Annealing a Genetic Algorithm for Constrained Optimization

F. Mendivil[1 2] and R. Shonkwiler[3]

Abstract

This paper considers the problem of adapting a Genetic Algorithm (GA) to constrained optimization problems, using a dynamic penalty approach as a type of annealing to force the search to concentrate on feasible solutions as the algorithm progresses. We suggest two different methods for ensuring almost sure convergence to a globally optimal (feasible) solution. The first of these involves modifying the GA evolution operators to yeild a Boltzmann-type distribution on populations. The second incorporates a dynamic penalty along with a slow annealing of acceptance probabilities. We prove the almost sure convergence of both of these methods.

Key Words: Genetic Algorithms, Constrained Optimization, Simulated Annealing

1 Introduction

Let f be a real-valued function defined on the finite domain Ω. By transformation if necessary, we may assume without loss of generality that $f : \Omega \to \mathbb{R}$ is strictly positive, i.e., $f(x) > 0$ for points $x \in \Omega$. Some (strict) subset of the points of Ω may be designated as *infeasible*. The purpose of this report is to discuss Genetic Algorithm based methods for finding the global optimum of f constrained to the feasible points of Ω. We distinguish between the function f to be optimized and the *fitness* ϕ of the algorithm, which is clearly taken to be a function of f. Our basic approach is to vary ϕ dynamically as a function of "time" t, taken as a non-decreasing function of the iteration count of the algorithm. Therefore $\phi_x(t)$ is a function of both $x \in \Omega$ and run time $t = 1, 2, \ldots$. An alterate approach taken by some authors is to vary the mutation rate (for example, see [1]).

A Genetic Algorithm is a Markov chain defined on the set Γ of populations over Ω. A population i of Ω is a fixed sized multi-subset of Ω, that is a subset of Ω of a given cardinaliy, say K, possibly having repeated members,

$$i = \{x_1, x_2, \ldots, x_K\} \subset \Omega.$$

By the *fitness* Φ_i of a population i we usually mean the maximal fitness of its members,

$$\Phi_i(t) = \max_{x \in i} \phi_x(t). \tag{1}$$

[1]Support for this work from the Natural Sciences and Engineering Research Council of Canada (NSERC)

[2]Department of Mathematics and Statistics, Acadia University, Wolfville, NS, B4P 2R6

[3]School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332

In some instances, another definition of the fitness of a population might be convenient.

Our basic penalty mechanism is simple. Let $M : \Omega \to [0, \infty)$ be a measure of feasibility. That is, $M(x) = 0$ if and only if $x \in \Omega$ is feasible and $M(x) > 0$ if x is infeasible. For our results, it is possible for $M(x) \in \{0, 1\}$. Further, let β_t be an increasing sequence of positive numbers. Given $x \in \Omega$ and $t \in \mathbb{N}$, we define the *attenuated fitness* $\phi_x(t)$ to be

$$\phi_x(t) = e^{-M(x)\beta_t} f(x). \tag{2}$$

This clearly has the effect of imposing an increasing penalty on infeasible states as the iteration count t increases.

It is useful to point out the analogy with simulated annealing (SA). SA uses a *cooling schedule* where temperature is some function of the iteration count, $T = c(n)$, often taken as $c(n) = C/\log(n)$. Then the probability of an unfavorable transition is given by

$$e^{-\Delta E/(kT)} = e^{-\frac{\Delta E \log(n)}{kC}} = (1/n)^{\Delta E/(kC)}. \tag{3}$$

Matching analogous parts of the penalty in (2) with (3), we have M correpsonding to $\Delta E/(kC)$ and β_t corresponding to $\log(n)$.

For a given Genetic Algorithm and two populations i, j, we use $p_{ij}(t)$ to denote the probability of a one-step transition from population i to population j at time t (as our probabilities will in general be time-dependent). We define $\mathcal{F} \subset \Gamma$ as those populations containing only feasible states, $\mathcal{I} \subset \Gamma$ as those popluations containing only infeasible states and $\mathcal{M} \subset \Gamma$ as all other states (that is, each population in $\mathcal{M}$ contains at least one feasible and one infeasible state).

Related Prior Work

The problem of optimization under a constraint is clearly a classical problem and has generated a huge amount of research. For a very nice overview of evolutionary approaches to optimization in general see [2], and in particular chapter 9 for a discussion of constraint-handling techniques. The paper [3] also has some good discussion on constraint-handling.

