Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 10, October 2015, pg.27 – 34

RESEARCH ARTICLE

Hybrid Genetic Algorithm and Mixed Crossover Operator for Optimizing TSP

Smita Sharma[1], Girdhar Gopal[2]

Department of Computer Science & Applications, Kurukshetra University[1, 2]

Kurukshetra, INDIA

smita148.sharma@gmail.com [1], girdhar.gopal@gmail.com [2]

***Abstract*- Genetic Algorithms (GAs) are the search algorithms and optimization techniques based on the mechanics of natural selection and natural genetics. They sort out interesting areas of a space quickly but without guaranteeing more convergence. So GA may be mixed with various local problem-specific search techniques to form a hybrid that will combine the globality and parallelism of GA with more convergence behavior of local search technique. In this paper, a simple genetic algorithm and a genetic algorithm with changing crossover is used. This paper proposes hybrid algorithms in which hill climbing is applied on each individual selected by selection operator for reproduction. The experiments have been conducted using three different inputs from TSPLIB provided by Heidelberg University and implementation is carried out using MATLAB. The result shows that the proposed algorithms perform better than the simple genetic algorithm in terms of producing more optimal results.**

Keywords: Crossover, Hill Climbing, Hybrid Genetic Algorithm, Travelling Salesman Problem.

I. INTRODUCTION

Genetic algorithms were developed as problem independent search algorithms [1, 2, 3] that simulate the biological evolution to search for an optimal solution to a problem. The basic principles of this technique were first given by John Holland and are well described, in [4]. GA is a method for moving from one population of "chromosomes" to a new population by using a kind of "natural selection" in combination 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 [5].

Genetic algorithms are inspired from biological genetics model and so most of its terminology has been borrowed from genetics. The resemblance between natural and GA terminology is given as [1]:

TABLE 1

Resemblance between Natural and GA Terminology

Natural	**Genetic Algorithm**
Chromosome	String
Gene	Feature, or character
Allele	Feature value
Locus	String position
Genotype	Structure
Phenotype	Parameter set, alternative solution

Genetic algorithm uses an iterative process to create a population. The algorithm stops, when the population converges towards the optimal solution or any termination criterion are met. A gene is said to have converged if 95% of the population have same value for that gene [4]. The population is said to have converged when all the genes in a chromosome have converged. The simple genetic algorithm has the following steps:

- *Initialization*

  GA begins its search from a random population of solutions which is called as initial population. Each solution is represented using a chromosome, which is just an ideal representation. Many coding schemes have been proposed by different researchers to accomplish this task, including binary encoding, hexadecimal encoding, octal encoding, permutation encoding, value encoding and tree encoding [1]. The selection of encoding scheme is problem dependent.

- *Evaluation*

  The fitness value of the each chromosome is evaluated by using the fitness function. The fitness corresponds to evaluation of how good the chromosome is.

- *Selection*

  Selection is done randomly with a probability depending on the fitness of individuals. This allocates more copies of the chromosomes in to mating pool with higher fitness values and thus imposes the survival-of-the-fittest mechanism [6]. The main idea of selection is to prefer best solutions than the poor ones. Various selection procedures have been proposed by different researchers to accomplish this idea, including roulette-wheel selection, rank based selection and tournament based selection [1].

- *Recombination*

  Crossover and mutation are applied on the mating pool individuals. Crossover operator is applied to the mating pool with the hope that it creates a better offspring by combination of meaningful building blocks. Various crossover operators are uniform crossover, N-point crossover, partially mapped crossover, order crossover, cycle crossover [1]. Mutation performs a random walk in the domain of a candidate solution.

- *Replacement*

  The new population created by recombination replaces the original chromosomes of the parental population. Various replacement techniques are proposed like elitist replacement and steady state replacement [1].

If GA has been designed well, the population will definitely converge to optimal solution to the problem. GA is a robust technique and can deal with a wide range of problem areas, including those also which are difficult for other methods to solve. Rather GAs are not guaranteed to find the global optimum solution, but they can find “acceptably good” solutions to problems “acceptably quickly”. Where specialized techniques exist to solve particular problems, they out-perform GA in both speed and accuracy of the result. The main motivation of GA is where no such technique exists. Even where problem specific techniques work well, improvements have been made by hybridizing them with GA [5].

