VISUALISING ENSEMBLE DIFFERENTIAL EVOLUTION ALGORITHM

Project Report

Submitted by

ASHWIN NAIR [CB.EN.U4CSE18211]
VIJAY SWAMINATHAN [CB.EN.U4CSE18467]

Under the guidance of

Dr. C. Shunmuga Velayutham
(Vice Chairperson, Department of Computer Science & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

Amrita Nagar PO, Coimbatore - 641 112, Tamilnadu

ABSTRACT

Evolutionary algorithms(EAs) are a powerful class of search algorithms used for optimization problems. They draw their inspiration from Darwin's Theory of evolution. They are generally used in case of NP hard problems as they don't have either a polynomial time running algorithm nor a polynomial time check algorithm for the NP solution.

Since EAs are stochastic in nature, it is very difficult to analyze and understand their search dynamics. For instance, in the case of machine learning algorithms we can use visualisation plots to better understand how our models are performing. Same isn't the case with evolutionary algorithms by virtue of their stochasticity. An even tougher task is to analyze the working of ensemble evolutionary algorithms. Simply put, ensemble learning clubs more than two algorithms together and supposedly gives us a better result than what those individual algorithms would have given us.

Analysis of the above-mentioned algorithms like EA and ensemble EA is predominantly done with a mathematical/theoretical modelling, where we try to predict the convergence of the algorithm thereby guaranteeing us the optimised result. Another method is through empirical analysis, where EAs are simulated for a given problem a number of times and analyze the performance. The latter is a data driven approach and depends heavily on the data.

Yet another method of analysis is using visualisation techniques. These

would help us analyze the exploratory and exploitative aspects of the algorithms, a very thin line to balance in the field of evolutionary algorithms.

In this project, we propose to Visualize Ensemble Differential Evolution (DE) algorithms (which are a class of EAs). The visualization is intended to understand and analyze the complementary nature of ensemble DE algorithms.

The visualization will employ the existing graph theoretic approaches extended to ensemble algorithms.

TABLE OF CONTENTS

LIST OF FIGURES

List of Algorithms

ABBREVIATIONS

STN	Search Trajectory Networks
LON	Local optima network
CN	Complex Networks
PCA	Principal Component Analysis
MPEDE	Multi-population ensemble DE
DE	Differential Evolution
EA	Evolutionary Algorithm

LIST OF SYMBOLS

$\vec{x}$	Vectors

Chapter 1

INTRODUCTION

1.1 Problem Definition

We propose to visualize ensemble Differential Evolution (DE) algorithms (which are a class of Evolutionary Algorithms (EAs)) by employing existing graph theoretic approaches extended to ensemble algorithms. The visualization is intended to understand and analyze the complementary nature of ensemble DE algorithms. EAs (Eiben and Smith (2015)) are a powerful class of search algorithms used for optimization problems. They are generally used in case of NP hard problems as they don't have neither a polynomial time running algorithm nor a polynomial time check algorithm for the NP solution. The steps involved in EA are: **initialization, selection, genetic operators (crossover and mutation), and termination**

EAs are stochastic algorithms. Their search dynamics are hard to analyze due to their stochastic nature. Apart from theoretical and empirical methods, visualization is a potential method to analyse EAs.

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Figure 1.1: The general scheme of an evolutionary algorithm in pseudocode

In Differential EA the mutation is done using 3 randomly chosen vectors from the population and then adding the weighted difference of 2 vectors to the 3rd.

$$\vec{x}' = \vec{x} + F \cdot (\vec{y} - \vec{z})$$

Ensemble methods use multiple learning algorithms to obtain better predictive performance than what could have been obtained from any of the constituent learning algorithms alone. Simply put, ensemble learning clubs more than two algorithms together and supposedly gives us a better result than what those individual algorithms would have given us.

Chapter 2

LITERATURE SURVEY

2.1 Local optima networks (LONs) and Disconnectivity graphs aka Barrier trees

Local optima networks (LONs) (Ochoa et al. (2008)) are a compressed model of fitness landscapes where nodes are local optima and edges represent possible transitions between optima with a given search operator.

