Global Journal of Enterprise Information System

DOI: 10.18311/gjeis/2016/15779

A Genetic Algorithm based Approach in Predicting and Optimizing Sickle Cell Anaemia

Amita Jain[1] and Charu Gupta[2*]

[1]Computer Science and Engineering Department, Ambedkar Institute of Advanced Communication Technologies and Research, Government of NCT of Delhi, India; amitajain@aiactr.ac.in

[2]Computer Science and Engineering Department, Bhagwan Parshuram Institute of Technology, GGS Indraprastha University, Delhi, India; charugupta0202@gmail.com

Abstract

A good amount of information is hidden in medical data which can be analyzed using various computational techniques. Meta-heuristics play a vital role in producing optimal or near optimal results to complex problems. Genetic algorithms are a robust adaptive optimization method based on biological principles. The research is important because it is necessary to detect and cure certain diseases like Sickle Cell Anaemia which prove to be fatal many a times, if not taken care of. In the proposed work we present the application of genetic algorithm (differential evolution) to predict sickle cell anaemia and optimize the results. We also propose a new crossover operator in differential evolution. The algorithm gives us optimized values of the following blood components (parameters)- HB, RBC and MCH. With these values it can be shown that patients whose functional value is less than or equal to the best value suffer from sickle cell anaemia. With such an approach for data analysis these patients can be cured on time. Further we conclude that data mining algorithms can make it easier and less time consuming to predict and optimize the parameters.

Keywords: Differential Evolution, Genetic Algorithm, Meta-heuristics, Sickle Cell Anaemia

1. Introduction

Computers have certainly changed and turned our lives in all ways. They have revolutionised all fields of science, be it, chemistry, biology, physics, mathematics, social sciences, engineering. In medicine, computers have changed the way of diagnosis of a disease. On a ground level, when computers are able to give decisive answers to problems in view of given symptoms, it surely reflects the amount of advancement in the field of computer and computing technology. The basic unit of working is the development of a software (can be thought of as several thousand lines of code) which is responsible to carry out these decisions given some input parameters. The design of algorithm plays a vital role in the development of an algorithm. A high level algorithmic framework called as meta-heuristics, which help to build strategies for developing heuristic optimization techniques, are employed. The term was coined by Glover (1986) and combines the Greek prefix meta- (metá, beyond in the sense of high-level) with heuristic (from the Greek heuriskein or euriskein, to search) [15]. For complex problems metahueristics are able to provide a more viable solution. They are best fit to be applied to real life optimization problems.

Genetic algorithm (GA) are a type of metahuetristics which are governed by laws of genetics. GA's find optimal solution to complex problems[1]. In this method, some individuals are generated containing various properties and features[1-3]. In order to diversify the population, mutation and crossover are used to produce generation with different traits[1-3,9,11-13]. These two operations play a vital role in capturing the randomness in data. Several approaches have been used to propose crossover and mutation operators[9,17-21]. The process of reproduction continues till a population of most fit individuals is generated. It is interesting to note that, GAs search the solution space in iterative manner while other derivative methods search only a single point. Moreover, the difference also lies in the way the genetic algorithms and other derivatives use the transition rules. GA's use probabilistic rules whereas deterministic approach use deterministic rules[15]. In this paper, we discuss an application of Differential Evolution[16,18-20],

*Author for correspondence

a popular genetic algorithm to optimize the diagnosis of sickle cell anaemia. Sickle cell anaemia is a type of anaemia where the patient develops crescent (sickle) shaped Red Blood Cells (RBC). This is a genetic disease for which no subtle cure is available. The associated problems with this disease can be minimised if detected early. The medication can also help relieve pain and prevent further aggravation[5]. For more technical details[4,6,14] can be read on sickle cell anaemia

2. Related Work

The concept of evolutionary algorithms (EA) was to apply ideas from the theory of natural selection to navigate through large search spaces[21]. The problem with EA's of "getting stuck" with local optimum was effectively dealt with novel search heuristics. With studies in[9,12,13,17-22] we could understand the novel search strategies such as particle swarm optimization and differential evolution for numerical optimization that were hardly known outside the search heuristics. Improved differential evolution for automatic clustering has been studied in[18]. These methods gave an insight into a different approach where evolutionary technique produced effective results for clustering. Although there are other evolutionary techniques such as particle swarm optimization, simulated annealing, random search that have been studied for a while but differential evolution because of its robustness, ease of implementation and less use of control parameters proved to be a very effective optimization technique[17,19]

