Fireworks Algorithm for Graph Coloring

Raouan El Ghazi[1], Lamia Benameur[2], and Alami Chentoufi Jihane [3]
[1] Faculty of Science, Ibn Tofail University, BP 133, Kenitra, Morocco
[2]Faculty of Science, Abdelmalek Essaadi University, Tetouan, Morocco
[3] Faculty of Science, Ibn Tofail University, BP 133, Kenitra, Morocco

Keywords: Graph coloring problem (GCP), Graph theory, Firework algorithm (FWA), Optimization algorithm, Swarm intelligence, Metaheuristics.

Abstract: The graph-coloring problem has a long and illustrious history, and it is one of the most famous problems in the field of graph theory which consists of determining the minimum number necessary to color the vertices of any graph such that no two adjacent vertices have the same color. In this context, a set of metaheuristics are proposed to restrict the search for a solution. In this study, a new swarm intelligence optimization algorithm called fireworks algorithm FWA will be applied to graph coloring problems to speed up the optimization process, reduce the time, and improve overall performance.

1 INTRODUCTION

The mathematician Leonhard Euler starts coloring theory specifically in 1736 with the seven bridges of the Königsberg problem. The problem was a famous puzzle that concerns the possibility to walk in the Königsberg by crossing all bridges exactly once. Then the problem of the four colors always raises the question of whether four colors are sufficient to draw any map so that neighboring regions do not have the same color. Many types of research on the problem have focused on theoretical aspects such as planar graphs, triangle-free graphs, and random graphs.

The problem of the coloring of graphs has been studied by many authors and has developed in various disciplines like chemistry, biology, social sciences also in the field of operation research, such as register allocation (Chaitin;2004), frequency assignment (Aardel, Hoesel, Koster, Mannino & Sassano;2002), school and university timetabling (Burke &Newall ;1999) but also for its theoretical aspects. The GCP is closely associated with a classical NP-hard combinatorial optimization problem (Garey & Johnson; 1979).

Many algorithms have been proposed for this problem. The DSATUR algorithm (Brélaz, 1979) is one of the most well-known algorithms due to its efficiency and simplicity. Leighton proposed another popular algorithm in the same year. The Recursive Largest First (RLF) algorithm also the Bruch and-cut algorithm (Méndez-Diaz & Zabala; 2006). These methods can only resolve the small size. The optimization algorithms has become necessary used for large problems to find an approximate solution in a reasonable time. Heuristics generally make it possible to find acceptable solutions in a reasonable time. However, they do not offer any guarantee as to the optimality of the best solution found.

For this reason, the researchers are interested in heuristic approaches. Several metaheuristic methods had been applied to solve the GCP who seems the most promising way in finding exciting solutions. The most efficient heuristic approaches are local search methods like variable neighborhood search (Avanthay, Hertz& Zufferey; 2003), tabu search (Hertz & de Werra; 1987), and variable space search (Hertz, Plumettaz, and Zufferey; 2008). As well as hybrid population-based methods such as Genetic and hybrid algorithms (Fleurent & Ferland, 1996) and Hybrid evolutionary algorithms (Galinier & Hao, 1999).

Advised Fireworks algorithm has been increasingly popular in the blind optimization algorithms. This algorithm can be used to solve complex optimization problems. In this paper, we proposed the new swarm algorithm called fireworks.

FWA is a newly developed in swarm intelligence optimization algorithm based on the phenomenon of fireworks explosions in the sky at night. Y. Tan and Y. Zhu was proposed FWA in 2010 as an

El Ghazi, R., Benameur, L. and Chentoufi Jihane, A.
Fireworks Algorithm for Graph Coloring.
DOI: 10.5220/0010734500003101
In *Proceedings of the 2nd International Conference on Big Data, Modelling and Machine Learning (BML 2021)*, pages 366-370
ISBN: 978-989-758-559-3

optimization technique. Then it has been successfully applied in many fields and solving partial optimization problems, such as multi-satellite control resource scheduling, hardware/software partitioning, also, it has resolved image identification and spam detection, etc....