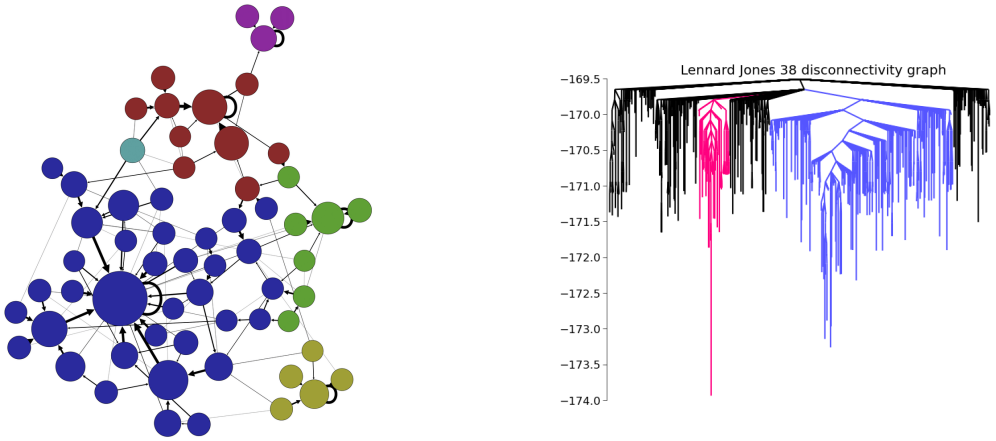


Figure 2.1: (Left) Local optima networks (LONs) (Right) Barrier Trees

Barrier trees(Flamm et al. (2002)) are graph-based modelling tools originated from the study of energy landscapes, which has also been applied to model the fitness landscapes of optimisation problems.

In LONs the goal is to model the structure of the fitness landscape, but instead the search behaviour of optimization algorithms.

2.2 Complex networks

Complex Networks (CN) (Zelinka and Davendra (2011)) graphs whose connections represent inter-actions amongst the individuals during all generations, vertices are indi-

viduals that are activated by other individuals, incrementally from generation to generation. In CN we model the interactions among candidate solutions in the population.

Figure 2.2: Complex Networks

2.3 Pareto front approximations and Trace generation plots

Pareto front approx.(Tusar and Filipic (2015); Fieldsend et al. (2019)) involves dimensionality reduction techniques to show algorithm progression in improving objective values over time. Trace generation plots show how the hypervolume value changes over generations.

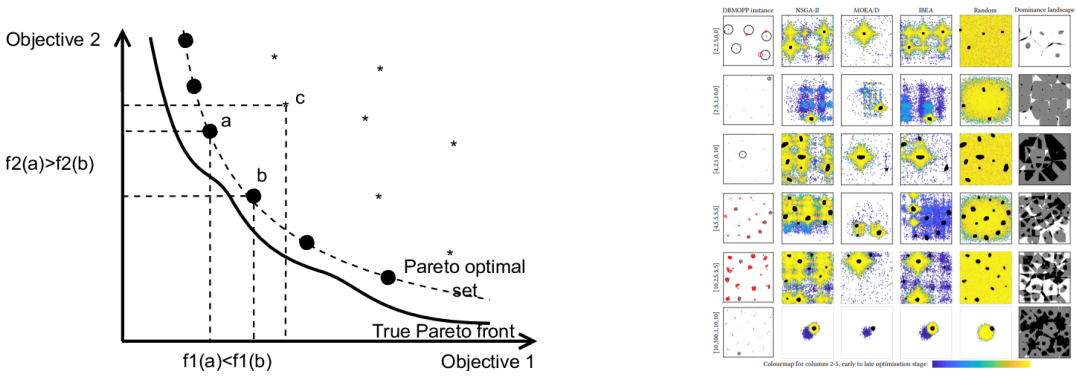


Figure 2.3: (Left)Pareto front approximations, (Right) Trace Generation Plots

2.4 Principal Component Analysis (PCA), Sammon mapping and t-distributed stochastic neighbour embedding

Dimensionality reduction techniques like Principal Component Analysis (PCA) (Collins (2003); Pohlheim (2006); Michalak (2019)), Sammon mapping, and t-distributed stochastic neighbor embedding can be used to map multi-dimensional solutions to lower dimensions to visualize how trajectories of solutions change over time.

With this approach, the positions of solutions relative to each other and the movement of individuals in a population can be visualized over an algorithm run using a sequence of 2-D frames or a 3-D stacking of 2-D frames.

