Discrete Artificial Electric Field Optimization Algorithm for Graph Coloring Problem

Yixuan Yu[1,2], Yongquan Zhou[1,2](✉), Qifang Luo[1,2], and Xiuxi Wei[1,2]

[1] College of Electronic Information, Guangxi University for Nationalities, Nanning 530006, China
l.qf@163.com

[2] Guangxi High School Key Laboratory of Complex System and Computational Intelligence, Nanning 530006, China

Abstract. Planar graph coloring is a classic NP - hard problem. So far there is no completely effective method to solve this problem. This paper presents a discrete artificial electric field algorithm to solve the graph coloring problem. The algorithm first codes according to the graph coloring problem to meet the requirements of the graph coloring problem, and then adds part of local search to improve the performance of the algorithm. The algorithm can effectively and accurately solve the coloring problem of plane graphs. Compared with several classical algorithms, the results show that the proposed algorithm has a smaller average number of iterations and a higher success rate in dealing with graph coloring problems, and it can also find the correct coloring scheme in real map problems.

Keywords: Artificial electric field algorithm · Discrete artificial electric field algorithm · Graph coloring problem · Metaheuristic optimization

1 Introduction

The graph coloring problem is a classical NP problem. It is of great significance in network modeling, computer science, sociology and many other disciplines. At the same time, it also has applications in many fields such as electronic information, bioscience and Internet [1]. The graph coloring problem can be described as finding a vertex scheme with the least number of colors, and in the simplest case, making the colors of any two adjacent vertices different. It can be used to solve combinatorial optimization problems, such as minimum dominance set and maximum coverage problems. Edges and vertices in the graph can be used to show a variety of connections and programs in physical, natural, social and PC systems [2]. Such as course schedule problem [3], multiprocessor task scheduling [4], frequency allocation in mobile wireless networks [5] and chemical storage.

Artificial Electric Field Algorithm (AEFA) [6] is an optimization algorithm based on physical Coulomb electrostatic force law proposed by Indian scholars Anita and Anupam Yadav in 2019. In artificial electric field algorithm, each individual in the

D.-S. Huang et al. (Eds.): ICIC 2022, LNAI 13395, pp. 876–890, 2022.
https://doi.org/10.1007/978-3-031-13832-4_70

population is regarded as an electric particle, and the position of each particle represents a candidate solution to the problem. These particles constantly change their position under the action of electric field force, so as to find the optimal position of the individual, which is the optimal solution to the problem. The proposed algorithm has been widely used in many fields such as scheduling, classification, power system, feature selection, and prediction. Recent studies show that artificial electric field algorithm is superior to other metaheuristic algorithms in some aspects. For example, artificial electric field algorithm has been used in high-order matching graph problem [7], quadratic distribution problem [8], adjustment of fractional order PID controller of magnetic levitation system [9], detection of white blood cells in approximate blood [10]. Power system economic load distribution [11] and other fields have been successfully applied, and has excellent performance. Artificial electric field algorithm has gradually developed into one of the mainstream algorithms in the field of intelligent optimization.

This paper presents a discrete artificial electric field algorithm to solve the graph coloring problem. Firstly, a discrete artificial electric field algorithm is proposed to meet the requirements of the graph coloring problem by redefining the position and velocity representation of charged particles and the position updating rules. The local search part is added to speed up the convergence of the algorithm and improve the development ability of the algorithm. In the experimental part, 5 randomly generated maps with different numbers of regions are used to test the effectiveness of the proposed algorithm, and finally, three real maps are selected to verify the performance of the algorithm. According to the experimental results, this version of discrete artificial electric field algorithm is feasible and advantageous in solving the planar graph coloring problem.

2 Discrete Artificial Electric Field Optimization Algorithm

Artificial electric field algorithm has outstanding application in many fields. This paper proposes a discrete artificial electric field algorithm based on GCP problem. Firstly, the code is coded according to the graph coloring problem, then the position and speed update formula of each particle are redefined to meet the requirements of the graph coloring problem, and then the part of local search is added to improve the performance of the algorithm.

