INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

A Survey on Hybrid Genetic Algorithm

Ms. Meenu[1] **, Mr. Amit Verma**[2]

[1]M.Tech Student, [2] Assistant Professor & Head
CSE Department
[1,2]Samalkha Group of Institution, Kurukshetra University

Abstract*: Genetic algorithms are able to combine other techniques with it to find the best of the combination. In this paper, hybrid genetic algorithms are reviewed by reviewing the different ways in which the local search method and a genetic algorithm can be combined for optimality. Basically local search and genetic algorithm are two complement solutions. Genetic algorithms performs good in finding global searching because they are capable of quickly finding promising regions, but they take relatively long time to find the optima in those regions. Local search ability to find the local optima with high accuracy complements the genetic algorithms global view of the solutions space.*

Keywords: *Evolutionary computation, Genetic Algorithm, Genetic Local search algorithms, Hybridization, Hybrid Genetic Algorithms.*

1. INTRODUCTION OF GENETIC ALGORITHM

The term "genetic algorithm'" (GA) is applied to any search or optimization algorithm that is based on Darwinian principles of natural selection. Genetic Algorithm is a population-based search and optimization method which mimics the process of natural evolution. Genetic Algorithms (GAs) were invented by John Holland in the 1960s and were developed by Holland (1975). Holland's GA is a method for moving from one population of "chromosomes" to a new population by using a kind of "natural selection" together with the genetics inspired operators like crossover, mutation, and inversion as explained in Fig 1. A chromosome contains a group of numbers that completely specifies a candidate during the optimization process. For example, when finding the root of a polynomial, the candidates are complex numbers. One choice of chromosome could consist of two numbers - the real part and the imaginary part - to completely specify a candidate. The selection operator chooses those chromosomes in the population that will be allowed to reproduce, and on average the fitter chromosomes produce more offspring than the less fit ones. Crossover exchanges subparts of two chromosomes, roughly mimicking biological recombination between two chromosomes For example, a child of two parents *2+3i* and *5+4i* could be *2+4i*; mutation randomly changes the allele values of some locations in the chromosome, a chromosome representing *2+5i* after mutation, could represent *2+5.06i*; and inversion reverses the order of a contiguous section of the chromosome, thus rearranging the order in which genes are arrayed. Typically, genetic algorithms use crossover, mutation and reproduction to provide structure to a random search.

In this paper, It is specified what the term Hybridization means in genetic algorithms. The review of genetic-local hybrid algorithm provides a view to the factors affecting the performance of such type of hybrid algorithms.

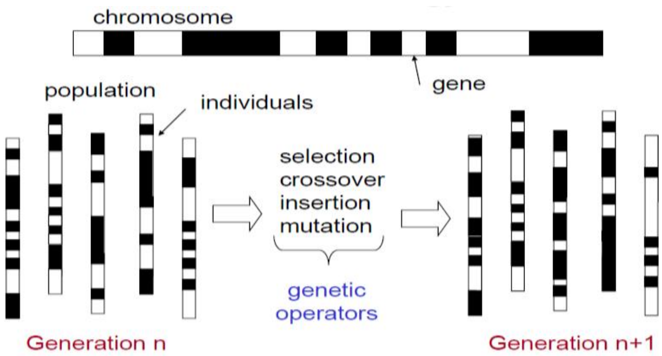

Figure 1: GA Terminology

Those factors are discussed and a framework is suggested with some parameters which need to be taken into consideration when one designs a hybrid algorithm with local and global search methods.

2. HYBRID GENETIC ALGORITHM

The performance of a genetic algorithm depends on the mechanism for balancing the two conflicting objectives, Exploitation (Exploiting best solutions found so far) and Exploration (exploring the search space for promising solutions). The power of genetic algorithms comes from their ability to combine both exploration and exploitation in an optimal way. The difficulty of finding the best solution in the best found region accounts for the genetic algorithm operator's inability to make small moves in the neighborhood of current solutions. So if one uses some local optimization algorithm for making good balancing between global exploration and local exploitation, then the algorithm can easily produce solutions with high accuracy. Genetic algorithms are very fast to locate the region where the global optimum lies, but they take long time to find the exact local optima in a region. So a combination of genetic algorithm and local search method is applied, which can speed up the search for finding the global optima. Incorporating a local search within a genetic algorithm can improve the search performance on the condition that their roles cooperate to achieve the optimization goal.