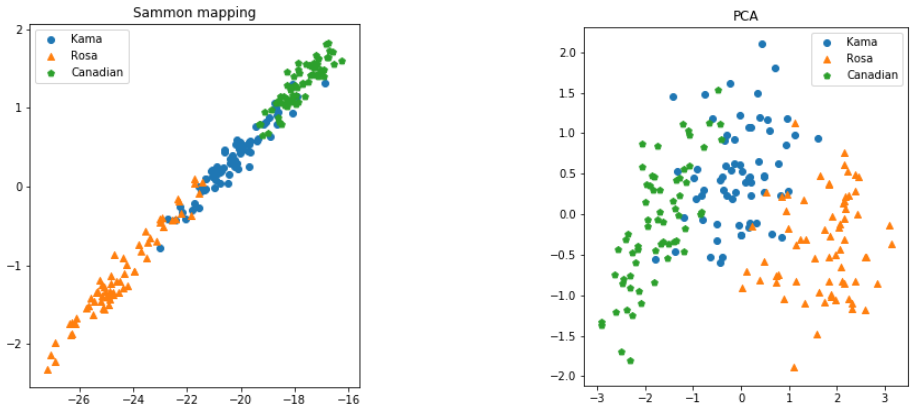


Figure 2.4: (Left) Sammon Mapping, (Right) PCA

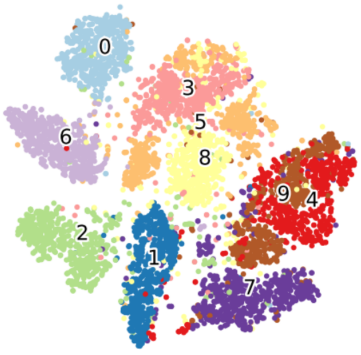


Figure 2.5: T Distributed Neighbour Embedding

2.5 Search Trajectory Networks (STN)

An STN (Ochoa et al. (2021)) is a directed graph STN = G(N, E), with node set N, and edge set E. Where Node is a location in a search trajectory of the search process being modelled. Edges are directed and connect two consecutive locations in the search trajectory. Edges are weighted with the number of times a transition between two given nodes occurred during the process of sampling and constructing the STN. A Search Trajectory is a Given sequence of representative solutions in the order in which they are encountered during the search process,

STNs are graph theoretic data driven visualization methods to visualize the search behavior of metaheuristics methods. STN allow us to observe and quantify potential regions in the search space, this information gives new insights into the understanding of metaheuristic methods.

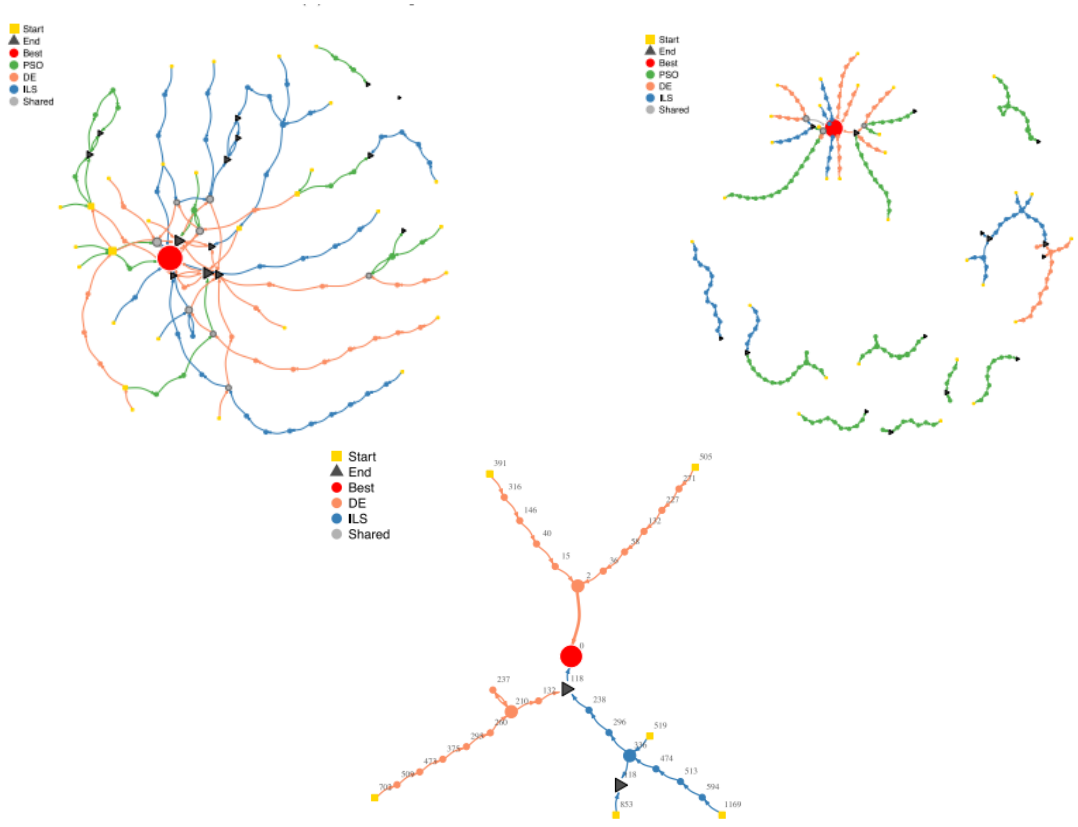


Figure 2.6: Search Trajectory Networks (STN)