With the interest of studying the application of evolutionary strategies a brief look at the basic factors/blood components (RBC, MCH and Hb) that are affected in Sickle cell anaemia are discussed here. MCH, acronym for Mean Corpuscular Haemoglobin, is the average amount of haemoglobin found in red blood cells. It is measured in picograms.

RBC stands for Red Blood Cells. They carry oxygen. Normal RBC range in Males is 4.7 to 6.1 million cells per micro liter (cells/mcL) and in females is 4.2 to 5.4 million cells/mcL. People suffering from anaemia have RBC in the range 2.37-3.73cells/mcl[6,14]

Table 1. Normal and anaemic range of RBC, MCH and Hb[6]

Blood components	Normal Range	Anemic Range
RBC(cells/mcl)	4.2 - 6.1	2.37 – 3.73
MCH (pg)	25.63 - 29.23	26.52 – 32.16
Hb (g/dl)	25.63 - 29.23	6.63-10.87

Hemoglobin abbreviated as Hb or Hgb, is the iron-containing oxygen-transport metalloprotein in the red blood cells. It is measured in g/dl. A short summary of the variation in the values of these factors is given in Table 1.

Genetic algorithms (GA) as a complete entity, in which knowledge of this emerging technology can be integrated together to form the framework of a design tool[2]. In the following section we describe differential evolution and propose a new crossover operator for better optimization of the results.

3. Proposed Work

Differential evolution (DE) is an optimization technique that is suitable for problems where objective functions are non-linear, non-differentiable, noisy, flat, multi-dimensional, have many local minima, multiple constraints or stochasticity. DE has emerged as a strong, robust, simple yet effective optimization technique. The concept with every evolutionary algorithm is fitness or objective function. In our work the objective function is a minimization function that attempts to minimize the distance between the actual and the predicted value. In the following section, we provide a brief description of the working procedure of differential evolution and the proposed modification in the crossover operator.

3.1 Working Procedure

The i^{th} individual vector (chromosome) of the population at time step (generation) t has d components (dimensions) i.e.

$$X_i(t)=[X_{i,1}(t), X_{i,2}(t),.....,X_{i,d}(t)] \quad (1)$$

For each individual $X_k(t)$ of the current population S_{pop}, DE randomly samples three other individuals, i.e., $X_i(t)$,$X_j(t)$ and $X_m(t)$, from the same generation. It then computes the difference of $X_i(t)$ and $X_j(t)$, scales it by a factor f and creates a trial offspring X_{off} by adding the result to $X_m(t)$. Many strategies for creating trial offspring have been summarized in[19]. The strategy can be chosen according to the requirements of the application and we chose to use "DE/best/1". Thus, for n^{th} component of each individual,

$$\overbrace{\begin{matrix} X_{best,n(t)} + f(X_{i,n(t)} - X_{j,n}(t), & \text{if rand}(0,1) < Cr \\ X_{k,n}(t), & \text{otherwise} \end{matrix}}^{X_{off}} \quad (2)$$

If the new offspring, X_{off}, yields a better value of the objective function, it replaces its parent in the next generation otherwise the parent is retained.

