

# Annealing a Genetic Algorithm for Constrained Optimization<sup>1</sup>

F. Mendivil<sup>2</sup> <sup>3</sup> and R. Shonkwiler<sup>4</sup>

## Abstract

In this paper, we adapt a genetic algorithm for constrained optimization problems. We use a dynamic penalty approach along with some form of annealing, thus forcing the search to concentrate on feasible solutions as the algorithm progresses. We suggest two different general-purpose methods for guaranteeing convergence to a globally optimal (feasible) solution, neither of which makes any assumptions on the structure of the optimization problem. The former involves modifying the GA evolution operators to yield a Boltzmann-type distribution on populations. The latter incorporates a dynamic penalty along with a slow annealing of acceptance probabilities. We prove that, with probability one, both of these methods will converge to a globally optimal feasible state.

*Key Words:* Genetic algorithms, constrained optimization, simulated annealing, Markov chain, convergence proofs

## 1 Introduction

Let  $f$  be a real-valued function defined on the finite domain  $\Omega$ . By transformation, if necessary, we may assume without loss of generality that  $f : \Omega \rightarrow \mathbb{R}$  is positive, i.e.,  $f(x) > 0$  for each point  $x \in \Omega$ . Some (strict) subset of  $\Omega$  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  $\Omega$ . We distinguish between the function  $f$  to be optimized and the *fitness*  $\phi$  of the algorithm, which clearly should be taken to be a function of  $f$ . Our basic approach is to vary  $\phi$  dynamically as a function of “time”  $t$ , taken as a non-increasing function of the iteration count of the algorithm. Therefore  $\phi_x(t)$  is a function of both  $x \in \Omega$  and run time  $t = 1, 2, \dots$ . An alternate approach to ensuring convergence, which is taken by some authors, is to vary the mutation rate (for example, see [1], where they prove both uniqueness of a stationary distribution and asymptotic convergence to this stationary distribution based on a slow decay of mutation probabilities applied to a simple GA).

---

<sup>1</sup>The authors would like to thank the anonymous referee, who read the paper very carefully and made many suggestions, which substantially improved the quality of the final paper.

<sup>2</sup>Support for this work from the Natural Sciences and Engineering Research Council of Canada (NSERC).

<sup>3</sup>Professor, Department of Mathematics and Statistics, Acadia University, Wolfville, NS, B4P 2R6, CANADA

e-mail: franklin.mendivil@acadiau.ca, Corresponding author

<sup>4</sup>Professor, School of Mathematics, Georgia Institute of Technology, Atlanta, GA, 30332, USA

e-mail: shenk@math.gatech.edu

We view a genetic algorithm as a Markov chain defined on the set  $\Gamma$  of populations over  $\Omega$ . A population  $i$  of  $\Omega$  is a fixed sized multi-subset of  $\Omega$ , that is a subset of  $\Omega$  of a given cardinality, say  $K$ , possibly having repeated members,

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

By the *fitness*  $\Phi_i$  of population  $i$  we usually mean the maximal fitness of its members,

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

In some instances, another definition of the fitness of a population might be convenient. All we require is that populations with maximal fitness should contain at least one individual of maximal fitness. This is because the process uses population fitness; so, if the process finds a maximally fit population, we want this population to contain at least one optimal individual.

Our basic penalty mechanism is simple. Let  $M : \Omega \rightarrow [0, \infty)$  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  $M(x) \in \{0, 1\}$ , but this is certainly not necessary. Further, let  $\beta_t$  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

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

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

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

Matching analogous parts of the penalty in (2) with (3), we have  $M$  corresponding to  $\Delta E/(kC)$  and  $\beta_t$  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 populations 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 deleting 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 augmenting the algorithm with an annealing criterion, 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. However, as a general purpose algorithm our approach does not take any features of a given problem into account and thus in specific instances can easily be out-performed by special purpose algorithms. In particular, we make no use of problem features, such as smoothness, convexity or special geometric properties. It is exactly these features which allow many constraint-handling algorithms (such as those discussed in chapter 9 of [2]) to function efficiently. Our algorithms, in contrast, are useful in situations such as combinatorial optimization, where there might not be a natural way to “fix” an infeasible solution. In some of these problems just finding one feasible solution can be a challenge (such as the traveling salesman problem with time windows [4, 5]); so “fixing” an infeasible solution may be computationally challenging.