2.1 Discrete Position Representation

In AEFA, the positions of charged particles correspond to possible solutions to a problem. When coding GCP problem with n maps, this paper implements a relatively simple and effective coding method. According to the four-color guess problem, which has been proved by numerous scientists, any planar graph can be filled with four colors. The numbers 0, 1, 2 and 3 are used to represent the four different colors. A viable code is the sequence $(X_1, \ldots, X_i, \ldots, X_n)$ of 0, 1, 2, and 3, with integers between $X_i \in [0, 1]$.

Example 1: For a GCP problem of size 6. A hypothetical code Numbers for 1 to 0, 2, 1, 0, 2, 3 said being shaded areas for 0 color, Numbers for 2 being shaded area for no. 2 colors, number 3 being shaded area as the no. 1 color, number 4 area also being shaded for 0 color, Numbers for five areas are also shading for no. 2 colors, The area numbered 6 is colored as color 3.

A possible solution is a permutation of the integers 0, 1, 2 and 3, with the corresponding numbers corresponding to the corresponding region's color label.

2.2 Fitness Function

For a given undirected connected planar graph G, the graph can be simplified into a point set $V(G)$ and an edge set $E(G)$, which is defined as follows:

$$V(G) = \{v_1, v_2, v_3, \ldots, v_n\} \tag{1}$$

$$E(G) = \{e_1, e_2, e_3, \ldots, e_m\} \tag{2}$$

where n is the number of points (that is, the number of regions in the map), and m is the number of edges (that is, the number of adjacent edges between regions in the map). In this way, we can describe the number of points and edges on a picture with an association matrix, which is expressed as:

$$a_{ij} = \begin{cases} 1, \text{ if } v_i v_j \in E(G) \\ 0, \quad other \end{cases} \tag{3}$$

Thus, the fitness function can be expressed as

$$f(R) = \sum_{x=1}^{n} \sum_{y=1}^{n} \text{conflict}_{xy} \tag{4}$$

$$\text{conflict}_{xy} = \begin{cases} a_{xy} \; adjacent\ matrices\ are\ of\ the\ same\ colors \\ 0 \quad adjacent\ matrices\ are\ different\ colors \end{cases} \tag{5}$$

2.3 Charged Particle Velocity Representation

The velocity of a charged particle is expressed as an I by N two-dimensional matrix, and the elements in the matrix represent the probability that the dotted particle will change color. The initialization of velocity V_N^i is the same as the initialization of charged particle position, which is randomly generated by the integers 0, 1, 2, 3.

2.4 Location Update

Location updates are an essential part of global search. The updated speed can be calculated according to the updating formula of the artificial electric field algorithm. The speed of the charged particle is processed by Eq. (6) to change the color probability of the particle with points. In order to better adapt to the coding, the color updating process according to the speed is shown in Algorithm 1.

$$si = 0.5/\left(1 + e^{-v}\right) \tag{6}$$

Algorithm 1: Location update

```
si = 0.5/(1+e^-V )
if      rand() > 0.5 & rand < si
              S(i) = mod(S(i)+0,4);
else if rand() <= 0.5 & rand < si
              S(i) = mod(S(i)+1,4);
else if rand() <= 0.5 & rand >= si
              S(i) = mod(S(i)+2,4);
else if rand() > 0.5 & rand >= si
              S(i) = mod(S(i)+3,4);
endif
```

2.5 Local Search Operator

The local search part is added to the artificial electric field algorithm to enhance the exploration ability of the algorithm. The main purpose of local search is to find a better solution in the neighborhood space. It depends on the number of collisions between a node and its neighbors. The number of conflicting nodes can be obtained from Eq. (7). The number of conflicting nodes is then converted into a conflict factor by the Sigmoid function (Eq. (8)). If the collision coefficient is larger than the randomly generated number between 0 and 1, it will be assigned a higher probability of changing the color number (Eq. (9)). The discrete local search formula is as follows:

$$\mathrm{Cr}_j = \sum_{k=1}^{n} \mathrm{conflict}_{jk} \tag{7}$$

$$Cf_j = \frac{1}{1 + e^{-Cr_j + 2}} \tag{8}$$

$$S(i,j) = \begin{cases} best(1,j) \text{ if } Cf_j > rand() \\ S(i,j) \qquad other \end{cases} \tag{9}$$