Differential evolution is selected for optimization procedure because of its effectiveness and ease of adaptation to the objective function. The scope for improvement in DE lies in the adaption of its scale factor *f* and crossover rate *Cr*. Simple adaptive and self-adaptive variants for *f* and *Cr* have been devised to improve the performance of the algorithm without causing any additional burden[21,22]. Instead of a constant scale factor and crossover rate, studies propose variations in these parameters[18-22]. The random variation in scale factor has been studied in[20] where the scale factor *f* is varied in a random manner in the range [0.5, 1]. The behaviour of DE is influenced both by the mutation and the crossover operator (*F* and *Cr* respectively).

```
Generate S_pop individuals of the initial population pseudo-randomly;

for i=1 to S_pop
        compute f(X_i);
end-for
for i=1 to S_pop

/* mutation*/
select three individuals X_i(t), X_j(t) and X_best(t),
generate rand(0,1);                    /* random number generated for computing i^th scale factor
value*/
compute f_i = 0.5 * (1+rand(0,1));

/* for every i^th individual value, the generated random number gives a different scale factor
value  in the range [0.5,1]*/

compute X_off = X_best,n(t) + f_i (X_i,n(t) – X_j,n(t) );

/* X_best(t) value is chosen from the current population that minimizes the objective function & is
problem dependent.
X_i,n and X_j,n are chosen randomly from the population(forecasted enrollment values)*/

/* crossover*/
X_off = x_off;
for j = 1 to n
                generate rand(0,1);
                compute Cr_i = 0.5 [(f_av/f_min)- f_i + rand(0,1)];
/* f_av is the mean scale factor value (0.75) and f_i is the scale factor value for i^th individual*/

                if rand (0,1) < Cr_i ;
X_off = X_i;
                end-if
end-for

/*selection*/
if f (X_off) < f (X_i)

/* if the offspring yields a better fitness value then it replaces the parent in next generation*/

        X_i = X_off ;
end-if
end-for
```

Figure 1. Differential Evolution pseudocode.

3.2 Proposed Modification in Crossover Operator

Crossover operation is used to increase the potential diversity of a population. The two basic types of crossover variants are-binomial and exponential. In binomial case the *Cr* parameter explicitly determines the probability of a vector to be replaced by a mutated one whereas in exponential the *Cr* parameter decides how many vectors will be mutated. The detailed description of these variants can be found in[22]. A Linearly decreasing crossover rate Cr with time from 1.0 to 0.5 has been studied in[18]. The time variation of *Cr* was expressed in terms of the maximum & minimum crossover rate, maximum and minimum number of iterations. We propose to vary this crossover rate in a random manner using

$$\mathrm{Cr_i} = 0.5 * \left[\frac{\mathrm{f_{av}}}{\mathrm{f_{min}}} - \mathrm{f_i} + \mathrm{rand}(0,1) \right] \quad (3)$$

Here, f_{av} is the mean value of scale factor, f_{min} is the minimum value and f_i is the value of scale factor for i^{th} candidate solution. If the adaptive crossover rate of the i^{th} solution is close to Cr_{min} (Cr_{min} = 0.5) then Cr_i tends to be a small value and if it it close to C_{rmax} (Cr_{max} =1.0) then Cr_i tends to be large. So, we use this variable crossover rate Cr_i for the i^{th} candidate solution for iteration *k*.

For the sake of clarity, the pseudo-code highlighting working principles of the modified DE are shown in figure 1. As usual, the main phases of modified DE are – mutation, crossover and selection. During mutation, three individual vectors $X_i(t)$, $X_j(t)$ and $X_m(t)$ are randomly chosen from the current population S_{pop}, where the difference of $X_i(t)$ & $X_j(t)$ is scaled by a variable scale factor f_i and the result is added to $X_{best}(t)$ [$X_m(t)$] which is the best individual vector, highly dependent on problem under consideration, (Qin *et al.*, 2005) yielding the best fitness value for i^{th} population vector. After mutation, the crossover rate is computed for the i^{th} vector using eq. (4). If the offspring vector (X_{off}) successfully satisfies the condition in eq. (3) then, it is admitted to the next generation otherwise the parent vector is retained.

4. Experimental Analysis and Results

Dataset for 37 patients where the values of HB (haemoglobin values), RBC(red blood cells) value and MCH(Mean Corpuscular Haemoglobin) values[7] are taken. Dataset which contains upper bounds and lower bounds of HB, RBC and MCH of healthy people and those suffering from sickle cell anaemia[6].

