Volume 5, Issue 6, June 2015 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Knowledge based Memetic Replacement Operator in Genetic Algorithm

Jaideep Atri
Assistant Professor, S. A. Jain College,
Ambala City, Haryana, India

***Abstract**— Travelling salesman problem, known as NP-hard problem, is among the most classical and challenging problem for researchers. It is difficult to compute this type of problems as it results in exponential complexity. Many heuristic computation methods have been proposed for solving this type of problem. In this paper, the focus of the researcher is on a hybrid method that combine two heuristic optimization approaches, Genetic Algorithms (GA) and Hill Climbing (HC). In the proposed GA-HCR algorithm, genetic algorithm is conducted to provide the diversity in routes. Thereafter, Hill Climbing based replacement operation is performed instead of generational replacement to go out of local optima. The GA-HCR is simulated on two different instances from TSPLIB provided by Heidelberg University and the results are compared with GA based generational replacement algorithm using two different selection operators namely, Roulette wheel and Rank Selection. The implementation has been carried out using MATLAB and result shows that GA-HCR algorithm outperforms the existing algorithm in terms of producing more quality solution and better convergence rate.*

***Keywords**—Genetic Algorithm, Hill Climbing, Hybrid genetic algorithm, Memetic Algorithm, Replacement.*

I. INTRODUCTION

Genetic Algorithms, Genetic Programming, Evolutionary Strategies, Evolutionary Programming are the heuristic; population based evolutionary optimization algorithms [1] that improve a set of solution iteratively by using some biologically inspired mechanisms like crossover, natural selection, mutation and survival of fitness. Among all existing evolutionary algorithms, Genetic Algorithm (GA) is the best known branch that is easy to use and implement as it uses only a simple scalar performance measure and does not require or use any derived information. Genetic algorithms are defined as adaptive optimization algorithms that mimic the process of evolution, genetics and Darwin theory of natural selection [2]. In order to find the global solution of a problem, genetic algorithm can simultaneously evaluates a population points in the search space. For more than four decades Genetic algorithms have been applied to variety of Optimization problem like Travelling salesman, benchmark dejong’s function, Protein Synthesis etc. A generic genetic algorithm starts with a population of fixed length strings that are generated randomly. The string is the collection of genes called as chromosome that is generated according to the type of problem using specific encoding scheme. Selection operator is used to form the best mating pool that improves the quality of population. Two operators’ crossover and mutation are used to bring diversity in the population as well as to generate the better offspring’s. Chromosomes having higher fitness value has more probability to pass on from generation to generations. Algorithm converges to a good solution, after several iterations.

The generic code of Genetic Algorithm is given below:

Step 1: Initialization: Decide the appropriate problem specific encoding scheme and generate initial population P randomly
Step 2: Evaluation of Fitness: Calculate the fitness value of each individual in P.
Step 3: Selection: Select the individuals (individuals having better fitness values) using a suitable selection technique to generate a mating pool M.
Step4: Crossover: Perform recombination operation on the selected individuals with probability Pc to create the offspring’s population O.
Step 5: Mutation: Perform mutation on the new individuals (chromosomes) Population O with probability Pm to generate a new mutated set O’
Step 6: Replacement: Replace the Population P with the good offspring O’
Step 7: Termination: If termination criterion is satisfied, then halt the algorithm
Else
Iterate through steps 2 to 7

The performances of genetic algorithms depend on the balancing between the exploration and exploitation techniques. Exploitation means to find out the better solution using already available knowledge and Exploration is to investigate unknown and new area in search space. In actual practice, the population size is finite that influences the performance of genetic algorithm and in case of multimodal search space it mainly leads to the problem of genetic drift. Incorporating a local search method within the genetic operators can introduce new genes than can overcome the problem of genetic drift and accelerate the search towards global optima [3]. A combination of genetic algorithm and a local search method is called as hybrid genetic algorithm or memetic algorithm. In memetic algorithms, local search and knowledge can be incorporated at any stage like initialization, selection, crossover and replacement. The finite population can cause genetic algorithm to produce solutions of inferior quality. Genetic algorithm is unable to make small moves in the neighborhood of current solution that makes it unable to locate the best solution in the best region [4]. Using a knowledge based local search method within the framework of genetic algorithm can improve the exploiting ability of search without limiting its exploring ability [3]. Genetic Algorithm mainly, lead to premature convergence if the diversity is very low, which is not desirable.

