Evolution Strategy Programming (ESP)

EBADA A. SARHAN IRAKY H.KHALIFA
Faculty of Computers & Information
Helwan University

MOHAMED S. EMAM
Faculty of Computers & Information
Zagazig University

Present by: Mohamed S. Emam

Abstract: *Evolutionary Algorithms are search algorithms based on the Darwinian metaphor of "Natural Selection". Typically these algorithms maintain a population of individual solutions, each of which has a fitness attached to it, which in some way reflects the quality of the solution. The search proceeds via the iterative generation, evaluation and possible. This paper presents a new Self-Adaptive Evolutionary Algorithms technique called Evolution Strategy Programming (ESP) which is a combination of Evolution Strategy (ES) and Evolutionary Programming (EP). Evolutionary Algorithms rely on two genetic operators — Crossover and Mutation, in case of real parameter representation (ES & EP), experiments show that mutation is more powerful than crossover so we concentrate our work on mutation. ESP performs mutation process by the same method of Evolution Strategies but with extra rule called random adaptation rule. We perform in this paper a comparison between standard (μ,λ)-Evolution Strategy and Evolution Strategy Programming (ESP) on a highly multimodal function (Function after Fletcher and Powell).*

Key-Words: *Evolutionary-Algorithms, Evolution-Strategy, Evolutionary-Programming, Evolution-Strategy-Programming,, Mutation, Random-Adaptation.*

1. Introduction

As the history of the field suggests there are many different variants of Evolutionary Algorithms. The common underlying idea behind all these techniques is the same: given a population of individuals the environmental pressure causes natural selection (survival of the fittest) and this causes a rise in the fitness of the population. Given a quality function to be maximized we can randomly create a set of candidate solutions, i.e., elements of the function's domain, and apply the quality function as an abstract fitness measure — the higher the better. Based on this fitness, some of the better candidates are chosen to seed the next generation by applying recombination and/or mutation to them. Recombination is an operator applied to two or more selected candidates (the so-called parents) and results one or more new candidates (the children). Mutation is applied to one candidate and results in one new candidate. Executing recombination and mutation leads to a set of new candidates (the offspring) that compete — based on their fitness (and possibly age) — with the old ones for a place in the next generation. This process can be iterated until a candidate with sufficient quality (a solution) is found or a previously set computational limit is reached [2].

In this process there are two fundamental forces that form the basis of evolutionary systems.

- Variation operators (recombination and mutation) create the necessary diversity and thereby facilitate novelty, while
- selection acts as a force pushing quality.

The combined application of variation and selection generally leads to improving fitness values in consecutive populations. It is easy (although somewhat misleading) to see such a process as if the evolution is optimizing, or at least "approximizing", by approaching optimal values closer and closer over its course. Alternatively, evolution it is often seen as a process of adaptation. From this perspective, the fitness is not seen as an objective function to be optimized, but as an expression of environmental requirements. Matching these requirements more closely implies an increased viability, reflected in a higher number of offspring. The evolutionary process makes the population adapt to the environment better and better [3].

Let us note that many components of such an evolutionary process are stochastic. During selection fitter individuals have a higher chance to be selected than less fit ones, but typically even the weak individuals have a chance to become a parent or to survive. For recombination of individuals the choice of which pieces will be recombined is random. Similarly for mutation, the pieces that will be mutated within a candidate solution, and the new pieces replacing them, are chosen randomly. The general scheme of an Evolutionary Algorithm can is given in Figure 1 in a pseudo-code fashion; Figure 2 shows a diagram.

```
BEGIN
   INITIALISE population with random
   candidate solutions;
   EVALUATE each candidate;

   REPEAT
       1 SELECT parents;
       2 RECOMBINE pairs of parents;
       3 MUTATE the resulting offspring;
       4 EVALUATE new candidates;
       5 SELECT individuals for the next
         generation;

  UNTIL ( TERMINATION CONDITION is
             satisfied )
END
```

Fig. 1 The general scheme of an Evolutionary Algorithm in pseudo-code

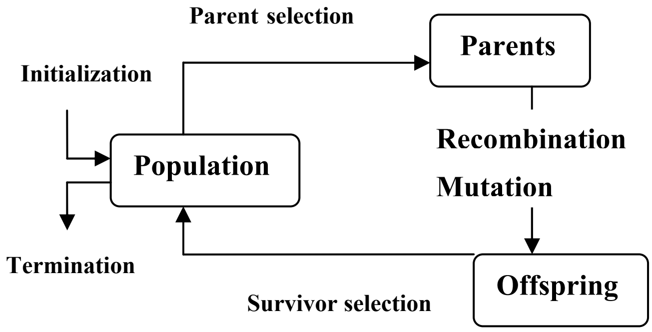

Fig. 2 The general scheme of an Evolutionary Algorithm as a flow-chart

2. Evolution Strategy

Evolution strategies are similar to genetic algorithms in that both attempt to find a (near-)optimal solution to a problem within a search space (all possible solutions to a problem) without exhaustively testing all solutions. While evolution strategies are a joint development of Rechenberg, Biernert, and Schwefel, who did preliminary work in this area in the 1960s at the Technical University of Berlin (TUB) in Germany.

