left\| \sum_{l=k_0}^{k_0+k-1} (\phi(a, x_l) - J(a)) \right\|$$

exists and is $o_\mu(1)$.

C5 J has a finite number of local minima, and attains its global minimum at some point a^* which is in the interior of A^0, and the ODE

$$\dot{a} = -\mu \left. \frac{\partial J}{\partial a} \right|_{a(t)} \qquad ; \qquad a(t_0) = a_0 \qquad (9)$$

is Lagrange stable.

C6 $\mu = o_\alpha(\alpha)$ and $\mu = o_\beta(\beta)$.

Let $a^* \in A$ be the global minimiser of J and let $A^* \subset A^0$ be the basin of attraction of a^* in (9). The probability of initialising a member in A^* is $\sigma = D_a(A^*)$.

Because of the online nature of the SGD algorithm which underpins the HCGD algorithm, the estimate parameter $a_k^{\hat{n}}$ defined by the algorithm will (in general) not converge to a^* or any other point, but will always "jiggle about". Therefore a more general notion of convergence is needed. This more general notion will be denoted γ-convergence in what follows.

In section V sets $C_\gamma(a^*)$ are defined such that $C_{\gamma_1}(a^*) \subset C_{\gamma_2}(a^*)$ whenever $\gamma_1 < \gamma_2$ and $C_\gamma(a^*) \to \{a^*\}$ as $\gamma \to 0$. These sets are used to define the notion of γ-convergence as follows: The parameters $a_k^{\hat{n}}$ γ-converge to a^* if there exists $j \geq 0$ such that $a_k^{\hat{n}} \in C_{(1+3\gamma)\gamma}(a^*)$ for all $k \geq j$. For the purposes of this definition, the significance of the sets $C_\gamma(a^*)$ lies in the fact that the sets converge to $\{a^*\}$, rather than in the particular exact definition of the sets.

Let $\hat{k}$ be the time when the HCGD algorithm γ-converges to a^*. That is, let

$$\hat{k} := \min\{j \geq 0 : a_k^{\hat{n}} \in C_{(1+3\gamma)\gamma}(a^*) \text{ for all } k \geq j\}.$$

We have the following results about $\hat{k}$. (Simpler versions, that are more interpretable appear in section V.

Theorem 4: Consider the HCGD algorithm with Assumptions *C1* to *C6*. For any $\eta \in (0,1)$ there exists $\gamma_0 > 0$ such that for any $\gamma < \gamma_0$ there exists α_0, β_0 such that if $\alpha < \alpha_0$ and $\beta < \beta_0$ there exists $\tau_0 > 0$ such that if $\tau \geq \tau_0$ then

R1. $\hat{k} \leq \min\{k : a_k^{\hat{n}} \in C_{\frac{5}{3}\gamma}(a^*)\}$;

R2. There exist T_1, $T_2 > 0$ such that

$$Pr\{\hat{k} \leq k\} > \left(1 - [(1-\eta)(1 - (1-\eta)\sigma)]^N\right)$$
$$+ [(1-\eta)(1 - \sigma)]^N \frac{1-[(1-\eta)(1-(1-\eta)\sigma)]^{N-1}}{1-[(1-\eta)(1-\sigma)]^{N-1}} \times$$
$$\left(1 - [(1-\eta)(1 - \sigma)]^{\left\lfloor \frac{k-T_1}{T_2} \right\rfloor (N-1)}\right) ;$$

R3. There exist t_γ^*, t_γ^{loc}, $\tilde{t}_\gamma^*$, $\tilde{t}_\gamma^{loc}$ such that $\tilde{t}_\gamma^* \rightarrow t_\gamma^*$ and $\tilde{t}_\gamma^{loc} \rightarrow t_\gamma^{loc}$ as $\gamma \rightarrow 0$, and

$$\frac{(1-\eta)^2\sigma - N[1 - (1-\eta)(1 - \sigma)]^2}{(N-1)[1 - (1-\eta)(1 - \sigma)]^2}t_\gamma^{loc} + t_\gamma^* \leq \mathbb{E}(\hat{k}) \leq \tag{10}$$
$$\frac{\eta + (1-\eta)\sigma - (1-\eta)^4\sigma^2}{(N-1)(1-\eta)^4\sigma^2}(\tilde{t}_\gamma^{loc} + 1) + \tilde{t}_\gamma^* + 1.$$

Theorem 4 does not provide a practical method for choosing the parameters used by the HCGD algorithm. However, it does prove that suitable quantities exist, and illuminates the dependencies between the parameters. In particular, the flatness parameter γ should be chosen first, and can be chosen as small as desired, provided it is smaller than an upper bound determined by the average cost function. The cost and gradient step size must then be chosen smaller than an upper bound determined by the flatness γ. The choice of α and β sets a lower bound on the transition time τ and an upper bound on the parameter step size μ. The congregation size N can be chosen independently of all of the other algorithm parameters.

V. SIMPLIFIED APPROXIMATE ANALYSIS

In this section we derive results concerning the time to convergence for a simplified case, which involves a reduction of the HCGD algorithm to the continuous time equivalent. The results parallel results *R1* to *R3* of Theorem 4. This enables us to understand the heuristic behaviour of the HCGD algorithm, without being distracted by the online and approximate aspects of the algorithm. In order to extend the results to the HCGD algorithm itself, the arguments must be extended using the averaging results in Theorem 3. Full details appear in the extended version of this paper.

The simplification takes the form of the following assumptions, which are not valid in the online optimisation context of the problem, but which do not change the overall behaviour of the algorithm significantly.

1. The parameter estimates are continuous functions of time, updated according to continuous time gradient descent on the average cost function J;

2. The exact value of the average error function J and its gradient $\frac{\partial J}{\partial a}$ are used instead of the estimates Φ and Γ for ascertaining the fitness and flatness of members in the congregation;

3. The transition time τ is equal to zero;

4. The estimates are not updated if they enter a region where $\left\|\frac{\partial J}{\partial a}\right\| < \gamma$.

Under these assumptions, the cost $J(a^{\hat{n}}(t))$ is non-increasing, since member $\hat{n}$ evolves according to gradient descent and the value of $\hat{n}$ is changed whenever the cost of a new member is less than the cost of the member currently chosen. In order to discuss further the behaviour of the simplified algorithm, some definitions and assumptions must be made. These definitions will also be used in the analysis of the HCGD algorithm in the next section.

For any value of γ, the region where the gradient of the cost function J is less than γ is denoted F_γ:

$$F_\gamma := \left\{ a \in \mathbb{R}^m : \left\| \frac{\partial J}{\partial a} \bigg|_a \right\| \leq \gamma \right\}.$$

The region F_γ is deemed to be flat, and parameter estimates are restarted as soon as they enter F_γ. In the limit $\gamma \to 0$, the set F_0 contains only the only the critical points of J, so that $\mathrm{vol}(F_\gamma) \to 0$. There exists some constant γ_1 such that for all $\gamma < \gamma_1$, each connected component of F_γ contains at least one critical point of J, and only contains multiple critical points of J if these points are themselves connected. Thus, for these small values of γ, F_γ is confined to small regions around the local minima and the boundaries of the basins of attraction of the local minima in (9).
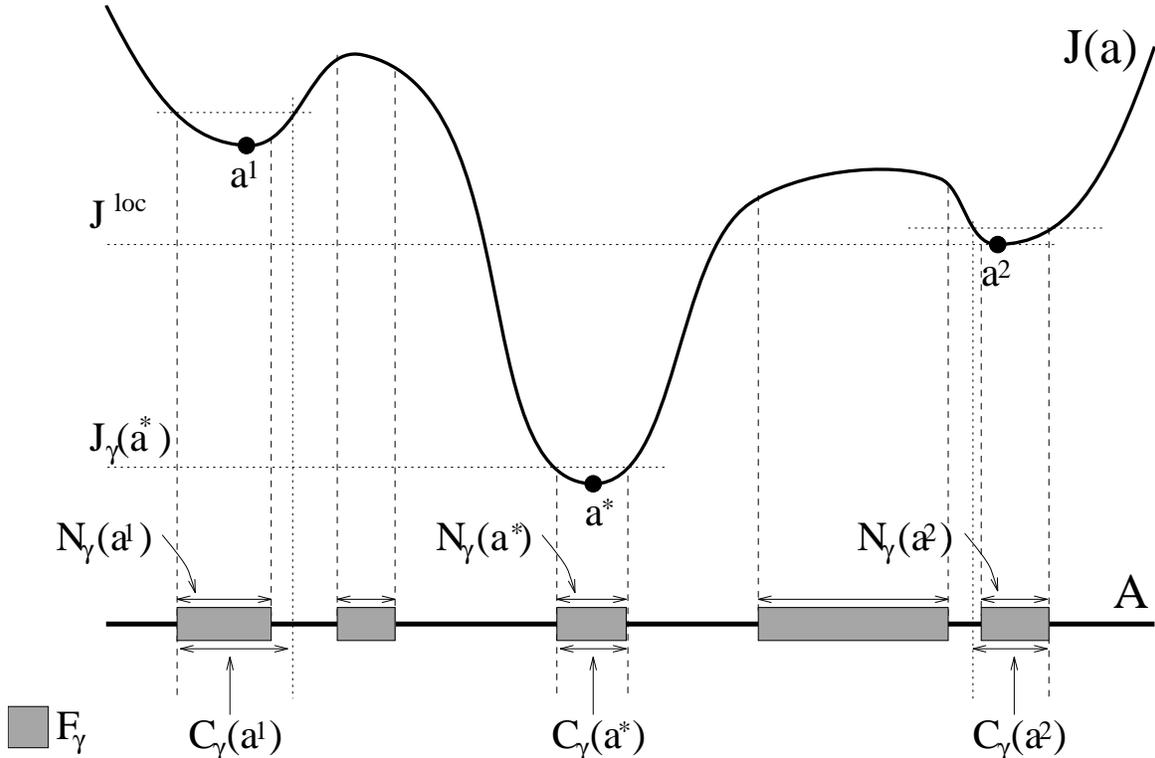

Fig. 1. An average error function J and the corresponding flat region F_γ and contour sets $C_\gamma(a^{loc})$ for a one dimensional parameter space A.

