In Proceedings of the 2011 IEEE Congress on Evolutionary Computation 2011.

Measure-Theoretic Evolutionary Annealing

Alan J Lockett
Department of Computer Science
University of Texas
Austin, Texas, USA
Email: alockett@cs.utexas.edu

Risto Miikkulainen
Department of Computer Science
University of Texas
Austin, Texas, USA
Email: risto@cs.utexas.edu

***Abstract*—There is a deep connection between simulated annealing and genetic algorithms with proportional selection. Evolutionary annealing is a novel evolutionary algorithm that makes this connection explicit, resulting in an evolutionary optimization method that can be viewed either as simulated annealing with improved sampling or as a non-Markovian selection mechanism for genetic algorithms with selection over all prior populations. A martingale-based analysis shows that evolutionary annealing is asymptotically convergent and this analysis leads to heuristics for setting learning parameters to optimize the convergence rate. In this work and in parallel work evolutionary annealing is shown to converge faster than other evolutionary algorithms on several benchmark problems, establishing a promising foundation for future theoretical and experimental research into algorithms based on evolutionary annealing.**

I. Introduction

Genetic algorithms typically operate in a Markov fashion, with the population for each generation constructed stochastically from the prior population only. As a result, genetic algorithms can discover and then forget high quality regions within the search domain. By forgetting prior generations regardless of quality, the algorithm can fail to exploit crucial information, resulting in suboptimal performance. This problem can be alleviated by selecting individuals for reproduction over the entire pool of previously observed solutions. However, a genetic algorithm with blind non-Markovian selection can become trapped by local optima early on. This issue can be mitigated by combining genetic algorithms and simulated annealing in a novel manner to produce an evoutionary algorithm that solidly outperforms both genetic algorithms and simulated annealing.

This paper proposes evolutionary annealing, a global optimization algorithm for arbitrary measure spaces that can be alternately viewed as a genetic algorithm with non-Markovian selection or as a method of performing simulated annealing without the Metropolis algorithm. Evolutionary annealing introduces a new annealed selection operator, exploiting a connection between the average effect of proportional selection and the annealed Boltzmann distributions used in simulated annealing. Although many genetic algorithms have previously employed the Boltzmann distribution for selection (1; 2; 3), REA is distinct from these approaches in that it can select any member of any prior population. Evolutionary annealing is distantly related to Estimation of Distribution Algorithms (EDAs), since it builds a global model of the annealing distributions for the fitness function (4; 5). However, whereas EDAs build models based solely on the best members of the immediately prior generation, evolutionary annealing maintains a model based on the entire history of observation.

Theoretically, evolutionary annealing converges asymptotically to the true global optima of the fitness function. The proof is given in Section IV-A. Experimentally, evolutionary annealing converges quickly on a bank of standard benchmarks. This paper includes results for six standard benchmark problems in Section V. These benchmarks include multimodal and non-separable problems that are difficult for many optimization methods. However, based on its efficient sampling, evolutionary annealing performs well in a comparison with Differential Evolution (DE), Correlated Matrix Adaption Evolution Strategies (ES), and Particle Swarm Optimization (PSO). In parallel work, another variant of evolutionary annealing has also been shown to perform well against a wider bank of competitors (6).

II. Background

Before introducing evolutionary annealing, some background in simulated annealing is presented, followed by a short discussion of the theory of proportional selection in genetic algorithms that motivates the global selection mechanism of evolutionary annealing.

A. Simulated Annealing

Simulated annealing is a general optimization algorithm that employs properties of statistical mechanics to locate minima of a given fitness function (7; 8). At the core of the simulated annealing algorithm is the Boltzmann distribution. At time n, simulated annealing samples from a distribution given by

$$\mathcal{A}_n^f(dx) = \frac{1}{Z_n} \exp\left(-\frac{f(x)}{T_n}\right) dx, \tag{1}$$

where f is the fitness function, Z_n is a normalizing factor known as the *partition function*, and T_n is a sequence of temperatures with $T_n \to 0$. The sequence T_n is known as the *cooling schedule*. The distribution $\mathcal{A}_n^f$ will be referred to as an *annealing distribution* in this paper. Simulated annealing samples from $\mathcal{A}_n^f$ repeatedly using the Metropolis algorithm (9; 10). The process begins with a proposed solution x. At each time step, a proposal distribution $\mathbb{Q}$ is used to sample x_n. The proposed solution x is replaced with x_n with probability $\exp\left(-\max\left\{0, f(x) - f(x_n)\right\}/T_n\right)$. For

each fixed temperature T_n the algorithm will converge to a sample from $\mathcal{A}_n^f$. As $n \to \infty$, $\mathcal{A}_n^f$ converges in probability to a distribution that samples directly from the optimal points of f (7).

Subject to conditions on the cooling schedule, simulated annealing can be shown to converge asymptotically to the global optima of the fitness function (11; 12). For combinatorial problems, Hajek showed that simulated annealing converges if the cooling schedule is set according to $T_n \propto 1/\log n$ (11). In practice, simulated annealing has been used effectively in several science and engineering problems. However, its sensitivity to the proposal distribution and the cooling schedule means that it is not a good fit for all optimization problems.

Surprisingly, traditional genetic algorithms are connected with simulated annealing through an analysis of the average performance of a genetic algorithm with proportional selection. This connection is exposed by trivial manipulations of a previous result of Mühlenbein and Mahnig (3); the details are discussed next.

B. Expected Proportional Selection

Many genetic algorithms employ *proportional selection*, where individuals in the prior population are selected proportionally to their observed fitness. Much like simulated annealing, proportional selection sharpens the fitness function implicitly with each generation, so that on averaging over population trajectories the selection operator asymptotically places probability one on the optima of the fitness function (3). The following argument for discrete spaces is derived from Mühlenbein and Mahnig; analogues to this result hold in arbitrary measure spaces. For the purposes of this section, the goal is to maximize f, a positive fitness function.