2.6 The Exchange Operator

After adding the local search operator, the search ability of the algorithm is improved, but because of its limited search ability, it is difficult to explore the map containing a large number of regions. Therefore, an exchange operator is added in the local search process to enhance the local search capability of the algorithm. There are two kinds of exchange operators, one is single point exchange, the other is subsequence exchange. The operation mode is as follows:

Example 2: The sequence before the exchange was 0, 2, 1, 0, 2, 3. After the exchange of the positions of the second and sixth particles, the sequence becomes 0, 3, 1, 0, 2, 2.

For subsequence swapping, a random subsequence of length N is first selected and then flipped.

Example 3: The sequence before the swap is 0, 2, 1, 0, 2, 3. After the position of the second to sixth particles is flipped, the sequence changes to 0, 3, 2, 0, 1, 2 to obtain a new sequence.

The specific implementation steps of DAFEA to solve the planar graph coloring problem can be summarized as the pseudo-code shown in Algorithm 2.

Algorithm2. Discrete artificial electric field algorithm

Initialize a population $(X)$of n particles with random solutions.

Each charged particles represents one coloring sequence.

Initialize the vector $counter = zeros(1,n)$; it is used to record the generation of unimproved particles.

Set *limit=5;*

Find the best solution ***B*** in the initial population

Define a switch probability $p \in (0,1)$

Define a stopping criterion

while $(fitness(best)! = 0 \,\&\&\, t < MaxGeneration)$

  for i=1:n(all n flowers in the population)

    if $rand > p$

      (Artificial Electric Field Algorithm search process)

      Evaluate objective score values for each particles.

      Calculate total force and acceleration.

      Update velocity Vi of the ith particle.

      if $counter(1,i) == limit$

        %Jump out of local optimum

        Generate a new solution using Handle Maximum Conflict codes operation

        $counter(1,i) = 0;\ best\_local$

      end if

else
(Local search process)
Carry out Discrete Local Learning Operation.
Obtain the solution $tempX_1$;
Carry out Local Swap Operation.
Obtain the solution $tempX_2$
Carry out Local Sub-sequence Reverse Operation.
Obtain the solution $tempX_3$
Find the best solution among $tempX_1, tempX_2, tempX_3$ and assign to $best\_local$;
$x_i = best\_local$
end if
Handle Maximum Conflict Nodes Operation;
Evaluate the new solution x_i ;
if $fitness(x_{i,G+1}) < fitness(x_{i,G})$
update particle x_i ;
$counter(1,i) = 0;$
else
$counter(1,i) = counter(1,i) + 1;$
end if
end for
Find the current best solution $\boldsymbol{B}$
end while
Output the best solution found.

3 Simulation Experiment and Result Analysis

The operating environment of this experiment is as follows: Operating system: Windows 10; Processor: Intel(R) Core (TM) I7-9700 CPU; Frequency: 3.00 GHz; Memory: 16.0 GB (15.9 GB available); Programming tool: Matlab R2019(a).

3.1 Comparison of Each Algorithm Performance

The local discrete artificial electric field algorithm proposed in this paper is compared with the mainstream swarm intelligence algorithms PSO [12], EPSO, DE [13, 14], EODE [15] and LDFPA [16] respectively. EPSO is a particle swarm algorithm with the same operator, and the average number of iterations and success rate are used to compare their optimal performance. Set the maximum iteration times of the algorithm to 10000 generations; Population size: 50; Independent run: 20 times. To test the performance of the algorithm, randomly generate planar graphs of 7, 10, 20, 30, 54 and 100 regions, and if a method can get the correct coloring sequence in 10,000 generations, then it should be considered a successful run. If a method does not get the correct coloring sequence at 10,000 iterations, it should be considered a failed method. The maximum number of iterations, minimum number of iterations, average number of iterations and success rate are listed in the table recording the experimental results. Table 1 shows the numerical results of different algorithms for maps with different region numbers. In the table, the decimal part of the number is dropped for simplicity.

