Evolutionary Algorithms

final project paper

Temple University

Professor: Dr. Pei Wang

Student: Guilherme Mendes Marques de Oliveira

- **Introduction**

This work aims to bring together knowledge along with discussions in the Evolutionary Algorithms topic of Artificial Intelligence, but in addition, it will also bring any related subjects such as biology and mathematics to contribute to a more complete discussion of the main topic. The purpose of this paper is also to only focus on a theoretical approach to the studied problem.

In nature there are several processes that occur naturally and owns different degrees of complexity. This paper will focus on the biology of populations and how is that natural phenom relates to the mathematics of algorithms achieving the point to solve real world problems.

Several nature processes carry loads of different important characteristics which allows the system that is the universe to work, along with the life in it to maintain it self and propagate it's genes to the next generations. There is a complex process that happens in the background to allow populations to strive in the harsh conditions provided by the environment, which opens the possibility for us to try understand it and, eventually, apply the same ideas since they're somehow useful.

- **Main**

  - **A biology of populations overview**

In the biology studies, there is a subset known as evolutionary biology, which studies the phenoms that produces such a broad diversity of lives on earth. However, for the scope of this work we are more interested in, yet, a deeper level inside those studies, in which we could highlight cell biology, population biology, experimental evolution and evolutionary theory.

A cell as independent microorganism has it's own properties which define it as an individual different from others inside any other population: the genetic material which forms its DNA. The DNA is composed of a very detailed description of the individual, defining all characteristics from that individual and also making it unique among other individuals. Cells also have other important abilities which such as ways of breathing, feeding itself, reproducing and so on. However, some cells might have different aptitudes for each of those basic life tasks, allowing it to be better or worse at something which might change its ability to live. As an example, a cell which has problems for digesting the nutrients it eats is doomed to die, which, as explained later, wouldn't be considered as a “fit” (by the fitness function) for that environment. Going further into this, a population of cells is inserted into some environment which requests of it different abilities to be able to survive due to the hostile conditions of that habitat such as enemies (as a virus), competition between the same population for space, light and food; and so on. This leads life to be a more complex situation in which an individual need, in order to survive, be able to successfully accomplish different tasks, therefore, it must have a natural ability for accomplishing such doings which will be determined by it's DNA.

Still in the populations problem, each individual can reproduce, before perishing, by combining its DNA with another cell, producing a new individual which has characteristics from both parents but also, at the same time, has some randomness factor involved in it's DNA, making it not as simple as a combination 2 different pieces of DNA. This randomness factor, in biology is called crossing-over, and it explains why two human brothers do not look exactly the same (unless they are perfect twins, which is a different case). The reproduction process is repeated for the maintenance of the population or even

for the growth of the population.

Each new generation is based on all the past generations in terms of DNA. According to the evolutionary theory, there will be a process of natural selection for every population, allowing only the "*survival of the fittest*" individuals, the "*[...] preservation of favored races in the struggle for life [...]*", both chunks of phrase wrote by the english biologist *Herbert Spencer* (1820 – 1903) in his book *Principles of Biology* (1864).

Evolutionary algorithms is a class of population-based algorithms, which means, in other words, algorithms which simulates biological organisms evolution by the creation of a generation which is subject to a fitness function which filters those elements from the population. The ones that survive are used as base for the next generation and the process is repeated until some fitness level is reached or until the limit number of generations was reached.

- **A evolutionary algorithms (EA) overview**

Evolutionary algorithms are population based optimization algorithms. That means, algorithms that simulates life's behavior justifying why the biological populations overview is important. There are concepts taken from that natural process and applied on EA.