There are many possibilities for handling constraints in evolutionary algorithms including deleating infeasible solutions, attempting to "repair" or "project" an infeasible solution to the set of feasible solutions, and designing the algorithm to maintain feasibility, to name just a few. Our approach is very simple in that infeasible solutions are treated the same as feasible solutions and the dynamic penalty will asymptotically force them to be removed from the population. In addition, by adding an extra "annealing" to the algorithm, the process is guaranteed to converge not only to the set of feasible solutions but also to the set of feasible and globally optimal solutions.

The problem of designing a population-based evolutionary algorithm with a Boltzmann invariant distribution has also previously been investigated (in particular, see [4, 5, 6]). In [5], Variation 1 of the algorithm leads to a Boltzmann distribution on populations with the fitness of a population defined as the sum of the fitnesses of the individuals. Our method as outlined in Sections 4 and 5 is close to this method and provides another technique to ensure reversibility (using *trits*). We discuss the method in [6] further at the start of Section 4 and compare it with our method.

2 Convergence of penalized GA to feasible states

In this section, we discuss the problem of using a Genetic Algorithm for a constrained optimization problem. Our approach will be to incorporate a dynamic penalty attenuating the fitness of the infeasible states so that in the limit all the infeasible solutions will be poor performers in comparison to any feasible state.

There are two basic goals in using a dynamic penalty approach to solve a constrained optimization problem:

1. Ensure that the “solution” is a feasible state.

2. Ensure that the “solution” is an optimal state.

Here the “solution” is that point of the state space which is identified by the algorithm as being optimal.

Clearly the second goal is a stronger condition than the first (at least, if the dynamic penalty is working correctly). We start with the first condition, as it is the simpler one to ensure. However, we point out that even the first goal is not always a trivial task. For some combinatorial problems finding even one feasible solution can be an NP-hard task (for example, the TSP problem with time constraints).

We say an attenuated fitness GA is *irreducible* if the unattenuated version of the GA is irreducible as a Markov Chain. This means that it is possible for the GA to transition from any population to any other population in a finite number of steps.

We will say that an attenuated GA is *asserting* if there is some $\delta > 0$ so that for any population i containing an infeasible individual x there is another population j retaining all the feasible individuals from i but where x has been replaced by a feasible individual and for which $p_{ij} > \delta$. Further, we say the chain is *infeasible diminishing* if the probability of a transition to a population with more infeasible individuals than the present population decays to 0 as $t \to \infty$.

The interest in these definitions is that in order that the “solution” will be feasible, it is sufficient that a GA is asserting and infeasible diminishing. If the selection or removal phase of the GA depends on ratios of (attenuated) fitness, then the GA will most likely be infeasible diminishing. Many implementaions of the mutation operation will allow the creation of any individual in one step. In this case, the GA will be asserting (as with positive probability an infeasible individual can be exchanged for a feasible one).

Theorem 2.1 *Let X_n be the Markov Chain associated with an attenuated GA which is asserting and infeasible diminishing. Then asymptotically as $n \to \infty$, $P(\ X_n \in \mathcal{F}\ ) \to 1$.*

Proof: If K is the population size, then by the assumption that the GA is asserting we see that

$$\Pr(X_{n+i} \in \mathcal{F} \text{ some } i = 0, 1, 2, \ldots, K | X_n) > \delta^K > 0$$

for any n and state (population) X_n. However, then this means that with probability one we have $X_n \in \mathcal{F}$ infinitely often. The fact that the GA is infeasible diminishing means that $\Pr(X_{n+1} \notin \mathcal{F} | X_n \in \mathcal{F}) \to 0$ as $n \to \infty$. These two facts imply the result. ∎

Notice that since the state space is finite (our blanket assumption in this paper), there is some time t^* so that for any fixed $t > t^*$, the attenuated fitness of any infeasible state is worse than the fitness of any of the goal (optimal) states. Thus, if one fixed the attenuation factor at this level, an unconstrained GA would find the same feasible goal as the constrained GA. Of course, the problem is that one does not in general know when this time t^* occurs and thus needs to continues the attenuation.

However, just because the GA limits to feasible states does not necessarily guarantee that it will find the optimal solution. What may happen is that the feasible states comprise a collection of "islands" which are surrounded by infeasible states. If the problem is deceptive and the dynamic penalty is not carefully controlled then it is possible to get stuck in the wrong component with high probability and have the GA never sample an optimal state. In the language of Markov Chains, the limiting chain has multiple ergodic components and the chain gets stuck in one of these components which does not contain an optimal state.