There is an opportunity in hybrid optimization to capture the best of both (Genetic Algorithm and Local Search) schemes [1]. This opportunity depends on the design details of the hybrid genetic algorithm. There are several issues that need to be taken into consideration when designing a hybrid genetic algorithm. The hybrid algorithm should strike a balance between exploration and exploitation to solve global optimization problems.

One can apply GA initially to find the region where optima exist, and then refine those regions with local search to find the exact global optima. Performing local search on the GA's population can maintain diversity and reduce the problems like genetic drift [9].

The distance preserving crossover (DPX) to produce feasible solutions to solve TSP without losing diversity is proposed [2]. They used the non- sequential 4- change as a mutation operator for the same reason. Cycle crossover (CX), order crossover (OX), matrix crossover (MX), modified order crossover (MOX), edge recombination crossover (ERX), 2-opt operator, 3-opt operator and or-opt operators are examples of crossover and mutation operators which have been developed for TSP.

Combination of genetic algorithm with a cut- saturation algorithm was designed for the backbone design of communication networks [3]. They use a uniform crossover operator with a K-node-connectivity repair algorithm to repair infeasible offspring.

Ant colony optimization model was used for continuous search spaces as local search method to improve the quality of the solutions produced by a genetic algorithm in order to solve a real-world, heavily constrained, engineering design problem [4]. Before the algorithm begins it is to be determined the location of the nest. It should be a point in the search space which seems promising for free local search exploitation. They suggest finding it by utilizing a niching GA or a related strategy. Next they define a search radius R, which determines the extent of the subspace to be considered in each generation (cycle). Then initialize A(t) sends ants in various directions at a radius not greater than R; evaluate A(t) is a call of the objective function for all ants; acid_trail A(t) is proportionally (to the ants' fitness) adding trail quantity to the particular directions the ants have selected, send_ants A(t) sends ants by selecting directions using a Roulette wheel selection on the trail quantity and making a random step from the location of the best previous ant that have selected the same direction, evaporate A(t) is decrementing the trail.

Success of Genetic algorithm primarily depends on Initial population [1]; Beam Search (BS) was integrated with genetic algorithm to seed the initial population. They started with BS and find the best product line design, and use it in initial population as a member, and then generate remaining N-1 population in a random fashion. Then proceed as with genetic algorithm termination when stopping condition met. They proposed eight different types of combination of Regular GA, mutation, seeding with BS, GASM (GA with standard mutation), GASSM (GA with sorted representation and standard mutation), GAHM (GA with hybrid mutation), GASHM (GA with sorted representation and Hybrid mutation), GASMBS (GA with standard mutation and seeding), GASSMBS (GA with sorted representation, standard mutation and seeding), GAHMBS (GA with hybrid mutation and seeding), GASHMBS (GA with sorted representation, hybrid mutation and seeding). In this, they found that lower values of mutation will resulted in premature convergence, however, higher mutation (higher than 0.04) resulted in lack of significant improvement due to too much of random variation. Also they noted that all integrated techniques find their candidate earlier than the pure genetic algorithm based methods.

