

19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

Acceleration Based Particle Swarm Optimization for Graph Coloring Problem

Jitendra Agrawal, Shikha Agrawal

Department of Computer Science and Engineering, Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal, India

Abstract

The graph coloring problem is one of the combinatorial optimization problems. Although many heuristics and metaheuristics algorithm were developed to solve graph coloring problem but they have some limitations in one way or another. In case of tabu search, the algorithm becomes slow, if the tabu list is big. This is because lots of memory to keep the list and also a lot of time to travel through the list, is needed in each step of the algorithm. Simulated annealing has a big handicap when applied to graph coloring problem because there are lots of neighboring states that have the same energy value. The problem with ant colony optimization is that the number of ants that must be checked is n times bigger than other algorithms. Therefore, there will be a need of a large amount of memory and the computational time of this algorithm can be very large. A swarm intelligence based technique called as particle swarm optimization is therefore employed to solve the graph coloring problem. Particle swarm optimization is simple and powerful technique but its main drawback is its ability of being trapped in the local optimum. Therefore, to overcome this, an efficient Acceleration based Particle Swarm Optimization (APSO) is introduced in this paper. Empirical study of the proposed APSO algorithm is performed on the second DIMACS challenge benchmarks. The APSO results are compared with the standard PSO algorithm and experimental results validates the superiority of the proposed APSO.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).
Peer-review under responsibility of KES International

Keywords: Acceleration based Particle Swarm Optimization ; Combinatorial optimization problem; Graph coloring problem; Particle swarm optimization.

1. Introduction

Optimization is a challenging field that concerns with the finding of minima or maxima of functions. Today, optimization covers a wide variety of techniques from Operations Research, artificial intelligence and computer science, and is used in both scientific and industrial worlds. Optimization problems can be of different types such as multi objective optimization, multimodal optimization and combinatorial optimization [1]. Many optimization problems of both theoretical and practical importance relates them with the selection of best configuration of a set of finite object to achieve certain goals. Such problems are basically divided into two categories one is those where solutions are programmed with real valued variable and the other is those where solution is programmed with discrete variable. Combinatorial Optimization Problems are the one that belongs to the second category. A Combinatorial Optimization Problems (COP) [2] can be stated as among a finite set of possible solution we look for the best one minimum or maximum.

- Jitendra Agrawal. Tel.: +91-9425432199;.
E-mail address: jitendra@rgtu.net

The graph coloring problem is one of the combinatorial optimization problems and is NP hard [3]. It can be defined as for a given graph, find the least number of colors needed for coloring of graph such that no two adjacent vertices bear the same color. The least number of colors for coloring a graph G is called its chromatic number denoted by $\chi(G)$. Application of graph coloring includes map coloring, scheduling, radio frequency assignment, register allocation, pattern matching, Sudoku, timetabling and many more.

The paper is organized as follows, Section 2 reviews about the research works related to the graph coloring problem. In section 3, a brief discussion about the standard PSO is presented and section 4 describes the proposed APSO algorithm. The experimental results on the benchmark functions are given in Section 5 and conclusion of the paper is given in Section 6.

2. Literature Survey

Various approaches used by many researchers to solve the graph coloring problem are described as follows:

2.1. Heuristics Methods

Approaches based on heuristic methods were developed due to the high computational complexity of the graph coloring problem. Heuristic methods determine the suboptimal solutions in polynomial time because of their characteristics of accuracy and level of complexity of the graph. They can be of the certain types based on the idea of choosing a vertex first and then assigning an appropriate color. Welsh and Powell [4] proposed Largest-Fit (LF) method in which vertices of a high degree are colored in first priority. Despite of its simplicity this method is very effective. Matula et al. [5] proposed a heuristic method alike the LF method named as Smallest Last (SL) method. It is also based on the idea of color vertices with high degree first but it does not have certain faults of LF algorithm such as SL optimally color trees, cycles, unicyclic graph, wheels, complete bipartite graph. Although both the methods are easy to implement and fast by nature but they are quite inefficient in optimal coloring. To improve the efficiency of these sequential algorithms, interchanges are executed. By performing interchanges, a previously colored vertex is switched to another class, by allowing the current vertex to be colored without adding new color [5]. By performing interchanges, performance was improved but it is more time consuming. Another heuristic method proposed by Johnson [6] known as greedy independent sets (GIS) method which is an implementation of the maximum independent set algorithm. In this method, the vertices of graph are analyzed in a certain order. The vertex is assigned a color if the vertex is not adjacent to any vertex with the same color. Another technique introduced by Bralaz [7] was based on the idea of reordering the nodes at each stage known as Degree of Saturation (DSATUR) or Saturation LF (SLF). In this, saturation degree is the term by which next vertex to color is chosen. A vertex with maximum saturation degree is given the priority to be placed in the first legal color class. All the methods discussed above are based on the idea of choosing a vertex first and then assigning an appropriate color. However there is a more successful method proposed by Leighton [8] known as Recursive Largest Fit (RFL) which is based on the idea where each color is completed before introducing a new one. In this method, vertices of one class got selected at a time. Randomization is also carried to improve the performance of simple heuristics. This concept is reflected in the work proposed by Johnson [9] who introduced the XRLF method. In this method, for each color many candidate classes are created and one is selected with the least degree in the remaining graph.