For any critical point a^c of J, the largest connected component in F_γ which contains a^c is

denoted $N_\gamma(a^c)$. As $\gamma \to 0$, $N_\gamma(a^c) \to \{a^c\}$. Let

$$J_\gamma(a^c) := \max_{a \in N_\gamma(a^c)} J(a)$$
$$C_\gamma(a^c) := \{a \in \mathbb{R}^m : J(a) \leq J_\gamma(a^c)\}.$$

Then $J_\gamma(a^c)$ is the greatest value of the cost in the part of the flat region that is connected to the global minimum, and the boundary of $C_\gamma(a^c)$ is the contour line for J at the value $J_\gamma(a^c)$. Clearly, for $\gamma > 0$, $N_\gamma(a^c) \subset C_\gamma(a^c)$. From the definition of $a(t)$ in the gradient equation (9), if $a(t_1) \in C_\gamma(a^c)$, then $a(t) \in C_\gamma(a^c)$ for all $t \geq t_1$.

There exists $\gamma_2 \leq \gamma_1$ such that $C_\gamma(a^*) \cap F_\gamma = N_\gamma(a^*)$ for all $\gamma \leq \gamma_2$. This implies that a^* is the only local minimum in $C_\gamma(a^*)$. Choosing $\gamma < \gamma_2$ ensures that estimates which have γ-converged to a^* are are not in reality converged to some non-global local minimum.

Figure 1 shows an example of an average error function J and the corresponding flat region F_γ and contour sets $C_\gamma(a^{loc})$ for a one dimensional parameter space . Here there is a unique global minimum at a^*, and two non-global local minima a^1 and a^2. In this example each connected component of F_γ contains only one critical point of J, and $C_\gamma(a^*) \cap F_\gamma = N_\gamma(a^*)$.

The first time some member of the congregation enters $N_\gamma(a^*)$, it will become the selected member. Since the cost for the selected member for the simplified algorithm is always decreasing, the following holds:

Result 1 *If $a^{\hat{n}}$ enters $C_\gamma(a^*)$ at time $\hat{t}$, then $a^{\hat{n}}(t) \in C_\gamma(a^*)$ for all $t \geq \hat{t}$.*

Note that in this section the convergence to the set $C_\gamma(a^*)$ instead of the set $C_{(1+3\gamma)\gamma}(a^*)$ is considered. This is possible because of the simplified nature of the algorithm.

Estimates initialised in A^* will stay in A^* until they are restarted, and will γ-converge to a^* if they are allowed to evolve for a sufficiently long time. However according to assumption 4, estimates initialised in $F_\gamma \cap A^*$ are not updated so they do not γ-converge to a^*. Let A^*_γ be the set of all initial points in A^* such that the solution of (9) enters some other part of the flat region before converging to a^*

$$A^*_\gamma := \{a_0 \in A^* : a(t_1) \in F_\gamma \backslash N_\gamma(a^*), \ a(t) \text{ defined by (9)}, \tag{11}$$
$$t_1 = \min\{t : a(t) \in F_\gamma\}\}.$$

Then A^*_γ is a proper subset of $A^* \backslash (F_\gamma \backslash N_\gamma(a^*))$. Since $\gamma \leq \gamma_2$, $C_\gamma(a^*) \subset A^*_\gamma$.

Figure 2 shows a contour plot of J for an example with $A \subset \mathbb{R}^2$. In this example there are no non-global local minima in A, but there is a local maximum a^m and a saddle point a^s. The global basin of attraction and the effective global basin of attraction are shown.

The probability of initialising a member in A^*_γ is $\sigma_\gamma := Pr\{a_0 \in A^*_\gamma\} = D_a(A^*_\gamma)$. All members initialised in A^*_γ enter $N_\gamma(a^*)$ before being restarted, and no other members do, so the probability that any given estimate γ-converges to a^* before being restarted is σ_γ. In the limit $\gamma \to 0$, and $A^*_\gamma \to A^*$, so that $\sigma_\gamma \to \sigma$. In particular, for any $\eta > 0$ there exists γ_η such that $\sigma_\gamma \geq (1 - \eta)\sigma$ for all $\gamma \leq \gamma_\eta$.

In order to determine the probability that the algorithm has γ-converged at a particular time, it is necessary to know how long it takes individual solutions of (9) to be restarted. The estimates are restarted when they reach F_γ. Let $t_\gamma(a_0)$ be the time taken for an estimate initialised at a_0 to
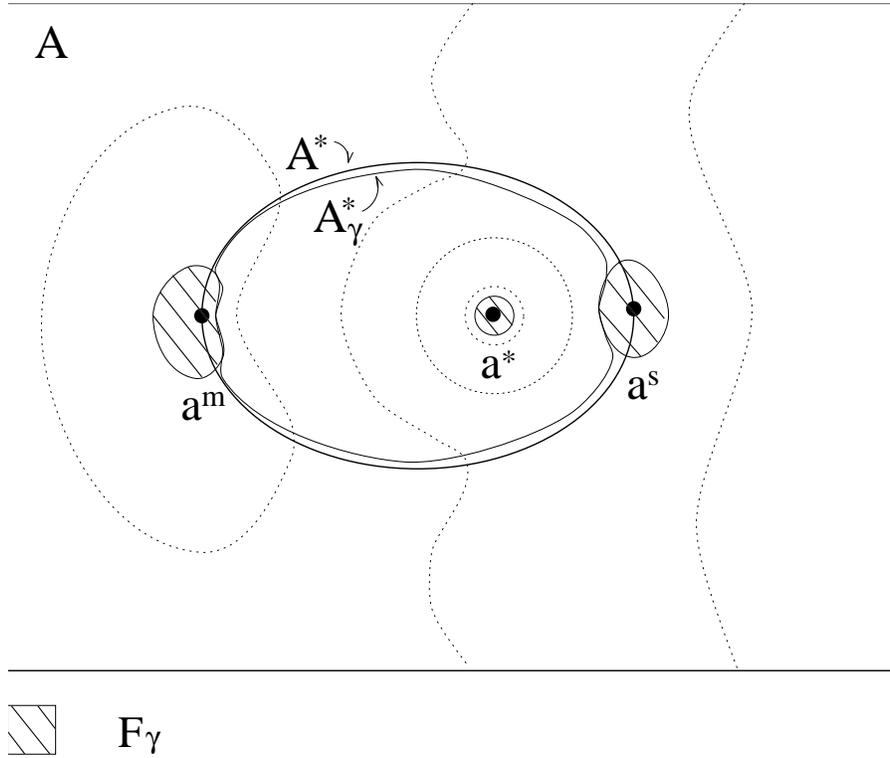
Fig. 2. Contour plot of an average error function J for a two dimensional parameter space A, showing the flat region F_γ and the effective basin of attraction $A_\gamma^* \subset A^*$. The dotted lines show the contours of J.

reach this set

$$t_\gamma(a_0) = \min \left\{ t - t_0 : a(t) \in F_\gamma, \ a(t) \text{ defined by } (9) \right\}.$$

Then $t_\gamma(a_0) \to \infty$ as either $\mu \to 0$ or $\gamma \to 0$.

The basins of attraction of the local minima are open sets, and the time to convergence for estimates initialised close to a local maximum of J is unbounded. However, for any estimate started in A_γ^* the gradient of J is bounded below until the estimate enters $C_\gamma(a^*)$, so the time to convergence is bounded. Denote this time by T_γ^*

$$T_\gamma^* := \max \left\{ t_\gamma(a_0) \colon a_0 \in A_\gamma^* \right\}$$

For estimates initialised in any compact subset of the basin of attraction A^{loc} of a local minimum a^{loc}, the time for convergence to a point in F_γ is bounded. For any $\eta \in (0, 1)$, choose compact sets $B^{loc} \subset A^{loc}$, such that the probability of initialising in $B_\eta = \bigcup_{a^{loc} \neq a^*} B^{loc}$ is $(1 - \eta)(1 - \sigma)$ and the maximum time to convergence for estimates initialised in B_η is minimized. This is possible since the probability of initialising in $\bigcup_{a^{loc} \neq a^*} A^{loc}$ is $(1 - \sigma)$. Let the maximum time to convergence for estimates initialised in B_η be denoted $T_{\gamma, \eta}^{loc}$.

$$T_{\gamma, \eta}^{loc} := \max \left\{ t_\gamma(a_0) \colon a_0 \in B_\eta \right\}.$$

Not all estimates take the maximum time to converge, so the probability that $\hat{t} \leq T_\gamma^*$ is greater than the probability that at least one of the members was initialised in A_γ^*. Thus

$$Pr\{\hat{t} \leq T_\gamma^*\} > \left(1 - (1 - \sigma_\gamma)]^N\right) \tag{12}$$

where the probability is over all restarts of the algorithm. Now consider the case that none of the first initialisations are in A_γ^*. If they are all in B_η, all but one of the members will be restarted by time $T_{\gamma,\eta}^{loc}$, and one of the new members may be initialised in A_γ^*. Thus the probability that $\hat{t} \leq T_{\gamma,\eta}^{loc} + T_\gamma^*$ is greater than probability that $\hat{t} \leq T_\gamma^*$ plus the probability that all of the members are initialised in B_η the first time, and then in the second set of initialisations one is initialised in A_γ^*. Using (12) gives

$$Pr\{\hat{t} \leq T_{\gamma,\eta}^{loc} + T_\gamma^*\} > Pr\{\hat{t} \leq T_\gamma^*\} + (1-\eta)^N (1-\sigma)^N \left(1 - (1-\sigma_\gamma)^{N-1}\right)$$

