

## Convalesce Optimization for Input Allocation Problem Using Hybrid Genetic Algorithm

<sup>1</sup>Mamta Madan and <sup>2</sup>Sushila Madan

<sup>1</sup>Department of Information Technology, Vivekananda Institute of Professional Studies, (GGSIPU), India

<sup>2</sup>Department of Computer Science, Lady Shri Ram College, Delhi University, India

---

**Abstract: Problem statement:** The purpose of this study was to describe categories of hybrid genetic algorithm and validate that the hybrid genetic algorithm converges to the optimal solution rather than a near optimal solution so that Hybrid Genetic algorithms can be used to solve real world problems and receive significant interest. **Approach:** We implemented the input allocation problem for a manufacturing unit firstly with pure genetic algorithm using Matlab's GA tool and then compared the results with hybrid genetic algorithm. **Results:** We observed that the results from applying only pure genetic algorithm to the problem were near optimal whereas when solved using hybrid genetic algorithm the results were significantly better and were optimal. **Conclusion:** The results presented by pure genetic algorithm and hybrid genetic algorithm are significant and validate that the hybrid genetic algorithm converges to the optimal solution rather than a near optimal solution.

**Key words:** Genetic algorithm, Lamarckian search, Baldwinian search, Matlab

---

### INTRODUCTION

Genetic Algorithms (GAs) are search algorithms that are conceptually based on the methods that living organisms adapt to their environment. These methods, known as natural selection or evolution, combine the concept of survival of the fittest with a structured yet randomized information exchange to form a search algorithm with some of the innovative flair of human search. In each generation, a new set of string structures is created from the fittest strings from the previous generation and occasionally a randomly altered new part. This process of exploiting historical data allows the GA to speculate on new search points and producing better solutions. Genetic algorithms were initially developed by John H. Holland, a professor of psychology and computer science at the University of Michigan. His research focused on what he called complex adaptive systems. Since their development, Genetic Algorithms have been used as optimization tools for complex problems that involve numerous variables or involve combinations of linear and, non-linear equations. As an optimization tool, the Genetic Algorithm attempts to improve performance leading to an optimal solution. In this process, there are two distinct steps (1) the process of improvement and (2) reaching the optimum itself. Of these two steps, the most important is the process of improvement (Goldberg,

1989). The genetic algorithm can be visualized as follows:

1. Produce an initial generation of Genomes using a random number generator.
2. Determine the fitness of all of the Genomes.
3. Determine which Genomes are allowed to reproduce.
4. Crossover the Genome pairs in the allowable population.
5. Pick the 2 fittest Genomes of the 2 parents and 2 children resulting from the crossover and add them to the next generation.
6. Produce random mutations through the next generation population.
7. Calculate the next generation's fitness, if achieved move to 8 else back to step 2.
8. Finish.

Genetic algorithm promises convergence but not optimally, although there is no guarantee of optimality, exponential convergence is assured. If Genetic Algorithm is run several times, it will converge each time possibly at different optimal chromosomes. Typically, Genetic Algorithm is coupled with a local search mechanism to find the optimal chromosome in the region. So if a hybrid genetic algorithm is used, it becomes easy to reach to the optimal solution. Hybrid genetic algorithm is discussed below in detail.

---

**Corresponding Author:** Mamta Madan, Department of information Technology, Vivekananda Institute of Professional Studies, (GGSIPU), India

**Hybrid genetic algorithm:** Hybrid Genetic Algorithm (HGA) is a novel genetic algorithm on some additional heuristics which improves the convergence rate of the algorithm as well as finds better solution. Genetic Algorithms are not good at identifying the optimal value of a chromosome for a problem but do very well in identifying the regions where those optima lie. Therefore a hybrid Genetic Algorithm is used-every ten generations, the user anneals the best 10% of the population. This has the effect of moving the top chromosomes in that generation to the local maximum in their region. Including a local search can also help combat with the genetic (Asoh and Muhlenbeim, 1994) Drift problem. Although a genetic algorithm can rapidly locate the region in which the global optimum exists, they take a relatively long time to locate the exact local optimum in the region of convergence. Therefore a combination of a genetic algorithm and a local search method can speed up the search to locate the global optimum.

