Volume 3, Issue 4, April 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Hybridization in Genetic Algorithms

Rakesh Kumar
DCSA, KUK, Haryana
India.

Girdhar Gopal
DCSA, KUK, Haryana
India.

Rajesh Kumar
DCSA, KUK, Haryana
India.

Abstract— Genetic algorithms facilitate the hybridization of other local search techniques to get the optimal solution. 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 are capable to find the local optima with high accuracy and fast convergence but suffers from the problem of foot hills. A perfect blending of genetic algorithm with local search techniques helps in exploitation as well as exploration. In this paper, an exhaustive survey of different hybridization techniques was carried out.

Keywords— Evolutionary computation, Genetic Algorithm, Genetic Local search algorithms, Hybridization, Hybrid Genetic Algorithms

I. Introduction

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) and his students and colleagues at the University of Michigan in the 1960s and the 1970s. 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. A chromosome contains a group of numbers that completely specifies a candidate solution during the optimization process [1]. The selection operator chooses the better chromosomes to create a mating pool that will participate in reproduction process. Crossover exchanges subparts of two chromosomes, roughly mimicking biological recombination between two chromosomes; mutation randomly changes the allele values of some locations in the chromosome, 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. Fig. 1 shows the basic terminology of Genetic Algorithms In this paper, a detailed study of hybridization of genetic algorithm with local search was carried out. The review of hybridization of genetic-local hybrid algorithm provides a view to the factors affecting the performance of such type of algorithms. Those factors are discussed and a framework is suggested with some parameters that need to be taken into consideration when one designs a hybrid algorithm with local and global search methods.

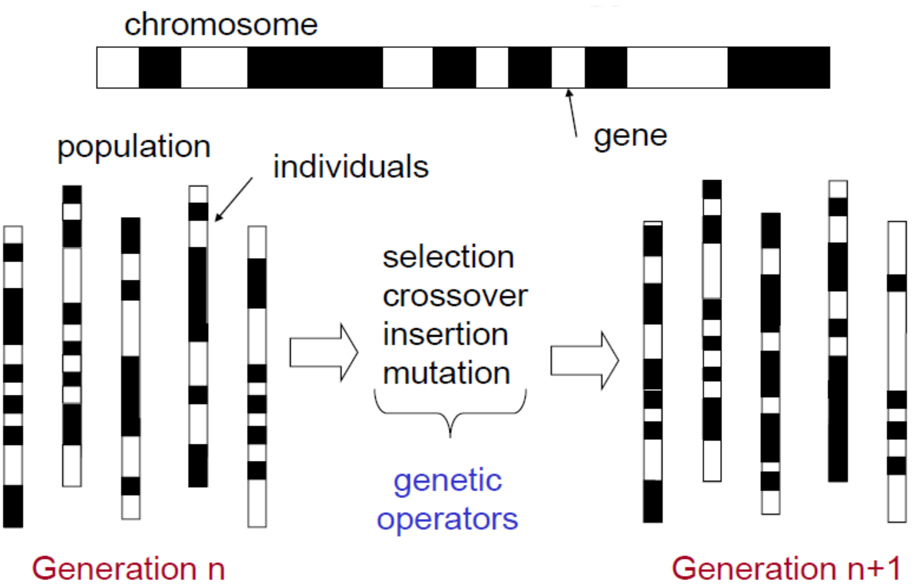

Fig.1: Basic GA Terminology

II. 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. Performing local search on the GA's population can maintain diversity and reduce the problems like genetic drift, [1]. 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, [2]. This opportunity depends on the design details of the hybrid genetic algorithm. They described several issues that need to be taken into consideration when designing a hybrid genetic algorithm.

The distance preserving crossover (DPX) to produce feasible solutions to solve TSP without losing diversity is proposed, [3]. 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 other 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, [4]. They use a uniform crossover operator with K-node-connectivity repair algorithm to repair infeasible offspring's. 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, [5]. Before the algorithm begins the location of the nest is to be determined. 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. Success of Genetic algorithm primarily depends on Initial population, [2]. Beam Search (BS) was integrated by them 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 with genetic algorithm untill termination criteria met. They proposed eight different types of combination of Regular GA, mutation, seeding with BS. 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, [6]. 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. Tournament selection scheme has been used 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.