2.6 Summary

STNs differ from LONs and Barrier Graph tools as the goal is not to model the structure of fitness landscapes, but instead the search behaviour of optimization algorithms. in STN, we model the trajectories of representative solutions across the search process. Moreover, it can be applied to any metaheuristic, not only to population-based ones, and merged STNs allow us to directly contrast the behaviour of different metaheuristics. Pareto front differ from STNs as they are visualising objective space, rather than solution space.

2.7 Software/Tools Requirements

Python, R

Chapter 3

PROPOSED SYSTEM

3.1 System Analysis

Module 1: We run the MPEDE algorithm on several functions in order to get the experimental data on how search optimisation works in these cases.

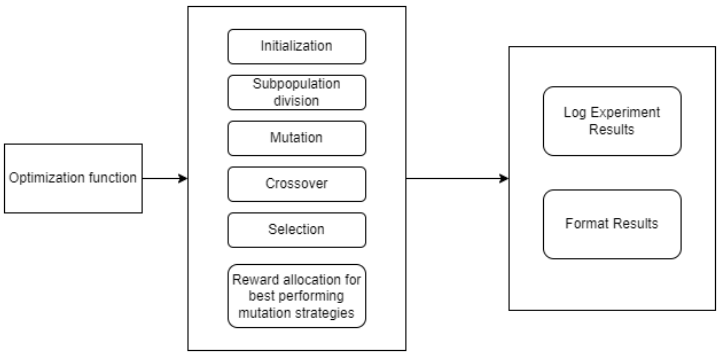


Figure 3.1: Module 1

Module 2: The objective of this module is to analyse the results using STNs on ensemble algorithms and validate them using suitable metrics.

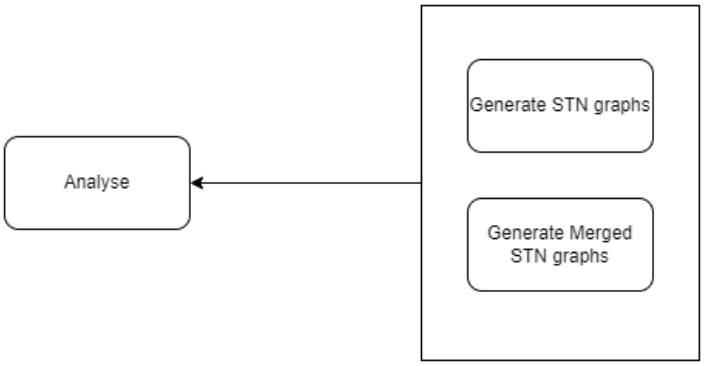


Figure 3.2: Module 2

3.1.1 System requirement analysis

Operating System: Windows, Linux, MacOS

Processor: i5 equvalent or higher

RAM: 8GB or higher

Hard Drive Space: 15MB

3.1.2 Module 1: Running experiments

OBJECTIVE

We run the MPEDE algorithm on several functions in order to get the experimental data on how search optimisation works in these cases.

DESCRIPTION

- First step involves choosing the optimization functions that are used for the analysis. These include Quadric (Schwefel 1.2), Michalewicz, Schwefel 2.26, Salomon, Rana.

Function	Definition and domain
Michalewicz	$f(\mathbf{x}) = -\sum_{i=1}^{D} \sin(x_i)\left(\sin(ix_i^2/\pi)\right)^{2p}, \quad x_i \in [0, \pi]$
Quadric	$f(\mathbf{x}) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2, \quad x_i \in [-100, 100]$
Rana	$f(\mathbf{x}) = \sum_{i=1}^{D} x_i \sin(\alpha)\cos(\beta) + \left(x_{(i+1)\mathrm{mod}D} + 1\right)\cos(\alpha)\sin(\beta)$, $D \geq 2, \quad \alpha = \sqrt{\lvert x_{i+1} + 1 - x_i\rvert}, \quad \beta = \sqrt{\lvert x_i + x_{i+1} + 1\rvert}$, $x_i \in [-512, 512]$
Salomon	$f(\mathbf{x}) = -\cos\left(2\pi\sqrt{\sum_{i=1}^{D} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{D} x_i^2} + 1$, $x_i \in [-100, 100]$
Schwefel 2.26	$f(\mathbf{x}) = -\sum_{i=1}^{D}\left(x_i \sin(\sqrt{\lvert x_i\rvert})\right), \quad x_i \in [-500, 500]$

Figure 3.3: Optimization functions

- Using Multi-population ensemble DE (MPEDE) Wu et al. (2016) for running the experiments. MPEDE uses 3 different mutation strategies.

