International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 12, Issue 3 [March. 2023] PP: 211-220

A Survey Paper on some Evolutionary Algorithms

E.S.SHAHEEN[1], M.A.El Sayed[2], D.A.Hammed[3], M.A.Elsisy[4]
[1,2,3,4]Basic ScienceDepartment., BanhaFaculty of Engineering, Banha University, Banha, EGYPT

ABSTRACT:*Evolutionary algorithms are an effective approach for solving complex optimization problems in various fields. This paper presents a survey about genetic, differential evolution, evolution strategies, and estimation of distribution algorithms. Pseudo-code and the basic flow chart of each algorithm are listed. Also, the improvement and hybridization of these algorithms are shown. The famous applications that are solved by these techniques are listed. The advantages and limitations of algorithms are presented.*
Keywords*: Meta-heuristics,Genetic Algorithm, Differential evolution, Evolutionary strategy, Estimation of distribution algorithms, Evolutionary algorithms, Multiobjectiveoptimisation.*

Date of Submission: 09-03-2023 Date of acceptance: 22-03-2023

I. INTRODUCTION

Metaheuristics algorithms are used for almost all engineering research fields. One of the most used families of metaheuristics is evolutionary computation(EC),these computations are inspired by population biology, genetics, and evolution. An algorithm from EC is called the evolutionary algorithm(EA). Evolutionary algorithms(EAs) are used for solving and dealing with very complex optimization problems, especially for non-deterministic polynomial time problems(NP). EAs don't require any prior knowledge about the structure of the problem, so these deal with the objective function as a black box at the beginning. They don't need to calculate the gradient or a Hessian matrix to find the optimum solution which allows us to handle problems with discontinuities. They used to get some information about the problem structure by generating a candidate solution and apply into the objective function to get the fitness value, this value is used to resample a better solution until meeting the stopping criteria[1].EAs are population-based algorithms that deal with noisy data, and also they deal with multiobjective optimization problems(MOOP). This paper is concentrated on the most popular main EAs like a genetic algorithm(GA), evolutionary strategy(ES), differential evolution(DE).

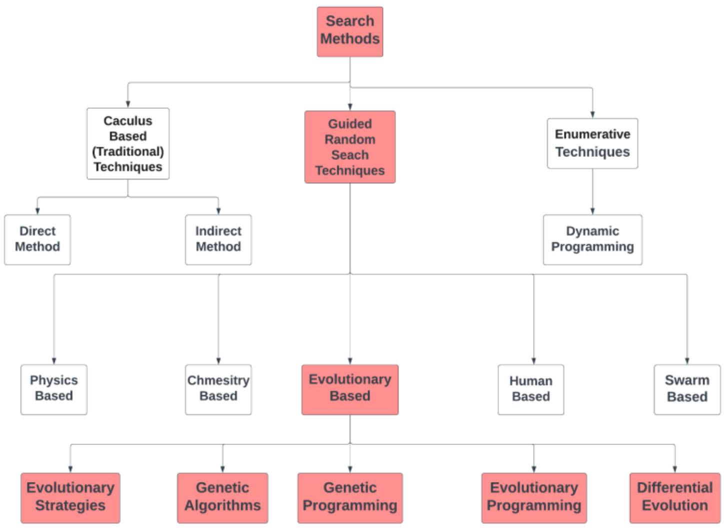

Fig. 1 Searching Methods

II. EVOLUTIONARY ALGORITHMS

EA is a stochastic search algorithm, it's a subclass of EC, Fig 1 shows EAs as one of the searching methods. First, generate a random population, then make iterations over three steps. The first step evaluates the fitness functions of all individuals in a population. Fig 2 represents the basic definitions for EAs.The second step, use fitness values from the first step to breed a new population of children. The third step joins parents with children to generate the next generation population. Algorithm 1 represents the pseudocode evolutionary algorithms. Evolutionary algorithms types are different from each other by how they perform Breed and Join. The Breed is divided into two parts. The first part is selecting parents from the old population and the second part is mutating,recombining,and crossover them in some way[2].