III. GENTIC AND LOCAL SEARCH

Hybrid genetic algorithms are based on the complementary view of genetic algorithms and local search methods. So there are several ways in which a local search can be incorporated in genetic algorithms. The main purposes of incorporation are as follows:

a. Capability Enhancement
b. Optimizing the Control Parameters

A. 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 will 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. In Genetic Algorithm design efficiency is a major concern in terms of the time needed to reach a solution of desired quality. 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, which was used to estimate the population size of genetic algorithms was shown, [7]. 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, [8].

Guarantee of Feasible Solutions, in highly constrained optimization problems is another factor. 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.
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.

B. Optimizing the Control Parameters

The setting of genetic algorithm control parameters is a key factor in the determination of the exploitation versus exploration trade-off. Other techniques can be used to monitor the behavior of a genetic algorithm in order to adapt its control parameters to improve the search performance. The ability of fuzzy logic to represent knowledge in imprecise and non- specific ways enables it to be used to reason on knowledge that is not clearly defined or completely understood. This ability makes fuzzy logic a suitable choice for adapting the control parameters of a genetic algorithm.

IV. 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 researchers while solving real-world problems are as follows:-

a. Balance between Global and Local Search
b. Local Search and Learning

A. Balance between Global and Local Search

According to Hybrid Theory, An optimization problem can be solved with a desired quality of solutions in two ways, [9]. 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 these two can be incorporate in many more complicated ways.

Mutation operator plays different role in hybrid algorithm than it plays in simple genetic algorithm, [10]. 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 genetic algorithm and 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 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.

B. Local Search and Learning

Local search methods use 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 search 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.

1) 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, [11]. 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.

2) 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, [12]. 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.

3) 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 shown 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, [13]. Many problems were used to compare the performance of different learning approaches, [14]. 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 yields 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, [15]. 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.

V. PARAMETERS OF HYBRIDIZATION

One goal of hybridization, which is effectiveness of search, can be achieved only when both algorithms combined in cooperative manner, otherwise this can be destructive too. The improper use of local search can waste the algorithm resources. The condition of appropriate use of both methods in addition to the condition of interacting in a cooperative way should be satisfied to perform the effective search. Researchers have proposed different methods to enable the mixing of both wisely or at least reduce the consequences of the improper use of the local search in the hybrid scheme. There are following parameters of local search which are useful in mixing with GA :-

a. Duration of Local Search
b. Probability and selection of Local Search
c. Frequency of local search

A. Duration of Local Search

On combinatorial domains, local search is performed till a solution converges to local optimum, while in continuous domains the local search is performed until its step length becomes too small, that may be before reaching the local optimum. Local search duration influences the balance between the global exploration and local refinement, [16]. A hybrid with long local search duration will execute fewer generations of the genetic algorithm than a hybrid with shorter local duration, if both terminate after the same number of function evaluations. Performing local search until a solution converges to a local optimum, which is referred to as complete local search, may lead to the loss of population diversity depending on the learning strategy used, [17]. Hybrid genetic algorithms that adopt the pure Lamarckian approach are more prone to loss of diversity than others which utilize other learning techniques. Applying a complete local search on costly function evaluations can also be expensive. However, there is a certain class of problems, decomposable fitness problems, where calculating the fitness of a solution given the fitness of its neighbor, is significantly less computationally expensive than computing its fitness from scratch, [18]. TSP is an example of this group of problems where computing the length of a tour that shares most of its edges with another tour, whose length is already known, is much cheaper than computing the length of a complete tour. They argued that hybrids are more suitable for problems exhibiting this property. A few studies have been conducted which investigate the optimal duration of local search. Hart found that using a short duration of local search produced the best results for the Griewangk functions, whereas a long duration produced better results for the Rastrigin functions. Very short and very long local search durations were experimented in a hybrid to optimize the drug-docking configuration. Both durations were found to yield similar performance, [10]. It was concluded that duration of local search is an important factor and hybrid genetic algorithms with long local searches will be most effective for nontrivial problems. The high cost of a complete local search on expensive function evaluations makes any improper use of the local search difficult to recover from. However, the recovering from any misuse of partial local search is still possible. Partial local search is more suitable for hybrids that decide on a global or a local approach depending on the current state of the search and the previous performance of both methods. In this case, when there is a possibility of misjudgment in some circumstances, the use of partial local search gives the hybrid a higher chance to recover from such errors than using a complete local search. However the partial local search expenses are possible to recover. It is better to decide about the local or global based on the current state of search and previous search results. Where there is a chance of misjudgment, the hybrid can give a chance to recover from those errors.