Table 1. Comparison of the experimental results for 5 regions

10	DAEFA	1	1	1	100%
	PSO	66	6	32	100%
	EPSO	47	5	26	100%
	DE	235	5	64	100%
	EODE	22	2	7	100%
	LGFPA	2	1	1	100%
20	DAEFA	12	2	4	100%
	PSO	5896	116	2418	40%
	EPSO	13	5	9	100%
	DE	2212	914	1554	50%
	EODE	1782	410	688	95%
	LGFPA	7	2	4	100%
30	DAEFA	83	4	24	100%
	PSO	10000	3426	2418	40%
	EPSO	4680	91	1569	100%
	DE	2212	914	1554	50%
	EODE	1782	410	688	95%
	LGFPA	32	8	18	100%
54	DAEFA	621	45	232	100%
	PSO	10000	10000	10000	0%
	EPSO	2743	8087	6226	20%
	DE	10000	10000	10000	0%
	EODE	10000	10000	10000	0%
	LGFPA	270	23	95	100%
100	DAEFA	5677	65	1243	90%

(*continued*)

Table 1. (*continued*)

	PSO	10000	10000	10000	0%
	EPSO	10000	10000	10000	0%
	DE	10000	10000	10000	0%
	EODE	10000	10000	10000	0%
	LGFPA	5794	81	1328	80%

Can be seen from Table 1, compared with other algorithms, DAEFA in to solve the problems of the regional plan of the small and medium-sized coloring success rate is 100%, also can prove DAEFA in addressing the problem of small and medium-sized effect is very good, and the other several algorithms in solving the problem of the area of a small number of floor plan can also meet the conditions of coloring, However, the number of iterations and accuracy are not as good as DAEFA. As the floor plan of the area number increasing, the dimensions of the problem are increasing, the algorithm put forward higher request, this time even decrease of several other contrast coloring the success rate of the algorithm under the specified number of iterations has been cannot provide meet the conditions of coloring solution, but DAEFA can also give the right color scheme. It can be found from the table that DAEFA can also find the correct solution algorithm when solving large-scale problems. Only LGFPA and DAEFA can find the correct solution in the 100block area. But DAEFA's success rate of 90% is higher than LGFPA's.

Figures 1, 2, 3 and 4 shows the convergence curves of DE, EODE, EPSO, LDFPA and DAEFA algorithms for maps with different number of regions (20 regions, 30 regions, 54 regions and 100 regions). These curves and the statistics listed in Table 1 indicate that DAEFA has a high rate of convergence. And its high level of search power and stability is easy to find.