Evolution strategies tend to be used for empirical experiments that are difficult to model mathematically. In this case, the system to be optimized is actually constructed. Evolution strategies are based on the principal of strong causality, which states that similar causes have similar effects. That is, a slight change to one encoding of a problem only slightly changes its optimality [5].

The $(\mu+\lambda)$-ES and (μ,λ)-ES

The $(\mu+\lambda)$-ES and (μ,λ)-ES were introduced by Schwefel, as we mentioned above the $(\mu+\lambda)$-ES is a natural extension of a multimembered evolution strategy $(\mu+1)$-ES, where μ individuals produce λ offspring. The new (temporary) population of $(\mu+\lambda)$ individuals is reduced by selection process again to μ individuals. On the other hand, in the (μ,λ)-ES the μ individuals produce λ offspring (consequently, λ>μ is necessary) and the selection process selects a new population of μ individuals from the set of λ offspring only [5].

Currently, the (μ,λ)-ES characterizes the state-of-the art in Evolution Strategy research and is therefore the strategy that I used in my study. As an introductory remark it should be noted that the major quality of this strategy is seen in its ability to incorporate the most important parameters of the strategy (standard deviations and correlation coefficients of normally distributed mutations) into the search process, such that optimization not only takes place on object variables, but also on strategy parameters [4].

3. Evolutionary Programming

The original Evolutionary Programming method used uniform random mutations on discrete underlying alphabets and, in its most elaborated form, a (μ+μ)-selection mechanism. Following on form the initial work of L. J. Fogel [Fog62, FOW66] this approach remained greatly underused for approximately thirty years.

Then, in the late 1980s D. B. Fogel (his son) extended Evolutionary Programming for applications involving continuous parameter optimization problems. Evolutionary Programming for continuous parameter optimization has many similarities with Evolution strategies: mutations are normally distributed and, what is more interesting, the more elaborated versions of Evolutionary Programming incorporate variances of mutations into the genotype (meta-EP), thus facilitating the self-adaptation of these parameters [3].

4. Evolution Strategy Programming

ESP is a new self-Adaptive Evolutionary Algorithms technique that combines the advantages of Evolution Strategy and Evolutionary Programming. In Evolution Strategies rely we have two methods plus and comma (μ+λ & μ,λ), the advantage of plus method is in the elitist process to preserve the best individual during the evolution process since the best μ individuals out of the union of parents and offspring survive, but each individual consists of two parameters, object parameter and strategy parameter so as we preserve the object parameters we also preserve the strategy parameters which is not useful for the self-adaptation during the evolution process. Comma method perform the inverse process of plus method, here best individuals may be lost since the best μ offspring individuals form the next parent generation (consequently, λ > μ is necessary), so the process of self-adaptation is gone well.

In Evolution Strategy Programming we use the (μ+μ) method to preserve the best individuals during the evolution process and in the same time we adapt the strategy parameters by making refresh to strategy part of all the individuals “that if the evolution process stop at stationary fixed points for certain times of generations” by assign a new random values (from intervals smaller than that in the beginning) to the strategy parameters of all the individuals in the population.

4.1 ESP Process

The process of Evolution Strategy Programming is the same as Evolution Strategies (ES) and Evolutionary Programming (EP) process. The mutation calculations were taken from (ES), selection was taken from (μ+λ)-Es, and there is no recombination as in (EP). The remaining operations are essentially the same for both methods. The extra rule added here is the random adaptation rule that occurs within the mutation method during the evolution process.

Each population member of the ESP was composed of two *n*-dimensional vectors. The first was the population vector $\vec{x} \in \Re^n$ and the second was a corresponding standard deviation vector $\vec{\sigma} \in \Re^n$ used in mutation. Thus each member a was constructed as $\vec{a} = (\vec{x}, \vec{\sigma})$. The population at any given time step of the interval is denoted as $P(t)$.