Proportional selection at the n^{th} time step is given by $\mathcal{S}_f^n(x) \propto f(x)N_x^{n-1}$, where $\mathcal{S}_f^n(x)$ is the probability of selecting x at time n, and N_x^n is a random variable indicating the number of copies of the solution x in the population at time n. Taking the expected value over N_x^n,

$$\mathbb{E}\left[\mathcal{S}_f^n(x)\right] \propto f(x)\mathbb{E}\left[N_x^{n-1}\right]. \tag{2}$$

The expected value on the left is also a probability distribution over x and therefore a selection rule, here termed *expected proportional selection*. It is possible to imagine an evolutionary algorithm where each successive population is sampled from just this rule. This algorithm is a one-stage, selection-only genetic algorithm; because expected proportional selection averages over all individuals, no variation is required.

In such an algorithm, if the initial population is selected uniformly at random (assuming this is possible), then $\mathbb{E}\left[N_x^0\right]$ is a constant, so

$$\mathbb{E}\left[\mathcal{S}_f^1(x)\right] \propto f(x). \tag{3}$$

By definition, $\mathbb{E}[\mathcal{S}_f^n(x)] = \mathbb{E}[N_x^n]/K$ where K is the population size, since N_x^n/K is just the proportion of the population taking the value x. Applying this fact to the recursion in Equation 2 yields $\mathbb{E}[S_f^n(x)] \propto f(x)^n$. Thus expected proportional selection sharpens the fitness function. Introducing $g(x) \equiv -\log(f(x))$,

$$\begin{aligned}\mathbb{E}\left[S_f^n(x)\right] &\propto \exp\left(-g(x)\right)^n \\ &= \exp\left(-\frac{1}{n^{-1}}g(x)\right)\end{aligned} \tag{4}$$

Comparing Equation 1 to Equation 4, expected proportional selection is found to have an annealing distribution on $-\log f$ with cooling schedule $T_n = n^{-1}$. Since the logarithm is monotonic, the maxima of f are the minima of g.

Expected proportional selection is not a feasible selection rule, because it requires total knowledge of the fitness function a priori. If such knowledge were possible, there would be no need for evolutionary computation; the optima would already be known. Genetic algorithms with proportional selection can be viewed as an approximation of this selection rule.

Evolutionary annealing exploits the theoretical relationship between simulated annealing and genetic algorithms to create a hybridized algorithm that merges qualities of both algorithms, as is described next.

III. Measure-Theoretic Evolutionary Annealing

This section defines the evolutionary annealing algorithm. The following notation is used throughout. Let $(X, \mathcal{F})$ be an arbitrary measure space, and let λ be a finite measure on $(X, \mathcal{F})$. Let $f : X \to \mathbb{R}$ be a fitness function which is to be minimized, and assume that f has all necessary integrability properties required by the formulae that follow. The notation (P_n) will represent a *stochastic population process*, that is, a sequence of populations generated by an evolutionary algorithm. Each population P_n contains a fixed number of individuals, denoted by $P_n = \left(P_n^k\right)_{k=1}^K$. The set A_n represents the set of all individuals up to time n, $A_n = \bigcup_{m \le n,k} \left\{P_m^k\right\}$. With these definitions, the basic algorithm can be defined.

A. Basic Algorithm

Evolutionary annealing consists of selection and variation phases. The population P_{n+1} is sampled one individual at a time in these two stages. In the selection phase, an element $a \in A_n$ is selected with probability

$$p_n(a) = \frac{\xi_n}{c_n(a)} \exp\left(-\frac{f(a)}{T_n}\right), \tag{5}$$

where T_n is a cooling schedule, ξ_n is a normalizing factor, and $c_n(a)$ is a weight intended to approximate the size of a region of X represented by the individual a, discussed below.

For the variation phase, evolutionary annealing requires a family of probability distributions $\{\nu_n^x\}_{x \in X}$ used to mutate selected points, so that given a selected point x, ν_n^x is used to vary x at time n. The choice of mutation distributions is essentially arbitrary, although the convergence theorems that follow will restrict this choice. In Euclidean space, Gaussians can be used, centered at x and with variation σ_n. In binary spaces, individual bits can be flipped with a probability dependent on n. The particular mutation distributions should be chosen based on the needs of the problem at hand. Some

results for a specific instantiation of evolutionary annealing with real vectors will be discussed in Section V.

Once an individual $a \in A_n$ has been selected with probability $p_n(a)$, then that individual is mutated according to ν_n^a in order to generate a new member of the population. That is, each individual in the population at time $n+1$ is sampled according to

$$P_n^k \sim \sum_{a \in A_n} p_n(a) \nu_n^a(dx). \tag{6}$$

Thus evolutionary annealing samples its populations from a sequence of mixture distributions with one mixing point located at each individual from prior populations. In this way, the selection is non-Markovian; the selected individual could come from any previous generation. The mixture probabilities $p_n(a)$ are chosen according to the annealing formula in Equation 5. Intuitively, if the temperature is fixed at a constant, as the number of mixing points increases and the variance of the mutation distribution decreases, the mixture distribution in Equation 6 converges to the annealing distribution $\mathcal{A}_n^f$ in Equation 1. It is commonly known that mixtures of Gaussians can model any sufficiently smooth distribution arbitrarily well if enough mixing points are used. It is also true that mixture distributions in general can model any probability measure arbitrarily well subject to certain conditions. A specific proof of convergence for evolutionary annealing is offered in Section IV-A; Theorem IV.1 states that evolutionary annealing converges in probability to the optima of f. Therefore P_n is successively sampled from better and better approximations to $\mathcal{A}_n^f$, and as $n \to \infty$, the population sequence P_n will increasingly focus on the optima of f. The rate of convergence will be taken up in Section IV-B.