The problem of designing a population-based evolutionary algorithm, with a Boltzmann invariant distribution, has also previously been investigated (in particular, see [6, 7, 8]). In [7], 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 4.1 and 4.2, is close to their method and provides another technique to ensure reversibility (using *trits*). In [7], they view a GA as a modified type of SA and show convergence based on this observation. However, there is no discussion of constrained problems in that paper. We discuss the method in [8] further at the start of Section 4.1 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 feasible state.

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

Clearly the latter goal is a stronger condition than the former. We start with the former condition, as it is the simplest of the two to ensure. However, we point out that even the former 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, see [5]).

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

It is sufficient for a GA to be asserting and infeasible diminishing in order for the “solution” to be feasible. If the selection or removal phases of the GA depend on ratios of (attenuated) fitness, then the GA will most likely be infeasible diminishing. Many implementations 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 as  $n \rightarrow \infty$ ,  $P(X_n \in \mathcal{F}) \rightarrow 1$ .*

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

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

for any  $n$  and state (population)  $X_n$ . However, 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  $\Pr(X_{n+1} \notin \mathcal{F} | X_n \in \mathcal{F}) \rightarrow 0$  as  $n \rightarrow \infty$ . These two facts imply the result.  $\square$

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 we fix 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 continue the attenuation.

However, the GA may still not find an optimal solution even if it asymptotically concentrates on feasible states. 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 an optimal solution

As an example, take the state space  $\Omega = \{0, 1, \dots, 7\}$  with states  $1, 2, \dots, 6$  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 crossover is performed on the bit representations of both pairs to obtain two offspring each (the crossover 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 constitute the new population completing the iteration. We start the GA in the feasible population  $\{0, 0, 0, 0\}$ .

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.

One shows this by appropriately partitioning the space of populations  $\Gamma$  and considering a “lumped” Markov chain with the optimal individuals all in one lumped state  $G$ . It can then be shown that the probability that the process be in state  $G$  at some  $t \geq N$  given that it not be in state  $G$  at time  $N$  is bounded from above by the sum  $C \sum_{t \geq N} 1/t^2$ . This comes because it is necessary to make two transitions, each of asymptotic order  $1/t$ .

## 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 adds only

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 works only 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 of 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 than the first method, but it does not have a predictable sequence of invariant distributions.

## 4 Method I: an annealed Boltzmann GA

In this section, we present the first of our two methods for ensuring the convergence to an optimal state of a GA with constraints. In this first method, we modify the GA operators in such a way as to generate a Boltzmann distribution. GAs with a Boltzmann distribution have certainly been investigated previously ([6] is an early example). The novelty in this section is the application to constrained optimization along with an annealing, which does the double duty of ensuring both a feasible and an optimal limiting state.

### 4.1 Boltzmann transition rules

One method for assuring that the limiting distribution, as  $t \rightarrow \infty$ , has unit mass on the state of global optima consists in 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 approximates 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 function  $\phi(t)$ ; as we fix  $t$ , we notationally suppress it when convenient.

The *Boltzmann* distribution on the space  $\Gamma$  of populations is the one for

which the probability of observing population  $i$  is proportional to  $\Phi_i(t)$ :

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

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 control the stationary distribution is by using the property known as *detailed balance*. Given the transition probabilities  $p_{ij}(t)$ , if  $\Pi(t)$  is some probability distribution on the states which satisfies

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

then  $\Pi(t)$  is the stationary distribution. Moreover, if (4) holds, where  $\Pi_i(t) \geq 0$  but not necessarily a probability distribution, then a normalized version of  $\Pi(t)$  is the stationary distribution. The idea is to ensure this relation with  $\Pi$  as the Boltzmann distribution. The detailed balance equation (4) is also known as the *reversibility 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 achieving a Boltzmann distribution. For other alternatives, see [6, 8]. In particular, [8] has a very similar theme to our method (as described in this section and the next section). In [8], the author also uses Boltzmann-like selection probabilities to ensure the Boltzmann distribution be the stationary distribution. Each iteration also changes only one population member at a time and there is also an annealing which forces the process to limit to optimal states in the limit. However, it is not so clear as to when their reversibility condition  $\widehat{R}$  (on page 296 of [8]) is satisfied, since this condition requires a version of the chain to be reversible for some unknown probability distribution. 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 (as the scheme in the example GA on page 296 in [8] does) is generally not reversible. Introducing a mutation operator, which can create an arbitrary individual, has the potential to re-introduce reversibility; however, again it is difficult to determine if the resulting chain is reversible with respect to some unknown distribution.