Evolutionary Local Search Algorithm (ELSA) is used in GSAT algorithm by incorporating Flip Heuristic Local search algorithm [5]. Flip heuristic is a local search algorithm that takes a randomly generated assignment as an input and yields another assignment that cannot be improved by flipping any variable. Evolutionary local search algorithm (ELSA) is a simple genetic algorithm with a local search method. Evolutionary local search algorithm consists of two main parts: local search and simple genetic algorithm. In this structure, local search exploit the promising solutions, as the genetic algorithm explores the search space for finding more promising solution candidates (selection and crossover) and at the same time avoiding from local maximum points (mutation). They use Tournament selection scheme between two individuals and uniform crossover is applied on two individuals with probability of 0.8. Mutation is applied on a gene of an individual with probability of 1/number of variables. Generational without elitism replacement scheme is used and population size is determined as 100. Whereas, with hybrid algorithm ELSA, the simple genetic algorithm part is implemented as given above except that population size is determined as 10 (for the sake of local search), uniform crossover is always applied, and mutation rate is taken as 0.5 (a real restart). GSAT and FH constitute the local search part with the basic properties as given above. Local search algorithms are applied in simple genetic algorithm after initialization, crossover, and mutation. According to experimental results, SGA gives very poor results in terms of Success Rate, Average Flip Evaluations to a Solution, and Average Flip to a Solution. GSAT and FH give better results in the viewpoint of Success Rate on the test instances where the number of variables is less than 100, and evolutionary local search algorithms become competitive with the local search methods on the test instances where the number of variables is higher than or equal to 100. Also, evolutionary local search algorithms find solutions with less number of accepted and evaluated flips on these test instances.

3. Study of Genetic and Local Search Algorithms

Hybrid genetic algorithms are based on the complementary view of genetic algorithms and local

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

search methods, where one is known as global search method and other is for locally optimizing the population. So there are several ways in which a local search can be incorporated in genetic algorithms or it is complemented to genetic algorithms

a. **Capability Enhancement**
b. **Optimizing the Control Parameters**

Capability Enhancement

Genetic Algorithm can be combined with local search methods in many different ways to optimize the overall search process. When genetic algorithm is combined with a search algorithm which is having local knowledge of problem, the overall search ability can be enhanced. The enhancement can be in terms of Quality and/or efficiency. This performance can also be improved by ensuring production of feasible solutions in the case of highly constrained problems. The efficiency of a local search in reaching a local optimum integrates the efficiency of a genetic algorithm in isolating the most promising basins of the search space. Therefore, incorporating a local search into a genetic algorithm can result in an efficient algorithm. The efficiency of the search can be enhanced in terms of the time needed to reach the global solution, and/or the memory needed to process the population.

Firstly, In Genetic Algorithm design efficiency is a major concern in terms of the time needed to reach a solution of desired quality.

Secondly, Population size is crucial in a genetic algorithm. It determines the memory size and the convergence speed in serial genetic algorithms and affects the speed of search in the case of parallel genetic algorithms. Efficient population sizing is critical for getting the most out of a fixed budget of function evaluations. The gambler's ruin model was shown, which was used to estimate the population size of genetic algorithms [6]. This model was used to show that population size depends on two parameters, which can be affected by incorporating local search. The two parameters represent the standard deviation of the population and the signal difference between the best and second best building blocks. If a local search method is incorporated in such a way as to reduce the standard deviation of the population and to increase the signal difference between the best and the second best chromosome, the resulting hybrid can be efficient even with small population sizes. The combined effect of probability of local search and learning strategy on the population size requirements of a hybrid is shown [7].

Third, Guarantee Feasible Solutions, in highly constrained optimization problems, the crossover and mutation operators of genetic algorithm generally produce illegal or infeasible solutions and hence waste most of the search time. This problem can be solved by incorporating problem-specific knowledge (Local Search). Problem-specific knowledge can be used either to prevent the genetic operators from producing infeasible solutions or to repair them.

Lastly, Operation Substitution, Genetic algorithms present a methodological framework that is easy to understand and handle. This framework can easily and openly incorporate other techniques. It is possible to utilize other techniques to perform one or more of the genetic algorithm operations like crossover operator, mutation operator or both.