III. GENETIC ALGORITHMS

The genetic algorithm is the most popular evolutionary algorithm, it was first proposed by Holland in 1970[3], GA uses recombination and mutation operators to get the solution to the problem, the solution is represented as a binary string each bit refers to the gene.A segment decoding function donates the i-th segment of an

individual.In GA we have a population of individuals (chromosome). Solving the problem using the objective function, and testing the solution gives the fitness value. This value is measuring the solution quality.The good fitness value the better chance to be selected by the next generation. GA has three main operators: Selection(new solution generated depends on the fitness value since of probability). crossover(parts of solutions exchanged between two solutions selected to crossover), and mutation(the value of a particular bit of binary string or gene changed based on probability). Fig 3 represents the basic flow chart of GA. Algorithm 2 shows the pseudocode of GA[4].

individual	a candidate solution
child and parent	a *child* is the Tweaked copy of a candidate solution (its *parent*)
population	set of candidate solutions
fitness	quality
fitness landscape	quality function
fitness assessment or evaluation	computing the fitness of an individual
selection	picking individuals based on their fitness
mutation	plain Tweaking. This is often thought as "asexual" breeding.
recombination or crossover	a special Tweak which takes two parents, swaps sections of them, and (usually) produces two children. This is often thought as "sexual" breeding.
breeding	producing one or more children from a population of parents through an iterated process of selection and Tweaking (typically mutation or recombination)
genotype or genome	an individual's data structure, as used during breeding
chromosome	a genotype in the form of a fixed-length vector
gene	a particular slot position in a chromosome
allele	a particular setting of a gene
phenotype	how the individual operates during fitness assessment
generation	one cycle of fitness assessment, breeding, and population re-assembly; or the population produced each such cycle

Fig. 2 The basic definitions of EAs

The advantages of GA are:
1. Using only information of objective function no need to derivative.
2. Support multi-objective optimization problems.
3. Robust to local minima or maxima (traps).
4. Easy to parallelize.
5. Working good for both continuous and discrete problems.

The limitations of GA:
1. The convergence of algorithms can be too fast or too small.
2. The quality of solution depends on selecting the parameters such as crossover, mutation, probability, size of population.
3. Cannot guarantee optimality.

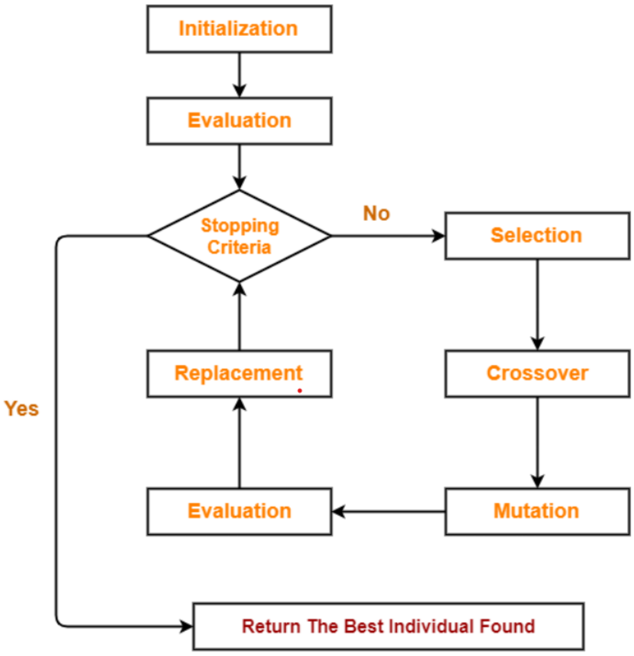

Fig. 3 The basic flow chart of GA

Algorithm 1 An Abstract Pseudocode Of Evolutionary Algorithm(EA)

$P \leftarrow$ Build Initial Population
$Best \leftarrow \{\}$
repeat
  Assess Fitness(P)
  for each individual $P_i \in P$ **do**
    if Best=$\{\}$ or Fitness(P_i) $<$ Fitness($Best$) **then**
      $Best \leftarrow P_i$
    end if
    $P \leftarrow$ Join(P,Breed(P))
  end for
until $Best$ is the ideal solution or timed out
return $Best$

Applications of GA: GA has enormous real world application such:Network Routing Protocol [5],CNN Architectures Using the Genetic Algorithm for Image Classification [6],using GA algorithms for optimizing machine learning models [7],GA application to optimization of AGC in three-area power system after deregulation [8] ,GA- based feature selection with application in handwritten character recognition [9].

