A Theoretical Comparison of Evolutionary Algorithms and Simulated Annealing*

William E. Hart
Applied and Numerical Mathematics Department
Sandia National Labs
P. O. Box 5800 - MS 1110
Albuquerque, NM 87185-1110
wehart@cs.sandia.gov

August 28, 1995

Abstract

This paper theoretically compares the performance of simulated annealing and evolutionary algorithms. Our main result is that under mild conditions a wide variety of evolutionary algorithms can be shown to have greater performance than simulated annealing after a sufficiently large number of function evaluations. This class of EAs includes variants of evolutionary strategie and evolutionary programming, the canonical genetic algorithm, as well as a variety of genetic algorithms that have been applied to combinatorial optimization problems. The proof of this result is based on a performance analysis of a very general class of stochastic optimization algorithms, which has implications for the performance of a variety of other optimization algorithms.

1 Introduction

This paper concerns the performance of algorithms that minimize an objective function of the form $f : S \rightarrow \mathbf{R}^n$, $|S| < \infty$. In particular, this paper concerns the relative performance of evolutionary algorithms (EAs) and simulated annealing (SA).

SA and EAs are optimization methods that have been successfully applied to solve combinatorial optimization problems [4, 9, 14, 23]. SA is inspired by an analogy with statistical cooling in metallurgy, where the annealing phase is composed of a heating phase followed by a phase of slow cooling. The resulting metal remains in a low-energy structure. The stochastic process used by SA is similarly controlled by a temperature

*This work was supported by the Applied Mathematical Sciences program, U.S. Department of Energy, Office of Energy Research, and was performed at Sandia National Laboratories, operated for the U.S. Department of Energy under contract No. DE-AC04-76DP00789.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

that is slowly cooled to guarantee that the final solution is at a near optimal solution. EAs are inspired by computational models of natural evolution, in which a population of individuals face selection pressure that eliminates individuals with low fitness from the population in successive generations. In the context of global optimization, the fitnesses of individuals is associated with the optimality of their solution in the search space. Thus EAs use selective pressure among a population of solutions to identify near optimal solutions.

Previously, comparisons between SA and EAs have been primarily confined to empirical evaluations on test problems (e.g. [3, 12, 22]). The relative performance of SA and EAs in these results is mixed, which suggests that SA and EAs may be relatively superior on different classes of problems. Theoretical investigations into SA and EAs have focused on the development of hybrid algorithms that employ techniques developed for both SA and EAs [18, 5]. These authors argue that the strengths of these two classes of algorithms can be combined. However, their arguments to not provide a basis for comparing the performance of SA and EAs themselves.

In this paper we theoretically analyze the relative performance of SA and EAs by comparing their probability of finding an optimal (or near optimal) solution after a given number of samples from a finite search space. In practice, the number of samples is directly related to the cost of these algorithms because each sample corresponds to a function evaluation and function evaluations for previously sampled solutions are not typically stored for future use.

Our analysis of the performance of SAs relies on a recent result by Ferreira and Žerovnik [7] that shows that under reasonable conditions the probability that SA finds an optimal solution is bounded away from one by a slowly decreasing function. A wide variety of EAs can be shown to have no such bound. Consequently, these EAs can be shown to have better performance than SAs after a sufficiently large number of iterations.

Our analysis of the performance of EAs relies on an analysis of a more general class of stochastic optimization algorithms that we call Markovian search algorithms. These algorithms use a Markov chain (with appropriate restrictions) to sample the search space and simply keeping track of the best solution seen. We show that Markovian search algorithms have better performance than SA and then describe how this result can be refined to account for the different costs associated with the Markov chain used to describe EAs.

This paper is organized as follows. Section 2 provides some background on EAs and SA. Section 3 reviews the bounds on SA proven by Ferreira and Žerovnik [7]. Section 4 defines and analyzes the performance of Markovian search algorithms. Section 5 refines this previous result to show that certain classes of EAs have better performance than SA for sufficiently large n. Finally, we discuss some limitations of these results and point to some natural extensions of this work.