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 [8] 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 (the *proposal* phase). This is followed by a roulette wheel selection, which could result in the new population being accepted or in no change to the population (the *selection* or *acceptance* phase). 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, the usual crossover mating and elimination might not be reversible. A simple example consists in 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  $\Omega$  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  $\Omega$  are real numbers simply by representing them in base three.

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

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

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

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

The binary operator  $\Delta$  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 perform 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 whether removing the given individual could decrease the fitness of the augmented population (note

that it could not increase the fitness). In particular, if it is impossible to decrease the fitness of the population no matter which individual (from the augmented sub-population) is removed, then we choose which individual to remove equally likely. On the other hand, if the population fitness could change, we remove individuals based on the possible fitnesses of the new population in comparison with the fitness of the current population.

Sort the numbers  $\phi_{x_1}, \phi_{x_2}, \dots, \phi_{x_K}, \phi_y$  and let  $\alpha$  be the largest and  $\omega$  be the next largest. Note that it is possible for  $\omega = \alpha$  if the two largest numbers are the same. We describe the transition probabilities based on a breakdown into cases.

**Case 1:** If  $\alpha = \omega$ , then we have  $\Phi_i = \alpha$  and the fitness of the new population will also be  $\alpha$ , independent of which individual is removed. In this case, we make the roulette wheel probabilities equal, that is, select which individual to remove from the augmented sub-population,  $\{x_1, x_2, y\}$ , equally likely, so with probability  $1/3$  each.

**Case 2:** If  $\alpha \neq \omega$  and  $\alpha \notin \{\phi_{x_1}, \phi_{x_2}, \phi_y\}$ , then  $\Phi_i = \alpha$  and the fitness of the new population will again also be  $\alpha$  no matter which element is removed. So, we again make the removal probabilities equally likely.

**Case 3:** If  $\alpha \neq \omega$  and  $\alpha \in \{\phi_{x_1}, \phi_{x_2}, \phi_y\}$ , then the fitness of the new population does depend on which individual is removed. In this case, either  $\Phi_i = \alpha$  or  $\Phi_i = \omega$  and similarly for the new population. Thus, we make the removal probabilities depend on which individual is to be removed and on  $\alpha$  and  $\omega$ . We define the roulette wheel probability for removing whichever point corresponds to  $\alpha$  (the most fit individual) as

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

and the roulette wheel probabilities for the other two are both set equal to

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

Note that in cases 1 and 2, the fitness of the new population is independent of which individual is removed from the sub-population and thus removal is equally likely. In case 3 this is no longer the situation, and thus we make the removal probabilities depend on both the fitnesses and on which individual is removed.

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,  $\alpha$ , and second,  $\omega$ , largest fitnesses of the augmented population must be determined, which is a very simple matter.

In some situations the probability  $p_{ij}(t)$  of a transition from population  $i$  to another population  $j$  is easily calculated. For instance, if  $i$  and  $j$  differ in more than one member, then  $p_{ij}(t) = 0$ . The other situation when  $p_{ij}(t) = 0$  is

also easily described. Suppose that  $i$  and  $j$  differ in exactly one member with  $y$  being unique to  $j$  and  $x_1$  being unique to  $i$ . For  $y$  to be generated, we must have that  $y$  differs in at most one trit from  $x_1 \Delta x_2$  for some  $x_2 \in i$ . If this is not true, then  $p_{ij}(t) = 0$  as there is no possibility for  $y$  to be generated from the population  $i$ . These are the situations in which  $p_{ij}(t) = 0$ .

The transition probabilities in the other situations are more difficult to describe. To explain this, again take  $y$  to be the member unique to  $j$  and  $x_1$  the one unique to  $i$ . Depending on  $i$  and  $y$ , there might be many choices of  $x_2$  which could generate  $y$  and thus would result in the population  $j$ ; so the transition from  $i$  to  $j$  involves a sum over all these possible choices. Similarly if  $i = j$ , then there are many possible ways this transition could occur, all of which involve generating some new element  $y$  and then removing it.