In this paper, the focus of researcher is on a hybrid method that combines two heuristic optimization approaches, Genetic Algorithms (GA) and Hill Climbing (HC). In the proposed GA-HCR algorithm, genetic algorithm is conducted to provide the diversity in routes. Thereafter, Hill Climbing based replacement operation is performed instead of generational replacement to move out of local optima. The GA-HCR is tested on two different TSP instances from TSPLIB provided by Heidelberg University and the results are compared with GA based generational replacement algorithm using two different selection operators namely, Roulette wheel and Rank Selection. The paper is distributed in following sections. Section 2 contains Literature review of researches performed by different researchers related to hybrid genetic algorithms. In section 3, hybrid algorithm is discussed. Hill Climbing search is described in section 4. Different operators algorithm used for comparison is discussed in section 5. Proposed GA-HCR along with its algorithm is provided in section 6. Implementation as well as results is specified in section 7. Conclusion & future work is discussed in last section.

II. LITERATURE REVIEW

Sivaraj et al [5] discussed about a novel approach that used tournament selection for initialization to improve the performance of genetic algorithm that aims at supplying better individuals in the beginning phase. The result shows that the selective initialization enhances the convergence velocity and produces more optimal solution than existing schemes used in generic genetic algorithm. Sharyar et al proposed a novel initialization approach which employs opposition based learning in order to generate initial population. The performed experiments over an inclusive set of benchmark functions demonstrated that replacing the random initialization with the opposition based population initialization can accelerate the convergence speed [6]. For multiple sequence alignment, Lee et al [7] developed a new algorithm by incorporating genetic algorithm with ant colony optimization. The performance of genetic algorithm has been enhanced by incorporating ant colony optimization (ACO) as local search. Genetic algorithm is able to provide the diversity of alignments in the proposed algorithm. After that, ant colony optimization is performed to remove the problem of local optima. Simulation result shows that the proposed algorithm has superior performance when it is compared to other existing algorithms.

A novel crossover operator that uses the principle of Tabu search has been proposed by Rakesh Kumar et al. The proposed crossover is compared with PMX and found that the proposed crossover results in better solution than PMX [8]. A hybrid genetic algorithm based on GA and Artificial Immune network Algorithm (GAIN) has been proposed by Antariksha for finding optimal collision free path in case of mobile robot moving in static environment filled with obstacles [9]. She concluded that GAIN is better for solving such kind of problems. To solve the maintenance scheduling problem, E. Burke et al. proposed a memetic algorithm based on Tabu search technique. The novel MA performs better and can be effectively applied to real problems [10]. For feature selection in volumetric data containing spatially distributed clusters of informative features in neuroscience application, Malin et al [11] proposed a memetic algorithm in which they concluded that the novel Memetic Algorithm recognized a majority of relevant features as compared to genetic algorithm. Manju Sharma and Sanjay Tyagi [12] proposed a new initialization based hybrid algorithm that supplies more fit individuals in the beginning phase using tabu search. The experiment has been conducted using TSP problem The Implementation result shows that the proposed Tabu initialization algorithm performs better than the existing initialization scheme.

III. MEMETIC ALGORITHMS

Although genetic algorithms can speedily find the region of global optima, however it takes a comparatively long time to locate the precise local optimum within the favorable region [13], [14]. So, incorporating problem specific learning in a genetic algorithm at different levels of genetic operation form a hybrid genetic algorithm [13].The technique of hybridization of problem specific knowledge and global genetic algorithm is memetic algorithm. Memetic Algorithm is driven by Dawkins notation of a meme. A meme is a block of information that reproduces itself as people exchange ideas [14]. A hybrid Algorithm combines the features of GA with many heuristic's search techniques like Ant colony optimization, simulated annealing, Hill Climbing etc. There are numerous ways to use the local search and genetic algorithm as complementary tools. Two renowned ways of hybridization are depends on the concepts of "Baldwin effect" [15] and "Lamarckism" [16]. Based on the Baldwinian search strategy, genotype itself remains unchanged but the local optimization interacts and allows the local search to change the fitness of individual. The disadvantage of Baldwin's is that it results in hindering effect [17]. According to Lamarckism, the characteristics acquired by individual during its lifetime may become heritable traits. According to this approach, during local optimization phase, fitness as well as the genotype of individuals is modified.