2 Background

2.1 A Canonical Evolutionary Algorithm

The *population* used by an EA consists of a N-tuple of solutions $x_i \in S$. Each solution x_i, called an *individual* of the population, is a feasible solution to the problem. The value $f(x_i)$ is said to be the *fitness* of the solution. Let $\{x_1^n, \ldots, x_N^n\}$ be the N individuals in a population at iteration n.

Figure 1 describes the steps of a canonical EA. While there are a number of details that distinguish EAs, this canonical EA captures the principle features of these algorithms. The exemplars of EAs are evolutionary programming (EP), evolutionary strategie (ES) and genetic algorithms (GAs). The principle feature distinguishing these EAs is the selection of operators used perform the evolutionary search as well as the search domains to which these EAs are typically applied. For detailed descriptions of these algorithms, see Fogel [8], Bäck and Schwefel [15], Goldberg [9] and Davis [4].

```
Select an initial population $\bar{x}^0 = \{x_1^0, \ldots, x_N^0\}$, $x_i^0 \in S$
Determine the values $f(x_i^0)$, $x_i^0 \in \bar{x}^0$
Repeat $n = 1, 2, \ldots$
    Perform selection
    Perform crossover with probability $\chi$
    Perform mutation with probability $\mu$
    Determine the values $f(x_i^n)$, $x_i^n \in \bar{x}^n$
    $x_n^* = \arg\min_{i=1,\ldots,N} f(x_i^n)$ and $y_n^* = f(x_n^*)$
    Perform replacement
Until some stopping criterion is satisfied
```

Figure 1: **Pseudo Code for Canonical Evolutionary Algorithms**

With high probability, the selection algorithm chooses the solutions from the current population with the lowest function values. Typically, the solutions with the highest function values are chosen with a low, non-zero probability. Consequently, an EA's search is concentrated in regions with low function values, though regions with high function values are still searched with low probability. Goldberg and Deb [10] review a variety of selection algorithms commonly used in GAs.

Crossover and mutation are evolutionary operators that are used to construct new solutions from the individuals chosen by the selection algorithm. The precise character of these operators is not necessary for purposes of the present analysis. Crossover is usually a binary operation that takes two individuals and generates one or two individuals. Mutation is usually a unary operation that takes one individual and generates another individual. For more details concerning these operators, see the references for the EAs mentioned above.

The replacement algorithm determines which members of the previous population are replaced by the individuals generated by the genetic operators. Often, the entire population or a randomly selected subset of the population is replaced by the new individuals. In *elitist* EAs, the replacement algorithm insures that the best individual in the previous population is kept in the current population if it is better than all of the newly constructed individuals.

2.2 Simulated Annealing

We assume that SA uses a neighborhood operator $N(s)$ to identify new solutions in S. $N(s)$ uses s to generate a solution in S according to a fixed distribution. Let $\mathcal{N}(s)$ be the set of possible neighbors generated by s, and let $\pi_s^N(s')$ be the probability that $N(s)$ generates s'.

Figure 2 describes the basic structure of SA. Given a temperature T, SA accepts new solutions according to the Boltzmann criterion. Let s' be a new solution generated from s using the neighborhood operator. If $f(s') \leq f(s)$ then s' is accepted. Otherwise, s' is accepted with probability

$$P(T, s, s') = e^{(f(s)-f(s'))/T}.$$

Consequently, the transition matrix $(p_{s,s'}(T))$, for all $s, s' \in S$, at temperature T is:

- $$p_{s,s}(T) = 1 - \sum_{s \neq s'} p_{s,s'}(T).$$

- If $s' \in \mathcal{N}(s)$ then
$$p_{s,s'}(T) = \begin{cases} \pi_s^N(s') & \text{if } f(s) \leq f(s') \\ \pi_s^N(s')P(T,s,s') & \text{if } f(s) > f(s') \end{cases} .$$

- If $s' \notin \mathcal{N}(s)$ then $p_{s,s'}(T) = 0$.

If T is annealed slowly enough, then it can be shown that the distribution of states sampled by SA asymptotically converges to the Boltzmann distribution [14, 23]. However, this convergence result requires that equillibrium is acheived for each temperature, which requires an infinite number of steps. Further, the temperature must be lowered very slowly to insure convergence. In practice, the temperature is reduced according to a temperature schedule, where m_i steps are performed at each temperature T_i, $i \geq 1$.