In practice, the population size is finite that influences the performance of genetic algorithm and leads to the problem of genetic drift that occurs mostly in case of multimodal search space. Embodying a local search method within the genetic operators can introduce new genes that can overcome the problem of genetic drift and accelerate the search towards global optima [7]. Addition of problem-specific knowledge also maintains a balance between exploration and exploitation. A combination of genetic algorithm and a local search method is called as memetic algorithm (MA) or hybrid genetic algorithm (HGA). There is an opportunity in hybrid optimization to capture the best of both (Genetic Algorithm and Local Search) schemes [8]. In hybrid genetic algorithms, knowledge and local search can be embodied at any stage like initialization, selection, crossover and mutation.

This paper incorporates hill climbing based local search after selection step. Two kind of genetic algorithms are used. One is simple genetic algorithm and the other one is genetic algorithm with changing crossover. Both the algorithms are then hybridized with hill climbing based local search. The performances of these algorithms are compared on benchmark travelling salesman problem (TSP) instances.

TSP is one of the well known combinatorial optimization problem in which we have to find the tour of all nodes that has the minimum total cost [9, 10]. As number of cities gets large, it becomes impractical to find the cost of every tour in polynomial time and that is why TSP is NP-Hard problem. Various methods of optimization have been used to solve the TSP such as Hill Climbing [11], Simulated Annealing [12], Tabu Search [13], Ant Colony [14], Particle Swarm [15], and Genetic Algorithm [16, 17] etc. Here we have used GA to solve TSP, and get good solution in reasonable time & establishing the degree of goodness.

The paper is organized in the following sections. In section 2, literature review is given on different researches related to hybrid genetic algorithms. In section 3, proposed algorithm is discussed. Implementation details and computational results are specified in section 4 and conclusion is given in section 5.

II. LITERATURE REVIEW

Several researchers have studied the performance of GA. The performance is usually evaluated in terms of convergence rate and number of generations to reach the optimal solution. David Goldberg [1] and Holland [2] by using k-armed bandit analogy showed that both exploration and exploitation are used by genetic algorithm at the same time. Due to certain parameters, it has been observed that, stochastic errors occur in genetic algorithm that leads to genetic drift [18]. Memetic algorithm with self adaptive local search was given by N. Krasnogor for TSP and also finds its successful application to Protein folding problem. Proposed memetic algorithm composed of genetic algorithm and Monte Carlo method. The proposed memetic algorithm has been compared with hill climber memetic algorithm, Boltzmann hill climber memetic algorithm and found to provide more optimized result [19]. Talbi *et al.* [20] concluded that Local Search contributes marginally to enhance the performance of GA.

Antariksha Bhaduri [21] proposed a hybrid genetic algorithm based on GA and Artificial Immune network Algorithm (GAIN) for finding optimal collision free path in case of mobile robot moving in static environment filled with obstacles. She concluded that GAIN is better for solving such kind of problems. Burke and Smith has been proposed a memetic algorithm that was based on Tabu search technique to solve the maintenance scheduling problem. The proposed memetic algorithm performs better and can be usefully applied to real problems [22]. Bjornsdotter and Wessberg [23] proposed a memetic algorithm for feature selection in volumetric data containing spatially distributed clusters of informative features in neuroscience application and concluded that the proposed memetic algorithm identified a majority of relevant features as compared to genetic algorithm. Sanusi *et al.* [24] investigated the performance of genetic algorithm and memetic algorithm for constrained optimization knapsack problem. Their analysis results showed that memetic algorithm converges faster than genetic algorithm and produces more optimal result and memetic algorithm is better suited to solve the NP-Hard problems. Sivaraj *et al.* discussed about a novel approach to improve the performance of genetic algorithm by using selective initialization that aims at supplying more fit individuals in the beginning. The result showed that the selective initialization enhances the convergence velocity and produces more optimal solution than existing schemes used in generic genetic algorithm [25].

Rakesh Kumar and Jyotishree proposed a novel crossover operator that uses the principle of Tabu search. Their proposed crossover retains the best combination of alleles and avoids the introduction of worst combination of alleles in the offsprings. This increases the average fitness of the population and improves the performance of genetic algorithms in terms of convergence and number of iterations. They compared the proposed crossover with partially matched crossover (PMX) and found that the proposed crossover yielded better results than PMX [26]. Rakesh Kumar and Jyotishree [27] also proposed a blended selection operator having balanced tradeoff between exploration and exploitation. The performance of this proposed selection is then compared with roulette wheel selection (RWS) and rank based selection (RS). They observed that the proposed blended selection outperformed both the selections.

