

# A NOVEL GREEDY GENETIC ALGORITHM TO SOLVE COMBINATORIAL OPTIMIZATION PROBLEM

M. A. Basmassi<sup>1,\*</sup>, L. Benameur<sup>2</sup>, J. A. Chentoufi<sup>1</sup>

<sup>1</sup> ISO Laboratory, Faculty of Sciences, Ibn Tofail University Kenitra, Morocco - basmassi.med.amine@gmail.com,  
j.alami@uit.ac.ma

<sup>2</sup> LIROSA Laboratory, Faculty of Sciences, Abdelmalik Essaadi University Tetouan, Morocco - lamia\_benameur@yahoo.fr

**KEY WORDS:** Greedy Sequential Algorithm, Genetic Algorithm, Combinatorial Optimization Problem, Graph Coloring, Computational Intelligence, Chromatic Number

## ABSTRACT:

In this paper, a modified genetic algorithm based on greedy sequential algorithm is presented to solve combinatorial optimization problem. The algorithm proposed here is a hybrid of heuristic and computational intelligence algorithm where greedy sequential algorithm is used as operator inside genetic algorithm like crossover and mutation. The greedy sequential function is used to correct non realizable solution after crossover and mutation which contribute to increase the rate of convergence and upgrade the population by improving the quality of chromosomes toward the chromatic number. Experiments on a set of 6 well-known DIMACS benchmark instances of graph coloring problem to test this approach show that the proposed algorithm achieves competitive results in comparison with three states of art algorithms in terms of either success rate and solution quality.

## 1. INTRODUCTION

Genetic algorithm is a metaheuristics that appertains to the computational intelligence algorithms, inspired from natural selection process, for solving both constrained and unconstrained optimization problems. The genetic algorithm (Mitchell, 1998) is commonly used to generate high-quality solutions. The process generally starts with a population of random generated individuals, then repeatedly modifies a population of individual solutions in an iterative process. The fitness is the worth of objective function that test how much the individual is near from the optimal solution. In each generation, the selection of individuals is done at random from the current population to be parents. These parents are used to produce new individuals, after that the mutation modified randomly a minority of them depending on the coefficient of mutation. Then, the new generation of candidate solutions is employed in the following iteration of the algorithm. Commonly, the algorithm end once a maximum number of generations has been reached, or a satisfying fitness stage has been produced for the problematic.

The genetic operators crossover and mutation do not maintain the adjacency constraint and in the most cases they violate it for that an algorithm to verifies and correct the affected new individuals is needed. The greedy sequential algorithm was the best solution for that.

The greedy sequential algorithm proposed in (Erciyes, 2013) to solve the graph coloring problem where, it labels the vertices of a graph according to some specific order and assigns the smallest possible index color to a vertex that has not been assigned to its neighbors. Such ordering is called a coloring heuristic. The palette contains a list of colors up to  $\Delta+1$ , where  $\Delta$  is the maximum degree. For each vertex  $V$ , an array where entries display the current allocated colors of the neighbors is

used. The algorithm begins with allocating the smallest index of accessible color to an uncolored vertex  $V$  from the palette that its neighbors have not been assigned. Once the color is allocated to  $V$ , the color is included to the array of neighbors for every neighbor of  $V$ , in order to avoid using this color in future (Erciyes, 2013).

The Graph Coloring Problem (GCP) is a well-known NP-complete problem. It includes both vertex coloring and edge coloring. Though, the graph coloring term in general goes to vertex coloring more than edge coloring (Jensen, 2013). For a specified set of vertices, that form a connected graph, the first aim is to colors vertices, on condition that no edge connects two identically colored vertices. It also aims to find the minimum number of colors to color the graph without violating the adjacency constraint. That number is called The Chromatic Number  $\chi(G)$  (Erciyes, 2013).

For a given graph  $(G)$ ,  $P(G,c)$  is the number of possible solutions for coloring the graph  $G$  with  $c$  colors and  $c=1,2,3\dots$ , Then, the chromatic number is given by the following formula:

$$\chi(G) = \min(c : P(G,c) > 0) \quad (1)$$

In this paper a new approach combine genetic algorithm and greedy sequential algorithm is presented to solve a combinatorial optimization problem. The graph coloring is computationally hard, for that it was chosen to find out how effective this approach is.

In the next section, a short view on prior works of genetic algorithm to solve GCP is presented. The proposed algorithm is presented in details in section 3. Finally, in section 4, experimental results are presented and compared to other methods reported in the literature.