To start understanding how EA works, imagine a starting population with a certain number of individuals (such as a 100, as an example). There is a function which calculates the fitness for each individual to be able to "survive that environment", to keep the biology analogy. Mathematically speaking, the fitness calculating is a way of producing a score of how "good" or "bad" the solution that individual carries within it's DNA. The DNA is a piece of information which every individual has, making it unique. Such piece of information is actually part of a possible solution to solve the problem this technique is being applied on. Getting back to fitness function, it then calculates a score of how close or how far a piece of solution contained on one individual's DNA is to the solution we are looking for. Of course, we don't have the desired solution before finishing the algorithm, that's actually the all point of running it, therefore, the fitness function must somehow be able to know what is a solution, but that depends specifically on the problem. As an example, if we consider a the problem of the traveler salesman, in which we are looking for a path on the graph which crosses all vertices in any order, without crossing the same vertex twice and also using the shortest possible path. The fitness function wouldn't know what is the solution for this problem, but it does know some properties of it, such as: a solution crosses all vertices without crossing a vertex twice. So, if we have a solution that matches this condition, then it is a valid solution but another solution that fills out the same pre-conditions but also have a shorter total path length should override the first found solution. This algorithm will not be used to get the optimum solution (it might give us the optimum, but it is not guaranteed), it is used for optimizations. The fitness function is then, a heuristic function that tries to predict how good that piece of solution is.

There is then a selection process in which the fitness function will "filter" those individuals which carry in their DNAs pieces of solutions that aren't good enough, eliminating them. This process is called selection due to the biology term *natural selection* which states *the survival of the fittest*. At this point there are two possibilities for the survivors, which will then either go over a phase of reproduction, recombination and mutation or over a phase of elitism. The first one, is the one in which two parents combine the solution they have, creating a new individual which is reproduction, the child individual will then have its new DNA assimilated by the combination of the two DNA pieces it receives from the parents and finally it will go over a mutation, which adds a little bit of randomness to

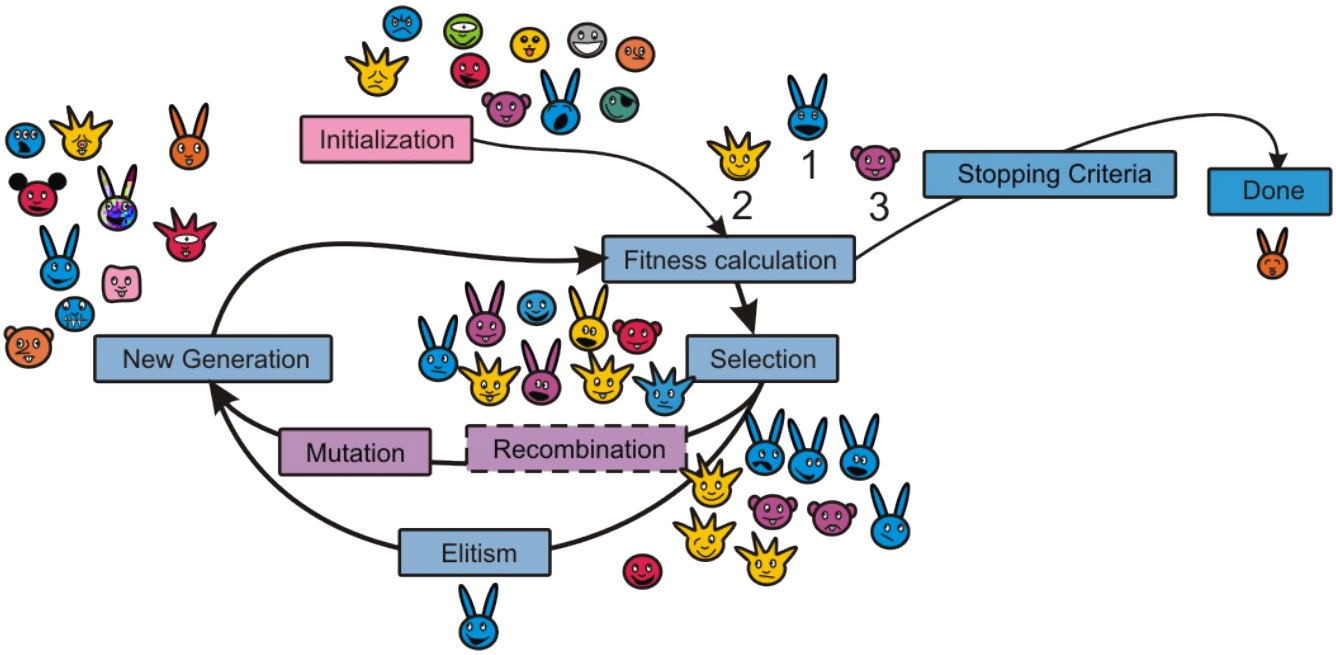
the child's DNA.