3 Review of Bounds on Simulated Annealing

In this section I review the analysis by Ferreira and Žerovnik [7] that bounds the probability of success of SA on an arbitrary problem. Chiang and Chow [2] give a related analysis that shows that the probability of reaching the optimal state is $1 - O(n^{-a})$, for some $a > 0$.

```
Compute a random initial state $s$
$n = 0$, $x_n = s$
Repeat $i = 1, 2, \ldots$
   Repeat $j = 1, 2, \ldots, m_i$
      Compute a neighbor $s' = N(s)$
      if $(f(s') \leq f(s))$ then
        $s = s'$
        if $(f(s) < f(x_n))$ then
          $x_n = s$
          $n = n + 1$
        endif
      else
        $s = s'$ with probability $p(T, s, s')$
      endif
   EndRepeat
EndRepeat
```

Figure 2: **Pseudo Code for Simulated Annealing**

Let K_1 be the set of states in S for which there is at least one downhill path of transitions defined by the neighborhood operator $N()$ that ends in a global minima. Let $R = |K_1| \,/\, |S|$. Let q_i be the minimum probability, at temperature T_i and taken over all states, that SA does not accept a state s' such that $f(s') > f(s)$. Formally,

$$q_i = \min_{s \in S} \left(1 - \sum_{s' \in A_s} p_{s,s'}(T_i) \right),$$

where $A_s = \{s' \mid f(s') > f(s)\}$.

Let $S^* = \{s \mid f(s) \leq f(s'), \forall s' \in S\}$. Ferreira and Žerovnik [7] prove the following lemma, which shows that performance of SA can be bounded by a function of n, the number of steps taken by the algorithm.

Lemma 1 ([7], Lemma 1) If $R < 1$ then $P(x_n \in S^*) \leq b(n) < 1$, where

$$b(n) = 1 - q_{i_n}^{\bar{m}_{i_n}} \Pi_{j=1}^{i_n - 1} q_j^{m_j} (1 - R),$$

and i_n is the index to the temperature at step n and $n = \bar{m}_{i_n} + \sum_{j=0}^{i_n - 1} m_j$.

This bound implies several other bounds on the probability of success of SA. In particular, it can be generalized for more practical situations in which a near optimal solution is desired. Let $S^\epsilon = \{s \mid |f(s) - f(s^*)| \leq \epsilon, s^* \in S^*\}$. Let $R' = |K_1'| \,/\, |S|$, where K_1' is the set of states in S for which there is at least one downhill path of transitions defined by the neighborhood operator $N()$ that ends with a state in S^ϵ.

Corollary 1 If $R' < 1$ then $P(x_n \in S^\epsilon) \leq b(n) < 1$, where

$$b(n) = 1 - q_{i_n}^{\bar{m}_{i_n}} \Pi_{j=1}^{i_n-1} q_j^{m_j}(1 - R'),$$

and i_n is the index to the temperature at step n and $n = \bar{m}_{i_n} + \sum_{j=0}^{i_n-1} m_j$.

4 Analysis of Markovian Search Algorithms

In this section I demonstrate general conditions for which stochastic optimization algorithms can be shown to be asymptotically better than SA. These conditions rely on an analysis of homogeneous finite Markov chains, which can be used to define stochastic optimization algorithms.

A homogenouse finite Markov chain describes the probabilistic steps that can be taken over a finite space. The probability of moving from state s_i to state s_j is p_{ij}, and these probabilities can be collected into a *transition matrix* $P = (p_{ij})$. For each entry, $p_{ij} \in [0, 1]$ and $\sum_{j=1}^{|S|} p_{ij} = 1$ for all $s_i \in S$. Given an initial distribution p_0, the distribution of the chain after the nth step is $p^n = p_0 P^n$. State s_j is said to be *accessible* from state s_i if there exists $k \geq 0$ such that $P_{ij}^{(k)} > 0$.

Given a set of goal states $S' \subset S$, we define a search algorithm that relies on a Markov chain to sample the search space. The key property of the Markov chain is that at least one goal state is accessible from every state in S. This ensures that the Markov chain will visit a goal state with high probability as the number of steps becomes large. Formally, the Markov chains that we consider are defined by the following assumption.