\* Corresponding author

## 2. PRIOR WORKS

Many genetic algorithm models have been introduced by researchers largely working from an experimental perspective. Many of these researchers are application oriented and are typically interested in genetic algorithms as optimization tools. In (Galinier, 1999), a combine genetic algorithm with a special crossover operator and tabu search algorithm was presented. The Branch-and-Cut algorithm based on an integer programming formulation is proposed for solving graph-coloring instances that are out of the reach of Dsatur method (Méndez-Díaz, 2006). Also, a memetic algorithm (Moscato, 1999) integrating a tabu search procedure with genetic algorithm to solve the GCP called MACOL algorithm is presented in (Zhipeng, 2010). Furthermore, many algorithms based on local search metaheuristic methods include Iterated Local Search (Chiarandini, 2002), Greedy Randomized Adaptive Search Procedure (Laguna, 2001) and Reactive Partial Tabu Search (Blöchliger, 2008) was implemented to resolve the GCP. Interested readers are directed to (Galinier, 2006) for an inclusive review of the local search approaches for graph coloring. Additionally, in (Agrawal, 2015), swarm intelligence-based technique called as particle swarm optimization is employed. In (Adegbindin, 2016), a recursive largest first-like algorithm gives good results, allowing to obtain a reduction of more than the half of the gap between the number of colors used and the best known upper bound of the chromatic number.

The GCP is one of the combinatorial optimization problems. While, a lot of computational intelligence techniques and heuristics algorithm were innovated to resolve GCP however they have several limitations in one way or another. Recently, a hybrid cultural-hierarchical clustering algorithm to solve the GCP is proposed in (Basmassi, 2019), this novel approach use a hybrid of hierarchical clustering and genetic algorithm to present the population space, the belief space is updated by the best individual's depending on their fitness and number of used colors.

This paper presents the resolution of the GCP by merging a ameliorated genetic algorithm, which uses the greedy sequential algorithm to improve the fitness of individuals in every generation to increase the convergence rate and toward the global optimal solution. The suggested method is performed on multiple instances of graphs imported from the DIMACS library (Trick, 2002) and the attained results are compared with others reported in the literature (Yüceoğlu, 2017), (Douiri, 2015) and (Zhou,2016).

## 3. THE PROPOSED APPROACH

The proposed approach novel greedy genetic algorithm (NGGA) combines genetic algorithm (GA) and greedy sequential algorithm (GSA) to solve the GCP. In NGGA, the genetic algorithm is selected as the framework that explores the search space. Because, the rate at which the algorithm can explore a space of possibilities is satisfying and adequate to the studied problematic. The greedy sequential algorithm is injected as operator in genetic algorithm to accelerate the search and improve chromosomes fitness.

The initial population is random, it contain feasible and non-feasible solution, then, parent selection which is based on the tournament selection, the crossover with three points, beside creep mutation and a sequential correction function, finally all chromosomes are evaluated and new parent populations is selected for another cycle of life.

The pseudo-code of the proposed method is as follow:

*Begin*

*Generate the initial population POP(0);*

*Evaluate POP(0);*

*Repeat*

*Select parents;*

*Generate new chromosomes using crossover;*

*Applied mutation on the new chromosomes;*

*Applied greedy sequential function;*

*Evaluate POP(t);*

*Until (Terminating condition is reached);*

*End;*

For a graph G, the chromatic number,  $\chi(G)$ , is the smallest number of colors to color G (Erciyes, 2013). The Calculation of  $\chi(G)$  is NP-complete.

**Initial Population:** The proposed algorithm begins with a population P of n feasible coloring graphs. This population can be acquired via any graph coloring algorithm which is capable to produce different appropriate colorings for a graph. In this case, initial population function is created to generate different chromosomes with different number of colors randomly.

**Parents selection:** The tournament selection is used as the selection method, where two individuals are randomly selected from the population, and then the best out of these is selected to become a parent.

**Crossover:** In this operator, two parents are selected and two children are produced using the genetic material of the parents. In this approach, a three points crossover is used.

**Mutation:** The uniform mutation is used to maintain and introduce diversity in the genetic population and it is applied with a small probability.

**Sequential correction algorithm:** This algorithm is mainly based on greedy sequential algorithm presented in the introduction. For each new child, all the genes that do not verify the constraints of the problem are checked. In this case, a different color from its neighbor is affected but it should be one of the used colors in the chromosomes, else no modification is performed.