Kumar *et al.* surveyed some hybrid algorithms which are quite useful in particular scenarios. They observed that there should be control on some parameters like duration of local search and frequency of local search when mixing GA and local search. The uncontrollable parameters will lead to resource wastage. They 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 [28]. Manju Sharma and Girdhar Gopal proposed a hybrid crossover operator that uses the knowledge concept along with the hill climbing principle. They compared its performance with PMX and OX on the TSP Oliver30. They found that hybrid crossover operator yields better result than PMX and OX [29]. Girdhar Gopal *et al.* have discussed effect of hill climbing in initialization of GA for solving optimization functions and concluded that if we start from a good population in genetic algorithm, we have more chances to get optimum result in less number of generations [30].

III. PROPOSED ALGORITHM

In this section, firstly two simple genetic algorithms with variation of crossover operator are discussed. After that hybridization is applied on both of the algorithms. In these algorithms, De Jong's guidelines, which is to start with a relatively high crossover probability ($Pc \geq 0.6$), relatively low mutation probability (Pm, 0.001-0.1), and a moderately sized population is used. The selections of parameter values are very depend on the problem to be solved [4, 31].

3.1 Simple Genetic Algorithms

First algorithm follows the basic steps of simple genetic algorithm that can be described as:

Algorithm 1: Simple GA with PMX

a) Initialization using permutation encoding.
b) Selection of fit individuals using rank based selection.
c) Partially Matched Crossover (PMX) is applied to the mate pool individuals to generate child chromosomes.
d) Elitism (best individual in old generation will go in next generation).
   Step b), c) and d) will be repeated population size times.
e) Interchange Mutation that swaps the positions of two genes.
f) Partial Replacement that selects the best individuals that will go in next generation from the new chromosomes and the parental population.

Second algorithm that is defined in this paper is genetic algorithm with changing crossover. In this algorithm the crossover operator is changed after half of the generation count. This algorithm has following steps:

Algorithm 2: Simple GA with OX+PMX

a) Initialization using permutation encoding.
b) Selection of fit individuals using same selection mechanism as in the first algorithm.
c) If generation_count <= n/2
   // n is total number of generations.
   Order Crossover (OX) is applied.
   else
   Partially Matched Crossover (PMX) is applied.
   Steps b) and c) will be repeated population size times.
d) Interchange Mutation swaps the positions of two genes with mutation probability 0.1.
e) Partial Replacement replaces the parental population with the best fit individuals of both the populations.

3.2 Hybrid Genetic Algorithms

Incorporating problem specific information in a genetic algorithm, at any level of genetic operation, forms a hybrid genetic algorithm (HGA) [32]. Two popular ways of hybridization depends on the concepts of "Baldwin effect" [33] and "Lamarckism" [34]. According to Baldwinian search strategy, the local optimization can interacts and allow the local search to change the fitness of individual but genotype itself remain unchanged. The disadvantage of Baldwinism is that it is slow. According to Lamarckism, the characteristics acquired by individual during its lifetime may become heritable traits. According to this approach both the fitness and genotype of individuals are changed during local optimization phase. Most of the memetic algorithms are based on Lamarckism approach of hybridization.

The proposed hybrid algorithms incorporates hill climbing local search after selection process in order to increase exploitation. In the proposed approach, members are selected using rank based selection and those have been used as initial point to carry out hill climbing search. In this approach, each individual is improved using hill climbing before passing to reproduction phase. Hill climbing is an optimization algorithm for single objective function. In this algorithm, a loop is performed in which the currently known best individual produces one offspring. If the fitness of new individual is better than parent it replaces it, otherwise stops the loop. Its algorithmic steps are given below:

Algorithm 3: Hill Climbing Algorithm

```
// A parent is selected
while (termination criteria is not satisfied) do
offspring <- neighbor (parent)
if (offspring is better than parent)
    parent <- offspring
    end if
end while
```

This algorithm will return best solution in the search area of the selected parent. So it makes the convergence faster towards global optima.