Example of non-convergence to optimal

As an example, take the state space $\Omega = \{0, 1, \ldots, 7\}$ with states $1, 2, \ldots, 6$ as infeasible and $f(0) = 2, f(7) = 3$ and $f(x) = 1$ for all infeasible states x. We take $\beta_t = \ln(t)$ so that $e^{-M\beta_t} = 1/t$ (so $M(x) = 1$ for all infeasible states). For the GA, take a four-population three-bit genetic algorithm as follows. An iteration starts by selecting, one at a time, four members of the current population via the roulette wheel method (that is, proportional to fitness). Those selected are paired off randomly and cross-over is performed on the bit representations of both pairs to obtain two offspring each (the cross-over point is equally likely to be between any two bit positions). Next, one mutation is performed with probability p. If a mutation is to occur, one of the four offspring is selected equally likely and one bit of the selected offspring is chosen at random and “flipped”. With or without mutation, the four offspring constitue the new population completing the iteration.

This GA is infeasible diminishing as the selection is proportional to fitness, so as the iteration count increases, the infeasible individuals become less likely to be selected for the next population. It is also asserting, as the bit representation of an infeasible individual has either one or two bits set to 1, so can be mutated to a feasible individual by flipping the correct bit.

However, it can be shown that if the GA has not found the optimal state (state 7) by iteration N, then the probability that it will ever be found decays to zero as N tends to infinity. The infeasible states form an ever deepening “valley” between the two feasible states and it becomes more and more difficult to traverse this “valley” as the algorithm proceeds.

3 Ensuring the convergence of a GA

In the following sections, we present two methods for ensuring the convergence of a GA (dynamic penalty or not) to the set of global optimal states. The first method consists of controlling the (time varying) limiting distribution by ensuring that it is the Boltzmann distribution while the second one only adds an *acceptance phase* to the GA.

Genetic algorithms plainly tend to favor more fit individuals over less fit ones. Thus, increasing the fitness of desirable individuals while decreasing the fitness of undesirable (or even infeasible) individuals during the course of a GA run is one way to try to ensure convergence. However, this really only works if the transitions (and thus the invariant distribution) depend in a predictable way on the fitness. That is, using changing fitness pressure to guide the GA usually requires a modification of the GA mating and mutation operators as well as the selection operator. However, once you do this, you might as well change them in such a way to make the invariant distribution easy to predict. This is the strategy we take with our first method of *Boltzmann transitions.* This strategy derives its motivation from Simulated Annealing, where by construction the (time varying) invariant distribution is the Boltzmann distribution.

Our second method also derives inspiration from Simulated Annealing, but this time only in the fact that it uses an *acceptance protocol* independent from the original GA, which is used as the *proposal process.* That is, we use the original GA operators to generate a new population from the old one and then use the acceptance protocol to decide whether to take the transition to the new population or to keep the old one. This method requires fewer changes to the GA algorithm, but it does not have a predictable sequence of invariant distributions.

4 Method I: Boltzmann transition rules

One method for assuring that the limiting distribution, as $t \to \infty$, has unit mass on the state of global optima is by controlling the stationary distribution of the chain at every value of t. Then it becomes a matter of increasing t sufficiently slowly in order that the chain closely approximate its stationary distribution each step of the way. In this section we keep the parameter t fixed and attempt to engineer the stationary distribution of the algorithm for that fixed time. As above, our control over the algorithm is via the fitness funcition $\phi(t)$; as we fix t, we notationally suppress it when convenient.

The *Boltzmann* distribution on the space Γ of populations is the one for which the probability of observing population i is proportional to $\Phi_i(t)$,

$$\Pr(i) = \frac{\Phi_i(t)}{\Sigma_{j\in\Gamma}\Phi_j(t)}.$$

The stationary distribution $\pi(t)$ of a standard genetic algorithm favors populations of greater fitness, but to what degree is normally difficult to predict. However if it were possible to arrange this to be the Boltzmann distribution then $\pi(t)$ would be easy to calculate and it would be simple to compare populations with respect to their sampling frequency.

One way to ensure the stationary distribution will be Boltzmann is if the property known as *detailed balance* holds. Given the transition probabilities $p_{ij}(t)$, if

$$\Pi_i(t)p_{ij}(t) = \Pi_j(t)p_{ji}(t) \qquad \text{for all } i, j \in \Gamma \tag{4}$$

holds, then Π is the stationary distribution. The idea is to ensure this relation with Π as the Boltzmann distribution. The detailed balance equation (4) is also known as the *reversiblility condition* in that a chain which satisfies (4) will look the same running backwards in time as running forwards (that is, if it has achieved its stationary distribution).