However, for our purposes it is not necessary to have an explicit form for  $p_{ij}(t)$  for all  $i$  and  $j$ . We will show that detailed balance holds by showing that it holds for each possible path from  $i$  to  $j$ , assuming that such a path does indeed exist.

**Theorem 4.1.** *Detailed balance, equation (4), holds for the tri-state transition scheme as described above (using  $\Delta$  as mating, one-trit mutation and the augmented population removal probabilities as in cases 1-3) with  $\Pi_i = \Phi_i$ . Thus a normalized version of the population fitnesses forms the stationary distribution.*

*Proof.* If  $p_{ij}(t) = 0$ , then  $p_{ji}(t) = 0$  as well and there is nothing to prove. If the old and new populations are the same, that is  $i = j$ , then this occurs when some new element  $y$  is generated and then removed. However, since the reverse path generates the same  $y$  and then removes it, the forward and reverse transitions along this path are the same and thus have exactly the same probabilities of occurring. Since this is true for every one of these paths and  $\Phi_i = \Phi_j$  we have that the result is true in the case  $i = j$ , that is

$$\Phi_i p_{ij} = \Phi_j p_{ji}.$$

Thus we assume that 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 the augmented population is  $a = \{y, x_1, x_2\}$ . To show detailed balance through  $a$  involves the consideration of several cases. If we fall into cases 1 or 2 (from above), then  $\Phi_j = \Phi_i$  and since the removal probabilities are  $1/3$ , it follows that the result is correct for these cases.

Before we continue with the remaining case 3, we introduce the constants  $c$  and  $\mu$ . Since we select the two parents equally likely from the population, the probability that any given pair be 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 with a specified component in a given value is

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

as there are  $L$  components each with three possible values.

Now for the remaining case 3, we have that only one member of the augmented sub-population  $\{x_1, x_2, y\}$  has fitness  $\alpha$  and no member of  $i \setminus \{x_1, x_2\}$  has fitness  $\alpha$ . We first consider the sub-case where  $\alpha = \phi_{x_1}$ . If  $y$  is removed, then  $j = i$  and detailed balance holds. If  $x_1$  is removed, then  $j = \{y, x_2, \dots, x_K\}$  and  $\Phi_j = \omega$  since  $\alpha$  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 removed 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 removed, then  $j = \{y, x_1, \dots, x_K\}$  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 removed on the reverse transition. Thus detailed balance holds if  $\alpha = \phi_{x_1}$ . Clearly the case  $\alpha = \phi_{x_2}$  is the same.

Finally, if  $\alpha = \phi_y$ , then either  $i = j$  (if  $y$  is removed) or  $i \neq j$  and  $\Phi_i = \omega$  and  $\Phi_j = \alpha$ . In the first situation, detailed balance holds. On the other hand, if  $y$  is not removed, then

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

since  $y$  is removed on the reverse transition and thus detailed balance holds.  $\square$

### Irreducibility

We consider the irreducibility of the chain for  $t < \infty$  here; the case for the limiting chain as  $t \rightarrow \infty$  is taken up below in Section 4.2. 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,  $i_0 = i, i_1, \dots, i_\ell = j$ , each of which contains  $x$ , such that  $p_{i_{k-1}, i_k}(t) > 0$ ,  $k = 1, \dots, \ell$ .

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. 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  $\ell$ th 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  $y' = x \Delta x'$  to be 0, giving the result  $y$ . In the selection phase replace  $x'$  with  $y$ , whose  $\ell$ th trit matches the  $\ell$ th trit of both  $x$  and  $z$ . Now, in subsequent steps, symmetric complement  $x \Delta y$  preserves this matching trit.

If the  $\ell$ th trit of  $x$  and  $z$  differ, then mutate the  $\ell$ th trit of  $y' = x \Delta x'$  to be either a) matching the  $\ell$ th trit of  $z$  if  $L - \ell$  is even or b) the symmetric complement of the  $\ell$ th trit of  $z$  and the  $\ell$ th trit of  $x$  if  $L - \ell$  is odd. In the selection phase, again replace  $x'$ . On subsequent steps the only change to this  $\ell$ th 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 additional trit is modified in a way to guarantee that at the final step all will match those of  $z$ .