A high-level algorithm for evolutionary annealing is shown in Algorithm 1. The algorithm articulates the process for sampling populations as previously described. Assuming that sampling ν_n^a and computing $c_n(a)$ do not add to the complexity, this algorithm has performance $O\left(N^2K^2\right)$. By storing the points in a database, it is possible to sample P_n in $O(K \log n)$ time. The most expensive elements computationally are the values of $p_n(a)$, which must be recomputed with each generation.

Evolutionary annealing at this level of abstraction has two parameters in addition to the choice of mutation distribution: the cooling schedule and the region weight $c_n(a)$. In light of (11), a default choice for the cooling schedule is given by $T_n^{-1} = \eta \log n$. Here η is a learning rate that scales the fitness function and thereby controls the aggressiveness of selection. A high learning rate focuses selection on the few best individuals and may restrict exploration of the space. A low learning rate allows promiscuous selection, slowing down refinement of previously discovered solutions but increasing the probability of escaping a local minimum. Again following (11), a possible value for η is $1/d$ where d is the largest depth of a local minima relative to its surroundings in the fitness landscape. In more complex spaces, different cooling schedules could be considered. There may also be a benefit to

Algorithm 1 Algorithm to Approximate a Fitness Measure

```
N, the number of generations
K, sample points (population size) per generation
P_1^k ~ λ/λ(X), the initial population
A_0 ← ∅, all points from all generations
for n ← 1 to N do
  A_n ← ⋃_k P_n^k ∪ A_{n−1}
  ξ_n ← 0
  for m ← 1 to n do
    for k ← 1 to K do
      Compute c_n(P_m^k), T_n
      p_n(P_m^k) ← c_n(P_m^k) exp(−f(P_m^k)/T_n)
      z_n(a) ← ξ_n + p_n(P_m^k)
      ξ_n ← ξ_n + z_n(a)
    end for
  end for
  for k ← 1 to K do
    r ← ξ_n × Uniform(0, 1)
    y ← argmin_{a∈A_n} {z_n(a) : r < z_n(a)}
    P_{n+1}^k ← a sample from ν_n^y
  end for
end for
```

linking the variance of the mutation distribution to the cooling schedule, so that as the probability of selecting the current best individual decreases, the variance also decreases to enable refined exploration of the immediate region around the current best. The effect of parameter settings is explored further in Section IV-B.

The region weight $c_n(a)$ is present in Equation 5 to avoid a particular scenario of premature convergence. Once a good solution is discovered, Evolutionary annealing will devote increasing resources to exploring the neighborhood of that point. If these points are also good, then the probability of selecting more points in the same region will increase in a feedback loop. Within a few generations, almost all points selected will come from the immediate environment of these good points. If there is a local minimum in the vicinity, evolutionary annealing would likely become entrapped in that region. The region weight $c_n(a)$ is intended to serve as a measure of how many individuals have been previously sampled in the region surrounding the point a. If the search space X is a metric space, then $c_n(a)$ can simply count the number of other points in A_n contained within a small radius of a. For theoretical reasons relating to convergence, the radius used should decrease to zero as $n \to \infty$. For vector spaces such as $\mathbb{R}^d$, hypercubes are more efficient to compute. By dividing out the region weight in Equation 5, selection remains proportional to fitness rather than becoming inordinately sensitive to the particular sequence of sampled populations. The region weight also plays an important role in the proof of global convergence.

In non-metric spaces it remains to be seen how $c_n(a)$ should be set. It is also questionable whether a counting im-

plementation can be efficiently computed, especially in high-dimensional spaces. For this purpose, a partitioning approach for setting $c_n(a)$ is described next that can be computed efficiently in many spaces. This partitioning approach is also an important tool for the convergence proof in Section IV-A.

B. Partitioning the Space

To demonstrate convergence of evolutionary annealing, each of the mixing points $a \in A_n$ will be considered representative of a particular region of the search space X. Each successive set A_n will be associated with a partition $\{E_n^a\}_{a \in A}$ of disjoint sets such that $X = \bigcup_{a \in A_n} E_n^a$. The σ-algebra $\mathcal{F}$ is assumed to be rich enough to support such partitions based on any collection of finite points in X. With such a partition, the proper value for the region weight is $c_n(a) = \lambda\left(E_n^a\right)^{-1}$.

The justification for setting $c_n(a)$ to count the number of other points nearby as advocated in the previous paragraphs is as follows. If a point a has a large number of nearby points at time n, then the size of partitioned region E_n^a must be small, or else it would also contain other points. If the measure λ spreads weight evenly across the space, and if the partition region E_n^a is small, its measure, $\lambda(E_n^a)$, will be commensurately small. As a consequence, the counting approximation should only be used in spaces where the measure λ supports this analysis; elsewhere, a partitioning approach should be preferred. It is also important to note that while the counting approach correctly decreases probabilities that are too high, it nonetheless fails to increase those probabilities that are too low, since an isolated point will have a count of 1 regardless of how far it is from other points. So the counting approach is biased in favor of known good points.