The second possibility is the process known as elitism where individuals do not reproduce, because their DNA is considered good enough and that individual, which was a potential parent, is then carried over to the next generation along with the child individuals born from a reproduction process. But that depends on the implementation, elitist individuals might reproduce as well.

The new generation is only composed of the recently created children and the elitist individuals while the rest of the individuals died, however, it might change depending on the implementation. An argument for why they should die, is that they are only composed of smaller pieces of solution that were already taken into consideration, while any small and really good piece was saved by the elitism process.

From this point and on the cycle repeats from the fitness calculation and on, until the stopping criteria is eventually reached. The stopping criteria, again, depends on the specifics of the problem. If we still consider the last analyzed case of the traveler salesman problem, we could say that the stop criteria would include that every vertex was crossed (by default not twice or more each). We could add on top of that another criteria that specifies the maximum desired path length and the algorithm would only finish upon reaching a total length equals or lower than the goal. Of course that more strict values would increase the needed number of iterations the algorithm take to find a solution matching those stopping criteria.

Below there is an image that helps understand the just explained process:



- **EA implementation discussion**

Personally I believe implementations should allow elitist individuals to reproduce since they have high scored pieces of solutions and they should be added to other existing solutions but still remain available on their own. It is natural to think that if we have a larger number of individuals with a certain characteristic among the population, it becomes more likely that the final solution will present

the same characteristic as well.

Also past generations I believe it's better if they die indeed, allowing space for the new generation to compose the starting scenario of the next cycle, since it's likely that they would compose a high percentage of low profit individuals. This means that the solution pieces they have might be limited and not be part of the final solution at all, mostly for huge problems. In addition, adding so many generations together will increase significantly the final computational costs, which should be considered for huge problems.

- **Inner topics: Genetic Algorithms (GA)**

Genetic algorithms is a inner topic of evolutionary algorithms they are, however, still very similar. GA are based on techniques of inheritance, mutation, selection and crossing-over (a.k.a. recombination). The process is pretty much the same described in the above EA overview topic, but the image shown by that topic's end describes a GA process. The reason why this topic is just a piece of the whole EA is due to it's specifics. EA is just a class of algorithms based on any natural populations self-development, while GA's are the true populations DNA merge and recombination for the population's natural maintenance process. The classic works in this topic were made by John Holland.

- **Inner topics: Evolution Strategy (ES) overview**

This topic main idea is focused on adaptation and evolution inside the context of populations from EA. The process of creation of new generations remains the same as in GA, the life cycle of the population, however, changes. Such process now has the arisen of new individuals as a result from evolution only, instead of the entire reproduction, recombination, mutation and selection as before. The main idea remains simple, nonetheless, there are some complications in a more deep level of this field, but it's not covered on the scope of this overview.

- **Inner topics: Differential Evolution (DE) overview**

This is another sub topic inside EA which is focused on solving mathematical functions optimization problems. As expected, it does use evolutionary population based techniques to find answers in a different way than any other numerical method for differential equations: it doesn't require functions to be differentiable. A numerical method example is the quasi-newton one. DE, however, doesn't require a gradient vector as the mentioned method, justifying why it can solve problems for non differentiable equations. This topic, however, is not going to be covered in deep detail.

This method performs a space search with the agents of a population using mathematical methods to compare them with each case, until new positions are found for those individuals which satisfies a certain quality condition, indicating an improvement. Let *f* be a function which domain is defined as being any the real number of any dimension *d* equals or greater to 1. The DE method takes a candidate solution as an argument being a vector in which each position is a real coefficient number, then it produces a score for that number. The goal is to find a solution s_1 in which $f(s_1) \leq f(x)$ for all x in the search space to find the global minimum and find an solution s_2 in which

$f(s_2) \geq f(x)$ for all x to find the global maximum.

- **My own opinion**

At a first look evolutionary algorithms might seem a technique only useful for biology problems with populations, where a recursive DNA recombination of two existing individuals can create a new one which is more likely to contain a finest DNA, due to the natural selection. This picture must however, be abstracted more broadly for not only biology in which we have individuals and DNAs but also a more theoretical abstraction in mathematics.