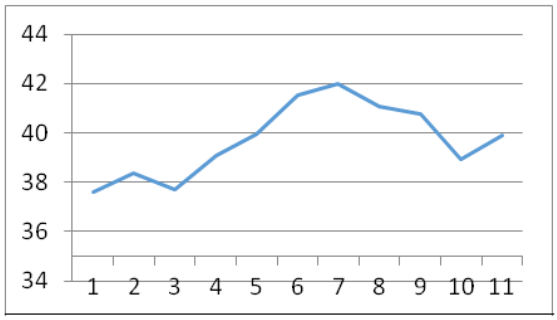

Figure 2. *(Generation Number(x axis) vs Average Fitness(y axis)).*

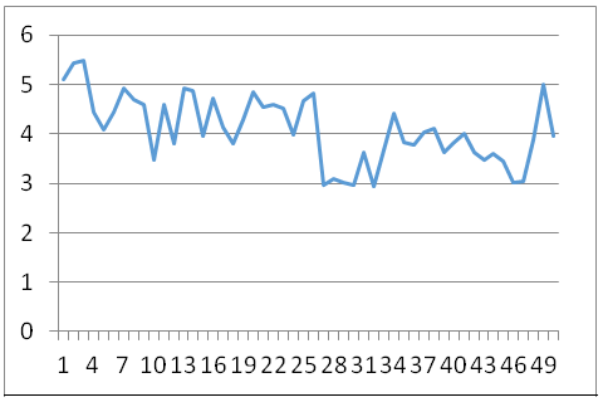

Figure 3. *(Generation Number(x axis) vs Standard Deviation (y axis)).*

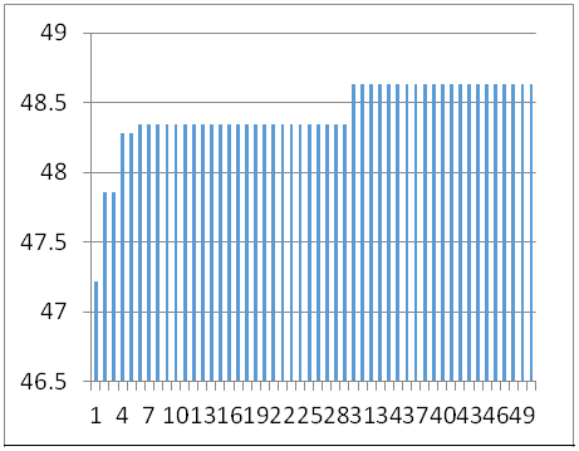

Figure 4. *(Generation Number(x axis) vs Best Value (y axis)).*

The results observed are depicted by the following graphs:

Where **every 1 unit of x axis = 5 generation no.**

The graph shows how average fitness varies with generation number implying that for every generation number a new fitness is calculated according to the objective function.

In figure 3 the graph shows how standard deviation varies with generation number which implies that for every generation number, a new mean is calculated and the deviation changes.

In figure 4 the graph shows how best value varies with generation number implying that for different generation numbers .i.e. for a particular range the best value remain same until a new and more optimized result is obtained after repetitive iterations.

The results include calculation of optimized values of HB, RBC and MCH. A best value of the object function is generated. Function value of dataset of patients is calculated. With the above results patients suffering from sickle cell anaemia can be identified. Patients whose function value is less than or equal to the best value suffer from sickle cell anaemia.

5. Conclusion

The research considers different attribute values involved in case of sickle cell anaemia patients. Optimized values of different attributes are calculated using differential evolution and according to an objective function the value using these attributes is calculated.

The research uses differential evolution technique i.e the steps are followed to find out substantial values of certain attributes for a person suffering from sickle cell anaemia. Differential evolution is best suited method for this purposed as it provides the best results in calculating optimized values. We also propose a new crossover operator for understanding the stochasticity.

In future the algorithm could be combined with different data mining techniques like neural networks or techniques like artificial intelligence to obtain more accurate results.