1) An initial population $P(0)$ of μ members is generated consisting of $\vec{a}_i = (\vec{x}_i, \vec{\sigma}_i)$, $\forall i \in \{1,\ldots,\mu\}$, where each x_{ij} of the vector $\vec{x}_i$ is a random value from the interval $[u_j,v_j]$ $\forall j \in \{1,\ldots,n\}$ and each σ_{ij} of the vector $\vec{\sigma}_i$ is a random value from the interval [0,c] $\forall j \in \{1,\ldots,n\}$ for a specified c > 0. Also the time step t is set to 0.
2) The fitness of each of the population members is determined such that $\Phi(\vec{a}_i) = f(\vec{x}_i)$, $\forall i \in \{1,\ldots,\mu\}$, where $f(\vec{x}_i)$ is some fitness function to be minimized. Also t is set to 1.
3) At this point a form of recombination could occur, but in this study performed this step it will be omitted.
4) For each member of the population $\vec{a}_i$ an offspring was created by mutation such that:

$$\vec{\sigma}'_i = \vec{\sigma}_i \exp(\tau' N(0,1) + \tau N_i(0,1)) \quad (1)$$
$$\vec{x}' = \vec{x} + \vec{\sigma}' N(0,1) \quad (2)$$

$\forall i \in \{1,\ldots,\mu\}$. Here N(0,1) is a Gaussian normally distributed random variable with mean 0 and variance $\sigma^2=1$ newly generated for every σ of every population member. Also, $N_i(0,1)$ is also a Gaussian normally distributed random variable with mean 0 and variance $\sigma^2=1$, but is only generated only once for each population member. The values τ and τ' allow for the evolution of the strategy parameters of the ES, EP, and ESP and are typically set, by Schwefel's suggestion, as follows [1]:

$$\tau \propto \left(\sqrt{2\sqrt{n}}\right)^{-1} \quad (3)$$
$$\tau \propto \left(\sqrt{2n}\right)^{-1} \quad (4)$$

Finally check the value of unsuccessful successive generations is reached to k times? If yes, then c is set to c/2 then refresh all the strategy parameters values of all the individuals in the population (i.e. each σ_{ij} of the vector $\vec{\sigma}_i$ is a new random value from the interval [0,c] $\forall j \in \{1,\ldots,n\}$, and $\forall i \in \{1,\ldots,\mu\}$, otherwise do nothing more.

5) The fitness of each offspring is determined such that $\Phi(\vec{a}'_i) = f(\vec{x}'_i)$, $\forall i \in \{1,\ldots,\mu\}$.
6) The new population $P(t)$ is created by (μ+μ)—selection mechanism as follows: The μ parent individuals produce μ offspring individuals sorted together, the best μ individuals from the sorted list of 2μ individuals (P(t-1) + offspring) survive to be the new population P(t) and so on.
7) The value for t is incremented by 1 and steps 3-7 repeated until an end condition is met such as a fixed number of generations have been completed.

4.2 ESP and (μ+λ)-ES similarities and differences

The major similarity between (μ+λ)-ES and ESP is that both systems maintain populations of potential solutions, make use of the selection principle of the survival of the best individuals, represent the individuals by the same way each individual a was constructed as $\vec{a} = (\vec{x}, \vec{\sigma})$, and use the same mutation process technique. There are two differences between these to approaches.

The first difference between ES and ESP is the recombination process where in ESP there is no use of recombination whereas ES use Recombination to generate the next generation of individuals.

The second difference is the rule of random adaptation for the strategy parameters of the individuals if the generation fixed on some best points in the solution space for a certain times of generations (k).

4.3 ESP and EP similarities and differences

Evolution Strategy Programming (ESP) technique is generalized to handle numerical optimization problems. It is quite similar to Evolutionary Programming (EP); they use floating point representation, the mutation is the key operator, and both of them do not use any recombination operators. The basic differences between Evolution Strategy Programming and Evolutionary Programming techniques can be summarized as follows:

- EP use a probabilistic selection (tournament selection), whereas ESP select the best μ individuals for the next generation,
- in Ep, fitness values are obtained from objective function values by scaling them and possibly by imposing some random alternation,
- mutation process as stated above is different since standard EP use uniform random mutation technique and meta-EP use the same mutation technique as in ES while ESP use the rule of random adaptation during the mutation process.

5. Test Function

We use a well-known function after Fletcher and Powell which is a highly multimodal function. It was introduced for the first time by Fletcher and Powell and also used by Schwefel in connection with Evolution Strategies. A three-dimensional plot of this function is shown in figure 3. The function after Fletcher and Powell is not symmetric, but instead the extrema are randomly distributed over the search space. This way, the objective function has no implicit symmetry advantage that might simplify optimization for certain algorithms [1].