We provide one mechanism for achiving a Boltzmann distribution. For other alternatives, see [4, 6]. In particular, [6] has a very similar theme to our method (as described in this section and the next). In [6], the author also uses Boltzmann-like selection probabilities to ensure the Boltzmann distribution is the stationary distribution. Each iteration also changes only one population member at a time. However, the scheme for ensuring reversibility (their symmetry condition on the neighborhoods) in their example GA seems to have a problem and we devised the "trit" representation (discussed in the next subsection) as a practical way to ensure reversibility. In particular, allowing only one parent of the "child" to be removed is necessary for reversibility. A scheme which allows the removal of an arbitrary member of the population is generally not reversible.

In fact, our method ensures both the reversibility of the Markov Chain and also yields a chain in which it is possible to go from any population to one consisting of only optimal states all the while not decreasing the population fitness. This makes the optimization problem have only one "basin", and gives the limiting chain only one ergodic component. We describe our methods in detail and provide proofs in part because the results in [6] cannot be applied to our situation, as our transitions do not satisfy their reversibility condition and our fitness function is time-varying, both because of the dynamic penalty and also because we anneal to force the process to find the optimal feasible states.

Tri-state elements or trits

The transition from one population to another, i to j, is conducted by constructing a proposed new population using "mating" and "mutation". This is followed by a roulette wheel selection which could result in the new population being accepted or in no change to the population. Cast in these terms, the process is reminiscent of that in simulated annealing. We define the process in such a way as to be reversible. As equation (4) shows, this is necessary for detailed balance to hold.

The usual way of representing individuals is by using a bit representation. However, then the usual cross-over mating and elimination might not be reversible. A simple example is combining the two parent bit-strings `01` and `10` to get an offspring of `00`. Then if we remove the parent `01` and retain `10` and `00`, it is impossible to generate `01` from the remaining population. Thus, this operation is not *removal reversible.*

Instead, suppose that the points x of Ω are L-tuples of the three element set $\{0, 1, 2\}$, in short, L-tuples of *trits.* This can be arranged when the points of Ω are real numbers simply by representing them in base three.

For two trits s, t we define the *symmetric complement* operator $\triangle$ as

$$s \triangle t = \begin{cases} s, & \text{if } s = t \\ \{0,1,2\} \setminus \{s,t\}, & \text{if } s \neq t. \end{cases}$$

For two vectors of trits x, y, we define $x \triangle y$ component by component.

Given a population i of size K, say $i = \{x_1, \ldots, x_K\}$, select two equally likely, say x_1 and x_2, and then use the operator $\triangle$ as the mating operator, $y' = x_1 \triangle x_2$.

The binary operator $\triangle$ has the property that from any two of the three points, x_1, x_2, or y', the third can be recovered component by component and thus we have:

Property 4.1 *The symmetric complement operation on trits is removal reversible.*

Returning to the proposal scheme, we now perfom a "mutation" operation. To do this, having constructed y', next select one of its L components equally likely and replace that component by a randomly selected trit. (Alternatively, several trits could be mutated in this way.) Denote the resulting mutated offspring by y; y together with x_1 and x_2 form an *augmented sub-population.* We remove one of these three via the roulette wheel method as described next. The basic philosophy is that the removal probability should depend on if removing the given individual could decrease the fitness of the population (note that it couldn't increase the fitness).

If $\Phi_i > \max(\phi_y, \phi_{x_1}, \phi_{x_2})$, make the roulette wheel probabilities equal, that is, select which individual to remove from the augmented sub-population equally likely.

If $\Phi_i \leq \max(\phi_y, \phi_{x_1}, \phi_{x_2})$ let α be the largest number and ω be the second largest number in the set $\{\phi_y, \phi_{x_1}, \phi_{x_2}\}$.

We define the roulette wheel probability for removing whichever point corresponds to α (the most fit individual) as

$$\frac{\omega}{2\alpha + \omega}.$$

The roulette wheel probabilities for the other two are both equal to

$$\frac{\alpha}{2\alpha + \omega}.$$

Notice that if $\alpha = \omega$ then all the transitions probabilities are equal to $1/3$.