4. HYBRID DESIGN ISSUES

A local search can be combined with genetic algorithm to propose a Hybrid Algorithm, which has the capability of extracting the best of both, with the condition of cooperating role of reaching the optimum goal. There are several issues that need to be considered when designing a hybrid genetic algorithm. Some of the issues faced by practitioners while solving real-world problems are as follows:-

A. Balance between Global and Local Search
B. Local Search and Learning

Balance between Global and Local Search

Hybrid Algorithm should strike a balance between exploration and exploitation, in order to be able to solve global optimization problems. According to Hybrid Theory, An optimization problem can be solved with a desired quality of solutions in two ways. The global search method can reach the goal alone or it can guide the search to the basin of attraction from where the local search method will lead to the desired solution. In genetic-local hybrid algorithm the main role of GA is to explore the whole search space to find the regions where optima lies or to the global optima. However the role of local search is to exploit the information gathered by the genetic algorithm. So combine the genetic algorithm with the local search to get the best of the exploring capability of GA, and efficiency of the local search to exploit and find the local optima.

Although the main aim of combining genetic algorithm and local search method is to get the best of exploring capability from the former, and the efficiency of reaching at local optima of the latter, these two can be incorporated in more complicated way.

Mutation operator plays different role in hybrid algorithm than it plays in simple genetic algorithm [8]. The local refinement requirement of mutation operator is carried over by the local search, so it can be optimized in global exploration also. The exploring ability of the genetic algorithm can be further improved by utilizing local search to ensure fair representation of different regions of a search. This can improve the ability of the genetic algorithm to direct the search to the most promising regions of the search space. Once the algorithm has guided the search to the basin of attraction of the global optimum, utilizing local search can further improve the search to produce an effective optimization algorithm. The first goal of the hybridization, which is the effectiveness of search, can be satisfied if a genetic algorithm and a local search method cooperate in the manner mentioned above. However, there are other more destructive forms of interaction. For example, the mutation and crossover operators can disrupt good and

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

complete local solutions which may waste algorithm resources and produce an inefficient search.
In addition the genetic operators can play role in systemically exploring the search space, they also perform some form of local search with relative low cost compared to the more accurate local search methods. The improper use of an expensive local search in a hybrid algorithm can waste algorithm resources. The algorithm should be able to decide wisely on both methods, especially when both can achieve the desired task, taking into account the benefits and costs of their utilization. The condition of an appropriate use of both methods in addition to the condition of interacting in a cooperative way should be satisfied in order to produce an effective and efficient search algorithm.

Local Search and Learning

Local search methods are used their local knowledge to improve the chances of an individual to be propagated to the next generation. This role of local search is same as the role of learning in evolution process, so local search is also viewed as learning process. The knowledge gained by local searching and its use in hybrid genetic algorithm has a great impact on the performance of algorithm. There are two basic models based on biological learning models to use the local knowledge in hybrid algorithm: Lamarckian approach and the Baldwinian approach. A third approach also used which is hybrid of these two approaches.

A. **Lamarckian Learning:** It is based on the inheritance mechanism, it inherit the acquired characteristics obtained through learning. The genetic structure of an individual and its fitness are changed to match the solution found by a local search method. This approach forces the genetic structure to reflect the result of the local search. In the Lamarckian approach, the local search method is used as a genetic operator called refinement genetic operator which modifies the genetic structure of an individual to enhance its fitness and places it back in the genetic population.
It is recognized as it never occurs in biological system due to non availability of mechanisms to accomplish it. But it can be simulated in computers to gain knowledge about the issues of general evolvability.
Lamarckian learning accelerates the search process, but it changes the genetic structure of individuals, which in turn disrupt the schema, so it can also result in premature convergence. A reverse mapping (phenotype-to-genotype) is required in this, which is computable in some problems, but in real-world problems it may be intractable. However, it is used in many hybrid algorithms to repair chromosomes to enhance the fitness and it is quite useful in TSP like problems.

B. **Baldwinian Learning**: It allows an individuals fitness to be increased to propagate it to next generation, without disrupting the genotype. Like, in natural evolution, learning does not change the schema, but increase the fitness for survival. The local search method uses the local knowledge to evaluate the fitness of individuals locally; one can use this local fitness to be used by global search to decide upon the fitness of individual. The Baldwin effect is same as Lamarckian effect in results, although it uses different mechanism.
How the Baldwin effect can transform the fitness landscape of a difficult optimization problem into a less difficult one is shown, and how the genetic search is attracted toward the solution found by learning [9]. They showed a good fact in favor of learning, that some aspect of environment are unpredicted, so it is better to leave them for learning process rather defining them in genetically. Their simulation supports the arguments of Baldwin and demonstrates that adaptive processes within the organism can be very effective in guiding evolution. The main limitation of the Baldwin effect is that it is only effective in spaces that would be hard to search without an adaptive process to restructure the space.