Assumption 1 Consider a finite Markov chain with transition matrix P defined over S. Let $S' \subseteq S$ be given. For all $s \in S - S'$, there exists $s' \in S'$ such that s' is accessible from s.

The following lemma shows that a Markov chain that satisfies Assumption 1 will sample a state in S' with high probability for sufficiently large n.

Lemma 2 Let $S' \subset S$ and $S_n = \{s_1, s_2, \ldots, s_n\}$ be a sequence of states sampled by a Markov chain that satisfies Assumption 1. Then there exists $\alpha \in \langle 0, 1 \rangle$ and n_0 such that for $n > n_0$

$$P\left(S_n \bigcap S' = \emptyset\right) \leq \alpha^n.$$

Proof. Let $p = \min_{p_{ij} > 0} p_{ij}$, where $P = (p_{ij})$ is the transition matrix of the Markov chain.

For all $s \in S$, consider the shortest sequence of transitions that terminates with a state in S'. By Assumption 1 such a sequence of transitions always exists. Let q be the length of the longest such sequence.

If $p = 1$ then a state in S' is reached in no more than q steps, so for $n > n_0 = q$,

$$P\left(S_n \bigcap S' = \emptyset\right) \leq \alpha^n.$$

for any $\alpha \in \langle 0, 1 \rangle$.

Now assume that $p < 1$. The probability of traversing any sequence of length q is at least p^q. It follows that after q steps the probability that a state in S' is not sampled is less than or equal to $1 - p^q$. Similarly, the probability that a state in S' is not sampled after kq steps is

$$P\left(S_n \bigcap S' = \emptyset\right) \leq (1 - p^q)^k .$$

Now for arbitrary n we have

$$P\left(S_n \bigcap S' = \emptyset\right) \leq (1 - p^q)^{\lfloor n/q \rfloor} \leq (1 - p^q)^{n/(2q)} .$$

Thus if

$$\alpha = (1 - p^q)^{1/(2q)}$$

then

$$P\left(S_n \bigcap S' = \emptyset\right) \leq \alpha^n$$

for $n > n_0 = q$. ■

We can now define a search algorithm that uses this Markov chain to find a goal state. In the context of global optimization, the goal states are typically S^* or S^ϵ.

Definition 1 A *(finite) Markovian search algorithm* samples the search space for a state in $S' \subseteq S$ according to a Markov chain that satisifies Assumption 1 and keeps track of the best solution sampled.

The class of Markovian search algorithms is quite broad. It includes a variety of EAs, as well as many other standard global optimization algorithms. The following theorem proves the remarkable result that *any* Markovian search algorithm eventually has a higher probability of success than SA.

Theorem 1 Let $\mathcal{A}$ be a Markovian search algorithm. If $R < 1$ then there exists a constant n_0 such that if $n > n_0$ then $P_{SA}(x_n \in S^*) < P_{\mathcal{A}}(x_n \in S^*)$.

Proof. If n is sufficiently large, then we know from Lemma 2 that $P_{\mathcal{A}}(x_n \in S^*) \geq 1 - \alpha^n$, for some constant $\alpha \in \langle 0, 1 \rangle$. From the definition q_i it follows that q_i tends to one as n increases. Thus there exists a k such that $q_k > \alpha$. Let

$$\begin{aligned} D &= \Pi_{j=1}^{k-1} q_j^{m_j} (1 - R) \\ m &= \sum_{j=1}^{k-1} m_j . \end{aligned}$$

Now $0 < D < 1$ and

$$\left(\frac{\alpha}{q_k}\right)^{\bar{n}} < \alpha^{-m} D$$

for sufficiently large $\bar{n}$. It follows that

$$\alpha^{(\bar{n}+m)} < q_k^{\bar{n}} \Pi_{j=1}^{k-1} q_j^{m_j} (1 - R).$$

If we take $n = \bar{n} + m$ and note that $q_k^{\bar{n}} < q_i^{\bar{m}_i} q_{i-1}^{m_{i-1}} \dots q_k^{m_k}$, $i = i_n$, then we have

$$\alpha^n = \alpha^{(\bar{n}+m)} < q_i^{\bar{m}_i} \Pi_{j=1}^{i-1} q_j^{m_j} (1 - R).$$