Similarly, the probability that $\hat{t} \leq rT_{\gamma,\eta}^{loc} + T_\gamma^*$ satisfies

$$Pr\{\hat{t} \leq rT_{\gamma,\eta}^{loc} + T_\gamma^*\} > Pr\{\hat{t} \leq (r-1)T_{\gamma,\eta}^{loc} + T_\gamma^*\}$$
$$+ \left(1 - (1-\sigma_\gamma)^{N-1}\right)[(1-\eta)(1-\sigma)]^{N+(r-1)(N-1)}$$

This recursive relationship gives

$$Pr\{\hat{t} \leq rT_{\gamma,\eta}^{loc} + T_\gamma^*\} >$$
$$\left(1 - (1-\sigma_\gamma)^N\right) + \left(1 - (1-\sigma_\gamma)^{N-1}\right) \sum_{i=0}^{r-1} [(1-\eta)(1-\sigma)]^{N+i(N-1)}$$
$$= \left(1 - (1-\sigma_\gamma)^N\right) + \left(1 - (1-\sigma_\gamma)^{N-1}\right)[(1-\eta)(1-\sigma)]^N \times$$
$$\frac{1-[(1-\eta)(1-\sigma)]^{r(N-1)}}{1-[(1-\eta)(1-\sigma)]^{N-1}}$$

Now at any time t, the probability that the selected member of the congregation has converged is equal to the probability that $\hat{t} \leq t$, which is greater than or equal to the probability that $\hat{t} \leq rT_{\gamma,\eta}^{loc} + T_\gamma^*$, where $r = \left\lfloor \frac{t-T_\gamma^*}{T_{\gamma,\eta}^{loc}} \right\rfloor$. Assuming $\gamma \leq \gamma_\eta$, we have

Result 2

$$Pr\{\hat{t} \leq t\} > 1 - (1 - (1-\eta)\sigma)^N$$
$$+ [(1-\eta)(1-\sigma)]^N \frac{1-(1-(1-\eta)\sigma)^{N-1}}{1-[(1-\eta)(1-\sigma)]^{N-1}} \times$$
$$\left(1 - [(1-\eta)(1-\sigma)]^{\left\lfloor \frac{t-T_\gamma^*}{T_{\gamma,\eta}^{loc}} \right\rfloor (N-1)}\right).$$

Taking the limit as $t \to \infty$ (as a consistency check) gives

$$Pr\{\hat{t} \leq t\} \to 1 - (1-(1-\eta)\sigma)^N + [(1-\eta)(1-\sigma)]^N \frac{1-(1-(1-\eta)\sigma)^{N-1}}{1-[(1-\eta)(1-\sigma)]^{N-1}}$$

When η (and hence γ) is allowed to decrease to zero, this limit increases to 1 as one would expect.

Finally, we calculate the expected time until the algorithm converges. Let $\hat{s}$ be the number of the first random initialisation such that the resulting estimate γ-converges to a^*. Members initialised outside A^* definitely won't converge to a^*, and members initialised inside A^*_γ definitely will, so $Pr\{\hat{s} = s\} = \sigma_\gamma(1 - \sigma_\gamma)^{s-1}$. Thus the expected value of $\hat{s}$ satisfies

$$\mathbb{E}(\hat{s}) = \sum_{s=1}^{\infty} s\sigma_\gamma(1 - \sigma_\gamma)^{s-1} = \frac{1}{\sigma_\gamma}.$$

From the choice of γ_2, we have $\frac{1}{\sigma} \leq \mathbb{E}(\hat{s}) \leq \frac{1}{(1-\eta)\sigma}$.

Let t^*_γ be the expected evolution time for an estimate initialised in A^*_γ, and let the expected value of $t_\gamma(a_0)$ for a_0 not in A^*_γ be t^{loc}_γ:

$$t^*_\gamma := \mathbb{E}\big(t_\gamma(a_0)|a_0 \in A^*_\gamma\big)$$
$$t^{loc}_\gamma := \mathbb{E}\big(t_\gamma(a_0)|a_0 \notin A^*_\gamma\big),$$

where the expectations are with respect to D_a. From the definition, it can be seen that t^*_γ is the expected time taken for an estimate to enter $N_\gamma(a^*)$. It may be greater than the expected time taken for an estimate to enter $C_\gamma(a^*)$, since $N_\gamma(a^*) \subset C_\gamma(a^*)$. Similarly t^{loc}_γ is the expected time taken for estimates outside A^*_γ to enter F_γ, where they can be restarted.

Assume $\hat{s} \leq N$. Then one of the members is initialised in A^*_γ at the outset, so the expected time until the algorithm γ-converges is $E(\hat{t}) = t^*_\gamma$. If $N < \hat{s} \leq 2N - 1$, then all of the members are first initialised outside A^*_γ. They are all restarted after an expected time of t^{loc}_γ, and one of them is restarted in A^*_γ, so the expected time until the algorithm γ-converges is $E(\hat{t}) = t^{loc}_\gamma + t^*_\gamma$. Similar argument shows that, for any $\hat{s}$, the expected time until the algorithm γ-converges is

$$\mathbb{E}(\hat{t} \,|\, \hat{s}) = \left\lceil \frac{\hat{s} - N}{N - 1} \right\rceil t^{loc}_\gamma + t^*_\gamma$$

Combining with the bounds on the expected value of $\hat{s}$ gives the following bound on the expected time until the simplified algorithm γ-converges.

Result 3
$$\frac{1 - N\sigma}{(N-1)\sigma}t^{loc}_\gamma + t^*_\gamma \leq \mathbb{E}(\hat{t}) \leq \frac{1 - (1-\eta)\sigma}{(N-1)(1-\eta)\sigma}t^{loc}_\gamma + t^*_\gamma.$$

For the purposes of the analysis it has been assumed that t^*_γ and t^{loc}_γ are known, however they would not be known in practice. Even calculating the bounds T^*_γ and $T^{loc}_{\gamma,\eta}$ requires very detailed knowledge of the cost surface J. From (9) it can be seen that T^*_γ, $T^{loc}_{\gamma,\eta}$, t^*_γ, and t^{loc}_γ are all proportional to $\frac{1}{\mu}$.

VI. COMPARISON WITH CGD ALGORITHM

Given that the HCGD algorithm is a modification of the CGD algorithm, an obvious problem is to compare the two. This turns out to be very difficult. Whilst it is straight-forward to do an empirical comparison on a few selected problems, such an exercise provides little insight. If

instead one attempts a theoretical comparison, one encounters a number of problems which we will discuss below. Nevertheless, a comparison is possible, although the results are not as clear cut as one would have hoped.

In order to make a comparison tractable, we will only compare the expected computation required for the two algorithms, and will make a number of approximations. We will use the limiting value of the expected time to convergence. If the parameters $\gamma, \alpha, \beta, \mu$ of the HCGD algorithm are sufficiently small, then $\hat{k}_N := \mathbb{E}(\hat{t})$, the expected number of iterations to convergence, approximately satisfies

$$\frac{1 - N\sigma}{(N-1)\sigma}t^{loc} + t^* \leq \hat{k}_N \leq \frac{1-\sigma}{(N-1)\sigma}(t^{loc} + 1) + t^* + 1, \qquad (13)$$

where t^* is the expected value of the maximum of τ and the time to converge for estimates starting in A^*, and t^{loc} is the expected value for estimates starting outside A^*. Since μ is small, t^* and t^{loc} are large, so that $t^* + 1 \approx t^*$. Here the time to convergence has been indexed by the number of members in the congregation. Since N members are updated at every iteration, the expected computation is proportional to $N\hat{k}_N$ as N changes.

In [2] it is shown that the expected computation for the CGD algorithm with two members is approximately

$$\frac{2K(1 - \sigma + \sigma^2)}{\sigma}, \qquad (14)$$

where K is the epoch length, which is the time that members are run until they are restarted. It is shown in [2] that the expected computation for a two member congregation is no more than twice the expected computation for an optimally sized congregation. The epoch length is chosen as a constant throughout the CGD algorithm, and (14) is derived under the assumption that K is sufficiently long that all members which are initialised in A^* have time to converge to a small neighbourhood around the global minimum before the end of the epoch. Thus K corresponds to the maximum time T^*_γ defined Section IV.

If $N = 2$, the bounds in (13) are no more than twice their minimum value, so we consider only two member congregations in comparing the two algorithms. We assume that the same values of μ and α are used for both algorithms. The parameters γ, τ, β appear in the HCGD algorithm but not the CGD algorithm, and the parameter K appears in CGD algorithm but not the HCGD algorithm. The choice of τ and β does not greatly affect the expected computation calculations. As γ increases in the HCGD algorithm, the time to convergence decreases, but the definition of convergence becomes weaker since $C_\gamma(a^*)$ increases. As K decreases in the CGD algorithm, the effective value of σ decreases, since less and less estimates have time to converge by the end of the epoch in the CGD algorithm.