Algorithm 2 partitions any measure space given a function for dividing a partition region between two separate points in the region. A partition is represented as a binary tree, with the root representing the entire space X and each branch partitioning X into two sets. The algorithm is initialized with a sequence of points $\{x_m\}_{m=0}^M \subseteq X$ to be partitioned (the mixing points), a tree $\mathcal{T}$ with X as the root node, and an assignment function k such that $k(m)$ is the leaf node of the tree assigned to the point x_m, or $\emptyset$ if no assignment has been made. The algorithm then loops through the mixing points, splitting the space where necessary to ensure that each leaf node contains exactly one mixing point. The algorithm relies on a domain-specific subroutine, $separate$. At the end of each iteration of the algorithm's main loop, each leaf node is assigned to exactly one mixing point. When a new mixing point is added, $separate$ partitions the leaf node to which it belongs into two new leaf nodes, each containing only one mixing point. The process of adding a single new mixing point to the tree requires only a tree traversal, so that at each generation, updating the partition requires $O(K \log NK)$ time, where NK is the number of points at the N^{th} generation. The counting approach, by contrast, requires worst case $O\left(N^2K^2\right)$ time, and therefore partitioning is potentially much less expensive computationally.

Algorithm 2 Algorithm to Generate a Partition Based On Grid Points

```
{x_m}_{m=1}^M ⊆ X, the mixing points
T ← {X}, the partition tree
k(i) ← ∅ for all i = 1, ..., M, node assignment function
for m ← 1 to M do
  N ← the leaf node in T such that x_m ∈ N
  if ∃j ≠ m s.t. k(j) = N then
    N_0, N_1 ← separate(x_j, x_m, N)
    T ← T ∪ {N_0, N_1}
    k(j) ← N_0, k(m) ← N_1
  else
    k(m) ← N
  end if
end for
```

In a vector space, such as $\mathbb{R}^d$, the function $separate$ can in many cases be given explicitly. Suppose that X is bounded above by $\{u_i\}$ and below by $\{\ell_i\}$ so that X has a rectangular shape. Each node in the partition tree will restrict the coefficient for exactly one of the basis vectors, say j. To maintain computability, it is necessary to require that $j < M < \infty$ for some M. That is, each set E_n^a in the partition is defined as a hyperrectangle on finitely many coordinates, with each step in the traversal of the partitioning tree adding a new coordinate value for some side of the hyperrectangle. So E_n^a can be represented as two vectors, u^a for the upper bounds, and ℓ^a for the lower bounds. Given the point $a \in X$ and a second point $x \in X$, E_n^a can be separated as follows. Let $k = \operatorname{argmax}_{i \leq M} |a_i - x_i|$; k is the index at which the rectangle E_n^a will be split. Suppose $a_k > x_k$ for the sake of simplicity; the opposite situation is handled analogously. Initialize $u^x \leftarrow u^a$ and $\ell^x \leftarrow \ell^a$. Then set $\ell_k^a \leftarrow \frac{1}{2}(a_k - x_k) + x_k$ and $u_k^x \leftarrow \frac{1}{2}(a_k - x_k) + x_k$. The regions E_{n+1}^a and E_{n+1}^x defined by these boundary vectors are then disjoint if the upper boundary is strict. Two vectors that are the same in the first M coefficients cannot be distinguished by this process, but that situation can usually be ignored. This separation algorithm can be efficiently implemented in many spaces of interest. The next section discusses the convergence properties of the algorithm.

IV. Convergence Properties

Subject to a reasonable set of conditions, evolutionary annealing with either proportional or tournament selection converges in probability to the set of optimal points for the fitness function. These conditions include: (1) the mutation variance must asymptotically decrease; (2) the annealing distributions must possess quasi-differentiability properties at the mixing points; (3) the fitness function must not be too irregular in the immediate neighborhood of the optima; and (4) the mutation variance and the temperature must decay slowly enough to guarantee full exploration of the space. With these conditions satisfied, evolutionary annealing converges to the optima. The convergence rate for evolutionary annealing is

highly sensitive to both the cooling schedule and the variance decay; the interaction of these parameters remains the subject of inquiry.

In this section, the preceding concepts are made rigorous, and a proof of convergence for evolutionary annealing is provided, followed by a discussion of convergence rates.

A. Convergence Proof

As mentioned above, the convergence proof requires conditions on the mutation variance, the annealing distributions, and the fitness function. First, convergence requires that the mutation distributions ν_n^a increasingly focus on the point a.

Definition 1. *A sequence of probability measures $\{\nu_n^a\}$ is said to be increasingly centered on a if for any decreasing sequence of sets $\{F_n\}$ such that $\bigcap_n F_n = \{a\}$, $\nu_n^a(F_n) \to 1$.*

This definition is satisfied if the distributions ν_n^a have mean a and variance decreasing to zero. Intuitively, mutation must cease asymptotically. Also, the mutation distributions must be well matched with the base measure λ in the sense that sets of λ-measure zero must also have ν_n^a-measure zero. This property is known as *absolute continuity* of ν_n^a with respect to λ.

The convergence proof makes use of the annealing distributions with respect to the measure λ. Specifically, define

$$g_n(x) = \frac{\exp\left(-f(x)/T_n\right)}{\int_X \exp\left(-f(x)/T_n\right)\ \lambda\left(dx\right)}. \tag{7}$$

and note that g_n is the density of an annealing distribution generalized to the space $(X, \mathcal{F}, \lambda)$, i.e. $\int_X g_n\, d\lambda = 1$. Set $g \equiv \lim_n g_n$. Now since $0 < g_n \leq 1$, it holds that $0 \leq g \leq 1$. Also, since g_n is bounded, $\int_X g\, d\lambda = \lim_n \int_X g_n\, d\lambda = 1$, so that g is also the λ-density of a probability distribution. Specifically, g assigns measure zero to all non-optimal points of f.

In order to guarantee that the mixture distributions used by evolutionary annealing are capable of approximating g_n, it is necessary that the densities g_n do not vary too quickly, i.e., that the fitness function does not oscillate wildly between infinitesimally close points. Formally, this concept can be defined based on the integrals of g_n on *nicely shrinking* sets. Nicely shrinking sets are a vanishing sequence of sets, each of which possesses interior points.