We organize this paper as follows: In section 2, we present a mathematical formulation of the graph-coloring problem. In section 3, an overview of the fireworks algorithm. Then in section 4, we describe our proposed method for GCP. In the end, we conclude and give some perspectives.

2 GRAPH COLORING PROBLEM

Let $G = (V, E)$ be a graph with a set of vertices V connected by a set of edges E. Given a positive integer k. A $\mathrm{k}-$coloring of G is a partition of V into k sets $\{V1, \ldots, Vk\}$, which called colors (i.e.,c: $v->$ $\{1, \ldots, k\}$), the graph coloring problem seeks to assign each vertex $v \in V$ such that no two vertices in the same set are adjacent.

If two adjacent vertices u and v have the same color, we say they called conflicting vertices, and the edge (u; v) ∈ E called a conflicting edge. The $\mathrm{k}-$ coloring called legal if there are no conflicting edges else, it is illegal.

Suppose that f is the objective function that associates with each coloring the number of conflicting edges. We can formulate the graph coloring problem as:

$$X(G) = \min_{x \epsilon X}\{f(x) : x \epsilon S\} \quad (1)$$

X: The set of coloring of the vertices $x \epsilon X$ can be uncolored vertices.

S: The set of realizable or legal coloring.

f: Number of colors of the coloring $x \epsilon X$.

We note $A(i,j)$ the matrix of conflict, and c (i): the associated color with a vertex i.

$$\mathrm{A}(i,j) = \begin{cases} 1 \text{ if } \mathrm{c}(i) = \mathrm{c}(j) \, and \, \{i,j\} \epsilon E \\ 0 \, otherwise \end{cases} \quad (2)$$

3 FIREWORKS ALGORITHM

When a fireworks display detonated. A shower of sparks will fill the local space. This explosion process is seen as a search in the local space around a specific point, and when we want to find a point X_j satisfying $f\,(x_j) = y$, we can continue to set off the fireworks in the search space until the optimal solution reached. .For each fireworks display, an explosion process is initiated and a shower of sparks fills the local space that surrounds it. The fireworks as well as the new sparks generated represent potential solutions in the research space.

We can describe the FWA algorithm as setting off N fireworks at N different locations, then we will evaluate the locations of the sparks obtained, the algorithm ends when the optimal location found. Otherwise, we will choose another N location and fireworks for the next generation. Many extensions of this algorithm saw the light. In 2013, a new method was presented to calculate the number of explosion sparks and the magnitude of the explosion of a firework display (J. Liu, S. Zheng, and Y. Tan, 2013), and an improved fireworks algorithm was proposed with five significant improvements. Later in 2014, an extension of the FWA was presented as the fireworks algorithm dynamic (Dynamic search in fireworks algorithm, 2014).

In order to ensure diversity and to balance global and local research, the explosion magnitude and population is generated among the new fireworks.

3.1 Explosion

After generating N fireworks randomly, then the fireworks, N generates sparks in the feasible space. The explosion operator is a key factor in the algorithm and plays a very important role. Indeed, the latter includes the explosion force, the explosion amplitude and the displacement operation

The number of explosion sparks of each firework is calculated as follows:

$$S_i = m.\frac{Y_{max} - f(x_i) + \varepsilon}{\sum_{i=1}^{n}\,(Y_{max} - f(x_i)) + \varepsilon} \quad (3)$$

Where m is a parameter that controls the overall number of sparks.

f(x) is an objective function, and x_{min} and x_{max} denote the limits of the potential space

Y_{max} Is the maximum (worst) value of the objective function among the n fireworks.

ε Indicates the smallest constant in the computer, and it used to prevent the denominator from becoming zero.

The solution is good (many sparks will be generated) if the difference between the best and the bad is within a smaller amplitude. To avoid the overwhelming effects of fireworks, limits on the number of sparks $\hat{S}_i$ are defined, as shown in the following equation

$$\hat{S} = \begin{cases} round(a.m) \\ round(b.m) \quad if \; S_i > bm \\ round(S_i) \end{cases} \quad (4)$$

Where a and b are constant.

After the number of sparks is determined, we will calculate the amplitude of a particular explosion to present the best fitness values as follows:

$$A_i = \hat{A}.\frac{f(x_i)-Y_{min}+\varepsilon}{\sum_{i=1}^{n}(f(x_i)-Y_{min})+\varepsilon} \quad (5)$$

$\hat{A}$: Denotes the maximum explosion amplitude.