**Categories of hybrid genetic algorithm:** When a local search method is added within a genetic algorithm, the performance of the Genetic Algorithm increases. There are several issues which should be taken care of when designing a hybrid genetic algorithm. The way by which information through local search is utilized within a hybrid genetic algorithm has a great impact on the performance of the search process. Two basic approaches based on biological learning models have been adopted to utilize local information: (a) The Lamarckian approach (Ei-Mihoub *et al.*, 2006) (b) The Baldwinian Approach. There is an opportunity in hybrid optimization to achieve to capture the best of both schemes (Lobo and Goldberg, 1997). Both of these schemes are described below.

**Lamarckian learning:** The Lamarckian approach is based on the inheritance of acquired characteristics obtained through learning. This approach forces the genetic structure to reflect the result of local search. The genetic structure of an individual and its fitness are changed to match the solution found by a local search method. In the Lamarckian approach, the local search method is used as a refinement operator that modifies the genetic structure of an individual and places it back. Lamarckian evolution can accelerate the search process of genetic algorithm (Whiteley *et al.*, 1994), on the other hands can disrupt the schema processing which can badly affect the exploring abilities of genetic algorithm, which may lead to premature convergence. Most of the hybrid genetic algorithms that repair chromosomes to satisfy constraints are Lamarckian and

the technique has been particularly effective in solving Travelling Salesman Problem (Julstrom, 2005).

**Baldwinian learning:** The Baldwinian learning allows an individual fitness to be improved by applying a local search, without changing the genotype. In this way it propagates its structure to next generation and thus follows the normal process of evolution. The Baldwinian approach in contrast to the Lamarckian approach does not allow parents to pass their acquired characteristics but only the fitness is retained. A local search method in the Baldwinian approach is usually used as a part of individual's evaluation process. The local search uses local knowledge to produce a new fitness score that can be used by global genetic algorithm to improve the results. The Baldwinian effect consists of the following two steps (a) Learning gives individuals the chance to change their phenotypes to improve the performance. (b) The second step is called genetic assimilation i.e., learning can accelerate the genetic acquisition of learned traits. Hinton and Nolan (1987) illustrated how the Baldwin effect can transform the fitness landscape of a difficult optimization problem into a less difficult one and how genetic search is benefited.

The selection of any form of learning in a hybrid genetic algorithm has a great impact on its performance. Various researchers have already worked on this and showed how it affects the performance on an optimization problem. Gruau and Whitley (1993) compared Lamarckian, Baldwinian and pure genetic algorithm in evolving the architecture that learn Boolean functions. They also concluded that any form of learning is better than pure genetic algorithm.

## MATERIALS AND METHODS

We used Matlab's Genetic Algorithm and Direct Search Toolbox both for implementing Genetic Algorithm and Hybrid Genetic Algorithm on Input Allocation problem. The fitness function was customized to this specific optimization problem. This fitness function will vary from one research problem to another. Results were achieved with different parameters such as objective function, display plots, Local search techniques, one may find about these parameters in Matlab GA toolbox documentation.

## RESULTS

Let us consider an input-Allocation Problem of a manufacturing concern which produces a product

consisting of two raw materials say A1 and A2. The production function is estimated as:

$$Z = f(x_1, x_2) = 3.6x_1 - 0.4x_1^2 + 1.6x_2 - 0.2x_2^2$$

Where:

- Z = The quantity (in tons) of the product produced
- $x_1$  and  $x_2$  = Designate the input amounts of raw materials A1 and A2

The company has RS 50,000 to spend on these two raw materials. The unit price of A1 is Rs. 10,000 and of A2 is Rs 5000. Determine how much input amounts of A1 and A2 be decided so as to maximize the production outputs (Taha, 1996).