B. Probability and selection of Local Search

In hybrid, a local search can be applied to either every individual in the population or a small fraction of the population. Because, applying the local search to every individual with heavy cost functions will waste the resources, compared to when it is only applied to those individuals which lies in some basin of attraction. Deciding upon the optimal fraction of the population which should perform local search, and the basis on which these individuals are chosen, has a great impact on the performance of a hybrid. The local search can be applied to individuals that fall in the same basin of attraction of the search space, whereby producing the same local optimum. Applying a local search to a large fraction of the population can limit exploration of the search space by allowing the genetic algorithm to evolve for a small number of generations. The possibility of applying local search on more than one individual from the same basin can be reduced by performing local search on only a small fraction of the population. This also lowers the chances of applying an unnecessary local search on individuals that fall in non- promising regions of the search space. Deciding upon the optimal fraction of the population which should perform local search, and the basis on which these individuals are chosen, has a great impact on the performance of a hybrid algorithm.

Impact of the fraction of the population that undergo local search on the performance of real-coded genetic algorithms was investigated, [19]. He found that a relation exists between this fraction, the population size and the performance of the hybrid. He also found that performing local search on small fractions could be more efficient when using larger populations and those large fractions can help to reflect the search space characteristics when using small populations. He concluded that a more selective use of local search could improve the efficiency of hybrids. Different techniques were proposed, such as tuning, distribution-based, fitness-based techniques, and local search potential, to decide on the optimal fraction of the population that should perform a local search, [19]. These techniques aim to reduce unnecessary local searches. However, they differ in the way they select individuals that perform the local search. Some of these techniques are Tuning Technique, Distribution-based Technique, Fitness-based Technique and Local Search Potential Technique.

C. Frequency of local search

This is the number of continuous generations performed by the genetic algorithm before applying one generation of local search. In traditional hybrid algorithm the frequency of local search is 1. A frequency of 2 to solve TSP was used, [11]. However, in a hybrid algorithm to solve the static correction problem, the genetic search algorithm was allowed to continue uninterrupted for 10 generations before applying a single iteration of waveform steepest ascent iteration to each individual in the population. This hybrid algorithm produced solutions with improved quality of 5% and additional savings in time compared with the traditional hybrid genetic algorithm.

The optimal frequency of local search is dependent on function and also varies with time, because it depends on the distribution of individuals in the population. Deciding when to use the local search and when the global genetic algorithm with probability matching is a big problem, [20]. They take it as a two-armed bandit problem, and make a model for this. This model is an adaptive model, which decides between the two searches on the efficiency of the both as the search progresses. They tried to answer the question, "when should the local search be used and when the global genetic algorithm should be used to achieve the maximum possible efficiency?"

A new heuristic algorithm for classical symmetric TSP was proposed and tested the performance against benchmark TSP problems, [21]. They presented overlapped neighborhood based local search algorithm to solve TSP and concluded that the proposed algorithm is superior in terms of average deviation and smallest deviation from optimal solutions.

A PBS blended selection operator was proposed, which has balanced trade-off between exploration and exploitation, [22]. They compare the performance on standard TSP problem with roulette-wheel and Rank selection techniques. The performance of PBS depends on the number of generations. In start of GA selection operator had explorative nature, as the search progress, selection pressure also increased and the nature of selection also changed to exploitative. The performance of PBS over other selection operators is superior.