C. **Hybrid Lamarckian-Baldwinian Models**: These models are designed with view of combining the best of both the above learning models. The combination can happen in two ways, one can combine the two learning's at individual-level, where some individuals evolve by Lamarckian model while some evolve by Baldwinian approach. The other level is gene-level, where a number of genes evolve by Lamarckian approach and some genes evolve by Baldwinian approach. It is showed by researchers that these hybrid schemes outperformed the individual models in many real-life problems. The adoption of any learning model has a high impact on the searching process. Many researchers used these models to investigate how these models affect the performance of hybrid algorithms by comparing results with pure genetic algorithms. 20% of the repaired solutions were replaced in hybrid algorithm to solve numerical optimization problems with nonlinear constraints [10].They conclude that neither of the pure strategy was found to be consistently effective. It was discovered that the 20% and 40% partial Lamarckian search strategies yielded the best mixture of solution quality and computational efficiency based on a minimax criterion (i.e. minimizing the worst case performance across all test problems instance). It was found that adaptation by Lamarckian evolution was much faster for neural networks than Darwinian evolution in a static environment [11]. However, when the environment changed from generation to generation, the Darwinian evolution was superior.
In conclusion, the use of pure Lamarckian, pure Baldwinian or any mixture of these two will affected by the representation, percentage of population performing in local search, local search method used and many other parameters.

5. REPLACEMENT OPERATOR OF GENETIC ALGORITHMS

Replacement is the last stage of any breeding cycle. Two parents are drawn from a fixed size population, they breed two children, but not all four can return to the population,

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS…..

so two must be replaced i.e., once off springs are produced, a method must determine which of the current members of the population, if any, should be replaced by the new solutions. Basically, there are two kinds of methods for maintaining the population; generational updates and steady state updates.

1. Generational replacement or Full replacement
   - Basic update
   - Derived (λ,μ) and (λ+ μ,μ) update
2. Steady state or online GAs use different replacement schemes
   - Replace Worst.
   - Replace Random.
   - Replace Parent.
   - Replace most similar (crowding).
   - Replace Best.

In generational genetic algorithms, two parents are selected from the current generation to create a child. This process is repeated n times (where n is the population size) until the children form a whole new generation. Only after this new generation is completely created are new parents selected from it to continue the process. Notice that the newly created child is placed in the next generation, rather than in the current one. If wishing to keep the population size constant, this means must choose a population member to replace with the new child. This, for obvious reasons, is called replacement. Since the entire population is replaced each generation, the only "memory" the algorithm has is from the performance of the crossover operator. If the crossover accurately conveys good genetic material from parents to offspring, the population will improve. If the crossover operator does not maintain genetic material, the population will not improve and the genetic algorithm will perform no better than a random search. A crossover operator that generates children that are more often unlike their parents than like them leads the algorithm to do more exploration than exploitation of the search space. In search spaces with many infeasible solutions, such scattering will more often generate infeasible rather than feasible solutions.

In a steady state GA (Overlapping Populations), a newly created child is inserted immediately into the current population. The steady-state genetic algorithm uses overlapping populations. In each generation, a portion of the population is replaced by the newly generated individuals. At one extreme, only one or two individuals may be replaced each generation (close to 100% overlap). At the other extreme, the steady-state algorithm becomes a simple genetic algorithm when the entire population is replaced (0% overlap). Since the algorithm only replaces a portion of the population of each generation, the best individuals are more likely to be selected and the population quickly converges to a single individual. As a result, the steady-state algorithm often converges prematurely to a suboptimal solution. Once again, the crossover and mutation operators are key to the algorithm performance; a crossover operator that generates children unlike their parents and/or a high mutation rate can delay the convergence. There are many methods of replacement ranging from replacing the worst population member, to replacing a member randomly chosen from the worst n% of population members, to replacing the worst population member that is the most similar to the child to insert. The last method mentioned is intended to maintain diversity by preventing the population from containing too many similar members. This GA uses the second method, randomly choosing a current member to replace from the worst 10%.