Algorithm 2 Pseudocode Of Genetic algorithm (GA)

```
popsize← desired population size
P = {}
for popsize times do
  P ← P ∪ { new random individual}
end for
Best ← {}
repeat
  for each individual P_i ∈ P do
    Assess Fitness(P_i)
    if Best={} or Fitness(P_i) > Fitness(Best) then
      Best=← P_i
    end if
  end for
  Q ← {}
  for popsize/2 times do
    Parent P_a ← SelectWithReplacement(P)
    Parent P_b ← SelectWithReplacement(P)
    Children C_a, C_b ← Crossoper(Copy(P_a),Copy(P_b))
    Q ← Q ∪ { Mutate(C_a),Mutate(C_b)}
    P ← Q
  end for
until Best is the ideal solution or timed out
return Best
```

IV. EVOLUTION STRATEGIES

Evolution strategies (ESs) were published by Rechenberg in 1973[10].ESs have differences compared to GA. In ESs the fitness values are not the decisive factor as in GA, the new generation size is not the same as the parent size but it depends on the two parameters (λ, μ).ESs operate on floating point vectors while GA operates on binary vectors. Also, the is another major difference between ES and GA in the selection process. In GA a certain number of individuals from the parent size is selected for the new population, but in ES it depends on parameters so when a sampling individual occurs during the selection process, the selection process in ES is deterministic unlike GA. In ES, the recombination process occurs first then selection, unlike GA. The crossover process may or may not be applied, so may only apply the mutation operator as shown in Fig 4.

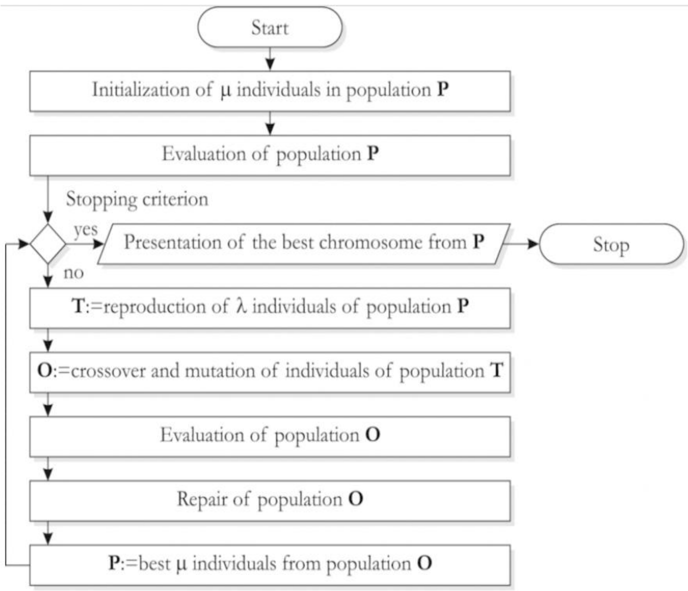

Fig. 4 The basic flow chart of ES

There are many types of ES: the first algorithm is a strategy (1+1), in this algorithm, we have only one base chromosome, in each iteration, a new individual one will be generated as a result of mutation. The second algorithm is an extension of (1+1) is ($\lambda + \mu$), this strategy has a larger number of individuals to get more diverse solutions to avoid local minima or maxima, at first starts with the parent population containing mu individuals with $\lambda \geq \mu$ a temporary population created by reproduction. The third algorithm is the strategy μ, λ) which is more used than (μ, λ)but it has the same operations except that the new population contains μ individuals selected from the best μ individuals of the population [11]. Algorithm 3 represents the pseudocode of ES.

The advantages of ES are:
1. Only need the objective function to evaluate.
2. Good for noisy data .
3. Get diversity solutions.
4. Used to approximate covariance and inverse hessian matrix.
5. Working well for noisy data.
6. Can access more points in search space.