Implementing the trit GA as described here takes only a modest additional effort. Forming the offspring $y' = x_1 \triangle x_2$ requires, in each component of the L-tuple, setting the trit of y' equal to that of x_1 and x_2 if they are the same or equal to the one missing if they are not. Then for roulette wheel selection both the first α and second ω largest fitnesses for the 3 member augmented population must be determined, which is a very simple matter.

The probability $p_{ij}(t)$ of a transition from population i to another population j is easily calculated. If i and j differ in more than one member, then $p_{ij}(t) = 0$.

Otherwise suppose that i and j differ in only one member. This member, y, of j must be the symmetric complement of two members of i, except possibly in one trit, with one of the members unique to i.

Since we selected the two "parents" equally likely from the population, the probability that any given pair is selected is

$$c = \frac{2}{K(K-1)},$$

(recall that K is the population size). Similarly, since we select which component to mutate equally likely, the probability that mutation results in with a specified component in a given value is

$$\mu = \frac{1}{3L}$$

as there are L components each with three possible values.

Now suppose x_1 and x_2 are selected as parents from population i and assume $\phi_{x_1} \geq \phi_{x_2}$ and y is generated as the offspring. Then

$$p_{ij}(t) = \begin{cases} c\mu\left(\frac{\omega}{2\alpha+\omega}\right), & \text{if } \phi_{x_1} \geq \phi_y \text{ and } x_1 \text{ is removed} \\ c\mu\left(\frac{\alpha}{2\alpha+\omega}\right), & \text{otherwise.} \end{cases}$$

In the first case notice that x_1 has been replaced by y and so $\Phi_j \leq \Phi_i$ (as $\phi_{x_1} \geq \phi_y$).

Theorem 4.1 *Detailed balance holds for tri-state transitions as defined above.*

Proof: If $i = j$ there is nothing to prove. Otherwise some member of i, x_1 say, has been replaced by some point, y. Therefore the augmented sub-population a will contain x_1 and y. There could be several paths from i to j in which the third member of the augmented sub-population differ. What we show is that each such path, through the same augmented sub-population, is reversible and satisfies detailed balance; hence the transition from i to j will as well.

Let x_1 and x_2 be selected for mating and let y be the mutated offspring; then $a = \{y, x_1, x_2\}$. To show detailed balance through a involves the consideration of several cases. First, if $\Phi_i > \max(\phi_{x_1}, \phi_{x_2}, \phi_y)$, then $\Phi_j = \Phi_i$ and since the removal probabilities are 1/3, it follows the result is correct for this case.

Next suppose $\alpha = \phi_{x_1}$. If y is selected out, then $j = i$ and detailed balance holds. If x_1 is selected out, then $j = \{y, x_2, \ldots, x_K\}$ and $\Phi_j = \omega$ since α is the fitness of x_1 and x_1 was removed. Hence

$$\Phi_i \; p_{ij} = \alpha c \mu \left(\frac{\omega}{2\alpha + \omega} \right)$$

and, since y is selected out on the reverse transition,

$$\Phi_j \; p_{ji} = \omega c \mu \left(\frac{\alpha}{2\alpha + \omega} \right).$$

Detailed balance holds as these are equal. If x_2 is selected out, then $j = \{y, x_1, \ldots, x_K\}$ and $\Phi_j = \alpha$. Hence

$$\Phi_i \; p_{ij} = \alpha c \mu \left(\frac{\alpha}{2\alpha + \omega} \right) = \Phi_j \; p_{ji}$$

since again y is selected out on the reverse transition Thus detailed balance holds if $\alpha = \phi_{x_1}$.

The remaining case $\alpha = \phi_y$ is similar and is omitted. ▮

Irreducibility

We consider the irreducibility of the chain for $t < \infty$ here; the case for the limiting chain as $t \to \infty$ is taken up below in Section 5. When $t < \infty$ the chain is irreducible even while maintaining fixed members of the population. Let i be any population and let x be a member of i of maximal fitness. Now let j be any population containing x, then there is a finite sequence of populations, $i_0 = i$, $i_1, \ldots, i_\ell = j$, each of which contains x, such that $p_{i_{k-1}, i_k}(t) > 0$, $k = 1, \ldots, \ell$.

To show this we only need to show that for each member z of j different from a member of i, it is possible to generate z in a finite number of steps from something in i different from x all the while maintaining x in the population. But this can be accomplished trit by trit as follows. Let x' be any other member of i. We describe what to do in the step where we are working on the ℓth trit, so there are $L - \ell$ trits left to fix. If the current trit of x and z match, say both are 0, then in the mutation phase, mutate the current trit of $y' = x \triangle x'$ to be 0, giving the result y. In the selection phase replace x' by y, whose ℓth trit matches the ℓth trit of both x and z. Now, in subsequent steps, symmetric complement preserves this matching trit.

