A brave new algorithm [⋆]

First Author$^{1[0000-1111-2222-3333]}$, Second Author$^{2,3[1111-2222-3333-4444]}$, and Third Author$^{3[2222--3333-4444-5555]}$

[1] Princeton University, Princeton NJ 08544, USA
[2] Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany
lncs@springer.com
http://www.springer.com/gp/computer-science/lncs
[3] ABC Institute, Rupert-Karls-University Heidelberg, Heidelberg, Germany
{abc,lncs}@uni-heidelberg.de

Abstract. At the beginning of this year one of the authors read ‘A brave new world‘ by Aldous Huxley. The Nobel describes a dystopia, which anticipates the development of breeding technology, and how this technology creates the perfect human race. Taking into account that when talking about genetic algorithms our goal is to achieve the optimum solution of a problem, and this book kind of describes the process for making the “perfect human”, we will try to work on this parallelism in this paper. The goal is to develop a Genetic algorithm based on the fecundation process of the book and compare it to other algorithms to see how it behaves. Investigating how the division in castes affects the diversity in the poblation.

Keywords: Evolutionary algorithm · Metaheuristics · Another keyword.

1 Introduction

At the beginning of the year one of the authors read the famous book by Aldous Huxley: A brave new world. The novel is a distopy that describes the development in reproductive technology, psychological manipulation and classical conditioning. The population is divided in *castes*, assigned since birth, where everyone knows and accepts their place. We are talking about a “optimum world“, whose optimization is based on the population and in the balance the division in castes creates, not in an individual.

When we talk about evolutionary algorithms the target is reaching the optimum solution for a problem., and this books perfectly describes the process through which they have reached the perfect human race. Thereforth we want to develop an algorithm based on the book’s fecundation proceses and compare its behaviour with other algorithms. Also, investigating how the division in castes affects the poblation’s diveristy.

[⋆] Supported by organization x.

2 Algorithm's Nature

As it was mentioned before, the algorithm is based in the optimization's process of the human race described on the book, thus we are talking about an algorithm based in the evolution of a population, it will follow the structure of evolutionary algorithms specifically a *genetic algorithm.* The book describes how they achieved the perfect human race working with an assembly line with different phases. This will be reflected with a *generational evolutionary algorithm* with selection, crossover, mutation and replacement operators.

The process begins in the ***Fecundation Room***, here the eggs are created and fertilized. Once the fertilization is finished all the eggs got to the ***Hatchery*** where the caste to which each individual will belong is decided. Huxley describes how the higher castes (*Alpha* and *Beta*) are suministred a higher amount of nutrients and hormones during the incubation. While the lower castes (*Gamma*, *Delta*, and *Epsilon*) are deprived of these elements, needed for the development. To imitate this "lack of nutrients", in the algorithm developed we will deprive the lower castes of the operators, they will only mutate. With all that has been mentioned, the castes will be developed in the following way :

- **Alphas**: in the books they are the most intelligents, the elite belongs to this group. They have responsibilities, they are ones that take decisions. In our implementation they will be reproduced with other individuals of the caste and they will evolve with all the operators.
- **Betas**: in the book they are less intelligents that the before mentioned and their main role is working in administrative tasks. In the implementation, the crossover will only be with individuals from the Alpha caste,
- **Gammas**: in the book they are subordinates, whose tasks require hability. In the implementation they will only mutate, but using local search
- **Deltas** and **Epsilons**: in the book both these castes are employees of the other castes and do repetitive works. In the implementation they will only have mutation by fixed segment.

With this structure in min the metaheuristic will be divided in the following phases :

- **Fecundation room**: the individuals are created in a randomized way.
- **Hatchery room**: in this phase we will divide the population in castes. We will do this following the fitness value of the individual as the criteria. Furthermore, each caste will have a different population percentage. Because in the book they mention that lower castes are produced with the *Bokanovsky's process*, where an ambryo its divided into 96 identical twins. In the algorithm this will be reflected in the population size, that will descende when the caste is higher.
- **Caste evolution**: each caste will follow a different process, as it was mentioned before

We are not talking about static castes, they are generated at the beginning of each generation. Let's imagine that we have a poplation size of ten, each individual with a fitness value. In the fist iteration the population will be divided following that value. After that, each individual will follow the evolution process corresponding to the caste. At the end of each generation all the chromosomes will be mixed, regardless of the caste. The next generation will start dividing this chromosomes in castes again.

3 Diversity analysis

Mantaining the *diversity* is crucial for avoiding the early convergence to local optima. Rosca [1] concluded that the poblations seem to get stuck in a local optima when the entropy didn't change or decreased drably in sequential generations. In genetic programming when we talk about diversity we refer to the structural differences such as the amount of different genotypes in the population or the singularity of the fitness values [2]. In this section we will analyze the diversity of our algorithm to study how division in castes affects.

There are different ways of calculating the diversity: genotypic diversity, fenotypic diversity, entropy, pseudo-isomorphism, edit distance, etc. Among which we chose *entropy*, that describes the distribution of the poblation around the different fitness values, and the *edit distance*, in which each individual it's evaluated against the best individual found so far. These metrics has been chosen since according to Burke [3] entropy and edit distance show a great correlation with the increment and decrement of the fitness value.

3.1 Diversity in Brave new algorithm

For these experiments we will use the configuration shown in Table 1. The fitness functions evaluated will be the Rastrigin function from the *Black-box optimization benchmarking* [4].