A difficulty now arises that we have to determine what values to set the various parameters in order to make a fair comparison between the two algorithms. One could argue that the optimal values of γ and K should be chosen, although we shall see that this does not lead to a conclusive result. Nevertheless, to this end, choose $\lambda > 0$, and let $N_\lambda = \{a : \|a - a^*\| < \lambda\}$. Let us say that the algorithms have converged when the best estimate satisfies $\hat{a} \in N_\lambda$. So in the expected computation expression for the HCGD algorithm, t^*_γ will be replaced with T^*_λ, the expected value of the maximum of τ and the time taken for estimates initialised in A^* to enter N_λ. That leads to

the requirement that $C_\gamma(a^*) \subset N_\lambda$. The effective value of σ is $\sigma(K) = Pr\{a_0 \in A(K)\}$, where

$$A(K) := \{a_0 \in A : a(K) \in N_\lambda \text{ where } a(t) \text{ is the solution of (9)}\}.$$

For each member of the congregation, the HCGD algorithm updates two m dimensional estimates (the parameter and the gradient), and one scalar estimate (the cost), whereas the CGD algorithm updates only the cost and the parameter. Therefore the number of calculations in each iteration of the HCGD algorithm is approximately double the calculations in each iteration of the CGD algorithm. In most situations where a congregational type of algorithm would be applied, the size of the basin of attraction of the global minimum would be small, so that the $\frac{1}{\sigma}$ terms in the expression for the expected computation are dominant. Therefore the two quantities

$$C_H(\gamma) := 4t_\gamma^{loc}\left(\frac{1}{\sigma} - 1\right) + 4T_\lambda^* \qquad (15)$$

$$C_C(K) := 2K\left(\frac{1}{\sigma(K)} - 1\right) \qquad (16)$$

approximately characterise the expected computation of the HCGD and CGD algorithms respectively.

A quick glance at these formulae confirms that the key difference between the HCGD and CGD algorithms is the t_γ^{loc} versus K terms, and this captures the idea that the HCGD algorithm will stop sooner sometimes. We have (unsuccessfully) attempted to construct cost functions for which for a given $c > 0$, the rather strong assertion that $\min_K C_C(K) > c \min_{\gamma:C_\gamma(a^*) \subset N_\lambda} C_H(\gamma)$ or conversely that $\min_{\gamma:C_\gamma(a^*) \subset N_\lambda} C_H(\gamma) < c \min_K C_C(K)$ holds. The difficulty is that by allowing the *optimal* value of the tunable parameters, we seem to neutralize any advantage that one algorithm has over the other. Furthermore, the optimal choice is never available in practice, and so it is a rather unrealistic comparison after all. We can easily give examples of cost functions for which, for particular choices of parameters, HCGD outperforms CGD, or *vice versa*, but that is not very satisfying either. We have not included details of this, as little insight is gained. The (effectively negative) result in the next section is some small consolation for our failure to provide a more useful comparison between the algorithms. Perhaps the one positive conclusion we can offer is that the HCGD algorithm may be preferable in the sense that there may be better grounds for the choice of γ (in terms of the quality of the desired solution perhaps) than on K; but of course there is no rigorous argument for this!

In the following lemmas we give 1 dimensional examples which show that J can be chosen to make the optimal values of these expected computation operators arbitrarily far apart, if favour of either algorithm.

Lemma 5: Assume $\lambda > 0$. For all $c > 0$ there exists $p > 0$ and a continuous cost function $J : [0, p] \to \mathbb{R}^+$ such that if $D_a = U(0, p)$ then the expected computation operators for the HCGD and CGD algorithms satisfy

$$\min_K C_C(K) > c \min_{\gamma:C_\gamma(a^*) \subset N_\lambda} C_H(\gamma).$$

Proof:

Fig. 3. The cost function defined in (VI), with $n = 2$.

Let $p = 2nR$ for some n to be determined and

$$J(a) = \begin{cases} (a - R)^2 & a \in [0, 2R] \\ R^2 - r^2 + (a - 2R - (2i-1)r)^2 & a \in [2(R + ir - r), 2(R + ir)] \end{cases} \quad \forall i \in \mathbb{N},$$

where $R = \lambda e^{4\tau}$, $r = \lambda e^{2\tau}$. Then $a^* = R$, $A^* = [0, 2R]$, and each $R + (2i-1)r$ is a local minimum of J. This cost function is depicted in Figure 3. The gradient of J satisfies

$$\frac{dJ}{da} = \begin{cases} 2(a - R) & a \in [0, 2R] \\ 2(a - 2R - (2i-1)r) & a \in [2(R + ir - r), 2(R + ir)] \end{cases} \quad \forall i \in \mathbb{N}.$$

Therefore $N_\lambda = C_{2\lambda}$. and $C_\gamma (a^{loc} = \{a : \|a - 3R\| \leq \frac{\gamma}{2}\}$. Differentiating and substituting in (9) gives

$$\ln\left(\frac{a(t) - a^*}{a_0 - a^*}\right) = -2\mu t \quad a \in [0, 2R]$$

$$\ln\left(\frac{a(t) - 2R - (2i-1)r}{a_0 - 2R - (2i-1)r}\right) = -2\mu t \quad a \in [2(R + ir - r), 2(R + ir)] \quad \forall i \in \mathbb{N}.$$

For the CGD algorithm, $A(K) = \{a_0 : \|a_0 - a^*\| e^{-2\mu K} \leq \lambda \text{ and } \|a_0 - a^*\| \leq R\}$. Since D_a is uniform on $[0, p]$,

$$\sigma(K) = \begin{cases} \frac{2\lambda e^{2K}}{2nR} & K \leq \frac{1}{2} \ln\left(\frac{R}{\lambda}\right) \\ \frac{2R}{2nR} & K > \frac{1}{2} \ln\left(\frac{R}{\lambda}\right). \end{cases}$$

Now $\frac{K}{\sigma(K)}$ is decreasing until $K = \frac{1}{2} \ln\left(\frac{R}{\lambda}\right)$, so

$$\min_K C_C(K) = \ln\left(\frac{R}{\lambda}\right)(n - 1) = 4\tau n - 4\tau.$$

For the HCGD algorithm, $\sigma = \frac{1}{n}$, and $t_\gamma(a_0) = \max\left\{\tau, \frac{1}{2} \ln\left(\frac{2\|a_0 - a^*\|}{\gamma}\right)\right\}$. Therefore

$$\begin{aligned} T_\lambda^* &= \int_0^{\lambda e^{2\tau}} \tau \frac{dx}{p} + \int_{\lambda e^{2\tau}}^R \frac{1}{2} \ln\left(\frac{x}{\lambda}\right) \frac{dx}{p} \\ &= \frac{\tau \lambda e^{2\tau}}{2nR} + \frac{1}{4nR}\left[x\left(\ln\left(\frac{x}{\lambda}\right) - 1\right)\right]_{\lambda e^{2\tau}}^{n\lambda e^{4\tau}} \\ &= \tau \ln(n) - \frac{1}{4} + \frac{1}{4n} e^{-2\tau} \end{aligned}$$

and similarly

$$t_\gamma^{loc} = \begin{cases} \frac{\tau r}{2nR} & r \le \frac{\gamma}{2}e^{2\tau} \\ \frac{1}{4nR}\left(r[\ln(\frac{2r}{\gamma}) - 1] + \frac{\gamma}{2}e^{2\tau}\right) & r > \frac{\gamma}{2}e^{2\tau}. \end{cases}$$

The second form of t_γ^{loc} decreases as γ increases, so $\min_{\gamma:C_\gamma(a^*)\subset N_\lambda} t_\gamma^{loc} = \frac{\tau r}{2nR} = \frac{\tau e^{-2\tau}}{2n}$. Substituting into (15) gives

$$\min_{\gamma:C_\gamma(a^*)\subset N_\lambda} C_H(\gamma) = 4\tau\ln(n) - 1 + \frac{(1 + 2\tau)e^{-2\tau}}{n}.$$

Clearly n can be chosen large enough to ensure that the result holds. ∎

Lemma 6: Assume $\lambda > 0$. For all $c > 0$ there exists $p > 0$ and a continuous cost function $J : [0, p] \to \mathbb{R}^+$ such that if $D_a = U(0, p)$ then the expected computation operators for the HCGD and CGD algorithms satisfy

$$\min_{\gamma:C_\gamma(a^*)\subset N_\lambda} C_H(\gamma) > c \min_K C_C(K).$$

Proof: Let $p = 4R$ and

Fig. 4. The cost function defined in (VI).

$$J(a) = \begin{cases} c(a - R)^2 & a \in [0, 2R] \\ (c - 1)R^2 + (a - 3R)^2 & a \in [2R, 4R], \end{cases}$$

where $R = 2\lambda$. Then $a^* = R$, $A^* = [0, 2R]$, and $3R$ is a local minimum of J. This cost function is depicted in Figure 3. The gradient of J satisfies

$$\frac{dJ}{da} = \begin{cases} 2c(a - R) & a \in [0, 2R] \\ 2(a - 3R) & a \in [2R, 4R]. \end{cases}$$

Therefore $N_\lambda = C_{2c\lambda}$ and $C_\gamma(a^{loc} = \{a : \|a - 3R\| \le \frac{\gamma}{2}\}$. Differentiating and substituting in (9) gives

$$\ln\left(\frac{a(t) - a^*}{a_0 - a^*}\right) = -2\mu ct \qquad a \in [0, 2R]$$

$$\ln\left(\frac{a(t) - 3R}{a_0 - 3R}\right) = -2\mu t \qquad a \in [2R, 4R].$$

For the CGD algorithm, $A(K) = \{a_0 : \|a_0 - a^*\|e^{-2\mu cK} \le \lambda$ and $\|a_0 - a^*\| \le R\}$. Since D_a is uniform on $[0, p]$,

$$\sigma(K) = \begin{cases} \frac{2\lambda e^{2cK}}{4R} & K \le \frac{1}{2c}\ln\left(\frac{R}{\lambda}\right) \\ \frac{2R}{4R} & K > \frac{1}{2c}\ln\left(\frac{R}{\lambda}\right). \end{cases}$$