**Solution:** We have used Genetic Algorithm and Direct Search Toolbox of Matlab. We need to create a fitness function which is given below:

- Fitness Function for the above Maximization problem:

$$\begin{aligned} \text{function } y &= \text{simple\_fitness}(x) \\ y &= 0.4 * x(1) * x(1) + 0.2 * x(2) * x(2) - 3.6 * x(1) - 1.6 * x(2) \end{aligned}$$

- Constraint Function for the Input Allocation Problem: This problem has certain constraints which also has to be specified which is given below:

$$\begin{aligned} \text{function } [c, ceq] &= \text{simple\_constraint}(x) \\ c &= 2 * x(1) + x(2) - 10; \\ ceq &= [] \end{aligned}$$

To execute this problem on Genetic Algorithm and Direct Search Toolbox, we need to create the function handler for the fitness function and constraint function which is created as follows on the command line:

ObjectiveFunction=@simple\_fitness

ConstraintFunction=@simple\_constraint

There are parameters which needs to be set e.g., Nvars = 2, which is the number of variables which is 2 (A1 and A2) for our problem. Set crossover as one point crossover. Set selection as Tournament Selection. In the stopping criterion fix number of population as 500. Set stall limit as infinity. Set plot interval to be best fitness so that it can plot only best fitted individual. After setting all these parameters for genetic algorithm, the results are shown in Fig. 1.

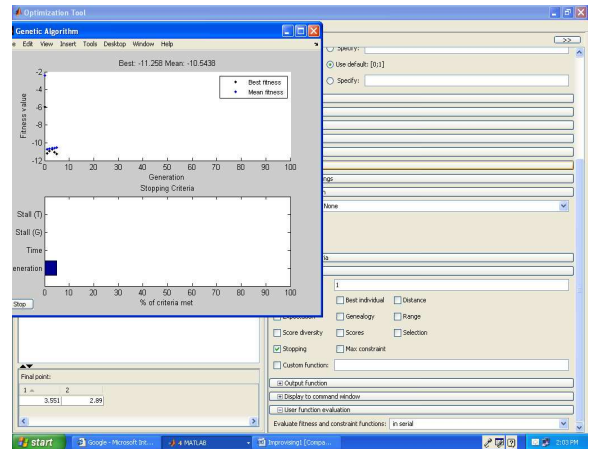


Fig. 1: Result with genetic algorithm

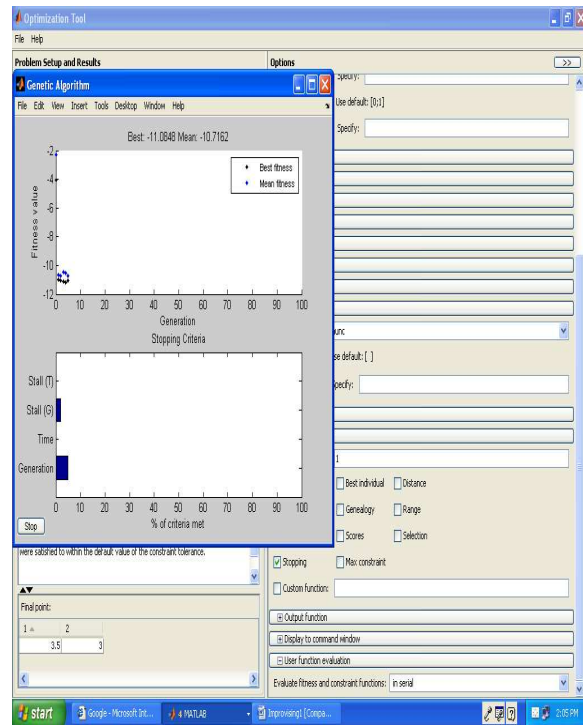


Fig. 2: Results with hybrid genetic algorithm

The solution we got from this toolbox is:  $x_1 = 3.551$  and  $x_2 = 2.89$ . The function value at these points came out to be 10.54 tons. Thus the solution was close to optimal solution.