IV. HILL CLIMBING

Hill climbing [18] is an optimization technique, belongs to the family of local search. It starts with a random solution to a problem, then attempts to locate a better solution by incrementally changing a single element of the solution. The algorithm keeps on repeating incrementally, till the modification done in the existing solution results in the favorable outcome.

Algorithm Hill Climbing (initial solution)
begin *p*:=initial solution
repeat generate an *b* ϵ *Neighbour(p);*
if fit(*b*) > fit(*p*)
then *p*:=*b*;
while fit(*b*) ≤ fit(*p*)
for all b ϵ *Neighbour(p);*
end

V. ALGORITHMS

This section discussed the different operators algorithms used for implementation.

Roulette wheel Selection

Procedure RWS (P, num)
// num = Total number of individual in population // mt_pool = mating pool size
// Cm_t = cumulative fitness
p=1, q=1, t=1
While (p <= num)
{
total = total + Fp //fitness of pth individual in population P
}
While (q<= mt_pool)
{
r=rand(0, total);
t=1;
While (t <= num)
{
Cumt = Cumt-1 + Ft
If (r <= cumt)
{
Select t^{th} individual
}
q=q+1
}
End procedure

Rank Selection(RS):

Procedure RS (P, num)
Set l=1, q=1, n=num, total=0
While (q<=n)
{
total =total+r_q // r_q is rank of qth individual in population P
}
Set q=1
While (q<=n)
{
RANKq=r_q/total // RANKq is selection probability of qth individual
}
While l <= mt_pool
{
r=rand(0,total);
Set q=1
While q<=n
{
Cm_q=Cm_q-1+RANKq //compute cumulative rank
If r<=Cm_q,
Select the individual q


```
}
 l=l+1
}
```

VI. PROPOSED HYBRID REPLACEMENT ALGORITHM

In this paper, the focus of the researcher is on a hybrid method that combines two heuristic optimization approaches, Genetic Algorithms (GA) and Hill Climbing (HC) at the replacement phase of generic cycle. The aim of applying hill climbing in the last phase is to increase the chance of passing better solution in the next generation via hybrid replacement strategy. Although the Genetic algorithm has the capability to locate the region, where the optimal solution exists, but it mainly leads to the problem of premature convergence and genetic drift. So, in order to locate the exact global optima and to make a perfect blending of exploration and exploitation the author combines the knowledge based local search at the replacement step of genetic algorithm.

The outline of proposed algorithm is as follows.

Procedure *GA(fitness, psize, Pc, Pm,tgen)*
// fitfn is fitness function for determining the phenotype.
// psize is the size of population in every generation
// Pc is fraction of population generated by crossover
// Pm is the mutation prbability
//tgen is total number of generations
P = randomly n individuals // initial generation is generated randomly
i=1
//define the next generation Q of psize
while *i <=tgen*
{
//Selection step:
S:= Select(P,Psize) // psize/2 individuals of P will be selected using selection method.
//Crossover step:
Q:= Crossover(S,psize,Pc) // Generates psize chromosomes with crossover probability Pc.
//Mutation step:
Mutation(Q,Pm) //Apply Mutation over chromosomes with mutation rate Pm
//Replacement step:
best_ind(i):=min(fitfn(Q)) // store best individual in population
Replace(Q) //Replaces old population using hybrid replacement strategies excluding best individual
i:=i+1
}
best:=min(best_ind) //finds best individual in all generations
end proc
Module for hybrid Replacement is
Replace(Q)
//Q is set of chromosomes generated as offspring
//P is parental generation of chromosomes
//H Hybrid chromosomal population
H:= Hill Climbing(Q) // Algorithm for hill climbing mentioned in section iv.
P:= H // Replace hybrid individual with parent population by Applying generational replacement
End

VII. IMPLEMENTATION & OBSERVATIONS

MATLAB code has been developed by author to analyze the performance of proposed GA-HCR algorithm using the same crossover and mutation operator. The code considers the benchmark TSP oliver30, Att48 problem [19]. Travelling salesman problem, known as NP-hard problem, is among the most classical and challenging problem for researchers. It is difficult to compute this type of problems as it results in exponential complexity**.** DNA sequencing, planning, logistics and manufacture of microchips are the applications of TSP. TSP problem is to find the Hamiltonian Path or shortest distance through a set of vertices, such that each vertex is visited exactly once [19].