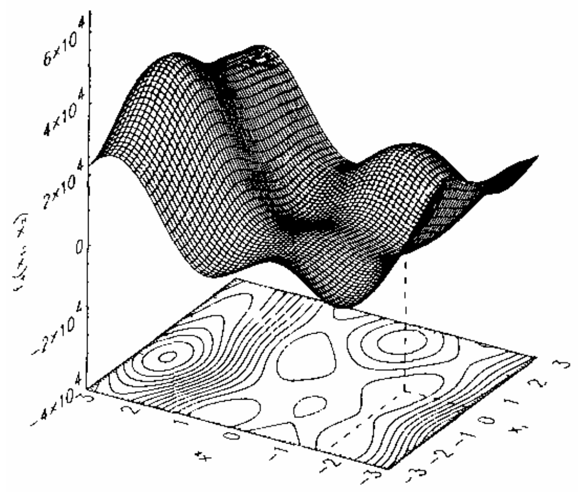

Figure 3: Three-dimensional plot of the function after Fletcher and Powell.

The random location of extrema is achieved by using random matrices A = (aij) and B = (bij) in the following description of the problem:

$$f\ (\vec{x}) = \sum_{i=1}^{n} (A_i - B_i)^2 \qquad (5)$$

$$A_i = \sum_{j=1}^{n} (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j)$$

$$B_i = \sum_{j=1}^{n} (a_{ij} \sin x_j + b_{ij} \cos x_j)$$

$$\vec{x}^* = \vec{\alpha}\,;\ \ f_4^* = 0\,;\ \ n = 5;\ \ -\pi \le x_i \le \pi$$

$$a_{ij}, b_{ij} \in [-100{,}100]\,;\ \ \alpha_j \in [-\pi, \pi]\,.$$

As Fletcher and Powell pointed out, there are up 2n extrema located in the search interval $|x_i| \le \pi$. In addition to A and B, the vector $\vec{\alpha}$ is also chosen at random. Here the values of A and B matrices used in our experiment [1]:

$$A = \left\{ \begin{matrix} -78 & 28 & 53 & -9 & 75 \\ 38 & 13 & -30 & 77 & 61 \\ -13 & -50 & -98 & 20 & -40 \\ -75 & 10 & -22 & -60 & -88 \\ 27 & 73 & 63 & 81 & 15 \end{matrix} \right\}$$

$$B = \left\{ \begin{matrix} -97 & -25 & -78 & -27 & 85 \\ -11 & -72 & 10 & -33 & -19 \\ 30 & 25 & -32 & -1 & 15 \\ 76 & 75 & 46 & 58 & 74 \\ 87 & -31 & -92 & -47 & 25 \end{matrix} \right\}$$

6. Experimental study

Concerning the population size the number of offspring individuals (λ) is adjusted to a common value of $\lambda = 100$ in order to achieve comparability of population sizes while at the same time limiting the computational requirements to a justifiable amount. This results in the following two algorithmic instances that are compared here:

- An Evolution Strategy that self-adapts n standard deviations but does not uses correlated mutations, Recombination is discrete, and the algorithm uses a (15,100)-selection mechanism.
- An Evolution Strategy Programming that self-adapts n standard deviation using k=30 and c=2. Using ($\mu+\mu$)-selection mechanism, evolves a population of $\mu = 100$.

The following two tables show a sample run from 50 total runs for both algorithms. We record the development of the best function value discovered so far during the evaluation process and we also record the time of arriving to the generation and as we show from table (1) and table (2) the convergence time rate in case of ESP is higher than the case of ES. Figure 4,5,6, and 7 explain the results existing in table (1) and table (2) The most remarkable result presented in table 1 is the high quality of the best value obtained from an ESP especially when compared to the small diversity of results yielded by.

ES		
Generation no	Best Fitness	Time (sec)
0	30025.35	0.18
15	1232.88	0.314
66	44.456	0.541
150	10.727	1.150
300	10.48	1.843
350	10.357	2.023
400	10.252	2.254
600	10.098	3.305
700	9.838	3.776
900	9.431	5.768

Table (1) evaluation of after Fletcher and Powell function using ES

ESP		
Generation no	Best Fitness	Time (sec)
0	30652.49	0.050
15	1041.39	0.121
66	10.80	0.361
150	0.296	0.772
300	0.00046	1.633
350	$< 1 \cdot 10^{-5}$	1.743
400	$< 1 \cdot 10^{-7}$	1.983
600	$< 1 \cdot 10^{-10}$	2.894
700	$< 1 \cdot 10^{-12}$	3.195
900	$< 1 \cdot 10^{-14}$	4.050

Table (2) evaluation of after Fletcher and Powell function using ESP

7. Conclusion

We propose an Evolution Strategy Programming as a new Evolutionary Algorithm technique, which combine the ES and EP techniques and add an extra rule to the mutation method used in ES and EP called random adaptation rule. And the method could solve robustly multi-modal functions that have strong local minima and deceptive landscape. This method robustly solves highly multimodal after Fletcher and Powell function with reaching to high quality best optimum value with an excellent convergence rate to than Evolution Strategies.