If the ℓth trit of x and z differ then mutate the ℓth trit of $y' = x \triangle x'$ to be either a) matching the ℓth trit of z if $L - \ell$ is even or b) the symmetric complement of the ℓth trit of z and the ℓth trit of x if $L - \ell$ is odd. In the selection phase, again replace x'. On subsequent steps the only change to this ℓth trit will be from taking the symmetric complement with x. Making the choice based on whether $L - \ell$ is even or odd insures that on the last step this trit will match the corresponding trit of z. In each iteration, one additionial trit is modified in a way to guarantee that at the final step all will match those of z.

Notice that x is preserved in all these steps and that each step can occur with positive probability.

5 Annealing the Boltzmann GA

As noted in the Introduction, we employ a penalty method for treating infeasible solutions, which takes the form of attenuating their objective values by a multiplicative factor of $e^{-M(x)\beta_t}$. As run-time t increases, the fitness of infeasible solutions must be recalculated, but the simple form of the attenuation makes this easy to do.

Attenuating the fitness of infeasible solutions is not enough to assure that the limiting chain converges to an optimal solution only that it should converge to a feasible solution (as discussed in Section 2). Thus, in addition to attenuation, the fitnesses of both feasible and infeasible solutions alike will be "cooled". We take the fitness of the inhomogeneous chain to be

$$\phi_x(t) = \left(e^{-M(x)\beta_t} f(x)\right)^t.$$

Now we consider the limiting chain as $t \to \infty$. Recall the definitions of $\alpha(t)$ and $\omega(t)$ as the largest and second largest fitnesses of the augmented subpopulation $\{x_1, x_2, y\}$ where again y is the individual which has been newly generated.

Lemma 5.1

1. *If* $\Phi_i(t) > \max\{\phi_{x_1}(t), \phi_{x_2}(t), \phi_y(t)\}$*, then removal is equally likely.*
2. *If* $\Phi_i(t) < \max\{\phi_{x_1}(t), \phi_{x_2}(t), \phi_y(t)\}$*, then, as* $t \to \infty$*, either* x_1 *or* x_2 *is selected out equally likely.*

3. *If $\Phi_i(t) = \max\{\phi_{x_1}(t), \phi_{x_2}(t)\} = \phi_y(t)$, then removal is equally likely.*
4. *If $\Phi_i(t) = \max\{\phi_{x_1}(t), \phi_{x_2}(t)\} > \phi_y(t)$, and $\phi_{x_1} = \phi_{x_2}$ then removal is equally likely.*
5. *If $\Phi_i(t) = \max\{\phi_{x_1}(t), \phi_{x_2}(t)\} > \phi_y(t)$, and $\phi_{x_1} > \phi_{x_2}$ then, as $t \to \infty$, either x_2 or y is selected out equally likely.*

Proof: (1) is by definition. In case (2), $\alpha(t) = \phi_y(t)$ and $\omega(t) < \alpha(t)$. Put $h = \omega(t)/\alpha(t) < 1$, then $h \to 0$ as $t \to \infty$ and we have

$$\frac{\omega}{2\alpha + \omega} = \frac{\frac{\omega}{\alpha}}{2 + \frac{\omega}{\alpha}} = \frac{h}{2 + h} \to 0$$

as $t \to \infty$. Therefore asymptotically y will not be selected out. At the same time,

$$\frac{\alpha}{2\alpha + \omega} = \frac{1}{2 + h} \to \frac{1}{2}$$

as $t \to \infty$ so asymptotically x_1 and x_2 are selected out equally likely. In cases (3) and (4), $\alpha(t) = \omega(t)$ so removal is equally likely. Case (5) is argued like case (2) with the roles of y and x_1 interchanged. ∎

Theorem 5.1 *Let $p_{ij} = \lim_{t\to\infty} p_{ij}(t)$ be the stepwise limiting transition probabilities.*

1. *If $\Phi_j(t) < \Phi_i(t)$ then $p_{ij} = 0$.*
2. *If j can be proposed from i and $\Phi_j(t) > \Phi_i(t)$ then $p_{ij} = 1$.*
3. *If j can be proposed from i and $\Phi_j(t) = \Phi_i(t)$ then $p_{ij} > 0$.*

Let C be the collection of populations containing a global optimizer and $R = \Gamma \setminus C$ be the remaining populations. Then C is the unique closed irreducible ergodic set for the limiting chain and R is a transient set.