Y_{min} = min (f (xi)) (i = 1, 2, …, n) is the minimum (best) value of the objective function among the N fireworks.

ε: has the same meaning.

Then, it is necessary to determine the sparks displacement in the explosion amplitude. Displacement operation is to make displacement on each dimension of a firework. Through the explosion operator, each firework generates a shower of sparks, helping to find the global optimal of an optimization function.

Algorithm 1: Generate a Gaussian spark

1. Calculate the fitness value f (xi) for each firework.
2. Select randomly z dimensions
   z = round (d. rand (0,1))
3. Calculate the coefficient of Gaussian explosion g = N (1, 1).
4. **For** each dimension $x_k^j \in$ {pre-selected z dimensions of x_k^j} **do**
   $$x_k^j = x_k^i.g$$
5. if $x_k^j < x_k^{min}$ or $x_k^j < x_k^{max}$ **then**
6. $x_k^j = x_k^{min} + -|x_k^j|\%(x_k^{max} - x_k^{min})$
7. **End if**
8. **End for**

To improve the diversity of the solutions and the local search ability; the Gaussian sparks strategy is introduced to produce sparks. The mapping rule will be carried out to map the spark to a new location within the feasible space. The sparks of the Gaussian explosion are shown in Algorithm 1.

3.2 Selection

After generating a shower of sparks, including explosion sparks and Gaussian sparks. Some of the generated sparks need to be selected and passed down to the next generation. In the selection strategy, we use the measurement of distance. Noted that the best spark is always kept for the next generation. Then, the other individuals are selected as:

$$P(x_i) = \frac{R(x_i)}{\sum_{j\in k} R(x_f)} \quad (6)$$

Where $R(x_i) = \sum_{j\in k} d(x_i, x_j) = \sum_{j\in k}\|x_i - x_j\|$ The distance between i^{th} solution and all of the other solutions.

K represents all of the solution's locations.

j∈ K means the position j belongs to set K.

Algorithm 2: The pseudo-code of the proposed algorithm

1. Randomly select N fireworks at N location
2. **While** terminal criteria = false, **do**
3. **For** all fireworkx_i∈ n do
4. Calculate the number and amplitude of sparks to be generated from the firework x_i; using Eq. (3), (5).
5. **End for**
6. **For** k=1
   // $\hat{\boldsymbol{m}}$ is the number of sparks generated by Gaussian mutation
7. Randomly selects firework that is not in n set;
8. Explode Gaussian spark (Algorithm 1)
9. **End for**
10. Select the best sparks from the next explosion generation;
11. Select the other sparks from both explosions using based on Eq.6.
12. **End While**

4 FIREWORKS ALGORITHM FOR GRAPH COLORING PROBLEM

For the correspondence with the fireworks algorithm, we have chosen as the representation of the explosion space the environment of the graph coloring problems while the fireworks and the sparks correspond to the different solutions from the coloration step.

The affinity between a fireworks display and the sparks generated by the fireworks display measures the degree of compatibility between them. Affinity is generally related to distance. However, for the coloring problem, a cost function is used also evaluation functions appropriate to the problem. This function has great importance because it allows examining if the solution is satisfactory or not.

In general, the cost functions have a combination of one of the following four components: space, execution time, communication. In our context, we consider the following parameters: execution time and occupied space.

4.1 The Implementation of the Fireworks Algorithm

There are several implementations of the fireworks algorithm. Each one is characterized by its adaptability to the problem to be solved. In this paper, we will implement the standard fireworks algorithm.

Several researchers have implemented the standard fireworks algorithm in several programming languages (Matlab, C, C ++, JAVA and python).

The algorithm we have proposed is an adaptation of the standard fireworks algorithm for the graph coloring problem. The application of this algorithm to an optimization problem is as follows: Fireworks and sparks represent the different solutions and the search space is the environment for graph-coloring problem.