The limitations of ES:
1. Need more time for convergence.
2. The quality of solution depends on selecting the parameters such as crossover, mutation, probability, size of population.
3. Need more memory to represent floating vectors.

Applications of ES: can be used for system parameter estimation[12],also can use evolution strategy with trust region called Trust Region Evolution Strategies [13],A Modified Covariance Matrix Adaptation Evolution Strategy for Real-World Constrained Optimization Problems[14].

Algorithm 3 Pseudocode Of ES (μ, λ)

```
μ ← number of parents selected
λ ← number of children generated by the parents
P = {}
for λ times do
  P ← ∪ {new random indivdual}
  Best ← {}
end for
repeat
  for each individual P_i ∈ P do
    Assesses Fitness(P_i)
    if best= or Fitness(P_i) > Fitness(best) then
      best ← P_i
    end if
    Q ← the μ individuals in P whose fitness are
    greatest
    P = {}
  end for
  for each individual Q_j ∈ Q do
    for λ/μ times do
      P ← P ∪ {Mutate(Copy(Q_j))}
    end for
  end for
until Best is the ideal solution or timed out
Require: best
```

V. DIFFERENTIAL EVOLUTION

Differential evolution(DE) was published by Storm in 1996 at IEEE international conference in evolutionary computation[15]. The total number of citations of DE has been approximately 22000 since 1996. DE is an evolutionary computation technique designed for multi-dimensional spaces.DE has two phases. First phase deals with the initialization of the population randomly.The second phase deals with the evolutionary behavior as mutation, crossover, and selection processes. The size of the mutation is depending on the current variance of the population. If the population is wide the mutation will make a large change. If they are concise in a region, the mutation will be very small[16]. The DE is considered an adaptive mutation algorithm.Mutation phase begins by taking a candidate solution from the population, then the algorithm takes three candidate solutions, after that calculate Vd , it then creates mutant vector Vm by adding the third vector to λVd where λ is a scaling factor.Fig 5 represents the behavior of DE.The crossover phase uses to maintain population diversity, DE generates a candidate solution which is a mixture between trail and target individual . Where Cr is the crossover factor and the value is in range(0, 0.51). The selection process is used to compare the values of target and candidate individuals for population selection. apply these individuals to the fitness function and take the best one as .Fig 6 represents the basic flow chart of DE.Algorithm 4 represents the Pseudocode of DE.