Now $\frac{K}{\sigma(K)}$ is decreasing until $K = \frac{1}{2c} \ln\left(\frac{R}{\lambda}\right)$, so

$$\min_K C_C(K) = \frac{1}{c} \ln\left(\frac{R}{\lambda}\right) = \frac{\ln(2)}{c}.$$

For the HCGD algorithm, $\sigma = \frac{1}{2}$, and for $a_0 \in A^*$ $t_\gamma(a_0) = \max\left\{\tau, \frac{1}{2c} \ln\left(\frac{2\|a_0 - a^*\|}{\gamma}\right)\right\}$. For large c, $R < \lambda e^{2c\tau}$, so $T_\lambda^* = \frac{2R\tau}{4R} = \frac{\tau}{2}$. For $a_0 \in A^{loc}$, $t_\gamma(a_0) = \max\left\{\tau, \frac{1}{2} \ln\left(\frac{2\|a_0 - a^*\|}{\gamma}\right)\right\}$ so

$$t_\gamma^{loc} = \begin{cases} \frac{2R\tau}{4R} & R \leq \frac{\gamma}{2} e^{2\tau} \\ \frac{1}{8R}\left(r[\ln(\frac{2r}{\gamma}) - 1] + \frac{\gamma}{2} e^{2\tau}\right) & R > \frac{\gamma}{2} e^{2\tau}. \end{cases}$$

The second form of t_γ^{loc} decreases as γ increases, so $\min_{\gamma:C_\gamma(a^*) \subset N_\lambda} t_\gamma^{loc} = \frac{\tau}{2}$. Substituting into (15) gives

$$\min_{\gamma:C_\gamma(a^*) \subset N_\lambda} C_H(\gamma) = 4\tau.$$

The result holds since $\tau \geq 1$. ■

VII. DENSITY OF QUADRATIC LEARNING PROBLEMS

Whilst the arguments above indicate that a J can always be found to give advantage to one of the particular algorithms, it may be argued that not all possible J can arise in "natural" learning problems. One might think that the way in which J arises in such problems imposes constraints on the range of J that are possible. We now show however that there is no constraint when J arises as the average cost in a learning problem with quadratic cost.

In order to prove the main result of this section we need to be able to simultaneously approximate a function J and its derivatives by polynomials with positive coefficients. We make use of the particular properties of Bernstein polynomials as captured in the following result:

Theorem 7: Let $A = [0,1]^m$. Suppose $J : A \to \mathbb{R}$ is continuous, nonnegative and $J_i'(a) := \frac{\partial}{\partial a_i} J(a)$ exists and is continuous for $i = 1, \ldots, m$. Then for all $\varepsilon > 0$ there exists $p \in \mathbb{N}$ and $c_{k_1, \ldots, k_m} \geq 0$ for all $k_1, \ldots, k_m \in \{0, \ldots, p\}$ such that

$$B_p(a) = B_p(J; a) = \sum_{k_1=0}^{p} \cdots \sum_{k_m=0}^{p} c_{k_1, \ldots, k_m} \prod_{i=1}^{m} a_i^{k_i} (1 - a_i)^{p - k_i}$$

satisfies

$$|J(a) - B_p(a)| < \varepsilon \quad \forall a \in A, \tag{17}$$

and

$$|J_i'(a) - B_p'(a)| < \varepsilon \quad \forall a \in A, \ i = 1, \ldots, m, \tag{18}$$

where $B_p'(a) = \frac{\partial}{\partial a_i} B_p(a)$.

Proof: Result (17) follows from Theorem 6.2.2 of [3] concerning Bernstein polynomials and the fact that the coefficients are given by $c_{k_1, \ldots, k_m} = \prod_{i=1}^{m} \binom{p}{k_i} J(\frac{k_1}{p}, \ldots, \frac{k_m}{p}) \geq 0$. Result (18) follows from lemma 8 below. ■

Lemma 8: Suppose $f : [0,1]^m \to \mathbb{R}$ is continuous and that $\frac{\partial}{\partial a_i} f(a)$ exists and is continuous for $i = 1, \ldots, m$. Let $B_p(f; a)$ be the pth multidimensional Bernstein polynomial approximant for f. Then for $i = 1, \ldots, m$,

$$\lim_{p \to \infty} \frac{\partial}{\partial a_i} B_p(f; a) = \frac{\partial}{\partial a_i} f(x)$$

uniformly in $[0,1]^m$.

Proof: This follows along identical lines to the proof of theorem 6.3.2 of [3] by using the multidimensional mean value theorem and the generalized Bernstein polynomials. ∎

We will also make use of a standard L^2 orthonormal approximation result [3, page 265]:

Theorem 9: Let $X = [0,1]^n$ and let $\psi_i : X \to \mathbb{R}$, $i \in \mathbb{N}$, be a set of orthonormal polynomials, i.e.

$$\int_X \psi_i(x)\psi_j(x)dx = \begin{cases} 0 & i \neq j \\ 1 & i = j. \end{cases}$$

For all $y : X \to \mathbb{R}$, $y \in L_2$, and all $\varepsilon > 0$, there exists $s \in \mathbb{N}$ and $c_1, \ldots, c_s \in \mathbb{R}$ such that

$$\int_X \left[y(x) - \sum_{i=1}^{s} c_i \psi_i(x) \right]^2 dx < \varepsilon$$

Our main result in this section is:

Theorem 10: Let $X \subset \mathbb{R}^n$, $A \subset \mathbb{R}^m$ be compact. For all $y : X \to \mathbb{R}$ and $J : A \to \mathbb{R}$ such that $y \in L^2$ and J is continuous, nonnegative and $J'_i(a)$ exists and is continuous, and all $\varepsilon > 0$, there exists a parametrisation $f : A \times X \to \mathbb{R}$ such that for all sequences (x_k) that cover X,

$$\left| J(a) - \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} [f(a, x_k) - y(x_k)]^2 \right| < \varepsilon \quad \forall a \in A. \tag{19}$$

and for $i = 1, \ldots, m$,

$$\left| J'_i(a) - \frac{\partial}{\partial a_i} \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} [f(a, x_k) - y(x_k)]^2 \right| < \varepsilon \quad \forall a \in A. \tag{20}$$

Proof: Assume w.l.o.g. $\varepsilon < \max_{a \in A} |J(a)| =: m$ and $\varepsilon < 1$. Choose a function $B_p(a)$ according to Theorem 7 to approximate J with error $\frac{\varepsilon}{3}$. There are $(p+1)^m$ monomial terms in B_p. Choose an orthonormal polynomial approximation $\sum_{i=i}^{s} c_i \psi(x)$ to approximate y with error $\frac{\varepsilon^2}{72m} < \frac{\varepsilon}{3}$.

Let $d = \max\{s, (p+1)^m\}$. Let $\lambda(a) \in \mathbb{R}^d$ be the vector with elements equal to the square roots of monomials in $B_p(a)$. That is for $j = \sum_{i=1}^{m} k_i(p+1)^{m-i}$, let $\lambda(a)_j = \left(c_{k_1, \ldots, k_m} \prod_{i=1}^{m} a_i^{k_i}(1-a_i)^{p-k_i} \right)^{\frac{1}{2}}$, and for $(p+1)^m < j \leq d$, let $\lambda(a)_j = 0$.

Let $\psi(x) \in \mathbb{R}^d$ be the vector with elements equal to the orthonormal polynomials $\psi_i(x)$ for $i \leq s$ and zero if $s < i \leq d$. Let $c \in \mathbb{R}^d$ be a vector with elements equal to the coefficients c_i defined by the orthonormal approximation to y for $i \leq s$ and zero if $s < i \leq d$. Thus $c = \int_X \psi(x)y(x)dx$.

Define the parametrization $f(a, x) = [\lambda(a) + c]^\top \psi(x)$. Then for any sequence (x_k) of points which cover X, and for any $a \in A$,

$$\left| J(a) - \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} [f(a, x_k) - y(x_k)]^2 \right|$$

$$= \left| J(a) - \int_X \left[(\lambda(a) + c)^\top \psi(x) - y(x) \right]^2 dx \right|$$

$$\leq \left| J(a) - \int_X \lambda(a)^\top \psi(x) \psi(x)^\top \lambda(a) dx \right|$$

$$+ 2 \left| \int_X \lambda(a)^\top \psi(x) (c^\top \psi(x) - y(x)) dx \right| + \left| \int_X \left[c^\top \psi(x) - y(x) \right]^2 dx \right|$$

$$\leq \left| J(a) - \lambda(a)^\top \int_X \psi(x) \psi(x)^\top dx \lambda(a) \right|$$

$$+ 2 \left(\int_X \left| \lambda(a)^\top \psi(x) \right|^2 dx \right)^{\frac{1}{2}} \left(\int_X |c^\top \psi(x) - y(x)|^2 dx \right)^{\frac{1}{2}} + \frac{\varepsilon}{3}$$

using the Schwarz inequality on the second term and Theorem 9 on the third term

$$\leq \left| J(a) - \lambda(a)^\top \lambda(a) \right| + 2 \left(\lambda(a)^\top \int_X \psi(x) \psi(x)^\top dx \lambda(a) \right)^{\frac{1}{2}} \left(\frac{\varepsilon^2}{72m} \right)^{\frac{1}{2}} + \frac{\varepsilon}{3}$$

$$\leq \frac{\varepsilon}{3} + 2 \left(2m \frac{\varepsilon^2}{72m} \right)^{\frac{1}{2}} + \frac{\varepsilon}{3}$$

and we have thus shown (19). In order to show (20), firstly observe that since $B_p(a) = \lambda(a)^\top \lambda(a)$, we have $B_p'(a) = 2\lambda_i'(a)^\top \lambda(a)$ where $\lambda_i'(a) = \frac{\partial}{\partial a_i} \lambda(a)$. Now for any $i = 1, \ldots, m$ and any $a \in A$ consider

$$\left| J_i'(a) - \frac{\partial}{\partial a_i} \lim_{K \to \infty} \frac{1}{K} \sum_{k=0}^{K-1} [f(a, x_k) - y(x_k)]^2 \right|$$

$$= \left| J_i'(a) - \frac{\partial}{\partial a_i} \int_X [(\lambda(a) + c)^\top \psi(x) - y(x)]^2 dx \right|$$

$$= \left| J_i'(a) - 2 \int_X [(\lambda(a) + c)^\top \psi(x) - y(x)] \lambda_i'(a)^\top \psi(x) dx \right|$$

$$= \left| J_i'(a) - 2 \int_X (\lambda(a) + c)^\top \psi(x) \psi(x)^\top \lambda_i'(a) + y(x) \psi(x)^\top \lambda_i'(a) dx \right|$$

$$= \left| J_i'(a) - 2\lambda(a)^\top \lambda_i'(a) - 2 \left[\int_X y(x) \psi(x)^\top dx - c^\top \right] \lambda_i'(a) \right|$$

$$= \left| J_i'(a) - B_p'(a) \right| < \varepsilon.$$

$\blacksquare$

Since both versions of CGD studied in this paper have behaviour governed by $J(a)$ and the gradient of $J(a)$, we have shown that "all" Js are possible.