Thus

$$P_{\mathcal{A}}(x_n \in S^*) \geq 1 - \alpha^n > 1 - q_i^{\bar{m}_i} \Pi_{j=1}^{i-1} q_j^{m_j} (1 - R) \geq P_{SA}(x_n \in S^*).$$

■

This result can be generalized to the case in which a near optimal solution is desired. The proof is identical, though the constant n_0 differs from that of Theorem 1.

Corollary 2 Let $\mathcal{A}$ be a Markovian search algorithm. If $R' < 1$ then there exists a constant n_0 such that if $n > n_0$ then $P_{SA}(x_n \in S^\epsilon) < P_{\mathcal{A}}(x_n \in S^\epsilon)$.

Finally, we comment on the condition $R < 1$ ($R' < 1$) that is required for these results. The following lemma demonstrates that this condition is satisfied if and only if the objective function contains a local minimum that is not globally optimal. Since stochastic global optimization algorithms are most appropriate for multimodal objective functions, this condition does not appear restrictive.

Definition 2 A *local minimum* is a state $s \in S$ such that $f(s) \leq f(s')$ for all $s' \in \mathcal{N}(s)$.

Lemma 3 $R < 1$ if and only if there is at least one local minima in S that is not a global optima.

Proof. Assume that there is a local minima that is not a global optima. From the definition of K_1, this state is not included in K_1, so $R < 1$.

Now if $R < 1$, then $S - K_1$ is not empty. Take any state $s \in S - K_1$ and consider a downhill path of transitions from s. At some point, this path terminates at a local minima. Since this path cannot lead to a local minima in K_1, this local minima must be in $S - K_1$. Consequently, there is at least one local minima in S that is not a global optima. ■

5 Analysis of Evolutionary Algorithms

It is quite natural to model EAs as Markov processes in which the state space contains the space of all possible populations, where each population is represented as a set of individuals. In particular, the Markov chains of GAs have been analyzed by a number of researchers [5, 6, 11, 16, 17, 20], since GAs have traditionally been applied to discrete search spaces (especially $\{0,1\}^L$).

As a consequence, we might expect the analysis of Markovian search algorithms to apply to certain classes of EAs. In fact this is the case. However, we need to prove modified forms of Theorem 1 and Corollary 2. These results assume that visiting n states in S incurs the same cost for both SA and the Markovian search algorithm.

However, this is not true for the Markov chains that model EAs, since each state is a vector of the states sampled by SA.

More formally, for Markov chains that model EAs the state space is $\bar{S} = \{(s_1, \ldots, s_N) \mid s_i \in S\}$. EAs simply need to find a population that contains one or more instances of optimal individuals. Consequently, we define $\bar{S}^* = \{(s_1, \ldots, s_N) \mid \exists s_i \in S^*, i = 1, \ldots, N\}$ and $\bar{S}^\epsilon = \{(s_1, \ldots, s_N) \mid \exists s_i \in S^\epsilon, i = 1, \ldots, N\}$. Note that if a state in $\bar{S}^*$ is sampled, then the canonical EA defined in Figure 1 has the property that $x_n \in S^*$.

Lemma 4 Let $\mathcal{A}$ be a Markovian search algorithm that costs N per transition. If $R < 1$ then there exists a constant n_0 such that if $n > n_0$ then $P_{SA}(x_{nN} \in S^*) < P_{\mathcal{A}}(x_n \in S^*)$.