This hill climbing algorithm is applied on both of the algorithms 1 & 2, so now the proposed algorithms have following steps:

a) Initialization
b) Selection
c) Hill Climbing Algorithm
d) Recombination
e) Replacement.

IV. IMPLEMENTATION AND RESULTS

In this paper, MATLAB code has been developed to find the performance of proposed hybrid algorithms and genetic algorithms. The performance of the algorithms is compared at three TSP instances: has been taken from TSPLIB. Three problem instances were taken for experiments from standard TSP library TSPLIB are Eil51, Eil76 and Eil101 where 51, 76 and 101 cities are included respectively [35].

The following parameters are used in this implementation:

- Population size: 50,100
- Number of generations: 250,500
- Selection method: Rank Based Selection (RS)
- Crossover Operator: Order Crossover (OX) and Partially Matched Crossover (PMX)
- Mutation: Interchange with mutation probability 0.01
- Replacement: Partial Replacement
- Algorithm ending criteria: Execution stops on reaching given number of generations.
- Fitness Function: Objective value of function.

The maximum number of generation is entered at runtime of program. The size of population is also entered at runtime of program.

Following are the tables to show the results and to observe the experiments.

TABLE 2

Minimum tour cost found with population size =50 and generations =250

Algorithm/ Instances	**SGA with PMX**	**SGA with OX+PMX**	**HGA with PMX**	**HGA with OX+PMX**
Eil51	1147	665	593	501
Eil76	1753	1079	882	784
Eil101	2495	1628	1266	1185

TABLE 3

Minimum tour cost found with population size =100 and generations =500

Algorithm/ Instances	**SGA with PMX**	**SGA with OX+PMX**	**HGA with PMX**	**HGA with OX+PMX**
Eil51	1089	546	563	461
Eil76	1660	892	761	648
Eil101	2165	1306	1110	887

The performance graphs in figures below show the minimum distance found by the algorithm in each generation. Fig. 1, 2 and 3 are for Eil51, Eil76 and Eil101 instances respectively. The figures are representing the result for population size= 100 and generations= 500. It can be seen from the graphs that the distance with proposed algorithms reduced towards optimal solution as the generation increased and finally converged at a certain generation.