Definition 2. *Given a sequence of sets $\{E_n^a\}_{n\in\mathbb{N}}$ that shrink nicely around a point a and a sequence of functions $\{g_n\}_{n\in\mathbb{N}}$ on a measure space $(X, \mathcal{F}, \lambda)$ such that each g_n is λ-integrable, the neighborhood average of g_n on E_n^a is given by*

$$g_n^\lambda(a) \equiv \lambda\left(E_n^a\right)^{-1} \int_{E_n^a} g_n d\lambda. \tag{8}$$

Definition 3. *On a measure space $(X, \mathcal{F}, \lambda)$, a sequence of λ-integrable functions g_n is approximated by its neighborhood average at a point a if for any sequence of nicely shrinking sets $\{E_n^a\}$*

$$\limsup_n \left|g_n(a) - g_n^\lambda(a)\right| = 0. \tag{9}$$

If the neighborhood average g_n^λ of a sequence g_n approximates the values of the sequence at a point well, then the neighborhood average can be used as a proxy for the function at that point. Approximation by the neighborhood average is a critical requirement for the convergence of evolutionary annealing, but is not overly restrictive in practical terms. This property is possessed by all continuous functions, but it is true for many discontinuous functions as well. In fact, only fitness functions that are chaotic at an infinitesimal scale are excluded by this requirement.

The next set of conditions pertains to the ϵ-optimal sets of f. Let f^* be the minimal value of f, and define $X_\epsilon \equiv \{x : f(x) < f^* + \epsilon\}$. X_ϵ includes all points in X that come within ϵ of the optimum. If the set X_ϵ has λ-measure zero for small values of ϵ, then the optima are isolated, and the mutation distributions for evolutionary annealing have zero probability of proposing the optima. In that situation, convergence is impossible.

A second pathological situation occurs when the boundary of the set X_ϵ is so jagged that it possesses positive λ-mass. In this case, the boundaries of X_ϵ can never be well approximated by a countable sequence of estimates.

A fitness function will be called *suitable* when these cases can be excluded. Additionally, suitability will be defined to account for the required integrability and neighborhood properties discussed above.

Definition 4. *A fitness function f is termed suitable whenever the following five conditions hold:*

1) *The minimum exists, i.e. $f^* > -\infty$.*
2) *The functions g_n are λ-integrable.*
3) *The functions g_n are well approximated by their neighborhood average.*
4) *The sets X_ϵ are $\mathcal{F}$-measurable, $\epsilon \geq 0$.*
5) *There exists a constant $\gamma > 0$ such that for all $\epsilon \in (0, \gamma)$, $\lambda(X_\epsilon) > 0$ and $\lambda(\partial X_\epsilon) = 0$.*

Finally, the cooling schedule and mutation variance must decay slowly enough to guarantee full exploration of the search space, or else the global optimum might be missed. Note that this requirement pertains to the shape of the mutation distributions and not just the variance. Specifically, let $E \subseteq X$ be an open region of the search space with positive λ-measure, and let E^c be its complement in X. Recall that A_n represents the set of all previously observed individuals at time n. Ultimately, to fully explore the space, there must be an n such that $A_n \cap E \neq \emptyset$ for each open set E. As a technical detail, X must be separable in order for this to be possible. Then X has a countable dense subset. If the space is fully explored, then $A_n \uparrow A$ where A is such a countable dense subset of X. An evolutionary annealing algorithm that meets satisfies this criterion will be termed *complete*.

The convergence theorem can now be stated. For convergence, evolutionary annealing is assumed to use the partitioning method to set the region weights $c_n(a)$. The proof makes liberal use of the partitions $\{E_n^a\}$.

Theorem IV.1. *A complete evolutionary annealing algorithm converges in probability to the minimal points of any suitable fitness function provided that the mutation distributions are increasingly centered on the mixing points.*

Proof: Fix $\epsilon, \delta > 0$ with $\epsilon < \gamma$. Without loss of generality, assume $\lambda(X) = 1$; if not, $\tilde{\lambda} \equiv \lambda/\lambda(X)$ will satisfy this equality. Let $\mathbb{G}_n(B) = \sum_{a \in A_n} p_n(a) \nu_n^a(B)$ be the distribution generating evolutionary annealing at time n. The desired result will follow if there exists an N such that for $n \geq N$, $\mathbb{G}_n(X_\epsilon) > 1 - \delta$.

Define the annealing distributions $\mathcal{A}_n(B) = \int_B g_n \, d\lambda$, and similarly $\mathcal{A}(B) = \int_B g \, d\lambda$. Now because $\mathcal{A}(X_\epsilon) = 1$ for all ϵ, it is sufficient to prove that $|\mathbb{G}_n(X_\epsilon) - \mathcal{A}(X_\epsilon)| < \delta$.

For convenience, let $\lambda_n^a(X_\epsilon) \equiv \lambda(X_\epsilon \cap E_n^a)/\lambda(E_n^a)$. Define $\tilde{\mathbb{G}}_n$ so that $\tilde{\mathbb{G}}_n(X_\epsilon) = \sum_{a \in A_n} p_n(a) \lambda_n^a(X_\epsilon)$. Since ν_n^a is increasingly centered on a, for n sufficiently large, $\nu_n^a(X_\epsilon \setminus E_n^a) < \delta/4$.