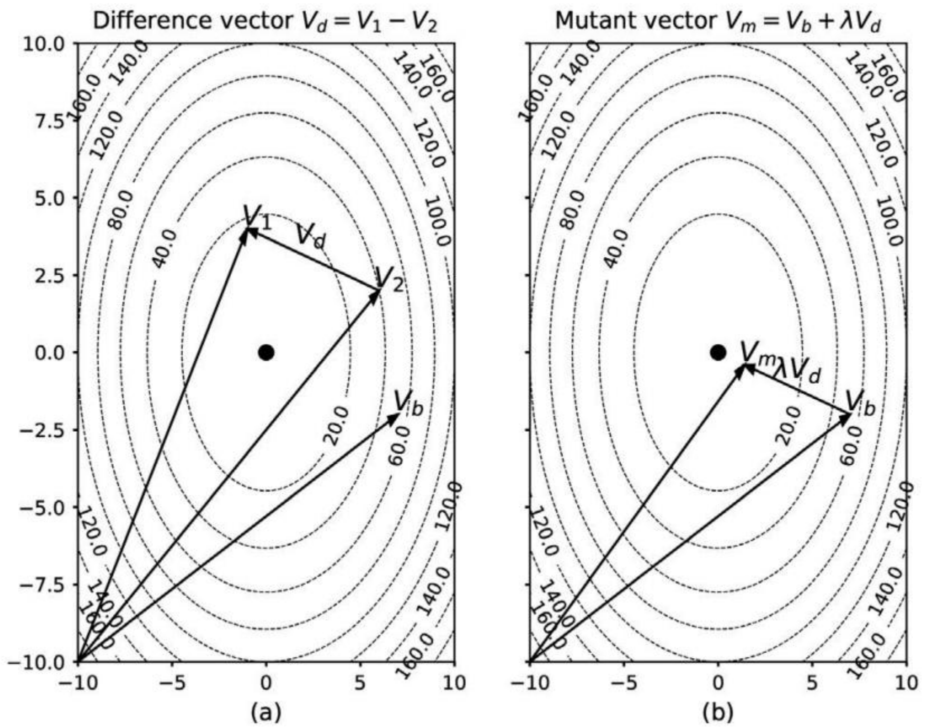

Fig. 5 Difference and mutant vectors.The dashed line represents the contours of an objective function

The advantages of DE are:

1. Easy to use,efficient memory utilization and low complexity.
2. Scaling factor),crossover rate and population size are only the control parameters for conical DE.
3. Can exit local minima with reasonable rate of convergence.
4. Heavily used for the exploration phase.
5. Good for multi-modal,multi-objective optimization problems.

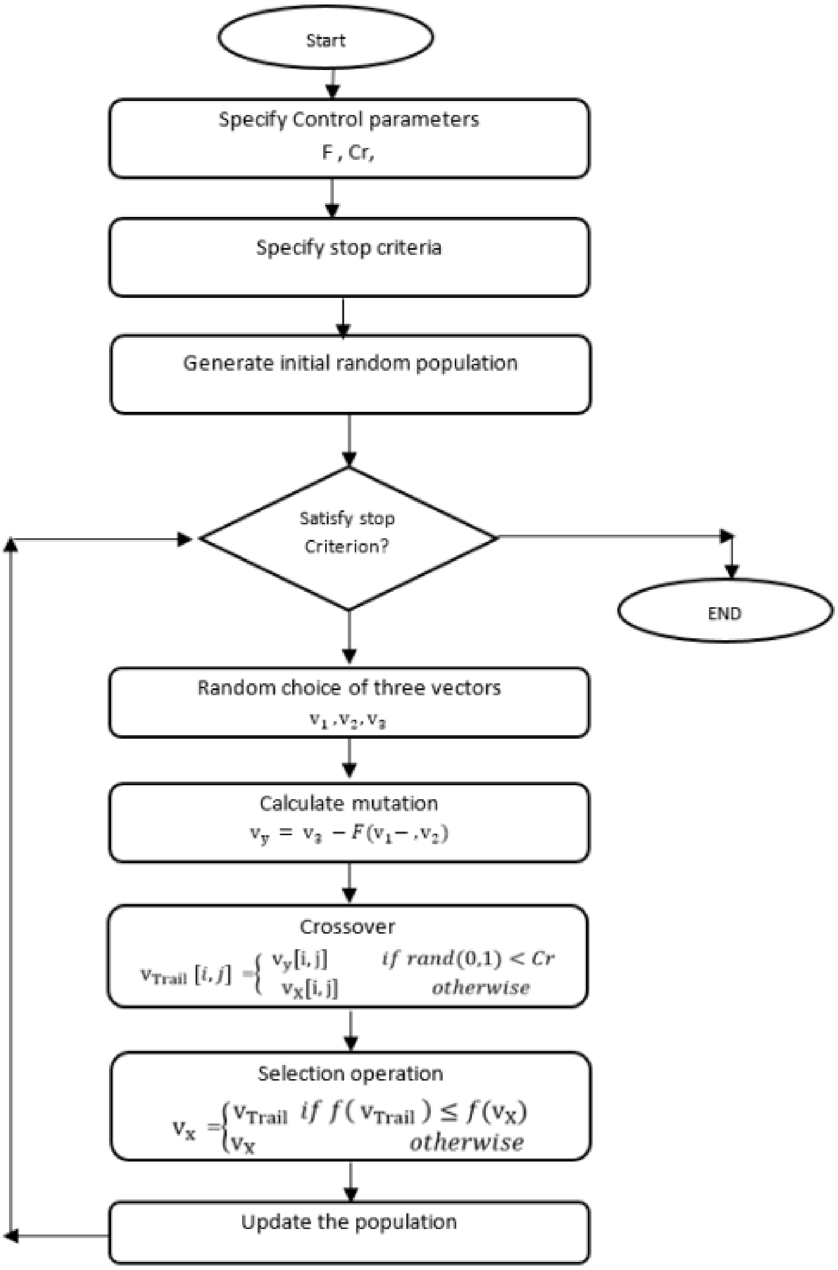

Fig. 6 The basic flow chart of Differential evolution.