Proof: The transition assertions follow easily from Lemma 5.1. Now let i be any population and j a population containing a global optimizer. Let x and z be individuals of maximal fitness for i and j respectively. From the irreducibility arguments of the previous section we may proceed trit by trit to assemble the trits of z all the while maintaining x in the population. Hence we need not attempt a transition to a less fit population. If we encounter a transition to a more fit population during the process, we take it with certainty. Now replace x with the individual of maximal fitness in this new and more fit population and continue. ∎

Theorem 5.2 *For any manner in which $t \to \infty$, the stepwise stationary distribution $\pi(t)$ converges, $\pi = \lim_{t\to\infty} \pi(t)$. Moreover the overall transition matrix*

$$P(0,t) = \prod_{\tau=0}^{t} P(t) = [p_{ij}(0,t)]$$

converges and to a limit independent of the starting distribution,

$$p_{ij}(0,t) = Pr(X_t \in j | X_0 \in i) = \pi_j$$

where $\pi_j = \lim_{t\to\infty} \pi_j(t)$.

Proof: For each fixed t, by our assumptions $\pi(t)$ is the unique stationary distribution. Furthermore, $p_{ij}(t) \to p_{ij}$ and the limiting transition matrix has only one ergodic component C consisting of those populations which contain a feasible globally optimal state. The result then follows from Theorem 1.1 in [7], which covers this special case. ∎

6 Method II: adding an acceptance protocol

Our second method of ensuring the convergence of a GA to the set of optimal states is to add an *acceptance protocol* to the original GA, while using the original GA as a *proposal process*. The proposal process generates possible new states for the chain, while the acceptance protocol decides whether to move the chain to the proposed new state or remain with the current state. We present the algorithm as a maximization algorithm, but minimization is similar.

In this section, we assume that the original GA is time independent and generates a Markov chain which is ergodic. We denote by $G = (g_{ij})$ the transition matrix generated by the original GA. Let $a_n = 1/\log(n+1)$ for $n = 2, 3, \ldots$. Then for any $k > 0$ we have

$$\sum_n a_{nk}^k = \sum_n \frac{1}{(\log(nk+1))^k} = \infty.$$

First, we describe our algorithm for unconstrained optimization (that is, with no differentiation between feasible and infeasible states). Thus, we have the function $f : \Omega \to \mathbb{R}$ which we wish to maximize, and we let the fitness function ϕ of the algorithm be equal to f (in the case of a constrained optimization with a dynamic penalty, ϕ and f differ). Then each step in the modified GA is:

1. Starting from the current population i, use G to generate a new population j
2. If $\Phi_j > \Phi_i$, then accept the transition to the new population j
3. If $\Phi_j \leq \Phi_i$, then with probability a_n accept the transition to the new population and otherwise remain with the old population.

This generates a non-homogeneous Markov chain X_n on populations Γ. Let $\mathcal{G} \subset \Gamma$ denote the collection of populations which contain at least one globally optimal state.

Notice that it really doesn't matter how one assigns a fitness to a population (rather than an individual), as long as populations with optimal fitness contain optimal states. That is, one can assign the fitness of a population to be the maximum fitness of its individuals, the sum of the fitnesses of the individuals, or many other variations. This will change what constitutes $\mathcal{G}$, but not the fact that an optimal individual has been found.

Theorem 6.1 *$P(X_n \in \mathcal{G}) \to 1$ as $n \to \infty$.*

Proof: Because we assume that G is ergodic, there is some k so that it is possible to reach a state in $\mathcal{G}$ from any starting state in Ω in at most k steps. Furthermore, since $\sum_n a_{nk}^k = \infty$, by Theorem 2 in [8] the desired result holds. ∎

The condition that $\sum_n a_{nk}^k = \infty$ is used in Theorem 2 from [8] to show that the non-homogeneous Markov chain X_n is weakly ergdic. The fact that the transition probabilities always favor fitness-increasing transitions is then used to show that the chain has a limiting distribution which is fully supported on the optimal states of the chain, here the optimal populations.