Also, because the algorithm is complete, there exists n large enough so that either $\mathring{X}_\epsilon \cap E_n^a = E_n^a$ or $\mathring{X}_\epsilon \cap E_n^a = \emptyset$ where as usual $\mathring{X}_\epsilon \equiv X_\epsilon \setminus \partial X_\epsilon$. Since $\epsilon < \gamma$, the measure of the boundary of X_ϵ can be ignored, and either $\lambda_n^a(X_\epsilon) = 0$ or $\lambda_n^a(X_\epsilon) = 1$. Similarly, $\nu_n^a(X_\epsilon \cap E_n^a)$ can be chosen to be within $\delta/4$ of either 0 or 1, since ν_n^a is increasingly centered on a and either $\nu_n^a(X_\epsilon \cap E_n^a) = \nu_n^a(E_n^a)$ or $\nu_n^a(X_\epsilon \cap E_n^a) = 0$, depending on whether $a \in \mathring{X}_\epsilon$. Therefore,

$$\begin{aligned}\left|\mathbb{G}_n(X_\epsilon) - \tilde{\mathbb{G}}_n(X_\epsilon)\right| &\leq \sum_{a \in A_n} p_n(a) \left|\nu_n^a(X_\epsilon) - \lambda_n^a(X_\epsilon)\right| \\ &\leq \sum_{a \in A_n} p_n(a) \nu_n^a(X_\epsilon \setminus E_n^a) \\ &\quad + \sum_{a \in A_n} p_n(a) \left|\nu_n^a(X_\epsilon \cap E_n^a) - \lambda_n^a(X_\epsilon)\right| \\ &< \frac{\delta}{4} + \frac{\delta}{4} = \frac{\delta}{2}. \end{aligned} \tag{10}$$

Thus $|\mathbb{G}_n(X_\epsilon) - \tilde{\mathbb{G}}_n(X_\epsilon)| < \frac{\delta}{2}$.

Next it will be shown that $|\tilde{\mathbb{G}}_n(X_\epsilon) - \mathcal{A}(X_\epsilon)| \to 0$ by using a martingale argument. Let $Z \equiv g$ be a random variable on $(X, \mathcal{F}, \lambda)$ and note that $\mathbb{E}(Z 1_{X_\epsilon}) = \mathcal{A}(X_\epsilon)$. Let $Y_n \equiv \sum_{a \in A_n} 1_{E_n^a} g_n(a)$ be a random process on the same space, so that $\mathbb{E}(Y_n 1_{X_\epsilon}) \approx \tilde{\mathbb{G}}_n(X_\epsilon)$ since $|p_n(a) - g_n(a) \lambda(E_n^a)| \to 0$ (recalling that $c_n(a) = \lambda(E_n^a)^{-1}$).

Let $\{\mathcal{E}_n^{A_n}\}$ be the filtration generated by the sequence of partitions $\{E_n^a\}$. Now consider the processes generated by conditioning on $\{\mathcal{E}_n^{A_n}\}$:

$$\begin{aligned}\tilde{Y}_n^\epsilon &= \mathbb{E}\left(Y_n 1_{X_\epsilon} \mid \mathcal{E}_n^{A_n}\right) \\ &= \sum_{a \in A_n} 1_{E_n^a} S_n(a) \lambda(X_\epsilon \cap E_n^a), \end{aligned} \tag{11}$$

$$\begin{aligned}Z_n^\epsilon &= \mathbb{E}\left(Z 1_{X_\epsilon} \mid \mathcal{E}_n^{A_n}\right) \\ &= \sum_{a \in A_n} 1_{E_n^a} \int_{X_\epsilon \cap E_n^a} g \, d\lambda. \end{aligned} \tag{12}$$

Note that $\mathbb{E}(\tilde{Y}_n^\epsilon) = \mathbb{E}(Y_n 1_{X_\epsilon})$ and $\mathbb{E}(Z_n^\epsilon) = \mathbb{E}(Z 1_{X_\epsilon})$ by the properties of conditional expectations.

It is also the case that $\tilde{Y}_n^\epsilon$ and Z_n^ϵ converge together because for $\eta > 0$

$$\begin{aligned}\left|Z_n^\epsilon - \tilde{Y}_n^\epsilon\right| &\leq \sum_{a \in A_n} 1_{E_n^a} \left|\int_{X_\epsilon \cap E_n^a} g \, d\lambda - g_n(a) \lambda(X_\epsilon \cap E_n^a)\right| \\ &= \sum_{a \in A_n} 1_{E_n^a} \left|\int_{X_\epsilon \cap E_n^a} [g(x) - g_n(a)] \lambda(dx)\right| \\ &\leq \sum_{a \in A_n} 1_{E_n^a} \left[\sup_{x \in E_n^a} \left|g_n^\lambda(x) - g_n(a)\right|\right] \lambda(E_n^a) \\ &\quad + \sum_{a \in A_n} 1_{E_n^a} \left|\int_{X_\epsilon \cap E_n^a} \left[g(x) - g_n^\lambda(x)\right] \lambda(dx)\right| \\ &< \sum_{a \in A_n} 1_{E_n^a} \left[\frac{\eta}{2} + \frac{\eta}{2}\right] = \eta. \end{aligned} \tag{13}$$

The inequalities hold because g_n is approximated by its neighborhood average at a, $X_\epsilon \cap E_n^a$ shrinks nicely, and $g_n \to g$ (and thus $g_n^\lambda \to g$).