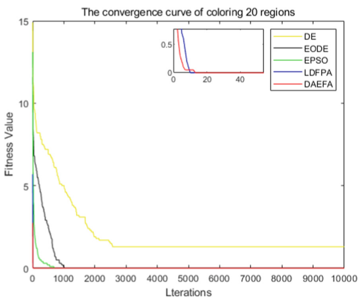

Fig. 1. The convergence curve of coloring 20 regions

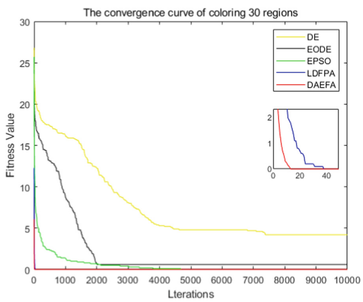

Fig. 2. The convergence curve of coloring 30 regions

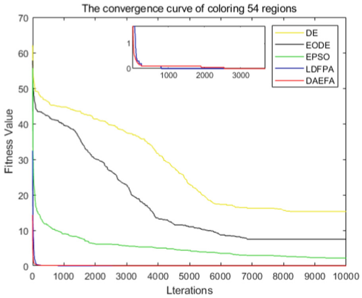

Fig. 3. The convergence curve of coloring 54 regions

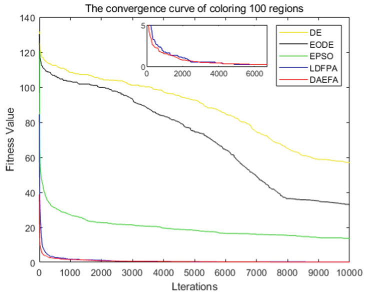

Fig. 4. The convergence curve of coloring 100 regions

3.2 Performance Verification of Real Map Coloring Problem

In Sect. 3.1, a large number of numerical experiments were carried out, but the results were not intuitive enough. Therefore, in this section, three real maps (China map, African map and counties map of Sichuan Province) are selected for coloring to verify the performance of DAEFA in a more intuitive way.

3.2.1 Chinese Map

Figure 5 is a map of China, including 34 provinces, municipalities and autonomous regions, numbered. Figure 6 is a solution of DAEFA to solve the coloring problem in China map. Figure 7 is the convergence curve of successfully finding the correct coloring sequence of Chinese map.

Fig. 5. Number the administrative regions for Chinese map

Fig. 6. The Chinese map after coloring

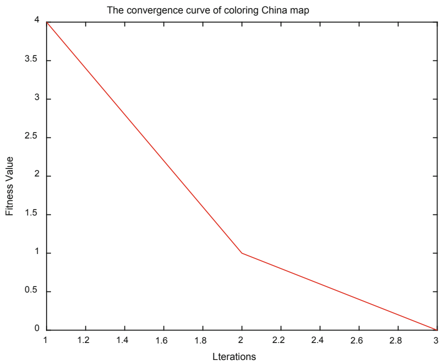

Fig. 7. The convergence curve of coloring Chinese map

The best individual coloring coding is: 0, 2, 1, 3, 3, 3, 1, 3, 0, 1, 2, 0, 1, 2, 0, 3, 0, 3, 1, 2, 0, 2, 0, 3, 2, 1, 3, 2, 0, 1, 2, 1, 0, 1. In Fig. 6, these 34 regions are colored by 4 colors without conflict as well.

3.2.2 African Map

We used DAEFA to find the correct coloring scheme for the African map. There were 54 countries in Africa and they were numbered. Figure 8 is the numbered African map.