3.1.3 Module 2: Analysing results

OBJECTIVE

The objective of this module is to analyse the results using STNs on ensemble algorithms and validate them using suitable metrics.

DESCRIPTION

- Gather data from the experiments and format it accordingly.
- Tabulate the information gathered from the data.
- Run the STNs for the gathered data and merge the graphs into a single graph.
- Analyse the graph and compare with tabulated results.

METRICS

- **ntotel** total number of nodes, ntotal, gives an idea of the amount of the search space that was explored.
- **nbest** The number of nodes with best found evaluation, nbest, indicates whether different locations of the search space evaluate to best-found fitness.
- **nend** The number of nodes at the end of trajectories (different than the best nodes), nend, indicates how likely it is for trajectories to end up in sub-optimal location.
- **best-strength** best-strength computes the incoming weighted degree of the best node(s). When there is more than one best node, this metric simply sums their incoming strengths. Values are between zero and one. This metric evaluates to one when all runs end in a best found solution. It provides a measure of the centrality and reachability of the best-found solution(s).
- **nshared** The number of shared nodes, nshared, indicates whether there are solutions or areas of the search space that tend to attract the trajectories of different algorithms.

3.2 System Design

3.2.1 Architecture Diagram

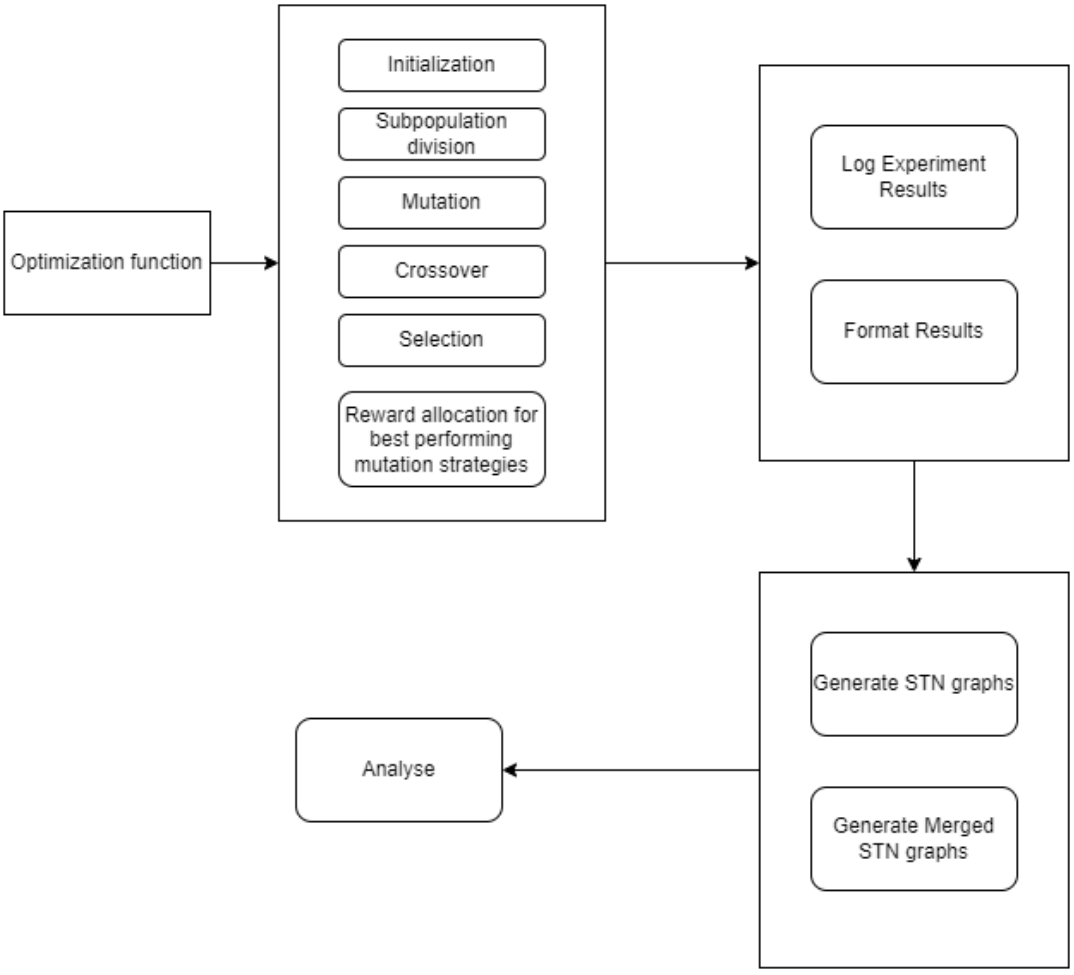


Figure 3.4: Architecture Diagram

Chapter 4

IMPLEMENTATION AND TESTING

Evolutionary algorithms are powerful metahuristics algorithms that are used to solve optimization problems. DE is one of the most efficient evolutionary algorithms currently in use. DE is a population-based stochastic search technique, in which mutation, crossover, and selection operators are utilised at each generation to move the population toward the global optimum. The performance of DE highly depends on the configuration of mutation strategy and control parameters, such as population size , scaling factor and crossover rate. So using different mutation strategies will make sure that the same algorithm can be used for different optimization problems. For this study we are using 5 optimization functions shown in Fig:3.3. For the experimentation we used Rana and Salomon in 3 dimensions, Michalewicz and Schwefel 2.26 in 5 dimensions and Quadric in 10 dimensions.