Proof. If n is sufficiently large, then we know from Lemma 2 that $P_{\mathcal{A}}(x_n \in S^*) \geq 1 - \alpha^n \geq 1 - \alpha^{n/N}$, for some constant $\alpha \in \langle 0, 1 \rangle$. From the definition q_i it follows that q_i tends to one as n increases. Thus there exists a k such that $q_k > \alpha^{1/N}$. Let

$$\begin{aligned} C &= \alpha^{1/N} \\ D &= \Pi_{j=1}^{k-1} q_j^{m_j}(1-R) \\ m &= \sum_{j=1}^{k-1} m_j. \end{aligned}$$

Now $0 < D < 1$ and
$$\left(\frac{C}{q_k}\right)^{(\bar{n}+m-1)N} < C^{-mN} D$$
for sufficiently large $\bar{n}$. It follows that

$$C^{(\bar{n}+m)N} < q_k^{(\bar{n}+m-1)N} \Pi_{j=1}^{k-1} q_j^{m_j}(1-R).$$

If we take $n = \bar{n} + m$ and note that $q_k^{(\bar{n}+m-1)N} < q_i^{\bar{m}_i} q_{i-1}^{m_{i-1}} \ldots q_k^{m_k}$, $i = i_{nN}$, then we have
$$C^{nN} = C^{(\bar{n}+m)N} < q_i^{\bar{m}_i} \Pi_{j=1}^{i-1} q_j^{m_j}(1-R).$$

Thus

$$P_{\mathcal{A}}(x_n \in S^*) \geq 1 - \alpha^n > 1 - q_i^{\bar{m}_i} \Pi_{j=1}^{i-1} q_j^{m_j}(1-R) \geq P_{SA}(x_{nN} \in S^*).$$

■

Corollary 3 Let $\mathcal{A}$ be a Markovian search algorithm that costs N per transition. If $R' < 1$ then there exists a constant n_0 such that if $n > n_0$ then $P_{SA}(x_{nN} \in S^\epsilon) < P_{\mathcal{A}}(x_n \in S^\epsilon)$.

Theorem 2 applies Lemma 4 to show that a wide variety of evolutionary algorithms are stochastically superior for large enough n. Let $\Gamma(\bar{x}^n) \subseteq S$ be the set of all possible individuals that could possibly be generated from $\bar{x}^n$ by the evolutionary operators.

Assumption 2 Consider an EA for which

1. $\forall \bar{s} \in \bar{S}, \Gamma(\bar{s}) = S$
2. $\forall s \in \Gamma(\bar{x}^n)$, if $f(s) < f(s'), \forall s' \in \bar{x}^n$ then $P(s \in \bar{x}^{n+1}) > 0$.

Assumption 3 Consider an EA for which

1. $\forall \bar{s} \in \bar{S}$ there exists a sequence of transitions $\bar{s} = \bar{s}_1, \bar{s}_2, \ldots, \bar{s}_j$ such that $\bar{s}_j \in \bar{S}^*$ and $s_{i+1} \in \Gamma(\bar{s}_i)^N$ for $i = 1, \ldots, j-1$.
2. $\forall s \in \Gamma(\bar{x}^n)$, $P(s \in \bar{x}^{n+1}) > 0$.

Theorem 2 Let $\mathcal{A}$ be an EA that can be modelled by a Markov chain that satisfies Assumption 2 or 3. If $R < 1$ then there exists a constant n_0 such that if $n > n_0$ then $P_{SA}(x_{nN} \in S^*) < P_{\mathcal{A}}(x_n \in S^*)$.

Proof. Assumptions 2 and 3 place restrictions on the transition matrix of the EA that ensure that it satisfies Assumption 1, where $S' = \bar{S}^*$. Given this, the result follows from Lemma 4 because the EA is a Markovian search algorithm.

If Assumption 2 is satisfied, then in a single transition every individual s can move to a state in S^*. Thus every state $\bar{s}$ can move to a state in $\bar{S}^*$. The second part of Assumption 3 guarantees that such a transition has positive probability, so Assumption 1 is satisfied.

If Assumption 3 is satisfied, then there exists a sequences of transitions that take every state in $\bar{S}$ to a state in $\bar{S}^*$. The second part of Assumption 2 guarantees that the probability of this sequence of transitions is positive, so Assumption 1 is satisified.

■

The restrictions imposed by Assumptions 2 and 3 are quite mild. Consequently, Theorem 2 applies to a large number of the EAs that have been previously considered. Assumption 2 restricts Theorem 2 to EAs for which the evolutionary operators can generate *any* solution in S. Consequently, the particular selection algorithm used in this class of EAs is irrelavant, since any solution can be used to generate a solution in S^*. This assumption allows for a very restrictive replacement algorithm, one that only needs to allow solutions to remain in the next population if they are better than the best solution in the current population.