The pseudo-code of the sequential correction algorithm function is given as follow:

*Begin*

*CH: New Chromosome.*

*Palette  $\leftarrow$  Get used colors in CH;*

*Repeat*

*V  $\leftarrow$  Select gene from CH;*

*CH  $\leftarrow$  CH  $\setminus$  {V};*

*if V does not satisfy the constraint then :*

*InValidColors  $\leftarrow$  Adjacent Gene Colors of V;*

*ValidColors  $\leftarrow$  Palette  $\setminus$  InValidColors;*

*If ValidColors  $\neq$   $\emptyset$  then :*

*V  $\leftarrow$  Any color from ValidColors;*

*End if;*

*End if;*

*Until (Terminating condition is reached);*

*End;*

**Child generation:** This population includes n-1 new childes chromosomes gets by the modified genetic process (where n is the population size) and the best individual from the parent generation (Elitism strategy), which guarantees that the best

solution obtained by the GA so far will be preserved among the generations.

#### 4. EXPERIMENTAL RESULTS

##### 4.1 Data

The data employed to test the proposed approach are derived from the DIMACS benchmarking graph collection. DIMACS is the Center for Discrete Mathematics and Theoretical Computer Science (Trick, 2002).

This collection is habitually used in experiments involving constraint satisfaction problems.

For the genetic algorithm used at this work, the crossover probability used is  $P_c=90\%$ , and  $P_m=1\%$  for the mutation probability. The algorithm is run for 100 generations and the population size used is 200.

##### 4.2 Results and discussion

The obtained results using the proposed approach are depicted in Table 1. Each line in the table corresponds to a particular graph. The first columns in Graph part indicate, respectively, the name, the number of vertices's, the number of edges and the chromatic number at the considered graph. The next columns displays the best optimal number and the success rate of the K[NGGA] that represent the obtained results using our suggested algorithm.

Graph				K[NGGA]	
Name	V	E	$\chi(G)$	K	Suc/15
DSJC125.1	125	736	5	6	15/15
Games120	120	638	9	9	15/15
Miles500	128	1170	20	20	13/15
DSJC125.5	125	3891	17	17	2/15
DSJC125.9	125	6961	44	44	7/15
David	87	406	11	11	15/15

Table 1. The NGGA results

A brief explanation was done to give an idea about algorithms chosen for comparing results.

In (Yüceoğlu, 2017), A column generation based algorithm is proposed to solve the robust graph coloring problem which is a generalization of the original graph coloring problem the algorithm is composed of two phases. The column generation phase which requires constantly generating columns with negative reduced cost, that corresponds to solve a pricing sub-problem. The pricing sub-problem is resolved at each iteration, it also generates new columns to expand restricted master problem or indicate that an optimal solution for the linear programming relaxation of master problem is obtained. In second phase, a branch and-bound method is employed along with the column generation procedure for each node of the branch-and-bound tree. This method is called branch-and-price. Information from the pricing sub-problem and the optimal solution to master problem is used to find an integer feasible solution.

In (Douiri, 2015) a hybrid genetic algorithm based on a local search heuristic called DBG. The initialization of the number  $k$  of colors is carried out by the DBG algorithm to give an approximate values of  $\chi(G)$ . Then the genetic algorithm is used to find and improve an optimal solution for GCP in experimental parameter the authors functioned with a

population equal to 120, a crossover probability  $P_c=0.7$ , a mutation probability  $P_m=0.15$  and iteration number varies between 250 and 2000.

In (Zhou,2016) an algorithm named reinforcement learning based local search (RLS) combines reinforcement learning and local search algorithms for solving grouping problems. The graph coloring problem was selected as the case of study to test the performance of the RLS approach.

The Table 2. display the best optimal number and the success rate of The K[Yüceoğlu], K[Douiri] and K[Zhou] representing respectively the best results of the used graphs for comparison in (Yüceoğlu, 2017), (Douiri, 2015) and (Zhou,2016).

Graph	K[Yüceoğlu]		K[Douiri]		K[Zhou]	
	K	Suc/15	K	Suc/15	K	Suc/15
Name	K	Suc/15	K	Suc/15	K	Suc/15
DSJC125.1	8	-	6	14/15	5	15/15
Games120	14	-	9	15/15	-	-
Miles500	30	-	20	10/15	20	15/15
DSJC125.5	26	-	17	8/15	17	11/15
DSJC125.9	66	-	44	7/15	44	15/15
David	17	-	11	14/15	-	-

Table 2. State-of-the-art results

The challenge tackled in this work is choosing common graphs in three recent state-of-arts to show how the proposed algorithm is flexible to give optimum result whatever the graph is.