VIII. CONCLUSIONS

An algorithm for online learning of an average cost function has been proposed. The algorithm is based on the use of a population of stepwise gradient descent algorithms which are periodically tested for fitness and restarted when deemed to be unfit. We have shown that the algorithm is globally convergent, by showing that the probability that the nominated estimate is in a small region of the global minimum increases with time, and the limiting value of the bound can be made arbitrarily close to 1. Moreover, we have found bounds on the expected number of iterations until the algorithm converges. Although not "practical", more can be said theoretically about our algorithm than about GAs applied to similar problems. The algorithm is naturally parallelizable.

The HCGD algorithm was devised as an alternative to a simpler congregational gradient descent algorithm presented in [2], with a view to reducing the expected computation necessary for global minimization. The CGD algorithm updates all members for a fixed time, and thus updates some members after they have converged, whereas the HCGD algorithm restarts members once they have converged to local minima. Thus the HCGD algorithm should converge with fewer iterations. However in order to determine when the estimates have converged, the HCGD algorithm updates an extra online estimate, so the computation at each iteration is increased. Further investigations will be required in order to determine whether the increased sophistication of this algorithm is warranted, since the comparison between the two algorithms is less than clear. The considerably more complex analysis (relative to [2]) certainly recommends against further complications/refinements to the algorithm!

Acknowledgement

REFERENCES

[1] K.L. Blackmore, Nonlinear Parameter Estimation in Classification Problems. PhD Thesis, Australian National University, 1995.

[2] K.L. Blackmore, R.C. Williamson, I.M.Y. Mareels, and W.A. Sethares. Online learning via congregational gradient descent. *Mathematics of Control, Signals and Systems*, **10**(4), 331–363, 1997. in March 1995.

[3] P.J. Davis. *Interpolation and Approximation*. Dover, New York, 1975.

[4] Z. Ding, R.A. Kennedy, B.D.O. Anderson, and C.R. Johnson Jr. Ill-convergence of Godard blind equalizers in data communication systems. *IEEE Transactions on Communications*, 39(9):1313–1327, 1991.

[5] S. Forrest. Genetic algorithms: Principles of natural selection applied to computation. *Science*, 261:972–978, 13 August 1993.

[6] C.R. Johnson Jr. *Lectures on Adaptive Parameter Estimation*. Prentice-Hall, New York, 1988.

[7] B. Widrow and M.A. Lehr. 30 years of adaptive neural networks: Perceptron, madeline, and backpropagation. In C. Lau, editor, *Neural Networks*, pages 27–53. IEEE Press, New York, 1992.

APPENDIX

I. COMPLETE ANALYSIS

In order to adapt the above analysis for the HCGD algorithm, the effect of each of the simplifications must be considered.

Following the analysis of the simpler algorithm, for any $\eta \in (0, 1)$, it is choose a constant γ_0 such that each connected component of F_{γ_0} contains exactly one critical point of J (or one connected set of critical points of J); $C_{\gamma_0}(a^*) \cap F_{\gamma_0} = N_{\gamma_0}(a^*)$; and $\sigma_{\gamma_0} \geq (1 - \eta)\sigma$. These relationships will hold for all $\gamma \leq \gamma_0$.

Since SGD is used instead of continuous time gradient descent of the average cost function, parameter estimates do not move perpendicular to the contour lines of J. In particular an estimate cannot be guaranteed to stay in the contour set $C_\gamma(a^*)$ once it enters the set. In the following lemma it is shown that, if the parameter step size is sufficiently small, a SGD parameter estimate stays in $C_\gamma(a^*)$ if it ever enters some smaller contour set.

Lemma 11: With assumptions *C1* to *C6*, let a_k be defined by (5) and choose γ_0 as above. For all $\gamma_2 < \gamma_1 \leq \gamma_0$, there exists μ_0 such that if $\mu \leq \mu_0$ and $a_0 \in C_{\gamma_2}(a^*)$ then $a_k \in C_{\gamma_1}(a^*)$ for all $k \geq k_0$.

Proof: Consider the derivative of the cost function J along trajectories of (9)

$$\dot{J}(a_{av}(t)) = -\mu \left. \frac{\partial J}{\partial a} \right|_{a_{av}(t)} \left(\left. \frac{\partial J}{\partial a} \right|_{a_{av}(t)} \right)^\top \quad ; \qquad a(t_0) = a_0. \tag{21}$$

Assume $\gamma < \gamma_0$. For all $a_0 \in C_{\gamma_2}(a^*)$, $a(t) \in C_{\gamma_2}(a^*) \subset C_{\gamma_1}(a^*) \subset A^*_{\gamma_0}$. The only critical point of J in $A^*_{\gamma_0}$ is a^*, and the gradient of J is greater than γ_0 in $A_{\gamma_0} \backslash N_{\gamma_0}(a^*)$, so there exists a constant $c > 0$ such that for all $a \in A^*_{\gamma_0}$

$$J(a) - J(a^*) \leq c \left. \frac{\partial J}{\partial a} \right|_a \left(\left. \frac{\partial J}{\partial a} \right|_a \right)^\top. \tag{22}$$

Then for all $a_0 \in C_{\gamma_2}(a^*) \subset A_{\gamma_0}$,

$$\dot{J}(a_{av}(t)) \leq -c\mu \left(J(a_{av}(t)) - J(a^*) \right) \quad ; \qquad a(t_0) = a_0. \tag{23}$$

Solving (23) gives

$$J(a_{av}(t)) - J(a^*) \leq \left(J(a_0) - J(a^*) \right) e^{-c\mu(t-t_0)}$$
$$\leq \left(J_{\gamma_2}(a^*) - J(a^*) \right) e^{-c\mu(t-t_0)}.$$

Now let $t_0 = k_0$. For any $k \geq k_0$,

$$J(a_k) - J(a^*) \leq J(a_{av}(k)) - J(a^*) + |J(a_k) - J(a_{av}(k))|.$$

The cost function J is Lipschitz continuous by Assumption *C3*. Thus result *1* of Theorem 3 shows that there exists some $o_\mu(1)$ function $l(\mu)$ and a constant $L > 0$ such that

$$J(a_k) - J(a^*) \leq \left(J_{\gamma_2}(a^*) - J(a^*) \right) e^{-c\mu(k-k_0)} + l(\mu)$$

for $k \leq \left\lfloor \frac{L}{\mu} \right\rfloor$. In particular, if μ is sufficiently small, $a_k \in C_{\gamma_1}(a^*)$ for $k \leq \left\lfloor \frac{L}{\mu} \right\rfloor$ and $a_{\left\lfloor \frac{L}{\mu} \right\rfloor} \in C_{\gamma_2}(a^*)$. The argument can now be repeated, with the initial condition for (9) replaced by $t_0 = \left\lfloor \frac{L}{\mu} \right\rfloor$, $a(t_0) = a_{\left\lfloor \frac{L}{\mu} \right\rfloor}$. The result follows. ∎

The online estimates of the cost and gradient are not exactly equal to the true values of the cost and gradient. Therefore the algorithm may decide that an estimate is in the flat region when in fact it isn't, and it may restart the current best estimate and keep an estimate with greater cost instead. In the following lemma it is shown that the algorithm parameters can be chosen in order to ensure that the cost and gradient estimates are arbitrarily accurate.