The limitations of DE are:
1. New siblings may have worse fitness than parents.
2. Can tune parameters to make equilibrium between exploration and exploitation phase but, it's not a good algorithm for the exploitation phase.
3. Don't work well for noisy optimization tasks [17].

Applications of DE: DE is used for many real life problems. Can use DE for automatic data stringing [18]. It is used for multi-modal multi-objective problems in reinforcement learning [19]. Als it has a good potential for image processing and AI applications as COVID-19 classification from CT images using multi objective differential evolution-based convolution neural networks[20].

Algorithm 4 Pseudocode Of Deferential evolution

$\alpha \in (0, 0.9]$ as a step size parameter
$p_c \in [0, 1)$ is the crossover probability parameter
$P \leftarrow$ random create the initial population of N
while termination criterion is not satisfied **do**
  for $i \in [1, N]$ **do**
    $x_i \leftarrow$ pick the i-th candidate solution
    x_a randomly select x_a where $a \neq i$
    x_b randomly select x_b where $a \neq i \neq b$
    x_c randomly select x_c where $a \neq i \neq b \neq c$
    $V_d \leftarrow x_a - x_b$
    $V_m \leftarrow x_c + \alpha V_d$
    i_r random integer $\in [1, n]$
    for each dimention $j \in [1, n]$ **do**
      $r_i \leftarrow$ random number $\in [0, 1]$
      if $r_i < p_c$ or $j = i_r$ **then**
        $V_t(j) = V_m(j)$
      else
        $V_t(j) = x_i(j)$
      end if
    end for
  end for
end while
return P

VI. ESTIMATION OF DISTRIBUTION ALGORITHM

Estimation of distribution algorithms (EDAs) was published by H. M¨uhlenbein in 1996 [21] Have been used for global optimization. This family of algorithms is much different compared to traditional EAs. There aren't any crossover or mutation operators, but EDA builds probabilistic modeling of good solutions in the searching space. After that, the new candidate solutions are chosen by the sampling of the model that extracted global statistical information. EDA uses reproduction techniques to find better solutions. These techniques are explained by the probabilistic model. There are two major branches of continuous EDAs. One is based on the Gaussian distribution model which is mostly used and the other model is based on histogram models. In the majority of research EDA is used for low dimensional (< 500) dimensions[22].Fig 7 represents the basic flow chart of EDA. Algorithm 5 represents the basic Pseudocode of EDAs.

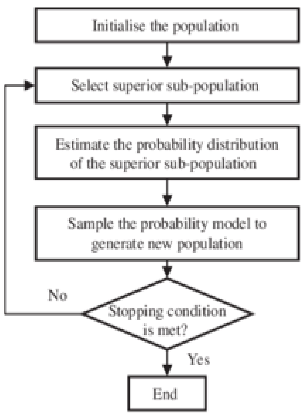

Fig.7 The basic flow chart of EDA

EDAs are categorized by the types of distribution that can encode. The first category of EDA is called Univariant models which is one of the simplest approaches to assume that the problem variables are independent. So that assumption means that the probability of any individual variable has not depended on any other variable as shown in Fig 8. Univariatemodel decomposes the probability of a candidate solution.

Tree based model is the second categorical of EDAs in which the conditional probability of availability may only depend on at most one other variable, its parent in a tree structure. Mutual information maximizing input clustering (MIMIC) uses a chain distribution to model interaction between variables. This works by using promising populations to calculate the mutual information between all pairs of variables, then starting with variables with minimum conditional entropy. After that the chain is added to a variable with maximum mutual information. This process is repeated until all variables are finished. The final tree contains only a single chain dependency with each parent consisting of only one child. If the model is completed the conditional probability of each parent is a calculation from the promising solutions. The new population solutions are generated by sampling the probability distribution functions by model [23].