Comparisons between these algorithms show that the NGGA gives better results than the column generation based algorithm in the all cases. Furthermore, results obtained by the hybrid genetic algorithms (Douiri, 2015) are the same results as the proposed algorithm for DSJC125.9 and Games120. Besides, the proposed algorithm have been successful to surpassed (Douiri, 2015) in the rate of trials that were run for DSJC125.1, Miles500 and David. Unless, DSJC125.5 where the proposed algorithm does not succeed to surpass it. Concerning (Zhou, 2016) the results show that the RLS algorithm exceed the proposed approach in DSJC125 and Miles500 graphs.

#### 5. CONCLUSION

In this paper, a Novel Greedy Genetic Algorithm is proposed to solve the combinatorial optimization problem different size problem instances. the hybridization of the algorithms used in this work show a harmonic combination, Genetic algorithm operators explore the space of solutions and the greedy sequential algorithm repair the affected solutions and intelligently adjusts its chromosomes to avoid local search and to reduce computation cost, all that contribute to a fast convergence and localization of the optimal solution. The computational experiments, carried out on a set of 6 benchmark graphs in three state-of-arts, show that our algorithm is able to provide strong performance on the majority of instances. The results obtained, using the proposed approach, are competitive compared to the current best-known results reported in state-of-art.

#### REFERENCES

Adegbindin, M., Hertz, A., Bellaiche, M., 2016: A new efficient RLF-like algorithm for the vertex coloring problem. Yugoslav Journal of Operations Research, 26(4), 441-456.

- Agrawal, J., Agrawal, S. 2015: Acceleration based particle swarm optimization for graph coloring problem. *Procedia Computer Science*, 60, 714-721.
- Basmassi, M.A., Benameur L., Chentoufi J.A., 2019: A Hybrid Cultural-Hierarchical Clustering Algorithm For The Graph Coloring Problem, *Journal of Advanced Research in Dynamical and Control Systems*, 11(6-SI), 1834-1842
- Blöchliger, I., Zufferey, N. 2008: A graph coloring heuristic using partial solutions and a reactive tabu scheme. *Computers & Operations Research*, 35(3), 960-975.
- Chiarandini, M., Stützle, T., 2002: An application of iterated local search to graph coloring problem. In *Proceedings of the Computational Symposium on Graph Coloring and its Generalizations*, 112-125.
- Douiri, S.M., Elbernoussi, S., 2015: Solving the graph coloring problem via hybrid genetic algorithms. *Journal of King Saud University Engineering Sciences*, 27, 114 -118.
- Erciyes, K., 2013: Distributed Graph algorithms for computer Networks. Computer Engineering Department Izmir University Uckuyular, Izmir, Turkey, Springer-Verlag London, 107-123.
- Galinier, P., Hao, J.K., 1999: Hybrid evolutionary algorithms for graph coloring, *Journal of combinatorial optimization*, 3(4), 379–397.
- Galinier, P., Hertz, A., 2006: A survey of local search methods for graph coloring. *Computers & Operations Research*, 33(9), 2547-2562.
- Jensen, T.R., Toft, B., 2011: *Graph Coloring Problems*, John Wiley & Sons.
- Laguna, M., Martí, R., 2001: A GRASP for Coloring Sparse Graphs. *Computational Optimization and Applications* 19, 165–178. <https://doi.org/10.1023/A:1011237503342>
- Méndez-Díaz, I., Zabala, P., 2006: A Branch-and-Cut algorithm for graph coloring, *Discrete Applied Mathematics*, 154(5), 826–847.
- Mitchell, M., 1998: *An Introduction to Genetic Algorithms*. Massachusetts Institute of Technology Press, Cambridge, Massachusetts, London, England.
- Moscato, P., 1999: *Memetic algorithms: a short introduction. New ideas in optimization*. McGraw-Hill Ltd., UK, GBR, 219–234.
- Trick, M., 2002: *Computational Series: Graph Coloring and its Generalizations*. Retrieved April 02, 2020, from <http://mat.gsia.cmu.edu/COLOR02/>.
- Yüceoğlu, B., Şahin, G., Van Hoesel, S.P.M. 2017: A column generation based algorithm for the robust graph coloring problem, *Discrete Applied Mathematics*, 217, 340–352.
- Zhipeng, L., Jin-Kao, H., 2010: A memetic algorithm for graph coloring, *European Journal of Operational Research*, 203(1), 241-250.
- Zhou, Y., Hao, J. K., & Duval, B. 2016: Reinforcement learning based local search for grouping problems: A case study on graph coloring. *Expert Systems with Applications*, 64, 412-422.