6. CONCLUSION

In this paper, some of the genetic-local hybrid techniques are reviewed, which are quite useful in particular scenarios. The ability quickly depends on the way of utilizing the information from both the searching mechanisms in both of them. The review shows that it is good enough to control some parameters when mixing both strategies like duration of local search, frequency of local search etc. Uncontrollable parameters will lead to resource wastage. Genetic algorithms and neighborhood search techniques will result in early findings of the optima. They both are good alone, but if one will combine them somehow in controllable environment, things can be done quite easily.

Although GA's are effective complete search algorithms with crossover and mutation operators, genetic algorithms can be improved using local search methods and they can be made competitive with others when the search space is too large to explore. Also, it is experienced that evolutionary local search algorithms can be improved when problem specific knowledge is incorporated and goal-oriented operators are used instead of blind operators in simple genetic algorithm part. These topics are decided as future work on finding an efficient evolutionary local search algorithm for the NP-Hard problems.

REFERENCES

[1] Martinez-Estudillo A., Hervas-Martnez C., Martnez-Estudillo F., and Garca-Pedrajas N., 2004, "Hybrid method based on clustering for evolutionary algorithms with local search," IEEE Transactions on Systems, Man and Cybernetics.

[2] Freisleben B. and Merz P., 1996, "New genetic local search operators for the traveling salesman problem," in the Fourth Conference on Parallel Problem Solving from Nature vol. 1141, Lectures Notes in Computer Science, H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, Eds. Berlin, Germany: Springer-Verlag, pp. 890–899.

[3] Konak A. & Smith A. E., 1999, "A hybrid genetic algorithm approach for backbone design of communication networks," in the 1999 Congress on Evolutionary Computation. Washington D.C, USA: IEEE, pp. 1817-1823.

[4] Bilchev G. and Parmee I. C., 1995, "The ant colony metaphor for searching continuous design spaces," in AISB Workshop on Evolutionary Computing, vol. 993, Lecture Notes In Computer

Science, T. C. Fogarty, Ed. Sheffield, UK: Springer Verlag, pp. 25-39.

[5] Levent Aksoy, Özgür Aydın Tekin, "Hybridization of Local Search Methods with a Simple Genetic Algorithm for the Satisfiability Problem"

[6] Harik G., Cantu-Paz E., Goldberg D. E. , and Miller B. l., 1999, "The gambler's ruin problem, genetic algorithms, and the sizing of populations," Evolutionary Computation, vol. 7, pp. 231 – 253.

[7] El-Mihoub T. , Hopgood A., Nolle L., and Battersby A., 2004, "Performance of hybrid genetic algorithms incorporating local search," in 18th European Simulation Multiconference (ESM2004), G. Horton, Ed. Magdeburg, Germany, pp. 154-160.

[8] Rosin C. D., Halliday R. S., Hart W. E., and Belew R. K., 1997, "A comparison of global and local search methods in drug docking," in the Seventh International Conference on Genetic Algorithms, T. Bäck, Ed. Michigan, USA: Morgan Kaufmann, pp. 221-228.

[9] Hinton G. and Nowlan S. J., 1987, "How learning can guide evolution," Complex Systems, vol. 1, pp. 495-502.

[10] Michalewicz Z. and Nazhiyath G., 1995, "Genocop III: a co-evolutionary algorithm for numerical optimization problems with nonlinear constraints," in 2nd IEEE International Conference on Evolutionary Computation, vol. 2. Perth, Australia IEEE, pp. 647-651.

[11] Sasaki T. and Tokoro M., 1997, "Adaptation toward changing environments: why Darwinian in nature?," in Fourth European Conference on Artificial Life, , Complex Adaptive Systems Series P. Husbands and I. Harvey, Eds. Brighton, UK: MIT press, pp. 145-153.