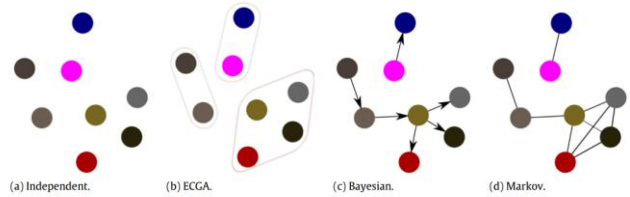

Fig.8 Examples of graphical models produced by different EDAs.

The advantages of EDAs are:

1. The ability to adapt their operators to the structure of the problem which has dynamically changed the model type while solving a problem.
2. Good for problems with some prior knowledge exploitation for example Bayes statistics can be used to bias model building in EDAs which lead to global optimum towards probabilistic models.
3. Reduced memory requirement which replaces current solutions by a probabilistic model. This allows us to solve large problems.

The limitations of EDAs:

1. More time consuming compared to simple search operators like tournament selection and two point crossover.
2. Sometimes it's difficult to adequate a probabilistic model, because many algorithms uses greedy techniques.

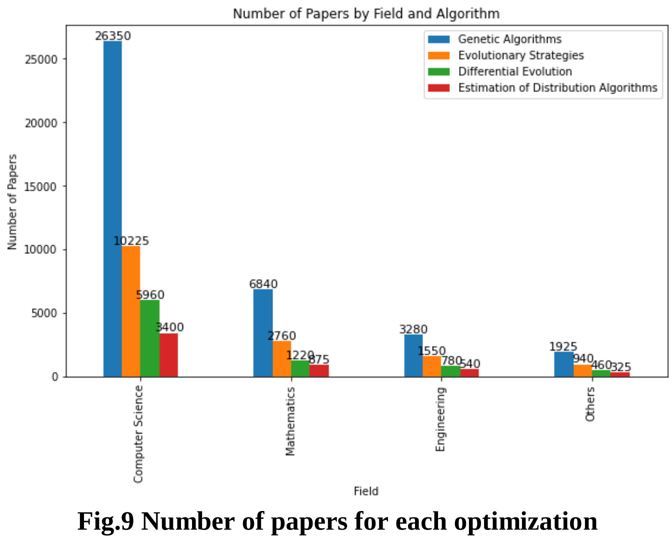

Fig.9 Number of papers for each optimization algorithms with its field.

VII. SUMMARY

The Fig 9 displays the number of papers published for four different algorithms (Genetic Algorithms, Evolutionary Strategies, Differential Evolution, and Estimation of Distribution Algorithms) across four different fields (Computer Science, Mathematics, Engineering, and Others). The height of each bar represents the number of papers published for the corresponding algorithm and field. The number of papers is displayed above each bar to facilitate easy comparison between the different fields and algorithms. The figure clearly shows that

Genetic Algorithms and Evolutionary Strategies have been the most popular algorithms, with the majority of papers published in the field of Computer Science.

VIII. CONCLUSION

This survey provided an overview of some evolutionary algorithms and their real-world applications. The pseudo-code implementation and the main properties of each algorithm are discussed. Many real-world applications are solved efficiently by evolutionary algorithms, as demonstrated in this paper. But real-world problems need new algorithms and hybridization between old ones to get more efficient techniques. Overall, this paper provides a valuable resource for researchers and practitioners who are interested in understanding and applying these algorithms in their work.

REFRENCES

[1]. Leonardo AzevedoScardua. Applied Evolutionary Algorithms for Engineers Using Python.CRC Press, 2021.

[2]. Sean Luke. Essentials of metaheuristics, volume 2.Lulu Raleigh, 2013.

[3]. Holland JH. Ga. Adaptation in natural and artificial systems. MIT Press, Cambridge, 1975.

[4]. AnnuLambora, Kunal Gupta, and Kriti Chopra.Genetic algorithm-a literature review. In 2019 international conference on machine learning, big data, cloud and parallel computing (COMITCon), pages 380–384. IEEE,2019.