4.1 Multi-population based ensemble DE (MPEDE)

MPEDE is an ensemble DE algorithm that combines various mutation strategies. In this implementation there are three mutation strategies namely "**current-to-pbest/1**", "**current-to-rand/1**" and "**rand/1**" as these are the robust and most commonly used mutation strategies.

Evidences show that “current-to-pbest/1” with an archive is very competitive in solving complex optimization problems, especially those with unimodal landscapes [67] or after one or more population members discover the global basin in a multimodal landscape. In contrast, the “current-to-rand/1” mutation strategy, which is applied without the aid of crossover operation, is particularly useful in solving rotated problems, as it is rotation-invariant [12]. The employed constituent mutation strategies are listed as below. It should be noted that the “rand/1” and “current-to-pbest/1” mutation strategies are used with the combination of binominal crossover while the “current-to-rand/1” strategy is used without crossover.

Mutation strategy 1: "current-to-pbest/1"

$V_{i,G} = X_{i,G} + F \cdot (X_p best_{,G} - X_{i,G} + X_{r1,G} - X_{r2,G})$

Mutation strategy 2: "current-to-rand/1"

$U_{i,G} = X_{i,G} + K \cdot (X_r 1_{,G} - X_{i,G}) + F \cdot (X_{r2,G} - X_{r3,G})$

Mutation strategy 3: "rand/1"

$V_{i,G} = X_{i,G} + F \cdot (X_{r1,G} - X_{r2,G})$

Initially, each mutation strategy obtains an indicator sub-population and the reward sub-population is randomly assigned to one of the three mutation strategies. Not the population is divided into 2 three indicator sub-population and one reward sub-population, After every certain number of generations, the mutation strategy that performed the best during the previous generations, is given the reward sub-population strategy as a reward. By using these steps, we ensure that the recently best performing strategy will be given more computational resources.

Algorithm 1 MPEDE

Require: four population namely pop1, pop2, pop3, pop4
  for G generations **do**
    $p1 \leftarrow currenttopbest(pop1, best_population)$
    $p2 \leftarrow curenttorand(pop2)$
    $p3 \leftarrow rand(pop3)$
    Find best candidate for current to pbest output

    if gen mod 5 == 0 **then**
      Get best candidate from p1, p2, p3
      Find best performing mutation
      Assign pop4 for the best performing mutation ($pop5 \leftarrow p1 + pop4$)
      Shuffle the pop5
      pop1, pop2 = split pop5
    end if
  end for

[1]