New innovation in crossover operator was proposed in genetic algorithm with Tabu search, [23]. They implemented it in TSP problem to find out two separate lists of city pairs, as best pair and worst pair. Then they used this knowledge in crossover operator such that the best pair is always maintained in new generation and the worst pair never retain to new generation. The result of this proposed crossover shows better results as compared to simple genetic algorithms.

A novel encoding scheme was proposed, [24]. There are lots of encoding schemes available with their own pros and cons. encoding schemes are either function of value or order of value or both. Further operators depend on the encoding scheme used like crossover and mutation. So they proposed a naive encoding scheme which is independent of function or value. It represents the genes in chromosome as its fitness contribution value. This encoding supports one-point crossover and PMX crossovers as they supported by binary and permutation encodings respectively. This encoding scheme can be applied to all the applications where initially binary encoding is implemented and would prove as efficient method of encoding

VI. CONCLUSIONS

In this paper, some of the hybrid algorithms are surveyed, which are quite useful in particular scenarios. Their ability depends on the way of utilizing the information from both the searching mechanisms i.e. GA and local search. It was observed 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 both were combined in controllable environment, things can be done quite easily and effectively. And as one will apply genetic algorithms to NP-Hard problems any slight improvement lead us to a better vision.

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 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 Hybrid genetic algorithm for the NP-Hard problems.

REFERENCES

[1] Holland J., 1975, "*Adaptation in Natural and Artificial Systems",* the University of Michigan.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] Goldberg D. E. and Voessner S., 1999, "*Optimizing global-local search hybrids*," in the Genetic and Evolutionary Computation Conference (GECCO 1999). Orlando, USA: Morgan Kaufmann, pp. 222-228.

[10] 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.

[11] Whitley D., Gordon S., and Mathias K., 1994, "*Lamarckian Evolution, the Baldwin effect and function optimization*," in Parallel Problem Solving from Nature - PPSN III vol. 866, Lecture Notes in Computer Science, Y. Davidor, H.-P. Schwefel, and R. Manner, Eds. Jerusalem: Springer- Verlag, pp. 6-15.

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

[13] 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.

[14] Houck C., Joines J., Kay M., and Wilson ., 1997, "*Empirical investigation of the benefits of partial Lamarckianism*," Evolutionary Computation, vol. 5, pp. 31- 60.

[15] 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.

[16] Ishibuchi H., Yoshida T. , and Murata T., 2003, "*Balance between genetic search and local search in memetic algorithms for multi-objective permutation flow-shop scheduling*," IEEE Transactions on Evolutionary Computation, , vol. 7, pp. 204- 223.

[17] Whitley D., Starkweather T., and Fuquay D. A., 1989, "*Scheduling problems and traveling salesman: the genetic edge recombination operator*," in the Third International Conference on Genetic Algorithms. Fairfax, USA, pp. 133 – 140.

[18] Radcliffe N. J. and Surry P. D., 1994, "*Formal memetic algorithms*," in Evolutionary Computing: AISB Workshop. Brighton, UK: Springer- Verlag, pp. 1-16.

[19] Hart W. E., 1994, "*Adaptive global optimization with local search*," Doctoral Dissertation. San Diego: University of California.

[20] Goldberg, D. E., 1989, "*Genetic Algorithms in Search, Optimization, and Machine Learning*." Addison-Wesley Publishing Company, Inc.,

[21] Lin S. and Kernighan B., 1973, "*An effective heuristic algorithm for the traveling salesman problem*," Operations Research, vol. 21, pp. 498-516.

[22] Kumar R., Jyotishree 2011. "*Blending roulette wheel selection & rank selection in genetic algorithms*." In Proceedings of 3rd International conference on machine learning and computing, V4, IEEE catalog number CFP1127J-PRT, ISBN 978-1-4244-9252-7, 197-202.

[23] Kumar R., Jyotishree, 2012, "*Novel Knowledge Based Tabu Crossover In Genetic Algorithms*", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 8, August 2012 ISSN: 2277 128X.

[24] Kumar R., Jyotishree, 2012, "*Novel Encoding Scheme in Genetic Algorithms for Better Fitness*", International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-1, Issue-6, August 2012.