The GA-HCR is tested on two different TSP instances from TSPLIB provided by Heidelberg University and the results are compared with GA based generational replacement algorithm using two different selection operators namely, Roulette wheel and Rank Selection

Parameters used for simulation are-

- Population size: 50
- Number of generations: Varies
- Encoding: Permutation encoding
- Selection: RWS, RS
- Crossover operator: Partially Mapped crossover operator.
- Mutation: Inversion Mutation

- Crossover probability (p_c=0.6)
- Mutation probability (p_m=0.01)

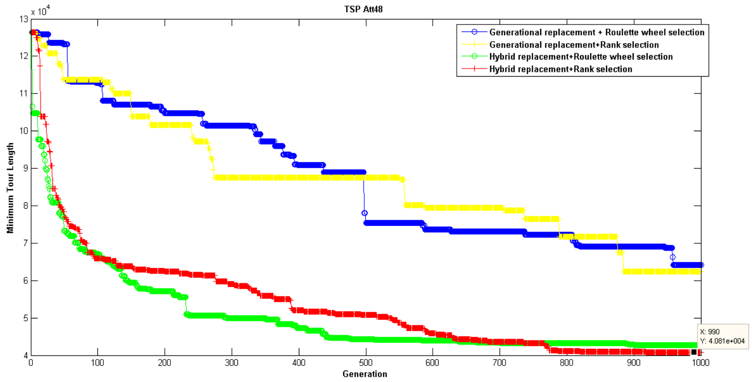

Fig. 1 Comparison of Minimum Tour length between in Att48 for 1000 generation

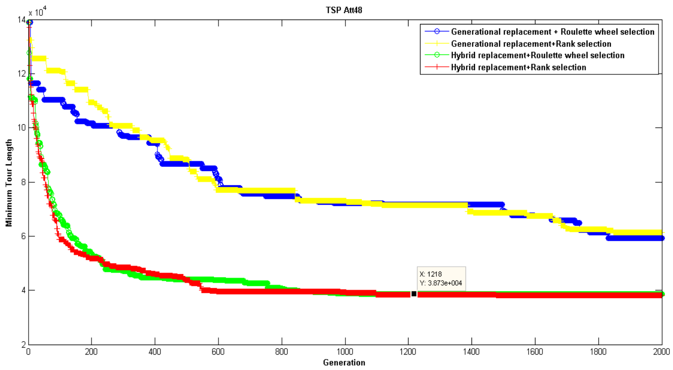

Fig. 2 Comparison of Minimum Tour length between in Att48 for 2000 generation

Figure 1 and Figure 2 depicts the minimum tour length for Att48 for 1000 and 2000 generations, respectively. Figure 3 and Figure 4 depicts the minimum tour length for Oliva30 for 1000 and 2000 generations, respectively. Table 1 depicts the comparison of minimum tour length for different generation (Gen).