[1] Github Link for implementation

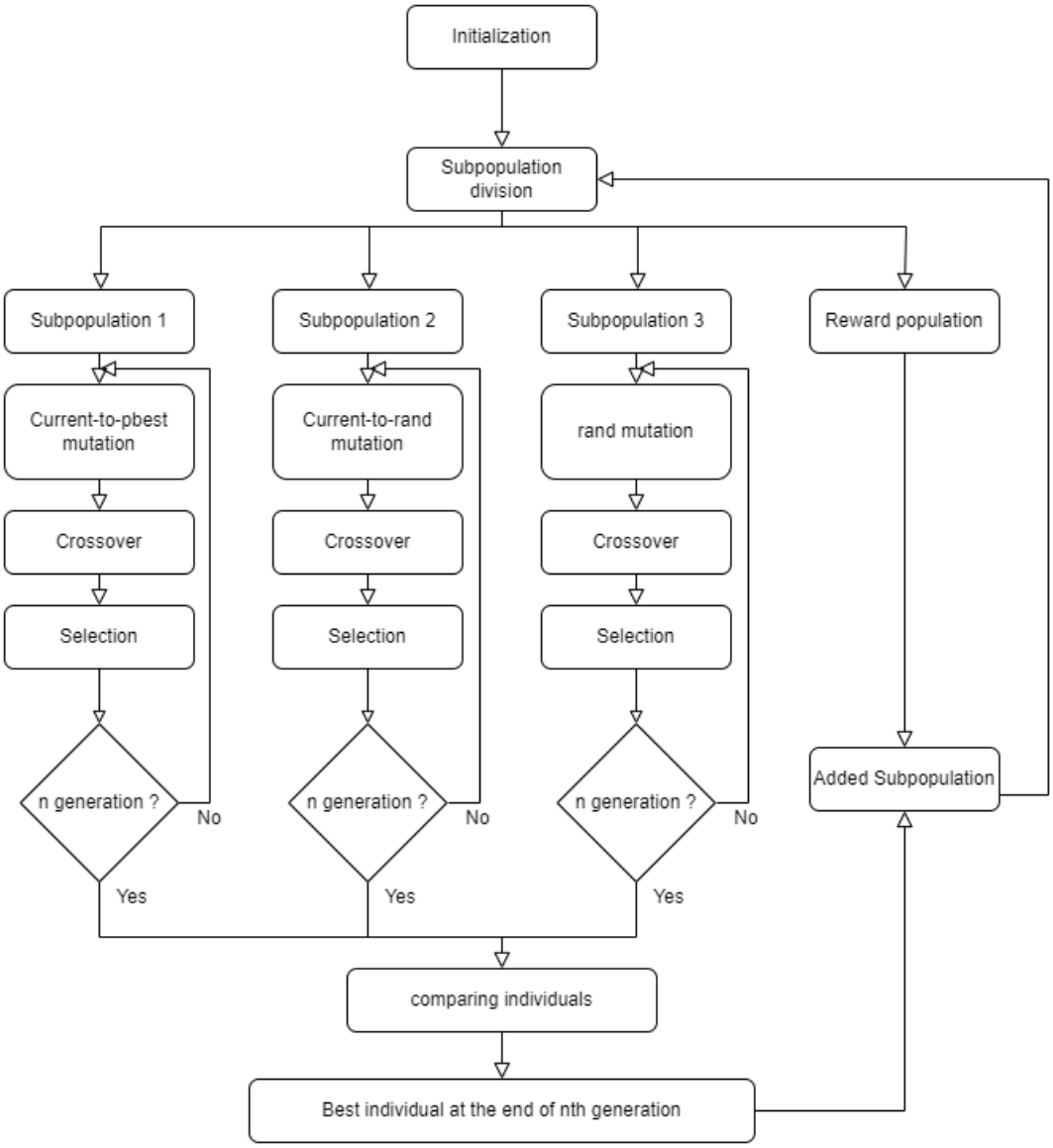


Figure 4.1: Flow Diagram

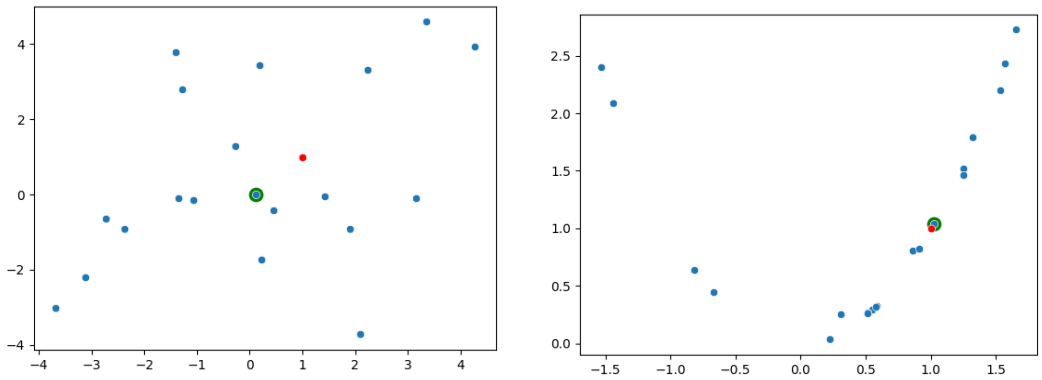
Chapter 5

RESULTS AND DISCUSSION

MPEDE is a state of the art algorithm, we use this as a frame of reference for other differential evolution algorithms. These metahuristic algorithms are very powerful but at the same time these are very stochastic in nature and involves a lot of trial and error. Visualizing the complimentary nature of the ensemble algorithms helps us in analysing the constituent algorithms or mutations in coherence.

Modified Quadric Benchmark function:

$y = 100 \cdot (x_2 - x_1^2)^2 + (1 - x_1)^2$



(a) Initial (b) Final

Figure 5.1: Red is the optimal solution, green is the best solution obtained by the algorithm

Chapter 6

CONCLUSION

In this project we've worked on analysing how ensemble differential evolution algorithms work in the search space. Suitable metrics do help in understanding how the algorithms perform.

However, visualising the working of these algorithms helps in analysing how much of the search space, the algorithms are covering and how they are converging towards the optimal solution. So we've used STNs or Search Trajectory Networks for the visualisation purpose. This form of visualisation has predefined metrics that are very useful for the validation of results.

The ensemble algorithm we've chosen is MPEDE, Multi Population Ensemble Differential Evolution algorithm, an ensemble algorithm with 3 separate mutation strategies. We've seen how MPEDE works on the optimisation functions so far in this project. In this phase we worked on implementing MPEDE and logging the data. In the next phase we will work on generating STNs using the same data and analysing them using suitable metrics.

Chapter 7

FUTURE ENHANCEMENT

Visualizing Ensemble models is very useful and informative, it gives valuable insights on how the different algorithms interact with each other over multiple generations. In our study we have used MPEDE with three different mutation strategies.

We argue, therefore, that STNs can be applied to analyse any metaheuristc. Future work will analyse real-world optimisation problems as well as scenarios where significant performance differences among algorithms are known to exist but are not well understood.