As we know that we are able to converge better and faster if we use hybrid genetic algorithm. So with these same parameters we incorporated the function for hybrid genetic algorithm with the name as fminunc. This function would automatically be activated where genetic algorithm terminates. The results are shown in Fig. 2.

For this problem the maximum function value evaluated is 10.7 tons which is the expected value. The value for variables were  $x_1 = 3.5$  and  $x_2 = 3.0$  which was the exact solution for the problem (Goldberg, 1998).

### DISCUSSION

As per problem statement in experimental study section, the material required for the manufacturing company with Pure Genetic Algorithm came to be  $A_1 = 3.551$  and  $A_2 = 2.89$ . The profit calculated was 10.4 tons which was not the optimal value as from the source of problem statement. The result for the same manufacturing unit for hybrid Genetic Algorithm was  $A_1$  as 3.5 and  $A_2$  as 3.0, profit come out to be 10.7 tons which was the exact solution expected. The above results demonstrates that Genetic Algorithm are able to reach to near optimal solution while if it mixed with any local search i.e., if it is made hybrid genetic algorithm, it converges to the optimal result.

### CONCLUSION

In this study, we have implemented hybrid genetic algorithm and try to explain how it can improve the efficiency of the given problem and produce an optimal instead of near optimal solution. This study focuses on the use of a local search algorithm with genetic algorithm to improve results. The ability of genetic-local hybrid to solve hard problems quickly depends on the way of utilizing local search information and on the balance between local and genetic algorithm. The approaches in the study (Goldberg, 1998) shows that hybridizing is one possible way to solve hard problems quickly, reliably and accurately without any human intervention. The basic success lies in choosing of how we merge a Local algorithm with the genetic algorithm.

### REFERENCES

Asoh, H. and H. Muhlenbein, 1994. On the mean convergence time of evolutionary algorithms without selection and mutation. *Lecture Notes Comput. Sci.*, 866: 88-97. DOI: 10.1007/3-540-58484-6\_253

Ei-Mihoub, T.A., A.A. Hopgood, L. Nolle and A. Batterby, 2006. Hybrid genetic algorithm a review. *Eng. Lett.*, 13: 124-137. <https://www.dora.dmu.ac.uk/handle/2086/1173>

Goldberg, D.E., 1989. *Genetic Algorithm in Search, Optimization and Machine Learning*. Addison-Wesley, Hardcover, ISBN: 0201157675, pp: 372.

Goldberg, D., 1998. *The race, the hurdle and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity*. University of Illinois at Urbana-Champaign, Urbana, IL 61801, IlliGAL Report No. 98007. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.48.4758>

Gruau, F. and D. Whitley, 1993. Adding learning to cellular development of neural network: Evolution and Baldwinian effect. *Evolut. Comput.*, 1: 213-233.

Hinton, G. and S.J. Nolan, 1987. How learning can guide evolution. *Complex Syst.*, 1: 495-502.

Julstrom, B.A., 2005. Comparing Darwinian, Baldwinian and Lamarckian search in a genetic algorithm for the 4-cycle problem. *Proceeding of the Genetic and Evolutionary Computation Conference*, Hardback, pp: 324. <http://sciencestage.com/d/1328805/comparing-darwinian-baldwinian-and-lamarckian-search-in-a-genetic-algorithm-for-the-4-cycle-problem-.html>

Lobo, F.G. and D.E. Goldberg, 1997. Decision making in a hybrid genetic algorithm. *Proceeding of the IEEE International Conference on Evolutionary Computation*, IEEE Xplore Press, USA., pp: 121-125. DOI: 10.1109/ICEC.1997.592281

Taha, H.A., 1996. *Operations Research An Introduction*. 6th Edn., Prentice Hall, ISBN: 10: 0132729156, pp: 916.

Whiteley, D., S. Gordon and K. Mathias, 1994. Lamarckian Evolution, the Baldwinian Effect and Function Optimization. In: *Parallel Problem Solving from Nature*, Davidor *et al.* (Eds.). Springer-Verlag, Berlin, pp: 6-15.