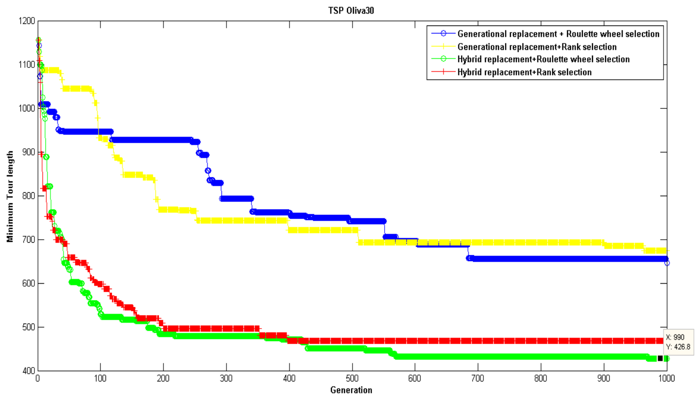

Fig. 3 Comparison of Minimum Tour length between in Oliva30 for 1000 generation

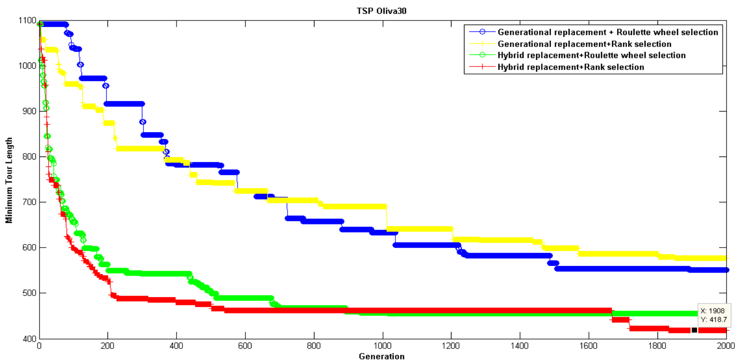

Fig. 4 Comparison of Minimum Tour length between in Oliva30 for 2000 generation

Optimal Tour [19] cities sequence in oliver30 is: 1 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 25 24 26 27 28 29 30 2 (Distance= 418.7406)
Optimal Tour [19] cities sequence in Att48 is: 1 8 38 31 44 18 7 28 6 37 19 27 17 43 30 36 46 33 20 47 21 32 39 48 5 42 24 10 45 35 4 26 2 29 34 41 16 22 3 23 14 25 13 11 12 15 40 9 (Distance= 10628)

Table 1 : Comparison of Minimum tour length for different Generation

Comparison (fmin)		**RWS**	**RS**	**Hybrid Replacement+ RWS**	**Hybrid Replacement+ RS**
Gen=50	**Oliver30**	837.6	893.2	795.6	769.8
	Att48	$1.349e^{+005}$	$1.132\ e^{+005}$	$7.925\ e^{+004}$	$7.529\ e^{+004}$
Gen=100	**Oliver30**	834.9	785.9	745	688.7
	Att48	$1.195\ e^{+005}$	$1.175\ e^{+005}$	$8.006\ e^{+004}$	$6.65\ e^{+004}$
Gen=200	**Oliver30**	793.7	732.8	659.9	580
	Att48	$1.046\ e^{+005}$	$1.041\ e^{+005}$	$6.01\ e^{+004}$	$5.48\ e^{+004}$
Gen=500	**Oliver30**	680	670.5	561.3	541.8
	Att48	$8.753\ e^{+004}$	$8.204\ e^{+004}$	$4.41\ e^{+004}$	$4.185\ e^{+004}$
Gen=1000	**Oliver30**	675	655.9	467.4	426.8
	Att48	$6.409\ e^{+004}$	$6.239\ e^{+004}$	$4.277\ e^{+004}$	$4.081\ e^{+004}$
Gen=2000	**Oliver30**	551.1	577	455.5	418.7
	Att48	$6.14\ e^{+004}$	$5.935\ e^{+004}$	$3.873\ e^{+004}$	$3.826\ e^{+004}$

It has been observed from the Figures and table that the proposed GA-HCR algorithms has outperformed generic genetic algorithm for different selection operators in terms of proper blending of exploration and exploitation. Incorporation of local search at the replacement step results in the exploitation of already explorated search space and prevents the algorithm fom premature convergence and genetic drift. In the proposed GA-HCR algorithm better individuals are pass on to the next generation that accelerates the search towards global optima.

VIII. CONCLUSION

In this paper, GA-HCR has been proposed as a new hybrid approach for optimizing problems like protein synthesis, TSP etc. GA-HCR integrates the concept of knowledge based local search at replacement phase of GA cycle. The proposed GA-HCR algorithm selects individuals throughout the search space that maintain perfect blending of exploration and exploitation in population and prevent the problem of premature convergence. In the proposed GA-HCR algorithm, genetic algorithm is conducted to provide the diversity in routes. Thereafter, Hill Climbing based replacement operation is performed instead of generational replacement to move out of local optima. The GA-HCR is tested on two different instances of TSP and the results are compared with GA based generational replacement algorithm using two different selection operators. The implementation result shows that GA-HCR algorithm outperforms the existing algorithm in terms of producing more quality solution and better convergence rate.
As a future work, author will concentrate on improving the efficiency of this knowledge based Hybrid approaches by exploring advanced concepts of genetic algorithms like parallel GA and by using other optimization technique like Ant Colony Optimization, Particle Swarm Optimization, and Simulated Annealing.