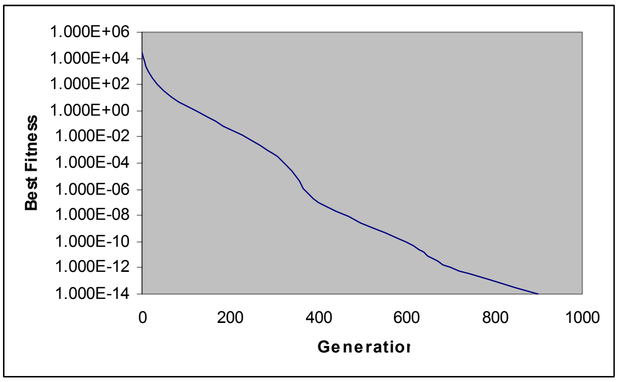

Fig. 4 experimental results of table (2), relation between generation and corresponding best value.

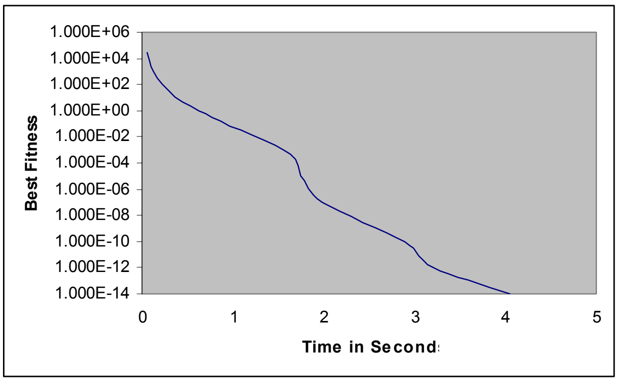

Fig. 5 experimental results of table (2), relation between best value and time of foundation.

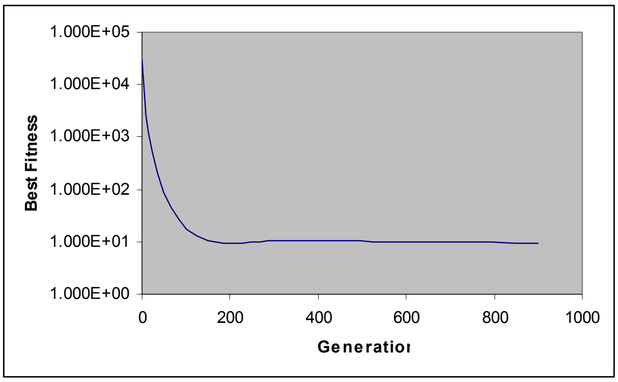

Fig. 6 experimental results of table (1), relation between generation and corresponding best value.

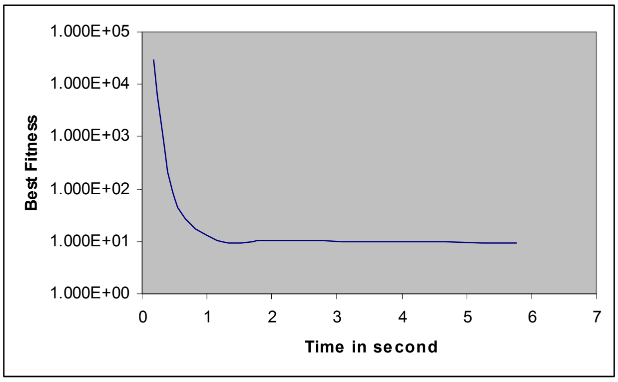

Fig. 7 experimental results of table (1), relation between best value and time of foundation.

8. **References**

[1] *Bäck: Evolutionary Algorithms in Theory and Practice. Oxford,1996*

[2] *Bäck, & H.-P. Schewefel: An Overview of Evolutionary Algorithms for Parameter Optimization, Evolutionary Computation, 1993*

[3] *Bäck, T., Fogel, D. and Michalewicz, Z. (eds in chief). Handbook of Evolutionary Computation. Oxford University, 1997*

[4] *H.-G. Beyer, The Theory of Evolution Strategies, (Springer, Heidelberg), 2000*

[5] *Michalewicz Z Genetic Algorithms + Data Structures = Evolution Programs (Berlin: Springer), 1999*