Table 1. Configuration parameters for the diversity exploration

Configuration parameters	Value
Chromosome size	20
Population size	100
Maximum generations	15
Alpha population percentage	10
Beta population percentage	20
Gamma population percentage	20
Delta population percentage	20
Epsilon population percentage	30
Mutation rate for all castes	10

In Figure 1 the data has been taken from the Rastrigin execution that returned the best fitness value. In the graphic on top we have the fitness value that resulted from each generation. As we can see the edit distance metric is closely related to the top graphic. The smallest edit distance is in the last generation, so we can conclude that the lower the edit distance the higher possibility of the algorithm to get stuck in a local optima.

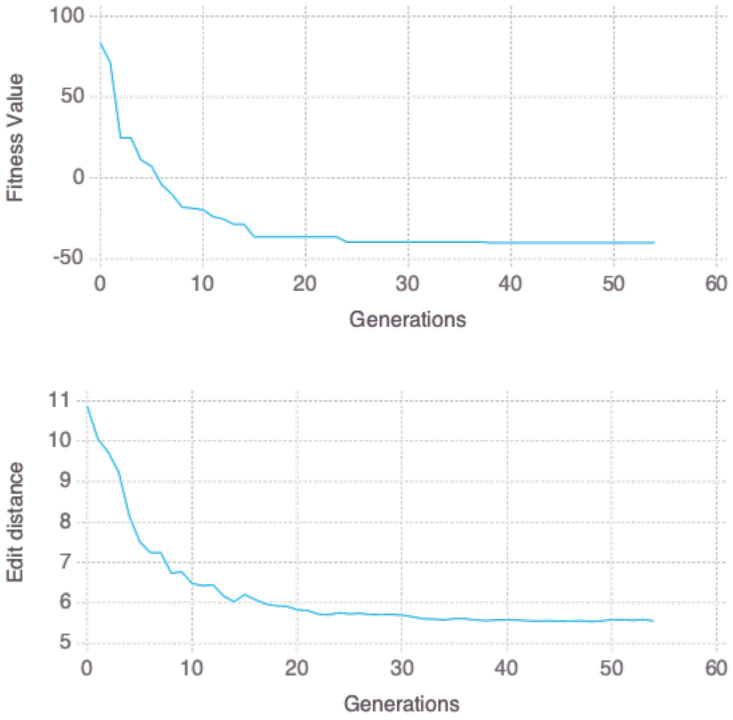
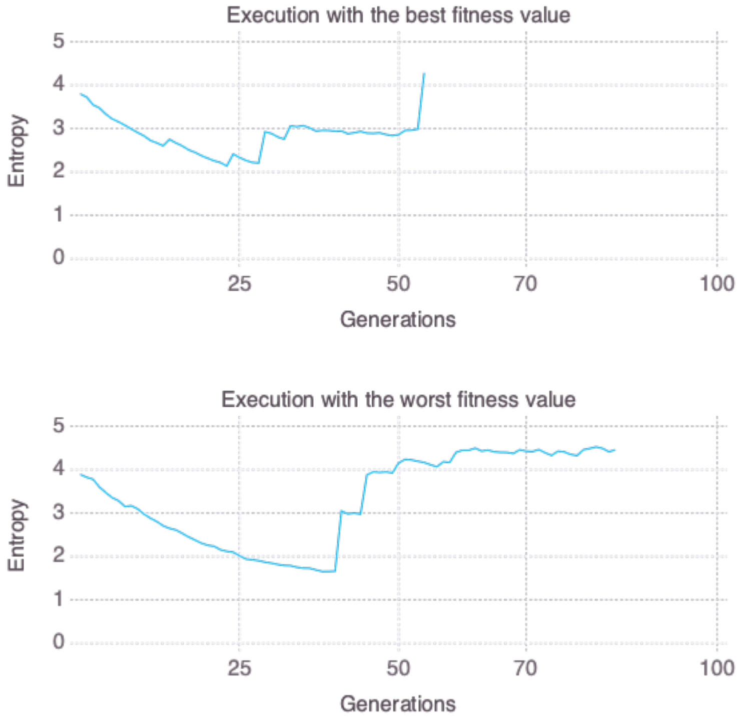


Fig. 1. Diversity in the execution with the best fitness value for the Rastrigin function

Now let's compare the entropy for the best and worst execution of the Rastrigin function to see if the entropy has anything to do with the algorithm outcome. As we know, algorithms seem to get stuck in local optima when entropy doesn't change or decrease drably in sequential generations. In Figure 2 we can see this case in the execution with the worst fitness value. The entropy decreases more thatn 2 points from generation 0 to generation 30. Whilst in the case of the best execution the entropy slope is less strong. Also in the execution with the best fitness we can see how when the entropy stays with similar values for multiple generations is when it gets stucked in local optima.


Execution with the best fitness value

Execution with the worst fitness value

Fig. 2. Comparation of the diversity between the execution with the best and the worst fitness value for the Rastrigin function

With this information we can now compare the diversity of our algorithm with the behaviour of a genetic algorithm without castes division.

References

1. J. P. Rosca: Entropy-driven adaptive representation. Proc. Workshop Genetic Programming: From Theory to Real-World Applications, 23–32 (1995)
2. J. R. McDonnell: Genetic programming exploratory power and the discovery of functions, 719–736 (1995)
3. Burke, Edmund and Gustafson, Steven and Kendall, Graham: Diversity in Genetic Programming: An Analysis of Measures and Correlation With Fitness. Evolutionary Computation, IEEE Transactions on **8**, 47–62 (2004)
4. Steffen Finck and Nikolaus Hansen† and Raymond Ros‡ and Anne Auger§: Real-Parameter Black-Box Optimization Benchmarking 2010: Presentation of the Noiseless Functions, (2014)
5. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
6. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
7. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
8. LNCS Homepage, http://www.springer.com/lncs. Last accessed 4 Oct 2017