We have proposed the fireworks algorithm to solve the problem of the coloring of graphs. The process proceeds as follows: We generate a random set of N fireworks FW constituting the initial population (The number of colors equal to k). Then calculate the cost of each FW using a cost function (objective function). Select the N best FW with respect to their cost function. The explosion of the best FW for a temporary population. Each time a better solution is found (A valid k-coloring is found) the population is updated. The Gaussian operation improves the generated population. Select the best fireworks from the temporary population then replaces the N fireworks by the improved FW. The algorithm stops when the fixed number of iterations that was carried out and the k-coloring is obtained.

The algorithm that we have proposed requires the choice of certain parameters, which are as follows:

4.1.1 Generation of the Initial Population

The fireworks algorithm generates an initial population that represents the set of possible solutions. The population is generated by the number of colors k. A vector of integers represents the initial population whose length is the vertex number.

1	2	3	4	5	6

Figure 1: Generation of the initial population

4.1.2 Objective Function (Fitness)

The Objective function or the adaptation function assigns each spark to a numerical value. This method ensures that the best individuals will be kept while the others are eliminated from the population. In our case, we try to minimize the number of conflicts (objective function) by modifying the color of a vertex.

4.1.3 Selection Operator

Selection operator is an operator that consists in choosing the best sparks and fireworks depends on the objective function to give birth to a new generation to apply the processes of explosion and mutation to them. We have limited our selection to a random choice of individuals that is done randomly, uniformly and without the intervention of the fit value. Each individual therefore has a uniform probability of being selected.

4.1.4 Mutation Operator

The mutation strategy allows reaching most of the points of the feasible domain to create new solutions. We introduce the Gaussian mutation to improve the diversity of a population. It consists of changing an invalid coloring $\alpha \epsilon \{1, \ldots, k\}$ for a vertex $i$by another different color. The mutation operator will evolve the new generation obtained. In the case of binary coding, it consists of inverting a random bit in a chromosome. A value bit at 1 takes the value 0 and vice versa.

1	0	0	1	1	1

1	1	0	1	1	1

Figure 2: Principle of mutation operation for binary coding

4.1.5 Replacement Operator

After the mutation step, a replacement method is used to generate a new population so that the size of that population remains constant. We keep at least the individual with the best performance from the next generation. The new generation is made up of the N/2 better and N/2 bad mutated fireworks. While taking into account the component memory.

After the replacement, some fireworks are eliminated and replaced by new ones generated randomly in order to diversify the population. The lower the affinity of fire, the greater its chance of being replaced.

The approach used is similar to that of classical evolutionary algorithms. In the context of combinatorial optimization problems, we can consider fireworks algorithms as evolutionary algorithms presenting particular operators (explosion, diversification).

Several stopping criteria can be chosen such as the execution time, the minimum fitness value, the number of generations. In our study, the chosen stopping criterion is the number of generations fixed in advance.

5 CONCLUSIONS

Graph coloring is a classic problem in graph theory and has attracted the attention of many researchers because of its multiple practical applications and the complexity of its resolution. It is difficult to say that there is a better algorithm for graph coloring; this is why our choice forced between the simplicity of execution. In this paper, we have proposed a Fireworks algorithm to solve the graph coloring problem. The objective is to have a minimum number of colors required to color a graph. In our approach, we have introduced a new feature, although it is similar to the principles of the algorithm. We found that the fireworks algorithm is influenced by factors such as population size, number of iterations.

REFERENCES

Avanthay, C., Hertz, A., and Zufferey, N., 2003. A variable neighborhood search for graph coloring. *European Journal of Operational Research,* 151(2) :379-388.

Alane, M, L, Renato, C.2018. Exact Algorithms for the Graph Coloring Problem. Revista de Informática Teórica e Aplicada - RITA -vol. 25, Num. 04 p57-73 https://doi.org/10.22456/2175-2745.80721.

Abdulmajeed, N.H. and Ayob, M., 2014. 'A firework algorithm for solving CVRP problem', *International Journal of Advancements in Computing Technology,* January, Vol. 6, No. 1, pp.79–86.

Brelaz, D., 1979. New methods to color vertices of a graph, *Communications of ACM 22 251–256.* https://doi.org/10.1145/359094.359101

Cédric, A. and Alain, H., and Zufferey, N., 2003, A variable neighborhood search for graph coloring, European Journal of Operational Research, Vol 151, Issue 2, pp. 379-388.