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

## 4.2 Annealing the Boltzmann GA

Having crafted the transition rules to ensure a Boltzmann distribution for any fixed  $t$ , we now add an annealing criterion. 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  $t \rightarrow \infty$ . Recall the definitions of  $\alpha(t)$  and  $\omega(t)$  as the largest and second largest fitnesses of the augmented population  $\{x_1, x_2, \dots, x_K, y\}$ , where again  $y$  is the individual which has been newly generated by perturbing the mating of  $x_1$  and  $x_2$ .

### Lemma 4.1.

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  $t \rightarrow \infty$ , either  $x_1$  or  $x_2$  is removed 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  $t \rightarrow \infty$ , either  $x_2$  or  $y$  is removed with equal probability.

*Proof.* Case 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  $h \rightarrow 0$  as  $t \rightarrow \infty$  and we have

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

as  $t \rightarrow \infty$ . Therefore asymptotically  $y$  will not be removed. At the same time,

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

as  $t \rightarrow \infty$ ; so asymptotically  $x_1$  and  $x_2$  are removed 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.  $\square$

**Theorem 4.2.** Let  $p_{ij} = \lim_{t \rightarrow \infty} p_{ij}(t)$  be the stepwise limiting transition probabilities.

1. If  $\Phi_j(t) < \Phi_i(t)$  for all large enough  $t$ , then  $p_{ij} = 0$ .
2. If  $j$  can be proposed from  $i$  and  $\Phi_j(t) > \Phi_i(t)$  for all sufficiently large  $t$ , then  $p_{ij} > 0$ . Further, if the proposed  $j$  contains an individual  $y$  with  $\phi_y > \Phi_i$ , then in the limit the new population will contain  $y$  with certainty.
3. If  $j$  can be proposed from  $i$  and  $\Phi_j(t) = \Phi_i(t)$  for all sufficiently large  $t$ , 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 4.1, but we illustrate with the proof of the first one. So, suppose that  $\Phi_j(t) < \Phi_i(t)$  for all sufficiently large values of  $t$ . Then population  $i$  has a unique element  $x_1$  with  $\phi_{x_1} = \Phi_i$ . To transition to population  $j$  with  $\Phi_j(t) < \Phi_i(t)$ , this element  $x_1$  must be removed, so we are in part 5 of Lemma 4.1. That is, asymptotically  $x_1$  is not removed and so  $p_{ij} = 0$ .

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.  $\square$

**Theorem 4.3.** For any manner in which  $t \rightarrow \infty$ , the stepwise stationary distribution  $\pi(t)$  converges,  $\pi = \lim_{t \rightarrow \infty} \pi(t)$ . Moreover, the overall transition matrix

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

converges to a limit independent of the starting distribution,

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

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

*Proof.* For each fixed  $t$ , by our assumptions  $\pi(t)$  is the unique stationary distribution. Furthermore,  $p_{ij}(t) \rightarrow p_{ij}$  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 [9], which covers this special case.  $\square$

## 5 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 an algorithm for maximization.

In this section, we assume that the original GA be time independent and generates a Markov chain which is ergodic. Most GAs generate ergodic chains, as usually just the mutation alone will allow the chain to move from any state to any other state in a finite number of moves and this is sufficient for ergodicity. We denote by  $G = (g_{ij})$  the transition matrix generated by the original GA. Let  $a_n = 1/\log(n+1)$  for  $n = 1, 2, 3, \dots$ . 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  $f : \Omega \rightarrow \mathbb{R}$  which we wish to maximize, and we let the fitness function  $\phi$  of the algorithm be equal to  $f$  (in the case of a constrained optimization with a dynamic penalty,  $\phi$  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  $\Gamma$ . Let  $\mathcal{G} \subset \Gamma$  denote the collection of populations which contain at least one globally optimal state.

Note that it really does not matter how one assigns a fitness to a population (rather than an individual), as long as a population with optimal fitness contains at least one optimal state. 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 it means to be an optimal population, but not the fact that an optimal individual has been found.