[5]. Bo Fan and JifengLuo.Spatially enabled emergency event analysis using a multi-level association rule mining method. Natural hazards, 67(2):239–260, 2013.

[6]. Yanan Sun, Bing Xue, Mengjie Zhang, Gary G Yen, and Jiancheng Lv. Automatically designing cnn architectures using the genetic algorithm for image classification. IEEE transactions on cybernetics, 50(9):3840–3854, 2020.

[7]. Khader M Hamdia, XiaoyingZhuang, and TimonRabczuk. An efficient optimization approach for designing machine learning models based on genetic algorithms. Neural Computing and Applications, 33(6):1923–1933, 2021.

[8]. Demiroren and HL Zeynelgil. Ga application to optimization of agc in three-area power system after deregulation. International Journal of Electrical Power & Energy Systems, 29(3):230– 240, 2007.

[9]. Claudio De Stefano, Francesco Fontanella, Cristina Marrocco, and A Scotto Di Freca. A ga-based feature selection approach with an application to handwritten character recognition.Pattern Recognition Letters, 35:130–141, 2014.

[10]. Ingo Rechenberg. Evolutionsstrategie.OptimierungtechnischerSystemenachPrinzipien der biologischen Evolution, 1973.

[11]. LeszekRutkowski. Computational. Intelligence Methods and Techniques; Springer: Berlin/Heidelberg, Germany, 2008.

[12]. Toshiharu Hatanaka, KatsujiUosaki, Hirokadzu Tanaka, and Yasuhiro Yamada.System parameter estimation by evolutionary strategy. In Proceedings of the 35th SICE Annual Conference.International Session Papers, pages 1045–1048. IEEE, 1996.

[13]. Guoqing Liu, Li Zhao, Feidiao Yang, Jiang Bian, Tao Qin, Nenghai Yu, and Tie-Yan Liu. Trust region evolution strategies. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 4352–4359, 2019.

[14]. Abhishek Kumar, Swagatam Das, and Ivan Zelinka. A modified covariance matrix adaptation evolution strategy for real-world constrained optimization problems. In Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, pages 11–12, 2020.

[15]. Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces.Journal of global optimization, 11(4):341–359,1997.

[16]. Millie Pant, HiraZaheer, Laura GarciaHernandez, Ajith Abraham, et al. Differential evolution: A review of more than two decades of research. Engineering Applications of Artificial Intelligence, 90:103479, 2020.

[17]. ThiemoKrink, BogdanFilipic, and Gary B Fogel. Noisy optimization problems-a particular challenge for differential evolution? In Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753), volume 1,pages 332–339. IEEE, 2004.

[18]. Ad´anJos´e-Garc´ıa and WilfridoG´omez-Flores. A survey of cluster validity indices for automatic data clustering using differential evolution. In Proceedings of the Genetic and Evolutionary Computation Conference, pages 314–322, 2021.

[19]. Zhihui Li, Li Shi, CaitongYue, Zhigang Shang, and Boyang Qu. Differential evolution based on reinforcement learning with fitness ranking. for solving multimodal multiobjective problems. Swarm and Evolutionary Computation, 49:234–244, 2019.

[20]. Dilbag Singh, Vijay Kumar, ManjitKaur, et al. Classification of covid-19 patients from chest ct images using multi-objective differential evolution–based convolutional neural networks. European Journal of Clinical Microbiology & Infectious Diseases, 39(7):1379–1389, 2020.

[21]. Heinz M¨uhlenbein and Gerhard Paass.From recombination of genes to the estimation of distributions i. binary parameters. In International conference on parallel problem solving from nature, pages 178–187. Springer, 1996.

[22]. Weishan Dong, Tianshi Chen, Peter Tiˇno, and Xin Yao. Scaling up estimation of distribution algorithms for continuous optimization. IEEE Transactions on Evolutionary Computation, 17(6):797–822, 2013.

[23]. Mark Hauschild and Martin Pelikan.An introduction and survey of estimation of distribution algorithms. Swarm and evolutionary computation, 1(3):111–128, 2011.