6. References

1. Zbigniew M. Genetic Algorithms + Data Structures = Evolution Programs. Third Edition, Springer. 1996.
2. Man KF, Tang KS, Kwong S. Genetic Algorithms: Concepts and Application. IEEE Xplore. 1996; 43(5):519–34.
3. Beasley D, Bull R, Martin RR. An Overview of Genetic Algorithms Part2. University Computing. 1993; 15(4):170–81.
4. Steinberg MH, Chui DH, Dover GJ, Sebastiani P, Alsultan A. Fetal hemoglobin in sickle cell anaemia a glass half full? US National Library of Medicine National Institutes of Health. 2014; 123(4):481–5. https://doi.org/10.1182/blood-2013-09-528067
5. Pediatr IJ. How to Reach Rapid Diagnosis in Sickle Cell Disease? US National Library of Medicine National Institutes of Health. 2010; 20(1):69–74.
6. Pathol JC, Sally D, Joan H, Milica B. Iron deficiency in sickle cell anaemia. US National Library of Medicine National Institutes of Health. 1983; 24(5):432–7.
7. Whitley D. A genetic algorithm tutorial. Statistics and Computing. 1994; 4(2):65–85. https://doi.org/10.1007/BF00175354
8. Akbari Z, Reza A. A multilevel evolutionary algorithm for optimizing numerical functions. International Journal of Industrial Electronics and Control. 2011; 2(3):419–30. https://doi.org/10.5267/j.ijiec.2010.03.002
9. Eiben AE, Raue PE, Ruttkay Z. Genetic algorithms with multi-parent recombination. Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature. 1994. p. 78–87. https://doi.org/10.1007/3-540-58484-6_252

10. Chuan-Kang T. On the Mean Convergence Time of Multi-parent Genetic Algorithms Without Selection. Advances in Artificial Life. 2005; 56(3):403–12.
11. Genetic Algorithm. Available from: http://en.wikipedia.org/wiki/Genetic_algorithm [Date accessed : 2-12-16]
12. Mitchell M. An Introduction to Genetic Algorithms. MIT Press. 1999.
13. Garg R, Mittal S. Optimisation Of Genetic Algorithm. International Journal of Advanced Research in Computer Science and Software Engineering. 2014; 4(4):534–8.
14. Sickle Cell Disease. Available from: http://en.wikipedia.org/wiki/Sickle-cell_disease [last accessed on: 2-12-15]
15. Ghaheri A, Shoar S, Naderan M, Hoseini SS. The Applications of Genetic Algorithms in Medicine. Oman Medical Journal. 2015; 30(6):406–16.
16. Paterlini S, Krink T. Differential evolution and particle swarm optimization. Computational Statistics and Data Analysis. 2006; 1220–47. https://doi.org/10.1016/j.csda.2004.12.004
17. Paterlini S, Minerva T. Evolutionary approaches for cluster analysis. In scomputing applications, Bonarini A, Masulli F, Pasi G editors. Berlin, Germany: Springer-Verlag. 2003; 167
18. Das S, Abraham A, Konar A. Automatic clustering using an improved differential evolution algorithm. IEEE Transactions on Systems, Man and Cybernetics- Part A: Systems and Humans. 2008a; 38(1). https://doi.org/10.1109/TSMCA.2007.909595
19. Das S, Abraham A, Konar A. Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspective. Studies in Computational Intelligence, Springer. 2008b; 116:1–38. https://doi.org/10.1007/978-3-540-78297-1_1
20. Das S, Konar A, Chakraborty UK. Two improved differential evolution schemes for faster global search. In Proceedings of ACM-SIGEVO GECCO, Washington, DC. 2005; 991–8. https://doi.org/10.1145/1068009.1068177
21. Tvrdik J. Differential evolution with competitive setting of control parameters. Task quarterly. 2007; 11(1-2):169–79.
22. Zaharie D. A comparative analysis of crossover variants in differential evolution. In Proceedings of the International Multi Conference on Computer Science and Information Technology. 2007. p. 171–81.
23. Storn R, Price K. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Space. Journal of Global Optimization. 1997; 341–59. https://doi.org/10.1023/A:1008202821328

Citation:

Amita Jain and Charu Gupta
"A Genetic Algorithm based Approach in Predicting and Optimizing Sickle Cell Anaemia",
Global Journal of Enterprise Information System. Volume-8, Issue-4, October-December, 2016. (http://informaticsjournals.com/index.php/gjeis)

Conflict of Interest:

Author of a Paper had no conflict neither financially nor academically.