The conditions imposed by Assumption 2 are satisfied by the canonical GA [17], which searches a space of binary vectors $\{0,1\}^L$ with a mutation operator that flips each bit along an individual with probability $\mu > 0$. The standard replacement algorithm for the canonical GA is to simply replace the entire population. This clearly satisfies the requirements of Assumption 2. The elitist GA considered by Suzuki [20] also satisfies Assumption 2. Assumption 2 also reflects the characteristics of EP and ES. These EAs use genetic operators that can generate any solution in the search space from any other solution. Furthermore, they use replacement algorithms that are no more restrictive than those permitted by Assumption 2. Consequently, Theorem 2 would apply to variants of EP or ES that are used for combinatorial optimization.

Assumption 3 restricts Theorem 2 to EAs for which every initial population can reach a population in $\bar{S}^*$ via some sequence of transitions. Further, the selection and replacement algorithms are restricted to allow any sequence of transitions to occur with positive probability. This assumption can be viewed as a generalization of Assumption 2 for the case where the genetic operators cannot necessarily generate a solution in S^* from every state in S. Examples of EAs that satisfy Assumption 3 are many of the GAs used to search through the space of permutations. These EAs have been applied problems such as the Travelling Salesman problem [19, 24], scheduling problems [19, 21], the clustering problem [1] and partitioning problems [13]. The limitation of the evolutionary operators for these EAs lies in the fact that the mutation operator is typically an operation like two-opt, which generates a small number of possible neighbors.

6 Discussion

Note that the analysis of EAs in Section 4 is independent of the particular formulation of evolutionary operators used by the EA. Consequently, the conditions in Theorem 2 may be satisfied by either mutation or crossover or both. This fact distinguishes this analysis from analyses of Markov chains of EAs that rely on mutation to prove performance guarantees (e.g. see Rudolph [17]).

Another interesting property of this analysis is that the neighborhood structure used by the Markov chain for the EA is independent of the neighborhood structure used by SA. Thus the steps used by the EA for mutation or crossover are not necessarily related to the steps used by SA. In practical terms, this means that the EA can encode the problem differently so long as the cardinality of the search space is preserved.

An important criticism of these results is that the value n_0 postulated by the results may be unreasonably large. Thus in practice SA may have better performance than EAs for all reasonable values of n. Further, it is not clear from this analysis whether it is necessarily true that $n_0 < |S|$. If $n_0 \geq |S|$ it is often possible to simply enumerate the space, in which case these results do not have practical relevance. Since the value n_0 depends upon the problem and neighborhood structures used by the SA and EAs, an important research issue is the identification of problems for which $n_0 < |S|$.

Note that this criticism does not apply if S is countably infinite. In fact, the results in Sections 3 and 4 can be generalized to handle the case when S is countably infinite. However, it is difficult to imagine many practical examples with this property. NP-complete problems typically have search spaces that are exponential in the size of the input, so for a given input the search space is finite. An example of a problem that might be reasonable is the grammar induction problem [25] where a grammar for an infinite set of sequences is being learned.

To conclude, we note that these results should naturally generalize to parallel EAs. Ferreira and Žerovnik [7] describe bounds for parallel versions of SA. Consequently, parallel GAs should naturally have a similar relationship with parallel SA. We also expect that the analysis of Markovian search algorithms can be used to prove that a variety of other search algorithms are superior to SA. An important step that is missing from the current analysis is accounting for general costs in these algorithms, which is necessary to generalize to a wide variety of other search algorithms.

Acknowledgements

I wish to thank Afonso Ferreira and Janez Žerovnik for clarifying their proof of the bound on the performance of simulated annealing. I also thank John DeLaurentis and David Wilson for their helpful discussions.

References

[1] J. N. Bhuyan, V. V. Raghavan, and V. K. Elayavalli, *Genetic algorithm for clustering with an ordered representation*, in Proc. of the 4th Intl. Conf. on Genetic Algorithms, 1991, pp. 408–415.