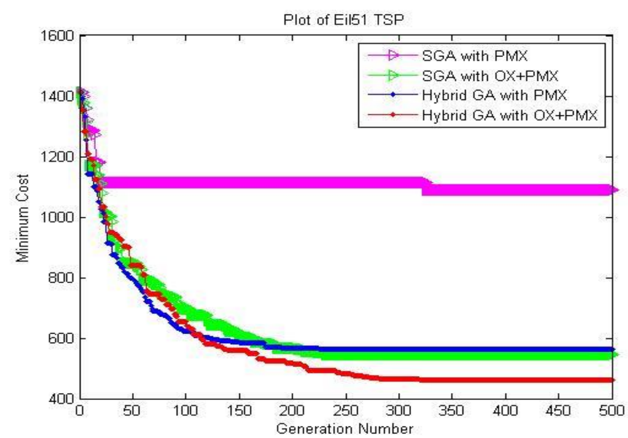

fig. 1. Comparison of minimum tour length for Eil51

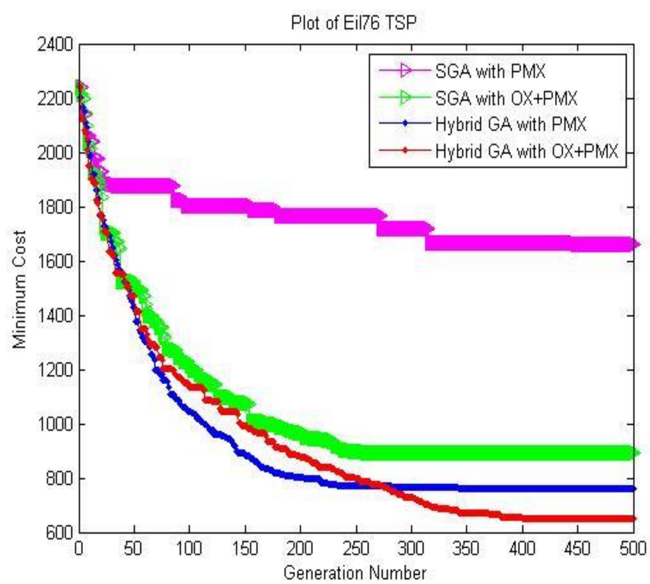

fig. 2. Comparison of minimum tour length for Eil76

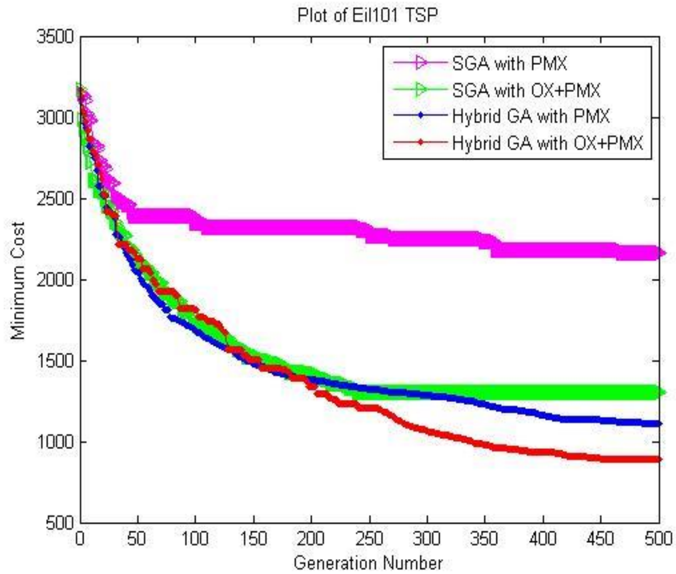

fig. 3. Comparison of minimum tour length for Eil101

V. CONCLUSION

The paper compares simple genetic algorithms and proposed hybrid algorithms on the given instances of travelling salesman problem in TSPLIB. It is found that the proposed hybrid algorithms provide better results than the simple genetic algorithms. The proposed algorithms use the concept of hill climbing local search after selection operation so as to allow the genetic algorithm to improve the exploiting ability of search without limiting its exploring ability. The proposed algorithms improve the performance in terms of convergence and optimal solution as well as maintain diversity in the population and solve the problems of premature convergence and genetic drift. The proposed algorithm can prove to be better for different NP- Hard problems also.

REFERENCES

[1] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, 13 ed., Addison-Wesley Publishing Company, 1989.

[2] J. H. Holland, Adaptation in Natural and Artificial Systems, Ann Arbor: University of Michigan Press, 1975.

[3] K. Man, K. Tang and S. Kwong, Genetic Algorithm: Concept and Designs, London: Springer Science & Business Media, 1999.

[4] K. A. De Jong, "An analysis of the behaviour of a class of genetic adaptive systems," University of Michigan, Ann Arbor, 1975.

[5] D. Beasley, D. Bull and R. Martin, "An overview of Genetic Algorithms: Part 1," *Fundamentals,* vol. 15, pp. 58-69, 1993.

[6] T. Baeck, D. Fogel and Z. Michalewicz, Evolutionary Computation 1: Basic Algorithms and Operators, CRC Press, 2000.

[7] W. E. Hart, "Adaptive global Optimization with Local Search," University of California at San Diego La Jolla, San Diego, 1994.

[8] A. Martinez-Estudillo, C. Hervas-Martinez, F. Martinez-Estudillo and N. Garcia-Pedrajas, "Hybridization of evolutionary algorithms and local search by means of a clustering method," *IEEE Transactions on Systems, Man and Cybernetics,* vol. 36, no. 3, pp. 534-545, 2005.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest and S. Clifford, Introduction to Algorithms, 2nd ed., MIT Press, 2009.

[10] E. Horowitz, S. Sahni and S. Rajasekarn, Fundamentals of Computer Algorithms, Galgotia Publications, 1999.

[11] I. Stojanovska and A. Dika, "Impact of the Number of Generations on the Fitness value and the Time Required to Find the Optimal Solution in Standard GA Applications," in *IEEE Coference*, 2010.

[12] P. Laarhoven and E. Aarts, "Simulated Annealing," in *Simulated Annealing: Theory and Applications*, Springer Netherlands, 1987, pp. 7-15.