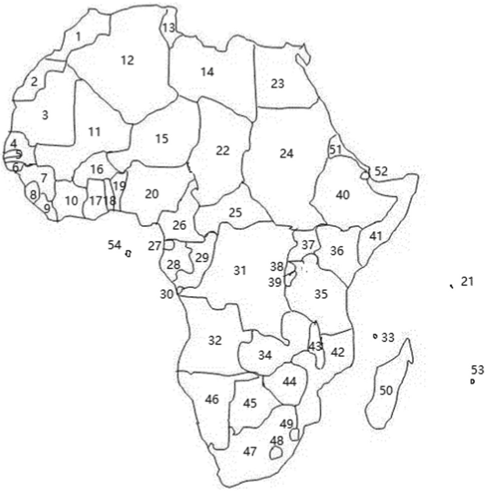

Fig. 8. Number the administrative regions for counties in African map

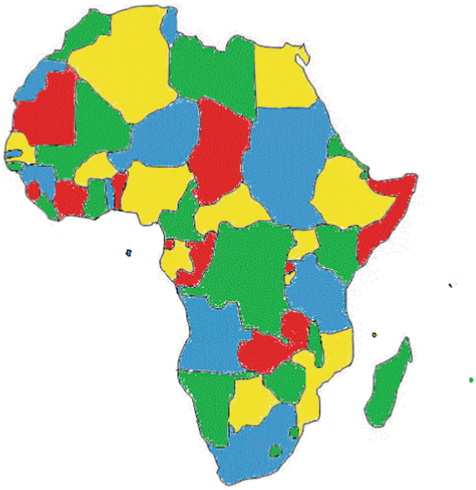

Fig. 9. The counties in African map after coloring

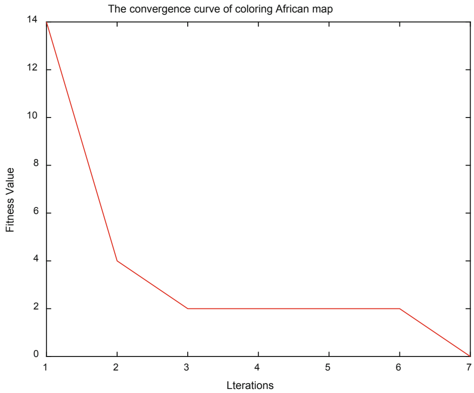

Fig. 10. The convergence curve of coloring counties in African map

The results show that DAEFA can provide a coloring scheme for this problem in a feasible time. Figure 9 is the map of Africa after coloring, and it can be found that there is no coloring conflict in this scheme. Figure 10 shows the convergence curve of successfully finding the correct coloring sequence of the African map.

The best individual coloring coding is: 0, 3, 1, 2, 3, 0, 3, 1, 0, 1, 0, 2, 3, 0, 3, 2, 0, 3, 1, 2, 2, 1, 2, 3, 2, 0, 1, 2, 1, 3, 0, 3, 2, 1, 3, 0, 2, 1, 2, 2, 1, 2, 0, 0, 2, 0, 3, 0, 0, 0, 0, 3, 0, 3. From Fig. 9, we can see that each area is painted with four colors. Every two areas that have a common boundary are filled with a different color.

3.2.3 Sichuan Province Map

We used DAEFA to find out the correct coloring scheme for the map of each county in Sichuan Province. There are 183 counties in Sichuan Province and they were numbered. Figure 11 is the numbered map of each county in Sichuan Province. Figure 12 shows DAEFA's coloring scheme to solve coloring problems in all counties of Sichuan Province. Figure 13 shows the convergence curve of successfully finding the correct coloring sequence of the map of each county in Sichuan Province.

Fig. 11. Number the administrative regions for counties in Sichuan province map

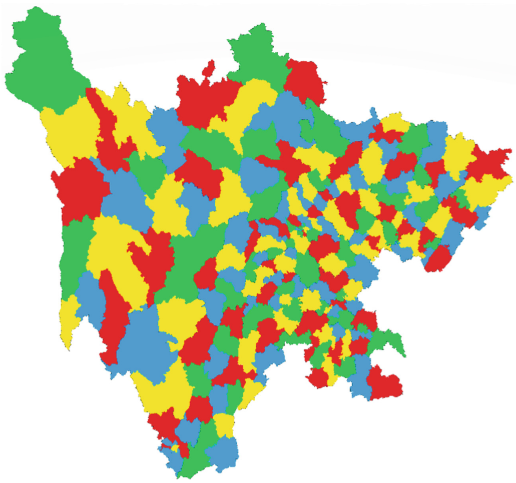

Fig. 12. The counties in Sichuan province map after coloring

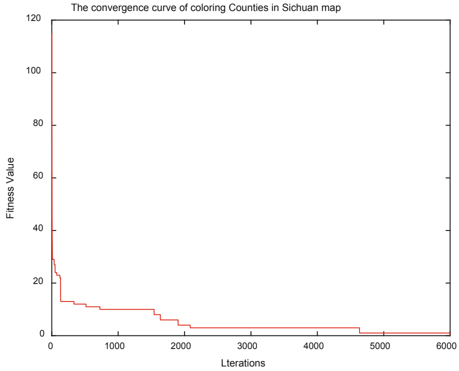

Fig. 13. The convergence curve of coloring counties in Sichuan province map

For the map of each county in Sichuan Province, DAEFA can find a correct coloring sequence, and it can be seen from Fig. 12 that there is no area of color conflict in the colored map. But because the number of areas is more, so the number of costs also increases, the success rate also decreases.