In mathematics this technique could be understood as an optimization problem in which the parents have part of the solution each, but when combined they can create a solution possibly better which involves both of the previous pieces but, yet, is not limited to it with the crossing-over process. As an example, in a graph in which we are interested on finding the shortest path, we have two individuals where the first contains the information that the shortest path should cross vertices A and C, but the second individual might only has the information it should cross vertex B. By combining them, one possible result we would get is a child with the information that the shortest path over that graph should cross vertices A, B and C.

We can abstract the evolutionary algorithms to a class of problems of both recursive and combinatory tasks, in which the trick consists of recursively creating new combinations based on previews results. Keeping that in mind, I've come to realize how close this subject is to the self repeating patterns called fractals. While fractals still are totally self repeating patterns, due to the similarities with EA, there might be possible to see them with a fixed dimension as EAs with a conservative DNA recombination, in which only the scale changes. Or also the opposite, EAs as fractals of two dimensions with non-constant pattern. More abstractly, the dimension level could be how much the population grows, leading us to think as the 2^{nd}-dimmension as a population in which every pair of parents have 4 children (assuming parents doesn't die), making the population raise by the factor of 2^{n-1}. Different dimensions, such as 1.25 (which for fractals, is possible, even though we can't picture it in real life, also, 1.25 is the dimension number for the natural self repeating pattern we see on rivers) we would have a slow increase of the population, which I believe it to be $(1.25)^{n-1}$ for this case. Both those functions presented considers the last generation as the only survivor and a starting population size of 1. The number 'n' is the desired number of the generations to be considered.

- **Related Topics**

Since this is a paper on the EA field, it's hard to say what are the related works, since it could be anything written that brings EA to a discussion. However, we might think about similar topics to the one on focus of this work, such as swarm algorithms.

Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, natural or artificial. The inspiration for creating algorithms like this comes, again, from nature, especially biology, just like EA. SI is composed of individuals that form a population, but this time, they are rather focused on interacting with other individuals from from the system it's in or with the environment it self. Each of those individuals follow simple rules or tasks, composing, one by one, the swarm system that, as a whole, behaves intelligently. As examples, we have algorithms in the SI topic such as the ant colony optimization, Bees algorithm and the particle swarm optimization.

- **Conclusion**

EA is a huge topic and sadly I couldn't cover it all within the time I had for this project, however, I still believe I've bring in a pretty good amount of information that was useful or interesting, somehow. The scope of this work only cover some of the topics I found out to be more important and at the same time more interesting, excluding genetic programming, topic which is not included here but I've read about and thought it could contribute to this paper.

There are plenty of different applications for EA, mainly when it comes about using an heuristic to produce good solutions in quite a short time length for a generic given problem. I have to say, of course, that EA is not the best technique for any kind of problem, since there are many different algorithms that were created specifically for their respective problems and, therefore, they should find better solutions in shorter time frames. By another hand, EA positions it self as a great generic technique, mostly for the cases in which there are no specific algorithms to solve the desired problems that could do the job any faster. As an example, there are plenty of graph related problems that belongs to classes of problems such as NP-complete or NP-hard where there is no “fast” algorithms for obtaining the optimum solution, they are all heuristic based, consequently, EA adds on top of that as a different heuristic approach to find optimizations to such problems.

- **References**

http://en.wikipedia.org/wiki/Evolutionary_algorithm

http://en.wikipedia.org/wiki/Genetic_algorithm

http://en.wikipedia.org/wiki/Evolution

http://en.wikipedia.org/wiki/Natural_selection

http://simple.wikipedia.org/wiki/Evolutionary_algorithm

http://en.wikipedia.org/wiki/Evolutionary_biology

http://en.wikipedia.org/wiki/Experimental_evolution

http://en.wikipedia.org/wiki/Survival_of_the_fittest

http://en.wikipedia.org/wiki/Fractal

http://en.wikipedia.org/wiki/Differential_evolution

http://mathworld.wolfram.com/DifferentialEvolution.html

http://rosettacode.org/wiki/Evolutionary_algorithm

http://en.wikipedia.org/wiki/Swarm_intelligence