[13] R. Thamilselvan and P. Balasubramanie, "A Genetic Algorithm with a Tabu Search (GTA) for Traveling Salesman Problem," *International Journal of Recent Trends in Engineering,* vol. 1, no. 1, pp. 607-610, 2009.

[14] M. Dorigo and L. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Transactions on Evolutionary Computation,* vol. 1, no. 1, pp. 53-66, 1997.

[15] R. C. Eberhart and J. Kennedy, "A New Optimizer using Particle Swarm Theory," Nagoya, 1995.

[16] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution programs, Berlin: Springer Science & Business Media, 1996.

[17] Z. Wang, H. Duan and X. Zhang, "An Improved Greedy Genetic Algorithm for Solving Travelling Salesman Problem," in *Proc of the 2009 IEEE Fifth International Conference on Natural computation (ICNC)*, 2009.

[18] D. E. Goldberg and P. Segrest, "Finite Markov chain analysis of genetic algorithms," in *Proceedings of 2nd International Conference on Genetic Algorithms Lawrence Erlbaum Associates*, Tuscaloosa, 1987.

[19] N. Krasnogor and J. Smith, "A memetic algorithm with self adaptive local search: TSP as a case study," 2000.

[20] E. G. Talbi, M. Rahoual, M. H. Mabed and C. Dhaenens, "A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop," Zurich, 2001.

[21] A. Bhaduri, "A mobile robot path planning using genetic artificial immune network algorithm," in *Proceedings of World Congress on Nature and biologically Inspired Computing*, 2009.

[22] E. K. Burke and A. J. Smith, "A memetic algorithm for the maintenance scheduling problem," in *International Conference on Neural Information Processing and Intelligent Information*, 2010.

[23] M. Bjornsdotter and J. Wessberg, "A memetic algorithm for selection of 3D clustered featured with applications in neuroscience," in *International Conference on Pattern Recognition*, 2010.

[24] H. A. Sanusi, A. Zubair and R. O. Oladele, "Comparative assessment of genetic and memetic algorithms," *Journal of Emerging Trends in Computing and Information Science,* vol. 2, no. 10, pp. 498-508, 2011.

[25] R. Sivaraj, T. Ravichandran and R. D. Priya, "Boosting performance of genetic algorithm through selective initialization," *European Journal Of Scientific Research*, vol. 68, no. 1, pp. 93-100, 2012.

[26] R. 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, 2012.

[27] R. Kumar and Jyotishree, "Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms," *International Journal of Machine Learning and Computing,* vol. 2, no. 4, pp. 365-370, 2012.

[28] R. Kumar, G. Gopal and R. Kumar, "Hybridization in Genetic Algorithms," *International Journal of Advanced Research in Computer Science and Software Engineering* , vol. 3, no. 4, pp. 403-409, 2013.

[29] M. Sharma and G. Gopal, "Hill Climbing Based Hybrid Crossover in Genetic Algorithms," *International Journal of Advanced Research in Computer Science and Software Engineering* , vol. 3, no. 4, pp. 468-473, 2013.

[30] G. Gopal, R. Kumar and N. Kumar, "Effect of Hill Climbing in Initialization of GA for solving optimization functions," *Journal of Environmental Science, Computer Science and Engineering & Technology* , vol. 4, no. 1, pp. 1-12, 2015.

[31] K. De Jong and W. M. Spears, "Using Genetic Algorithms to Solve NP-Complete Problems," in *Third International Conference on Genetic Algorithm*, San Mate. California , 1989.

[32] P. Moscato and C. Cotta, "A gentle introduction to memetic algorithms," in *Handbook of Metaheuristics*, Springer US, 2003, pp. 105-144.

[33] K. Ku and M. Mak, "Empherical analysis of the factors that affect the Baldwin effect: Parallel problem solving from nature," in *5th International Conference on lecture notes in computer science*, Amsterdam, 1998.

[34] G. M. Morris, D. S. Goodsell, R. S. Halliday, W. E. Hart, R. K. Belew, R. Huey and J. Olson, "Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function," *Journal of Computational Chemistry,* vol. 19, no. 14, pp. 1639-1662, 1998.

[35] G. Reinelt, "TSPLIB – A Travelling Salesman Problem Library," *ORSA Journal on Computing,* vol. 3, no. 4, pp. 376-384, 1991.