Lemma 12: With Assumptions *C1* to *C6*. let a_k be defined according to (5), and let Φ_k^n, Γ_k^n be defined according to (7) and (8) respectively, with initial conditions $\Phi_{k_0}^n = 0$, $\Gamma_{k_0}^n = 0$. There exists constants ε_0, B_ϕ and $B_{\partial\phi}$ such that for any positive $\varepsilon < \varepsilon_0$, there exists α_ε such that if $\alpha \leq \alpha_\varepsilon$ and $k \geq k_0 - \frac{1}{\alpha}\ln\frac{\varepsilon}{4B_\phi}$, then

$$|\Phi_k^n - J(a_k^n)| \leq \varepsilon.$$

Similarly, there exists β_ε such that if $\beta \leq \beta_\varepsilon$ and $k \geq k_0 - \frac{1}{\beta}\ln\frac{\varepsilon}{4B_{\partial\phi}}$, then

$$\left\| \Gamma_k^n - \frac{\partial J}{\partial a}\bigg|_{a_k^n} \right\| \leq \varepsilon.$$

Proof: The iterative definition of Φ_k^n in equation 7 can be rewritten in summation form as

$$\Phi_k^n = \alpha \sum_{j=k_0}^{k-1} (1-\alpha)^{k-j-1} \phi(a_j, x_j).$$

The parameter estimate a_k originates in the compact set A^0, and remains bounded since 9 is Lagrange stable. The input x_k is also bounded, so the value of the instantaneous cost is always bounded, by some constant B_ϕ. Thus

$$|\Phi_k^n| \leq \alpha \sum_{j=k_0}^{k-1} (1-\alpha)^{k-j-1} |\phi(a_j, x_j)|$$

$$\leq \alpha B_\phi \sum_{j=k_0}^{k-1} (1-\alpha)^{k-j-1}$$

$$\leq \alpha B_\phi \sum_{j=0}^{k-k_0-1} (1-\alpha)^j$$

$$= \alpha B_\phi \frac{1 - (1-\alpha)^{k-k_0}}{1 - (1-\alpha)}$$

$$\leq B_\phi$$

Similarly, the value of the gradient of the instantaneous cost is always bounded by some constant, denoted $B_{\partial\phi}$, and $\|\Gamma_k^n\| \leq B_{\partial\phi}$. Thus the cost and gradient estimates are bounded.

Let $\varepsilon_0 = \min\{B_\phi, B_{\partial\phi}\}$. Then for all positive $\varepsilon < \varepsilon_0$, $\ln\frac{\varepsilon}{4B_\phi}$ and $\ln\frac{\varepsilon}{4B_{\partial\phi}}$ are negative. Choose $j \geq k_0 - \frac{1}{\alpha}\ln\frac{\varepsilon}{4B_\phi}$, and define $k_0' = j + \lceil\frac{1}{\alpha}\ln\frac{\varepsilon}{4B_\phi}\rceil$. Then $k_0 < k_0' \leq j$ for all $\varepsilon < \varepsilon_0$.

Equation 7 can also be rewritten as

$$\Phi_{k+1}^n = \Phi_k^n - \alpha\left(\Phi_k^n - \phi(a_j^n, x_k)\right) - \alpha\left(\phi(a_j^n, x_k) - \phi(a_k^n, x_k)\right). \tag{24}$$

This has the form of equation 2, where the small parameter α replaces μ, and a_k in Theorem 3 is identified with Φ_k^n here. Identify $H(\Phi, x)$ with $\Phi - \phi(a_j^n, x)$, $h_k(\Phi, x)$ with $\frac{\alpha}{\mu}(\phi(a_j^n, x) - \phi(a_k^n, x))$, and $\beta(\alpha)$ with $\frac{\mu}{\alpha}$, and consider the initial time k_0'. Then if $k_0' \leq k \leq j$,

$$|h_k(\Phi, x)| \leq \frac{\alpha}{\mu}\lambda_\phi\|a_j^n - a_k^n\|$$
$$\leq \frac{\alpha}{\mu}\lambda_\phi\mu B_{\partial\phi}(j - k),$$

since $a_j^n = a_k^n - \mu\sum_{l=j}^{k-1}\frac{\partial\phi}{\partial a}\big|_{(a_l^n, x_l)}$. Using the fact that $k \geq k_0'$ gives

$$|h_k(\Phi, x)| \leq -\ln\frac{\varepsilon}{4B_\phi}\lambda_\phi B_{\partial\phi},$$

where the definition of a_j has been used, and then the fact that $k \geq k_0$. Therefore $h_k = O_\alpha(1)$, and $\beta(\alpha) = o_\alpha(1)$ by the Assumption *C6*.

The averaged ODE associated with (24) is

$$\dot{\Phi}_{av} = -\alpha(\Phi_{av} - J(a_j^n)), \tag{25}$$

with initial condition $\Phi_{av}(k_0') = \Phi_{k_0'}^n$. The ODE (25) has solution

$$\Phi_{av}(t) - J(a_j^n) = (\Phi_{k_0'}^n - J(a_j^n))e^{-\alpha(t - k_0')} \tag{26}$$

Using the definition of k_0' and the fact that $|\Phi_{k_0'}^n|$, $|J(a_j^n)| \leq B_\phi$ gives

$$|\Phi_{av}(k) - J(a_j^n)| \leq \frac{\varepsilon}{2}. \tag{27}$$

Equation 26 shows that $J(a_j^n)$ is the globally exponentially stable solution of (25), and result *2* of Theorem 3 applies. Therefore the solutions of (24) and (25) satisfy

$$\|\Phi_k^n - \Phi_{av}(k)\| \leq o_\alpha(1)$$

for all $k \geq k_0$. Combining with (27) shows that there exists α_ε sufficiently small that

$$|\Phi_j^n - J(a_j^n)| \leq \varepsilon.$$

The proof of the accuracy of the gradient estimate follows similarly. ∎

Let the value of the cost at the smallest non-global local minimum be denoted J^{loc}:

$$J^{loc} := \min\{J(a) : J(a) \text{ is a non-global local minimum of } J\}$$

If there are no non-global local minima of J, any value of $J^{loc} > J_{\gamma_0}(a^*)$ can be used. Let $\gamma \in (0,1)$. To prove convergence of the HCGD algorithm, we choose $\varepsilon < \frac{J^{loc} - J_{(1+\gamma)\gamma}(a^*)}{2}$ so that the cost estimate of $a \in C_{(1+\gamma)\gamma}$ is always smaller than the cost estimate for $a' \notin A^*$. We also choose ε sufficiently small to ensure that $a \in F_{(1+\gamma)\gamma}$ whenever the gradient estimate less than γ.

From Lemma 12 it can be seen that the transition time τ in the HCGD algorithm is needed in order to allow the cost and gradient estimates approach the true cost and gradient. In particular, the estimates are initialised at 0, so the algorithm would immediately restart every member if the transition time was removed. Because the transition time is nonzero, the evolution times of the members used in the calculation of results 1 and 2 must be increased, so that the maximum convergence times $T^{loc}_{\gamma,\eta}$ and T^*_{γ} must be replaced with $\max\{\tau, T^{loc}_{\gamma,\eta}\} + 1$ and $\max\{\tau, T^*_{\gamma}\} + 1$. The $+1$ term allows for the discrete index on the SGD estimates, as distinct from the continuous time index for true gradient descent estimates. Similarly the expected times must be replaced with the times

$$\mathbb{E}\left(\max\{t_\gamma(a_0), \tau\} \text{ given } a_0 \sim D_a, a_0 \in A^*_\gamma\right) + 1$$
$$\mathbb{E}\left(\max\{t_\gamma(a_0), \tau\} \text{ given } a_0 \sim D_a, a_0 \notin A^*_\gamma\right) + 1.$$

Finally, in the HCGD algorithm, estimates are updated even when they are in the flat region. This means that parameter estimates will continue to improve once they reach $N_\gamma(a^*)$ instead of stopping at its boundary. It also means that estimates initialised close to the edges of a basin of attraction, near a local maximum or a saddle point, may leave the component of F_γ near the boundary and converge to a local minimum before being restarted. This effectively increases the probability that a newly initialised estimate will converge to a^*—for continuous time gradient descent the probability would be between σ_γ and σ.

The probability that a member initialised in the flat region near the boundary of a basin of attraction is allowed to leave that region and converge to a local minimum decreases as the algorithm runs. This is because the current best estimate continues to increase (on average), so as soon as the transition time is up, if the estimate is still in the flat region, the estimate will be restarted. However it will never become zero as some estimates will always escape the flat region by the end of the transition time.

For the purposes of estimating the probability that the algorithm has converged at any time, it is not necessary to know whether estimates converge to local minima or stay near the boundaries of the basins of attraction before being restarted. It is sufficient to know whether estimates γ-converge to a^* or always stay larger than $J_{(1+3\gamma)\gamma}(a^*)$. Since $C_{(1+3\gamma)\gamma}(a^*) \cap F_{(1+3\gamma)\gamma} = N_{(1+3\gamma)\gamma}(a^*)$, we have $J_{(1+3\gamma)\gamma}(a^*) < J^{loc}$ for all $\gamma \leq \gamma_0$.

Let $p(\mu)$ be the probability that a newly initialised parameter estimate defined by the HCGD algorithm γ-converges to a^* before being restarted, and let $q(\mu)$ be the probability that the cost of a newly initialised parameter estimate remains greater than J^{loc} until it is restarted. The function $q(\mu)$ is greater than the probability that a corresponding SGD estimate converges to some nonglobal local minimum. In the following lemma it is shown that μ and γ can be chosen in order to make p arbitrarily close to σ and q arbitrarily close to $1 - \sigma$.

Lemma 13: Consider the HCGD algorithm with Assumptions *C1* to *C6*. For all $\gamma \leq \gamma_0$, there exists μ_0 such that if $\mu \leq \mu_0$ then

$$(1 - \eta)^2 \sigma \ \leq \ p(\mu) \leq 1 - (1 - \eta)(1 - \sigma)$$
$$(1 - \eta)(1 - \sigma) \leq \ q(\mu) \ \leq 1 - (1 - \eta)^2 \sigma.$$

Proof: **of Lemma 13** To find the lower bound on $p(\mu)$, recall the argument in the proof of Lemma 11. From equation 22 it is clear that there exists a constant t_γ such that for all $a_0 \in A^*$, $a_{av}(t) \in C_{(1+\gamma)\gamma}(a^*)$ for all $t \geq t_\gamma$. By result *1* of Theorem 3, $\|a_{av}(k) - a_k\| \leq l(\mu)$ for $k \leq \lceil t_\gamma \rceil$. Choose any compact subset $B^*_{(1+3\gamma)\gamma} \subset A^*_{(1+3\gamma)\gamma}$ such that the probability of initialising in $B^*_{(1+3\gamma)\gamma}$ is $(1 - \eta)\sigma_{(1+3\gamma)\gamma}$. If μ is sufficiently small, the parameter estimate enters $C_{(1+2\gamma)\gamma}(a^*) \supset C_{(1+\gamma)\gamma}(a^*)$. By Lemma 11, the parameter estimate remains in $C_{(1+3\gamma)\gamma}(a^*)$ for all $k \geq \lceil t_\gamma \rceil$.