**Theorem 5.1.**  $P(X_n \in \mathcal{G}) \rightarrow 1$  as  $n \rightarrow \infty$ .

*Proof.* Because we assume that  $G$  be ergodic, there is some  $k$  so that it is possible to reach a state in  $\mathcal{G}$  from any starting state in  $\Omega$  in at most  $k$  steps. Furthermore, since  $\sum_n a_{nk}^k = \infty$ , by Theorem 2 in [10] the desired result holds.  $\square$

The condition that  $\sum_n a_{nk}^k = \infty$  is used in Theorem 2 from [10] to show that the non-homogeneous Markov chain  $X_n$  is weakly ergodic. 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  $\Omega$  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  $\phi_x(t)$  again as in equation (2) 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, recalling that the only necessary property is that any population with maximal fitness should contain at least one optimal individual.

We initialize  $t = 0$ ,  $\beta_t = 1$  and our initial population  $i_0$ . Here  $\beta_t$  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  $\beta_t$ ) infeasible solution is still better than the best so far feasible solution, increment  $\beta_t$ , that is set  $\beta_{t+1} = \beta_t + 1$ . Otherwise, set  $\beta_{t+1} = \beta_t$ .
5. Increment  $t$ .

At a fixed value of  $\beta_t$ , this is (basically) the same as the algorithm for unconstrained optimization. Now, clearly at the beginning of a run the value  $\beta_t$  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 5.1, fixing the value of  $\beta_t$  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  $\beta_t$ , as the fitness landscape given by  $\Phi(t)$  depends on  $\beta_t$ . 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 situation in which 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 does not matter at what iteration this occurs; convergence to the global optimal is assured. Thus we have the following theorem.

**Theorem 5.2.** *The attenuated algorithm discussed above generates a Markov chain, which satisfies  $P(X_t \in \mathcal{G}) \rightarrow 1$  as  $t \rightarrow \infty$ .*

Note that in step 4, we only increment  $\beta_t$  by one. We could also increase  $\beta_t$  enough so that the best so far attenuated (at the new  $\beta_t$ ) infeasible would now be worse than the best so far feasible. This could possibly speed up the convergence of the algorithm as it would lead to the asymptotic value of  $\beta_t$  more quickly than simply incrementing by one each step. However, for a particular landscape, it might also complicate traversing the infeasible states and thus slow down the algorithm. It would be necessary to experiment with any particular optimization problem to find the best way to increase  $\beta_t$ . In the limit, the process will converge in either case.

## 6 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 example is compressed annealing from [11]. Instead of

having an incredibly slow compression schedule (where  $\lambda_k$  is required to grow slower than  $O(\ln(k))$ , see Section 3.3.1 in [11]), 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 Computation*, Vol. 1, No. 3, pp. 269-288, 1993.
2. Michalewicz, Z. and Fogel, D.B. *How to Solve It: Modern Heuristics*. Springer-Verlag, New York, New York, 2000.
3. Michalewicz, Z. and Attia, N. "Evolutionary optimization of constrained problems," Proceedings of the 3rd Annual Conference on Evolutionary Programming, pp. 98-108, 1994.
4. Ohlmann, J. Thomas and Barret W., "A Compressed-Annealing Heuristic for the Travelling Salesman Problem with Time Windows," *INFORMS Journal on Computing*, **19**, no. 1, pp. 80-90, 2007.
5. Savelsbergh, M., "Local search in routing problems with time windows," *Annals of Operations Research*, Vol 4, pp. 285-305, 1985.
6. Goldberg, D.E. "A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing," *Complex Systems*, Vol 4, Issue 4, pp. 445-460, 1990.
7. Mahfoud, Samir W. and Goldberg, David E., "Parallel recombinative simulated annealing: a genetic algorithm," *Parallel Computing* Vol 21, Issue 1, pp. 1-28, 1995.
8. Löwe, M., "On the Convergence of Genetic Algorithms," *Expositiones Mathematicae*, Vol 14, pp. 289-312, 1996.
9. Gidas, B., "Nonstationary Markov chains and Convergence of the Annealing Algorithm," *Journal of Statistical Physics*, Vol. 39, No 1/2, pp. 73-131, 1985.
10. Anily, S and Federgruen, A., "Simulated Annealing Methods with General Acceptance Probabilities," *Journal of Applied Probability*, Vol. 24, No 3, pp. 657-667, 1987.
11. 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.