4 Conclusions

Artificial electric field algorithm is a novel heuristic optimization algorithm, which can have excellent results in many fields. In order to solve the planar graph coloring problem, a DAEFA is proposed, in which the velocity is redefined as the probability of color change at this position, and local search operator and exchange operator are added to improve the development ability of the algorithm. Experiments show that the algorithm can effectively solve the planar graph coloring problem. Compared with other algorithms, it can be found that DAEFA has high speed, high precision and high accuracy in the optimization of coloring problems. The validity of DAEFA is verified. On the other hand, three real maps are selected for coloring experiments, which proves that DAEFA can find the correct coloring sequence, but with the increase of the number of regions, the success rate of the algorithm will gradually decrease. The large-scale graph coloring problem has higher requirements for algorithms, which need to be studied and experimented continuously.

In solving the graph coloring problem, DAEFA provides a new optimization method. Simulation experiments show that DAEFA has certain advantages over other swarm intelligence optimization algorithms. It is believed that with the continuous development and improvement of artificial electric field algorithm, artificial electric field algorithm will play a huge advantage in the field of artificial intelligence, such as combinatorial optimization, computational intelligence, data mining and so on.

Acknowledgment. This work is supported by National Science Foundation of China under Grants No. U21A20464, 62066005.

References

1. Barman, S.C., Pal, M., Mondal, S.: An optimal algorithm to find minimum k-hop dominating set of interval graphs. Discrete Math. Algorithms Appl. **11**(3), 1950016 (2018)
2. Bhosale, B., Ahmed, B., Biswas, A.: On wavelet based modeling of neural networks using graph theoretic approach. Life Sci. J. **10**(2), 1509–1515 (2013)
3. Burke, E.K., et al.: A graph-based hyper-heuristic for educational timetabling problems. Eur. J. Oper. Res. **176**, 177–192 (2007)
4. Giaro, K., Kubale, M., Obszarski, P.: A graph coloring approach to scheduling of multiprocessor tasks on dedicated machines with availability constraints. Discrete Appl. Math. **157**(17), 3625–3630 (2009)
5. Hao, J.K., Dorne, R., Galinier, P.: Tabu search for frequency assignment in mobile radio networks. J. Heuristics **4**(1), 47–62 (1998)
6. Anita, Yadav, A.: AEFA: artificial electric field algorithm for global optimization. Swarm Evol. Comput. **48**, 93–108 (2019)
7. Anita, Yadav, A.: Discrete artificial electric field algorithm for high-order graph matching. Appl. Soft Comput. **92**, 106260 (2020)
8. Sajwan, A., Yadav, A., Kumar, N., et al.: Development of discrete artificial electric field algorithm for quadratic assignment problems. In: Nigdeli, S.M., Kim, J.H., Bekdaş, G., Yadav, A. (eds.) ICHSA 2020. AISC, vol. 1275, pp. 411–421. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-8603-3_36
9. Demirren, A., et al.: Opposition-based artificial electric field algorithm and its application to FOPID controller design for unstable magnetic ball suspension system. Eng. Sci. Technol. Int. J. **24**(2), 469–479 (2020)
10. Sajwan, A., Yadav, A.: An intelligent model for the detection of white blood cells using artificial intelligence. Comput. Methods Programs Biomed. **199**(10), 105893 (2021)
11. Anita, Yadav, A., Kumar, N.: Application of artificial electric field algorithm for economic load dispatch problem. In: Abraham, A., Jabbar, M., Tiwari, S., Jesus, I. (eds.) SoCPaR 2019. AISC, vol. 1182, pp. 71–79. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-49345-5_8
12. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: ICNN 1995-International Conference on Neural Networks. IEEE (1995)
13. Qin, A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans. Evol. Comput. **13**(2), 398–417 (2009)
14. Valdez, F., Melin, P., Castillo, O.: An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms. Appl. Soft Comput. **11**, 2625–2632 (2011)
15. Jiao, W., Liu, G., Liu, D.: Elite particle swarm optimization with mutation. IEEE (2011)
16. Yang, X.-S.: Flower pollination algorithm for global optimization. In: Durand-Lose, J., Jonoska, N. (eds.) UCNC 2012. LNCS, vol. 7445, pp. 240–249. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32894-7_27