Because Z_n^ϵ is a Levy martingale, $Z_n^\epsilon \to \mathbb{E}\left(Z 1_{X_\epsilon} \mid \mathcal{E}_\infty^A\right)$, and the prior result then implies that $\tilde{Y}_n^\epsilon \to \mathbb{E}\left(Z 1_{X_\epsilon} \mid \mathcal{E}_\infty^A\right)$. By the properties of Levy martingales, it then holds that $\mathbb{E}\tilde{Y}_n^\epsilon \to \mathbb{E}\left(\mathbb{E}\left(Z 1_{X_\epsilon} \mid \mathcal{E}_\infty^A\right)\right) = \mathbb{E}(Z 1_{X_\epsilon})$. That is,

$$\tilde{\mathbb{G}}_n(X_\epsilon) \approx \mathbb{E}(Y_n 1_{X_\epsilon}) = \mathbb{E}(\tilde{Y}_n^{X_\epsilon}) \to \mathbb{E}(Z 1_{X_\epsilon}) = \mathcal{A}(X_\epsilon). \tag{14}$$

Putting it together, for n sufficiently large,

$$\begin{aligned}|\mathbb{G}_n(X_\epsilon) - \mathcal{A}(X_\epsilon)| &\leq \left|\mathbb{G}_n(X_\epsilon) - \tilde{\mathbb{G}}(X_\epsilon)\right| \\ &\quad + \left|\tilde{\mathbb{G}}_n(X_\epsilon) - \mathcal{A}(X_\epsilon)\right| \\ &< \frac{\delta}{2} + \frac{\delta}{2} = \delta, \end{aligned} \tag{15}$$

completing the proof. ∎

In sum, this theorem shows that evolutionary annealing is guaranteed to converge asymptotically arbitrarily close to the minima of the fitness function. The result is stronger than known results for simulated annealing and is generally independent of the learning parameters. However, less can be said regarding the rate of convergence, as will be discussed next.

B. Convergence Rates

An examination of the proof of Theorem IV.1 shows that there are three basic sources of approximation error: (1) the variance of the mutation distribution, (2) the accuracy of the neighborhood average, and (3) the speed of convergence for the annealing distributions, due to the cooling schedule. Of these, the variance and the cooling schedule are under the direct control of the practitioner. Implicitly, these two factors also control the accuracy of the neighborhood average. In order to set the cooling schedule and variance decay to maximize the rate of convergence, the effects of these three error sources must be carefully considered.

The first source of error is due to the difference $|\nu_n^a(X_\epsilon) - \lambda_n^a(X_\epsilon)|$. Convergence occurs because both of these measures ν_n^a and λ_n^a asymptotically become point masses, the former because of the variance decay and the latter because the mixing points eventually fill the search space. To minimize error, these two measures should be kept as close as possible for mixing points in the vicinity of the optima. As the algorithm begins to focus on a small group of optima, the partitions in that region will become smaller, and the variance of the mutation distribution should decrease at a similar rate. Notably, however, decreasing the variance also reduces the probability that the global optimum will be discovered if it has not already and if it is located sufficiently far from the current regions of focus. Also, when a new and better optimum is discovered after the search has already focused on other local optima, the exploration of the new optimum will proceed slowly if the variance has already decayed substantially. Therefore it may make sense to scale the variance to reflect the size of the partition region for the mixing point being mutated. In this way, larger variances will typically be employed in unexplored regions, whereas well explored regions will use a narrower variance, so that ν_n^a and λ_n^a are well-matched in general. The fact that the mixing points eventually fill the space guarantees that a dynamic and locally scaled variance decay schedule of this type will eventually become increasingly centered as required.

The second source of error pertains to the accuracy of approximating the average value of the annealing distribution by its value at the mixing points, due to the term $\left|g_n^\lambda(x) - g_n(a)\right|$. This error depends strongly on the fitness function. If the fitness function is relatively homogeneous and does not fluctuate at different rates in different regions of the search space, then it may be seen that this source of error strongly reflects the mass of the partition region, $\lambda(E_n^a)$. In a large region, the approximation $g_n(a)\lambda(E_n^a)$ is likely to differ more substantially from $\int_{E_n^a} g_n \, d\lambda$ than it would in a smaller region. Thus this source of error can perhaps be reduced by spreading the mixing points more evenly through the space in order to keep the partition regions uniformly small. This goal can be accomplished by using a high variance at the outset to guarantee full exploration of the space. At later stages, the use of a high variance is at odds with the need to match the mutation distribution ν_n^a with the measure λ_n^a, and thus the first two sources of error must be balanced and cannot be mutually eliminated. The accuracy of the average approximation is also affected by the cooling schedule. At high temperature, the approximation must be more accurate because g_n will vary less over the region E_n^a if T_n is large. Lowering the temperature increases this source of error by causing the function g_n to fluctuate more.

The third source of error concerns the speed of convergence of the annealing distributions due to the difference $|g - g_n|$. The faster the cooling schedule takes the temperature to zero, the faster this error will be minimized. But a fast cooling schedule will increase the error due to the neighborhood approximation. The speed of the cooling schedule must balance the need to minimize both the neighborhood approximation error and the annealing convergence error.

Overall, preliminary experiments showed that a logarithmic cooling schedule, e.g. $T_n^{-1} = \eta \log n$, works well in practice. Early results also suggest that the variance should start off quite large and decay exponentially fast. Also, it should be an effective approach to scale the variance locally based on the size of the partition region for the mixing point being mutated. The next section discusses experimental results.

V. Experimental Validation

Evolutionary annealing can be used to search for bit strings, real vectors, neural networks, Bayesian network structures, game strategies, programs, state machines, and any other structure that can be embedded within a measure space. Parallel to the theory development in this paper, specific implementations were defined and tested for real vectors. This paper displays results using the partitioning method for setting the region weight described in Section III-B. For results with the counting method in Section III-A against a larger bank of competing algorithms, see (6).

In Euclidean space, an instantiation of evolutionary annealing, named Real-space Evolutionary Annealing (REA), was tested on six common benchmarks: sphere, rosenbrock, ackley, whitley, modified shekel's foxholes, and griewank, all in five dimensions. REA employed a logarithmic cooling schedule, Gaussian variation, and an exponentially decaying variance set at an initially high value. The variance decay rate and the problem learning rates were set as described in parallel work (6); the only difference here is that the partitioning method was used for the region weights. For comparison, DE, ES, and PSO were also run on the same problems. All algorithms were run on 20 trials for 1000 generations with a population size of 100.