REFERENCES

1. Collins, T. D. (2003). "Applying software visualization technology to support the use of evolutionary algorithms." *Journal of Visual Languages Computing*, 14(2), 123–150.

2. Eiben, A. E. and Smith, J. E. (2015). *Introduction to Evolutionary Computing*. Springer Publishing Company, Incorporated, 2nd edition.

3. Fieldsend, J. E., Chugh, T., Allmendinger, R., and Miettinen, K. (2019). "A feature rich distance-based many-objective visualisable test problem generator." *Proceedings of the Genetic and Evolutionary Computation Conference*, 541–549.

4. Flamm, C., Hofacker, I. L., Stadler, P. F., and Wolfinger, M. T. (2002). "Barrier trees of degenerate landscapes." 216(2), 155–155.

5. Michalak, K. (2019). "Low-dimensional euclidean embedding for visualization of search spaces in combinatorial optimization." *Proceedings of the Genetic and Evolutionary Computation Conference Companion*.

6. Ochoa, G., Malan, K. M., and Blum, C. (2021). "Search trajectory networks: A tool for analysing and visualising the behaviour of metaheuristics." *Applied Soft Computing*, 109, 107492.

7. Ochoa, G., Tomassini, M., Vérel, S., and Darabos, C. (2008). "A study of nk landscapes' basins and local optima networks." GECCO '08, New York, NY, USA, Association for Computing Machinery, 555–562, <https://doi.org/10.1145/1389095.1389204>.

8. Pohlheim, H. (2006). "Multidimensional scaling for evolutionary algorithms—visualization of the path through search space and solution space using sammon mapping." *Artificial Life*, 12, 203–209.

9. Tusar, T. and Filipic, B. (2015). "Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method." *IEEE Transactions on Evolutionary Computation*, 19, 225–245.

10. Wu, G., Mallipeddi, R., Suganthan, P., Wang, R., and Chen, H. (2016). "Differential evolution with multi-population based ensemble of mutation strategies." *Information Sciences*, 329, 329–345 Special issue on Discovery Science.

11. Zelinka, I. and Davendra, D. (2011). "Investigation on relations between complex networks and evolutionary algorithm dynamics." *International Journal of Computer Information Systems and Industrial Management Applications*, 3, 236–247.