To see that the parameter estimate is not restarted before entering $C_{(1+2\gamma)\gamma}$, note that $A^*_{(1+3\gamma)\gamma} \subset A^*_{(1+2\gamma)\gamma}$. The averaged estimate remains in $A^*_{(1+3\gamma)\gamma}$ for all time so if μ is sufficiently small then a_k remains in $A^*_{(1+2\gamma)\gamma}$ for all time. By Lemma 12, if β is sufficiently small. $\Gamma_k \geq (1 + \gamma)\gamma$ for $a_k \in A^*_{(1+2\gamma)\gamma} \backslash N_{(1+2\gamma)\gamma}(a^*)$. Thus the algorithm will not decide that the member is flat before the parameter estimate enters $C_{(1+2\gamma)\gamma}$, for all estimates initialised in $B^*_{(1+3\gamma)\gamma}$. The probability of initialising in this set is $(1 - \eta)\sigma_{(1+3\gamma)\gamma} \geq (1 - \eta)^2 \sigma$. As discussed in the last section, the probability of convergence maybe higher than this, since estimates started in $A^* \backslash A^*_{(1+3\gamma)\gamma}$ may also be allowed to converge.

Similarly, the lower bound on $q(\mu)$ is derived by taking a union of sets B^{loc} in the interior of the basins of attraction on the non-global local minima, such that the probability of initialising in the union is equal to $(1 - \eta)(1 - \sigma)$. Result *3* of Theorem 3 shows that, if μ is sufficiently small, all estimates that originate in this set remain there, so the cost of the estimates must remain greater than J^{loc}.

The upper bounds follow from the fact that $J_{(1+3\gamma)\gamma}(a^*) \leq J^{loc}$, so that the events p and q are mutually exclusive. ∎

Updating members while they are in the flat region also increases the evolution time, since members can only be guaranteed to stay in the flat region until a better estimate appears so they can be restarted if they are in $\bigcup_{a^{loc}} N_\gamma(a^{loc})$. This set is the union of the flat neighbourhoods surrounding all of the local minima of J (including a^*). Therefore the time $t_\gamma(a_0)$ must be replaced with

$$\tilde{t}_\gamma(a_0) = \min \left\{ t - t_0 : a(t) \in \bigcup_{a^{loc}} N_\gamma(a^{loc}), \ a(t) \text{ defined by (9)} \right\}.$$

Now $\tilde{t}_\gamma(a_0)$ is not defined for a_0 in the basin of attraction of a saddle point or a local maximum, but the probability of choosing an initial value for which $\tilde{t}_\gamma(a_0)$ is not defined is zero because D_a is continuous. For each $a_0 \in \bigcup_{a^{loc}} A^{loc} \cup A^*$, there exists $\gamma(a_0)$ such that for all $\gamma < \gamma(a_0)$, $\tilde{(t)}_\gamma(a_0) = t_\gamma(a_0)$. Therefore for any set $B \subset A$, $\max\{\tilde{t}_\gamma(a_0) \text{ given } a_o \in B\} \to \max\{t_\gamma(a_0) \text{ given } a_o \in B\}$ and $\mathbb{E}\{\tilde{t}_\gamma(a_0) \text{ given } a_o \in B\} \to \mathbb{E}\{t_\gamma(a_0) \text{ given } a_o \in B\}$ as $\gamma \to 0$.

As $\gamma \to 0$, $\tilde{t}_\gamma(a_0) \to t_\gamma(a_0)$.

Proof: **of Theorem 4**

R1 Assume $a_j^{\hat{n}} \in C_{(1+2\gamma)\gamma}(a^*)$ for some j. Let $\hat{n} = n$ at time j. Assume $\gamma \le \gamma_0$. By Lemma 11, member n of the congregation remains in $C_{(1+3\gamma)\gamma}(a^*)$ until it is restarted. The assumption in Lemma 11 that $\mu < \mu_0$ can be replaced with $\alpha < \alpha_0$, since $\mu = o_\alpha(\alpha)$. It remains only to show that if member n is restarted, the new member is contained in $C_{(1+2\gamma)\gamma}(a^*)$.

Now assume that $a_k^n \in C_{(1+3\gamma)\gamma}$ is restarted, so $\Gamma_k^n < \gamma$. Lemma 12 applies, with $\varepsilon \le \min\{\varepsilon_0, \frac{1}{3}\gamma\}$, so $\Gamma_k^n < \gamma$ implies $a_k^n \in F_{(1+\gamma)\gamma}$. Now

$$
\begin{aligned}
a_k^n \in & C_{(1+3\gamma)\gamma}(a^*) \cap F_{(1+\gamma)\gamma} \\
= & \left(C_{(1+3\gamma)\gamma}(a^*) \cap F_{(1+3\gamma)\gamma}\right) \cap F_{(1+\gamma)\gamma}, \\
= & N_{(1+3\gamma)\gamma}(a^*) \cap F_{(1+\gamma)\gamma} \\
= & N_{(1+\gamma)\gamma}(a^*).
\end{aligned}
$$

Thus, by the choice of γ, $a_k^n \in C_{(1+\gamma)\gamma}(a^*)$. Since the algorithm decides to restart member n, then there must be another member m such that $\Phi_k^m < \Phi_k^n$. By Lemma 12,

$$
J(a_k^m) - \varepsilon \le J(a_k^n) + \varepsilon.
$$

Thus $J(a_k^m) \le J_{(1+\gamma)\gamma}(a^*) + 2\varepsilon \le J_{(1+2\gamma)\gamma}(a^*)$ if ε is sufficiently small. Now the retained best estimate is $a_k^m \in C_{(1+2\gamma)\gamma}$, as required.

R2 This result can be derived as for result 2 in the Section V, with a few minor corrections. The probability of initialising in A_γ^* is replaced by the probability of initialising in $B_{(1+3\gamma)\gamma}^*$ because the averaging result cannot guarantee that estimates originated arbitrarily close to the boundary of A_γ^* will converge. The convergence times $T_{\gamma,\eta}^{loc}$ and T_γ^* are replaced with

$$
\begin{aligned}
T_1 &= \max\{K, \max\{\tilde{t}_{(1-\gamma)\gamma,\eta}(a_0) \text{ given } a_0 \notin A^*\}\} + 1 \\
T_2 &= \max\{K, \max\{\tilde{t}_{(1-\gamma)\gamma,\eta}(a_0) \text{ given } a_0 \in A^*\}\} + 1.
\end{aligned}
$$

The times are changed to ensure that the gradient estimates become smaller than γ within the convergence time and stay in $F_{(1-\gamma)\gamma}$ after the convergence time. The K terms allow for the transition time and the $+1$ terms allow for the discrete index on the SGD estimates, as distinct from the continuous time index for true gradient descent estimates.

R3 Following the analysis for the simple case, let $\hat{s}$ be the number of the first initialisation for which the estimate γ-converges. From the definitions of p and q, $Pr\{\hat{s} = s\} = pq^{s-1}$, so Lemma 13 implies

$$
(1-\eta)^2 \sigma [(1-\eta)(1-\sigma)]^{s-1} \le Pr\{\hat{s} = s\} \le [1 - (1-\eta)(1-\sigma)](1 - (1-\eta)^2\sigma)^{s-1}.
$$

Therefore the expected value of $\hat{s}$ satisfies

$$
\frac{(1-\eta)^2 \sigma}{[1 - (1-\eta)(1-\sigma)]^2} \le \mathbb{E}(\hat{s}) \le \frac{\eta + (1-\eta)\sigma}{(1-\eta)^4 \sigma^2}.
$$

To complete the proof, upper and lower bounds on the expected time to convergence for estimates must be determined. For the lower bound, the convergence time is essentially the same as

for the simple algorithm, except that the time to converge to $F_{(1+\gamma)\gamma}$ is used because the gradient estimate may become less than γ if the true gradient is less than $(1+\gamma)\gamma$. For the upper bound the time to converge to sets around the local minima must be used. Let

$$
\begin{aligned}
t_\gamma^* &= \mathbb{E}\left(\max\{t_{(1+\gamma)\gamma}(a_0), K\} \text{ given } a_0 \sim D_a, a_0 \in A^*\right) \\
t_\gamma^{loc} &= \mathbb{E}\left(\max\{t_{(1+\gamma)\gamma}(a_0), K\} \text{ given } a_0 \sim D_a, a_0 \notin A^*\right) \\
\tilde{t}_\gamma^* &= \mathbb{E}\left(\max\{\tilde{t}_{(1-\gamma)\gamma}(a_0), K\} \text{ given } a_0 \sim D_a, a_0 \in A^*\right) \\
\tilde{t}_\gamma^{loc} &= \mathbb{E}\left(\max\{\tilde{t}_{(1-\gamma)\gamma}(a_0), K\} \text{ given } a_0 \sim D_a, a_0 \notin A^*\right).
\end{aligned}
$$

One problem remains—estimates originating in the region $A_0 \backslash \left(A_\gamma^* \cup B^{loc}\right)$ are not guaranteed to follow the continuous time parameter estimate, so the time until they converge is unrelated to $t_\gamma(a_0)$. Maybe it is bounded anyway? The probability of landing in this region goes to zero as γ goes to zero and you would expect that either the estimate would leave this region in a finite time or the gradient estimate would become zero in that finite time, so that either way the estimate gets restarted... ∎