



# Evolutionary Algorithms Applied to Landau-Gauge Fixing

J.F. Markham <sup>a</sup> T. D. Kieu <sup>b</sup>

<sup>a</sup> School of Physics, University of Melbourne, Vic 3052, Australia

<sup>b</sup> CSIRO MST, Private Bag 33, Clayton South MDC, Vic 3169, Australia

Current algorithms used to put a lattice gauge configuration into Landau gauge either suffer from the problem of critical slowing-down or involve an additional computational expense to overcome it. Evolutionary Algorithms (EAs), which have been widely applied to other global optimisation problems, may be of use in gauge fixing. Also, being global, they should not suffer from critical slowing-down as do local gradient based algorithms. We apply EA's and also a Steepest Descent (SD) based method to the problem of Landau Gauge Fixing and compare their performance.

## 1. Motivation

As SD is a local algorithm the time taken to converge goes as some power ( $> 1$ ) of  $V$  the lattice volume (critical slowing down) [1]. A solution to the problem is Fourier acceleration [2] but the cost of this goes as the cost of a FFT -  $V \log V$  at best. Once the gauge fixing condition has been met, Gribov copies may still exist. These can contribute to numerical errors in the measurement of various gauge dependent quantities, [3] [4] [5] [6] most notably propagators.

EA's on the other hand are global algorithms and should not suffer from critical slowing down. They are capable of finding a global maximum and so by suitable choice of constraint they can in principle eliminate Gribov copies.

## 2. Landau-gauge fixing on a lattice

For Landau-gauge we seek

$$\partial_\mu A^\mu = 0 \quad (1)$$

A sufficient (but not necessary) condition for this to be true is the maximisation of

$$F = \int d^4x A_\mu^g(x) A^{g\mu}(x) \quad (2)$$

where

$$A_\mu^g(x) = A_\mu(x) - \partial^\mu \chi(x) \quad (3)$$

Following [2] we do the following:

- Name the lattice version of  $\partial_\mu A^\mu$ ,  $\Theta$ , and the lattice version of  $F$ ,  $F_L$ .
- Implement SD.
- Compare the performance of SD with a competing algorithm in maximising  $F_L$ , using  $\Theta$  as a measure of convergence.

## 3. EA's applied to Landau-gauge fixing on a lattice

EAs solve global optimisation problems with natural selection and genetic style operations. The idea is to evolve successive generations of genomes with the goal of maximising their fitness [7] [8] [9]. We now describe the implementation of these operations to gauge fixing.

### 3.1. Coding

We encode the gauge transformations  $G(x)$  (*not* the gauge fields  $U(x)$ ). Each genome is made of group elements for each site of the lattice (1D is shown in Figure 1). Each group element is made of floats which represent the angles parameterising it.

- U(1) - just one angle.
- SU(2) - use the angles in O(3).

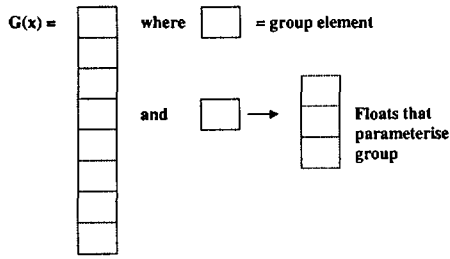


Figure 1. Encoding of groups

- $SU(3)$  factorise into 3  $SU(2)$  matrices acting on the three  $SU(2)$  subspaces in  $SU(3)$ .

### 3.2. Mutation

The floats are mutated by adding some normally distributed random numbers to them. The standard deviation of the distribution can vary according to the generation, genome and position within the genome.

### 3.3. Recombination

Cutting and splicing of the genome as used in Ref. [7] can be used. Best results are achieved by cutting on group boundaries and by having the genome duplicate the dimensionality and boundary conditions of the lattice. More successful methods are those which globally interpolate between genomes.

### 3.4. Fitness

The fitness of a genome is found by applying the  $G(\mathbf{x})$  that it represents to the original gauge fields and then evaluating  $F_L$  on the result as shown in Figure 2.

Genomes are selected from a population for breeding using algorithms which favour those with high fitness while at the same time maintaining diversity in the population. The last condition is to allow the population to escape local maxima.

## 4. Typical EA results

- The graphs that follow are not measures of flops.

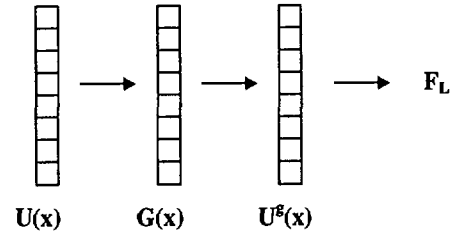


Figure 2. Evaluating the fitness of a gauge transformation

- One generation of EA takes more time than one iteration of SD.
- There is also the issue of memory. For an EA one must store many genomes, each of which has the size of a gauge transformation for the entire lattice.
- All lattices used are  $8^1$  with  $U(1)$  gauge fields with the exception of the last two figures which are  $16^1$  with  $U(1)$  gauge fields.