Now we turn to the situation where the state space Ω contains both feasible and infeasible states. We denote (as before) by $\mathcal{F}$ the populations with only feasible individuals. For this situation, we let $M(x)$ be a measure of the infeasibility of individual x, so that $M(x) = 0$ means that x is feasible while $M(x) > 0$ means that x is infeasible. We define

$$\phi_x(t) = e^{-M(x)t} f(x)$$

and define for a population i

$$\Phi_i(t) = \sum_{x \in i} \phi_x(t).$$

Again we mention that this particular definition of the fitness of a population is not necessary, many other definitions would do just as well. We choose this one, rather than the one we have used previously in the paper given in equation (1), mainly to illustrate this point that many variations will work.

We initialize $t = 0$, $\beta_t = 1$ and our initial population i_0. Here β_t will be used to keep track of our attenuation of infeasible solutions. Using this, our algorithm is:

1. Starting from the current population i_t, use G to generate a proposed new population j.

2. If $\Phi_j(t) > \Phi_{i_t}(t)$, then accept the transition to the new population j so that $i_{t+1} = j$.

3. If $\Phi_j(t) \leq \Phi_{i_t}(t)$, then with probability a_t set $i_{t+1} = j$ and otherwise $i_{t+1} = i_t$.

4. If the best (so far) attenuated (attenuated at the level given by β_t) infeasible is still better than the best so far feasible, increment β_t, that is set $\beta_t = \beta_t + 1$.

5. Increment t.

At a fixed value of β_t, this is (basically) the same as the algorithm for unconstrained optimization. Now, clearly at the beginning of a run the value β_t will be changing with almost every iteration. As the simulation proceeds, however, the attenuation will remain fixed for long periods of time, only changing when a new (and very good) infeasible state is found.

By Theorem 6.1, fixing the value of β_t will result in a chain which converges to the optimal solution for THAT specific problem instance. Note that this optimal solution might depend on the value of β_t, as the "fitness landscape" given by $\Phi(t)$ depends on β_t. However, there is some t^* so that for any $\beta_t > t^*$, the optimal solution will be the desired globally optimal feasible solution.

By irreducibility, the process will eventually visit every infeasible state, so eventually the attenuation will reach a point that the best attenuated infeasible state is worse than some feasible state. At this point the attenuation will freeze and the chain will behave like the unconstrained algorithm. Since a_t decays sufficiently slowly, it doesn't matter at what iteration this occurs; convergence to the global optimal is assured. Thus we have the following theorem.

Theorem 6.2 *The attenuated algorithm discussed above generates a Markov chain which satisfies* $P(X_t \in \mathcal{G}) \to 1$ *as* $t \to \infty$*.*

7 Closing Comments

The idea of updating the attenuation only when the process encounters a new (and better) infeasible state can be incorporated in many stochastic algorithms for optimization. One such example is Compressed Annealing from [9]. Instead of having an incredibly slow compression schedule (where λ_k is required to grow slower than $O(\ln(k))$, see Section 3.3.1. in [9]), one could only increase the pressure when necessary.

References

[1] DAVIS, T.E. and PRINCIPE, J.C., *A Markov framework of the simple genetic algorithm*, Journal of Evolutionary Computing, Vol. 1, No. 3, pp. 269-288, 1993.

[2] Michalewicz, Z. and Fogel, D.B. *How to Solve It: Modern Heuristics.* Springer-Verlag, 2000.

[3] Michalewicz, Z. and Attia, N. *Evolutionary optimizaion of constrained problems*, Proceedings of the 3rd Annual Conference on Evoutionary Programming, 1994.

[4] Goldberg, D.E. *A note on Boltzmann tournament selection for genetic algorithms population-orientetd simulated annealing*, Complex Systems, 4 (4):445-460, 1990.

[5] Mahfoud,, Samir W. and Goldberg,, David E, Parallel recombinative simulated annealing: a genetic algorithm, *Parallel Comput.* Vol 21, Issue 1, 1-28, 1995.

[6] LÖWE, M., *On the Convergence of Genetic Algorithms*, Expostiones Mathematicae, Vol 14, pp. 289-312, 1996.

[7] GIDAS, B., *Nonstationary Markov chains and Convergence of the Annealing Algorithm*, Journal of Statistical Physics, Vol. 39, No 1/2, pp. 73-131, 1985.

[8] ANILY, S and FEDERGRUEN, A., *Simulated Annealing Methods with General Acceptance Probabilities*, Journal of Applied Probability, Vol. 24, No 3, pp. 657-667, 1987.

[9] OHLMANN, J., and BEAN, J. and HENDERSON, S., *Convergence in Probability of Compressed Annealing*, Mathematics of Operations Research, Vol. 29, Issue 4, pp. 837-860, 2004.