Douiri, S., M., Elbernoussi, S.,2015. Solving the graph-coloring problem via hybrid genetic algorithms. *Journal of King Saud University - Engineering Sciences,27(1),114–118.* https://doi.org/10.1016/j.jksues.2013.04.001 .

Garey, M., Johnson D., 1979. Computers and intractability: A guide to the theory of NP-completeness, W.H. Freeman and Company, New York,

Galinier, P., Hao, J.K. 1999. Hybrid evolutionary algorithms for graph coloring. *J. Comb. Optim. 3(4):379-397,*

Hertz, A., Plumettaz, M. ,and Zufferey, N., 2008. Variable space search for graph coloring. *Disc. Appl. Math., 156(13):2551-2560.*

Jari,K., Teemu, N., Olli ,B., Michel G.,2007. An efficient variable neighborhood search heuristic for very large scale vehicle routing problems. *Computers & Operations Research,* Volume 34, Issue 9. pp 2743-2757.

Lü, Z., & Hao, J.-K. , 2010. A memetic algorithm for graph coloring. *European Journal of Operational Research*, 203(1),241–250. https://doi.org/10.1016/j.ejor.2009.07.016

Liu, J., Zheng, S. and Tan. Y., 2013 "The improvement on controlling exploration and exploitation of firework algorithm," *in Advances in Swarm Intelligence*, vol. 7928. Springer Berlin Heidelberg, pp. 11–23.

Malaguti, E., Monaci, M. & Toth, P., 2011. An exact approach for the Vertex Coloring Problem. Discrete Optimization, 8(2), 174-190.

Mendez, D., Zabala. P., 2006. A branch-and-cut algorithm for graph coloring. *Disc. Appl. Math.,* 154(5) :826-847. https://doi.org/10.1016/j.dam.2005.05.022 .

Mohamed. E, Raouan. E, Lamia. B, A. C. Jihane,2020. Fireworks Algorithm for Frequency Assignment Problem. NISS 2020: 4:1-4:6

Munoz, S., Teresaortuno, M., Ramírez, J., &, Yanez. J., 2005. Coloring fuzzy graphs. *Omega,* 33(3), 211–221. doi: 10.1016/j.omega.2004.04.006.

Mostafaie, T., Modarres Khiyabani, F., Navimipour, N. J. 2019. A systematic study on meta-heuristic approaches for solving the graph coloring problem. *Computers & Operations Research.* https://doi.org/10.1016/j.cor.2019.104850

Tan Y., Zhu Y., 2010 Fireworks Algorithm for Optimization. *Advances in Swarm Intelligence. ICSI 2010. vol 6145.* Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-13495-1_44

Wang, F., Z. Xu., 2011. Metaheuristics for robust graph coloring. Journal of Heuristics, 19(4), 529-548. https://doi.org/10.1007/s10732-011-9180-4

Wenwen, Y., Jiechang,W,.2017. Adaptive Fireworks Algorithm Based on Simulated Annealing, *13th International Conference on Computational Intelligence and Security*. pp. 371-375.

Taidi, Z., Benameur, L., Chentoufi, A., J.,2017. A fireworks algorithm for solving travelling salesman problem. I*nternational Journal of Computational Systems Engineering.* Vol. 3; https://doi.org/10.1504/IJCSYSE.2017.086740

Zhang, T., Yue, Q., Zhao, X., & Liu, G. ,2018. An improved firework algorithm for hardware/software partitioning. Appl Intell 49, 950–962. https://doi.org/10.1007/s10489-018-1310-3 .

Zheng, S., Janecek, A. and Y. Tan, "Enhanced Fireworks Algorithm," 2013 *IEEE Congress on Evolutionary Computation*,pp.2069-2077. https://doi.org/10.1109/CEC.2013.6557813

Zheng, S., Janecek, A., Li, J., and Tan, Y., 2014. Dynamic search in fireworks algorithm. *IEEE Congress on Evolutionary Computation (CEC)*, pp. 3222-3229, doi:10.1109/CEC.2014.6900485.