Figure 3 shows the statistics of a population as the algorithm progresses. The standard deviation of the fitness gives a measure of convergence (in addition to  $\Theta$ ). Figure 4 shows a direct comparison between SD and EA. On average SD performs better, and we have always found this to be the case.

The reason for this might be that there is no gradient information available to EA's. Close to convergence the genomes are doing a random walk (ie. without guidance) in the phase space of gauge transformations. Surprisingly, spectral decomposition of the gauged configurations suggests that the EA's are suffering from a similar problem that afflicts SD [2]. Examination of Figures 5 and 6 shows that the slow decay of low momenta modes in both algorithms is a problem.

## 5. Conclusion

In this preliminary study, our efforts have concentrated on improving the rate of convergence of EAs. This seems to go as some power of  $r/n$  where  $r$  is the distance from the actual maximum and  $n$  is the number of floats in a genome.

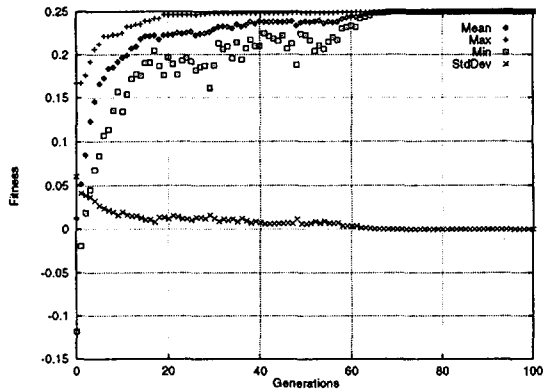


Figure 3. Fitness of Population vs Generations a typical plot for EA

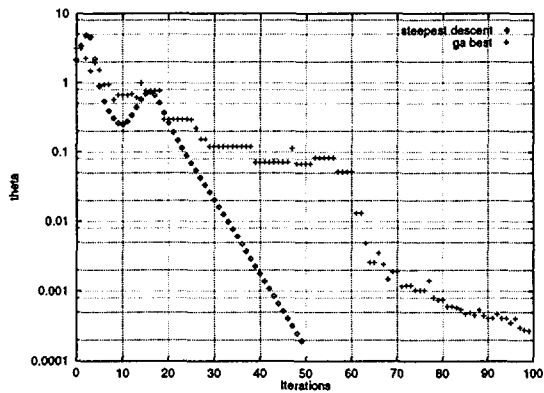


Figure 4.  $\Theta$  vs Generations - a comparison of the two methods for a typical run

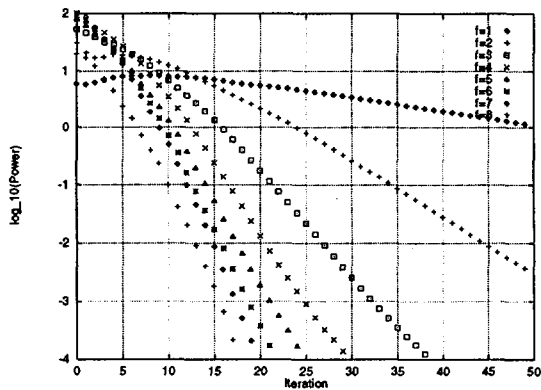


Figure 5. Power vs Iterations - Fourier components of successive gauged fields using SD

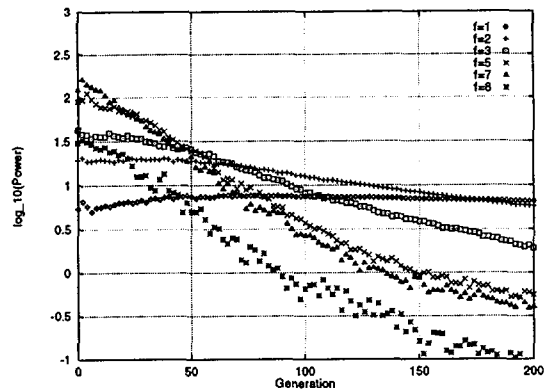


Figure 6. Power vs Generations - Fourier components of successive gauged fields using EA

Presently we are looking at their application to distinguishing between Gribov copies as in theory EAs should be able to find the global maximum effectively. Various hybrid methods combining EAs and local gradient methods – such as SD or its variations – are also being considered.

## REFERENCES

1. A. Cucchieri and T. Mendes *Nucl.Phys. B*471 (1996) 263-292
2. C. T. H. Davies et al *Phys. Rev. D*37, 1581 (1988)
3. A. Nakamura and M. Plewnia *Phys. Let. B* 255 2, (1991) 274
4. G. S. Bali, V. Boryakov, M. Müller-Preussker, F. Pahl *Nucl. Phys. B (Proc. Suppl.)* 42 (1995) 852-854
5. G. S. Bali, V. Boryakov, M. Müller-Preussker, K. Schilling *Phys.Rev. D*54 (1996) 2863-2875
6. A. Cucchieri *Nucl.Phys. B*508 (1997) 353-370
7. D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning* (Addison-Wesley Pub. Co., 1989).
8. T. Bäck, H. Schwefel *Evolutionary Computation* 1, 1-23 (1993)
9. D. B. Fogel *Evolutionary computation : toward a new philosophy of machine intelligence* (IEEE Press, 1995)