[2] T. Chaing and Y. Chow, *On the convergence rate of annealing processes*, SIAM J. Control and Optimization, 26 (1988), pp. 1455–1470.

[3] L. Davis, *Genetic Algorithms and Simulated Annealing*, Pittman, 1987.

[4] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[5] T. Davis and J. C. Principe, *A simulated annealing like convergence theory for the simple genetic algorithm*, in Proc. of the 4nd Intl. Conf. on Genetic Algorithms, 1991, pp. 174–181.

[6] K. A. De Jong, *An Analysis of the Behavior of a Class of Genetic Adaptive Systems*, PhD thesis, University of Michigan, Ann Arbor, 1975.

[7] A. G. Ferreira and J. Žerovnik, *Bounding the probability of success of stochastic methods for global optimization*, Computers Math. Applic., 25 (1993), pp. 1–8.

[8] D. B. Fogel, *Evolutionary Computation*, IEEE Press, Piscataway, NJ, 1995.

[9] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Co., Inc., 1989.

[10] D. E. Goldberg and K. Deb, *A comparative analysis of selection schemes used in genetic algorithms*, in Foundations of Genetic Algorithms, G. J. Rawlins, ed., Morgan-Kauffmann, San Mateo, CA, 1991, pp. 301–315.

[11] D. E. Goldberg and P. Segrest, *Finite Markov chain analysis of genetic algorithms*, in Proc. of the 2nd Intl. Conf. on Genetic Algorithms, 1987, pp. 1–8.

[12] L. Ingber and B. Rosen, *Genetic algorithms and very fast simulated reannealing - a comparison*, Mathematical and Computer Modelling, 16 (1992), pp. 87–100.

[13] D. R. Jones and M. A. Beltramo, *Solving partitioning problems with genetic algorithms*, in Proc. of the 4th Intl. Conf. on Genetic Algorithms, 1991, pp. 442–449.

[14] S. Kirkpatrick, C. Gelatt, Jr., and M. Vecchi, *Optimization by simulated annealing*, Science, 220 (1983), pp. 671–680.

[15] T. Bäck and H.-P. Schwefel, *An overview of evolutionary algorithms for parameter optimization*, Evolutionary Computation, 1 (1993), pp. 1–23.

[16] A. E. Nix and M. D. Vose, *Modelling genetic algorithms with Markov chains*, Annals of Mathematics and Artificial Intelligence, 5 (1992), pp. 79–88.

[17] G. Rudolph, *Convergence analysis of canonical genetic algorithms*, IEEE Transactions on Neural Networks, 5 (1994), pp. 96–101.

[18] m. Rudolph, *Massively parallel simulated annealing and its relation to evolutionary algorithms*, Evolutionary Computation, 1 (1993), pp. 361–383.

[19] T. Starkweather, S. McDaniel, D. Mathias, D. Whitley, and C. Whitley, *A comparison of genetic sequence operators*, in Proc. of the 4th Intl. Conf. on Genetic Algorithms, 1991, pp. 69–76.

[20] J. Suzuki, *A Markov chain analysis on a genetic algorithm*, in Proc. of the 5nd Intl. Conf. on Genetic Algorithms, 1993, pp. 146–153.

[21] G. Syswerda and J. Palmucci, *The application of genetic algorithms to resource scheduling*, in Proc. of the 4th Intl. Conf. on Genetic Algorithms, 1991, pp. 502–508.

[22] N. L. Ulder, E. H. Aarts, H.-J. Bandelt, P. J. van Laarhoven, and E. Pesch, *Genetic local search algorithms for the traveling salesman problem*, in Parallel Problem Solving from Nature, H.-P. Schwefel and R. Männer, eds., New York, 1990, Springer-Verlag, pp. 109–116.

[23] P. van Laarhoven and E. Aarts, *Simulated Annealing: Theory and Applications*, Reidel, 1987.

[24] D. Whitley and N.-W. Yoo, *Modeling simple genetic algorithms for permutation problems*, in Foundations of Genetic Algorithms 3, 1995, pp. 163–184.

[25] P. Wyard, *Context free grammar induction using genetic algorithms*, in Proc. of the 4th Intl. Conf. on Genetic Algorithms, 1991, pp. 514–518.