REFERENCES

[1] J. Holland, "*Adaptation in natural and artificial systems*", University of Michigan Press, Ann Arbor, 1975.

[2] D.E. Goldberg, "*Genetic algorithms in search, optimization, and machine learning*", Addison Wesley Longman, Inc., 1989, ISBN 0-201-15767-5.

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

[4] C. Reeves, "*Genetic algorithms and neighborhood search,*" in Evolutionary Computing, AISB Workshop, Lecture Notes in Computer Science, T. C. Fogarty, Ed. Leeds, UK: Springer-Verlag, vol. 865 pp. 115-130, 1994

[5] R. Sivaraj, T. Ravichandran and R. Devipriya " *Boosting performance of genetic algorithm through selective Initialization*.", European Journal Of Scientific Research, ISSN 1450-216X , vol. 68, No 1, pp 93-100, 2012.

[6] S. Rahnamayan, H. R. Tizhoosh and M. A. Salama, "*A novel population initialization method for accelerating evolutionary algorithms*", An International Journal Computers and mathematics with applications, vol. 53, pp 1605-1614, 2007.

[7] Z. J. Lee et al, "*Genetic Algorithm with Ant-Colony Optimization for multiple sequence Alignment*", Applied Soft Computing 8, pp 55–78 , 2008.

[8] Rakesh kumar and Jyotishree, "*Novel knowledge based tabu crossover in genetic algorithms*", International Journal of Advanced research in Computer science and software Engineering, vol. 2, No. 8, pp 78-82, Aug 2012.

[9] Antariksha Bhaduri, "*A mobile robot path planning using genetic artificial immune network algorithm*", Proceedings of World Congress on Nature and biologically Inspired Computing, NaBIC, IEEE, pp 1536-1539 , 2009.

[10] E. K. Burke and A. J. Smith, "*A memetic algorithm for the maintenance scheduling problem*", Proceedings of International Conf on Neural Information Processing and Intelligent Information, Springer, pp 469-473, 2010.

[11] Malin Bjornsdotter and Johan Wessberg, "*A memetic algorithm for selection of 3D clustered featured with applications in neuroscience*", Proceedings of International Conference on Pattern Recognition, IEEE, pp 1076-107, 2010.

[12] Manju Sharma and Sanjay Tyagi, "*Novel Knowledge based Selective tabu initialization in Genetic algorithm*", Vol 3, Issue 5, pp 764-770, May 2013.

[13] R. Dawkins, "*The selfish gene*", Oxford University Press, Oxford, 1976.

[14] K Ku., M. Mak, "*Empherical analysis of the factors that affect the Baldwin effect, Parallel problem solving from nature*", 5th International Conference Proceedings, lecture notes in computer science. Berlin, Heidelberg: Springer, PPSN 5, pp 481-90, 1998.

[15] G. M. Morris, G. M. Goodsell. , R. S. Halleday, W. E. Hartand , R. K. Belew et al., "*Automated docking using a Lamarckian genetic algorithm and an empherical bending free energy function*", Journal of Computational Chemistry, ISSN 19:1639-62, 1998.

[16] G. Mayley, "*Landscapes, learning costs and genetic assimilation*," Evolutionary Computation, vol. 4, pp. 213 - 234, 1996.

[17] Fred Glover, "*Future paths for integer programming and links to artificial intelligence*", Computers & Operations Research, vol. 13, No 5, pp 533–549, 1986. ISSN: 0305-0548.

[18] E G.Talbi, T.Muntean, "Hill-Climbing, Simulated Annealing and Genetic Algorithms: A Comparative Study and Application to the Mapping Problem", System Sciences, 1993, Proceeding of the Twenty-Sixth Hawaii International Conference.

[19] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