The number of successful trials for all algorithms is shown in Table V. REA was able to successfully optimize in substantially many trials on all benchmarks. On shekel's foxholes, REA was the most successful of all the algorithms tested, solving the problem on 7 out of 20 trials. On the other benchmarks, REA was successful more often than either ES or PSO. DE outperformed REA on ackley and whitley. However, REA is designed to be easily extensible to discontinuous functions and arbitrary measure spaces, whereas DE is not. Table V shows the number of function evaluations to attain success. REA converges faster than DE on most problems. Compared to ES, REA evaluates the fitness function a comparable number of times, but with better success rates.

Once good solutions were obtained, REA's ability to refine those solutions was less impressive, though still effective. The difficulty of refinement is a result of the exponentiation contained in the Boltzmann distribution. Under the exponent, once solutions are within 0.01 of the optimum, all solutions of this quality are likely to be selected roughly in equality even at low temperatures. This defect could be addressed by developing an annealing approach to tournament selection,

TABLE I
NUMBER OF SUCCESSFUL TRIALS (OUT OF 20, $\epsilon < 0.2$)

	sphere	rosenbrock	ackley	whitley	shekel	griewank
REA	20	20	13	16	7	20
DE	20	20	20	20	5	20
ES	20	20	11	10	1	20
PSO	20	4	3	6	1	2

TABLE II
FUNCTION EVALUATIONS BEFORE SUCCESS (IN THOUSANDS, $\epsilon < 0.2$)

	sphere	rosenbrock	ackley	whitley	shekel	griewank
REA	1	25	26	27	70	49
DE	2	17	62	63	80	9
ES	2	41	17	39	32	24
PSO	2	69	14	79	18	34

discussed in Future Work. See (6) for further discussion of experimental results.

VI. FUTURE WORK

Future experimentation evolutionary annealing will focus on training complex structures such as neural networks and game strategies. The purpose of defining evolutionary annealing at the level of abstraction in this paper is to provide a means for developing new algorithms to search in high-level spaces without having to reinvent the underlying evolutionary apparatus from whole cloth. Evolutionary annealing provides convergence guarantees as well as heuristics for setting learning parameters for a wide variety of search domains.

More work remains to be done in establishing the rate of convergence for evolutionary annealing beyond the heuristics provided in Section IV-A. For example, it is known that maximum likelihood estimates of mixture distributions with increasing mixing points approximate continuous distributions at a relatively fast rate of $C\left(\frac{\log n}{n}\right)^{0.25}$ (13). The distributions employed in evolutionary annealing are not the same, but similar performance may be hoped for on continuous fitness functions.

Finally, basic improvements can be made to the algorithm implementation. It may be possible to reduce the computational complexity from $O(N^2K^2)$ to $O(NK \log NK)$ through use of a tree-sampling algorithm rather than explicit updates of $p_n(a)$ from Equation 5. Additionally, it may be possible to develop a non-Markovian analogue to tournament selection that works in the context of annealed selection, protecting against the insensitivity of proportional selection to minor fluctuation in fitness near the optima.

VII. CONCLUSION

Evolutionary annealing leverages shared aspects of simulated annealing and genetic algorithms in order to produce a hybridized evolutionary algorithm that is provably convergent and amenable to mathematical analysis. This paper presented a convergence proof that enables a heuristic analysis of the convergence rates of the proposed algorithm. In parallel work, evolutionary annealing has been implemented for real vectors and shown to be effective at locating optima, and even more so at finding rough approximations to the global optima with few function evaluations. Evolutionary annealing and algorithms derived from it are therefore an exciting new prospect for experimental and theoretical research in evolutionary algorithms.

ACKNOWLEDGMENT

The authors would like to thank Gordan Žitkovič and Jesse Windle for their valuable advice in developing the proof contained in this paper. This research was supported in part by NSF under grants DBI-0939454 and IIS-0915038.

REFERENCES

[1] D. E. Goldberg, "A note on boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing," *Complex Systems*, vol. 4, 1995.

[2] I. Jeong and J. Lee, "Adaptive simulated annealing genetic algorithm for system identification," *Engineering Applications of Artificial Intelligence*, vol. 9, no. 5, pp. 523 – 532, 1996.

[3] H. Mühlenbein and T. Mahnig, "Mathematical analysis of evolutionary algorithms," in *Essays and Surveys in Metaheuristics, Operations Research/Computer Science Interface Series*. Kluwer Academic Publisher, 2002, pp. 525–556.

[4] M. Pelikan, D. Goldberg, and F. Lobo, "A survey of optimization by building and using probabilistic models," *Computational Optimization and Applications*, vol. 21, 2002.

[5] H. Mühlenbein, T. Mahnig, and A. O. Rodriguez, "Schemata, distributions, and graphical models in evolutionary optimization," *Journal of Heuristics*, vol. 5, 1999.

[6] A. Lockett and R. Miikkulainen, "Real-space evolutionary annealing," in *Proceedings of the 2011 Genetic and Evolutionary Computation Conference (GECCO-2011)*, 2011.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, 1983.

[8] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Statistical Science*, vol. 8, no. 1, 1993.

[9] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equations of state calculations by fast computing machines," *Journal of Chemical Physics*, vol. 21, no. 6, 1953.

[10] W. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, no. 1, 1970.

[11] B. Hajek, "Cooling schedules for optimal annealing," *Mathematics of Operation Research*, vol. 13, no. 4, 1988.

[12] R. L. Yang, "Convergence of the simulated annealing algorithm for continuous global optimization," *Journal of Optimization Theory and Applications*, vol. 104, no. 3, 2000.

[13] C. Genovese and L. Wasserman, "Rates of convergence for the gaussian mixture sieve," *Annals of Statistics*, vol. 28, no. 4, 2000.