2.2. Metaheuristic

Metaheuristic is another way to solve graph coloring problem. Metaheuristic can be defined as high level strategies for exploring search space by using different methods. In the context of graph coloring we can have two types of metaheuristic one is local search method and another is population based method. The local search method includes simulated annealing, tabu search and population based method includes genetic algorithm, ant colony optimization, particle swarm optimization.

2.2.1. Local Search Methods

Local search is a metaheuristic method for solving computationally hard optimization problems. Local search methods start with a complete assignment of a value to each variable and try to iteratively improve this assignment by improving steps, by taking random steps, or by restarting with another complete assignment. Local search methods can be of different types such as simulated annealing [29], tabu search [17], variable neighborhood search [34], variable search space [21], iterated local search [7] and large scale neighborhood search [40].

2.2.1.1 Simulated Annealing

In 1987, Simulated annealing (SA) was first applied to graph coloring problem by Chams, Hertz and Werra [31]. In this, initially a neighboring solution is selected. If the neighboring solution is better than the current solution it will be accepted as the starting solution. If it is not better it will be accepted with a certain probability that gradually decreases with a global parameter

called Temperature. Initially the temperature is high and it accepts all the solution but gradually temperature decreases which results in accepting only best solution. This process allows the algorithm to avoid local maximum. In 1991, Johnson et al. [10] intensively tested simulated annealing on random graphs. In order to evaluate and compare several graphs coloring heuristics, three simulated annealing heuristic based on three different strategies such as K-fixed penalty strategy, the proper strategy and the penalty strategy. Experiment shows that none of these strategies dominated clearly the others. In 2007, Szymon et al. [38] proposed parallel annealing (PSA). In this, the synchronous master slave model with periodic solution update is being used. Final results are compared with the Parallel genetic algorithm (PGA). Experiment shows that PSA achieved a similar performance level as the PGA. In 2009, Huberto et al. [23] proposed a way to transform the graph coloring problem into satisfiability (SAT) problem. Then a new approach that uses the threshold accepting algorithm (a variant of SA) and the Davis and Putnam algorithm was proposed. This is done because SA is not a complete algorithm and it's not always gets the optimal solutions. The resulting algorithm is a complete and obtains better results than the well known SA algorithm.

2.2.1.2 Tabu Search

In 1987, Hertz and Werra [1] were the first who applied tabu search (TS) known as Tabucol for finding a solution to the graph coloring problem using k-fixed penalty strategy. In this, instead of making a whole configuration tabu, the only single move is made. The algorithm is compared with simulated annealing and experiment shows that tabu search outperforms simulated annealing on different instances of random dense graph performance. In 1998, tabu search's peak performance is obtained by Dorne and Hao [14]. In their paper a generic tabu search is introduced for three coloring problems such as graph coloring, T coloring and set T coloring. In this the main concept is to integrate features such as greedy initialization, solution regeneration, dynamic tabu tenure, incremental evaluation of solution and constraint handling. Author compared the proposed algorithm with the well known DSATUR and found that the proposed algorithm is more efficient and robust. In 2008, Zufferey et al. [24] proposed a variant of Tabucol known as Partialcol which has a reactive component for adjusting the length of the time. Certain moves are excluded based on the fluctuations of the objective function and applied it with k coloring approach. The results are competitive when compared with other algorithms. In 2009, Porumbel et al. [11] proposed a new local search algorithm known as position guided tabu search (PGTS) heuristic which besides avoiding local optima also avoids revisiting candidate solution in previously visited regions. A learning process, based on a metric of the search space guides the tabu search towards yet uncensored regions. When compared with other better known local search algorithm PGTS proved to be effective.

2.2.1.3 Other Methods

Other local search methods include variable neighborhood search and variable search space. In 2003, Avanthay et al. [5] proposed a variable neighborhood search (VNS) algorithm which uses Tabucol as a local search operator and applies different types of perturbation operators. For analyzing the performance of the proposed VNS algorithm they compared the result obtained from Tabucol and the genetic hybrid algorithm (GH) proposed by Galinier and Hao which combine a tabu search with a genetic algorithm. They reported that VNS algorithm produces better results than Tabucol but VNS algorithm is not competitive with GH. In 2008, Hertz et al. [21] proposed variable search space (VSS-COL) as a new local search coloring heuristic. The VSS-Col heuristic alternates the use of three different local search heuristics that adopt different strategies: TABUCOL, PARTIALCOL, and a third tabu algorithm [20], which is not competitive by itself but complements adequately the two others. Experiment shows that VSS obtained better results when compared with local search heuristic.

2.2.2 Population Based Method

Population based methods are a kind of metaheuristic which deals with the set of populations. Population-based algorithms provide a natural, intrinsic way for the exploration of the search space. Population based methods are of different types such as genetic algorithm [25], ant colony optimization [13], particle swarm optimization [27].

2.2.2.1 Genetic Algorithm

Evolutionary algorithms have been adapted in the context of the graph coloring problem. The first algorithm based on genetic and evolutionary principles was developed in 1991 by Davis [30]. In this paper, genetic algorithm (GA) is used to encode solution as permutations of the vertices known as order based encoding. Then a greedy algorithm is applied to evaluate a solution. Experiment shows that this algorithm is competitive with greedy algorithm but when compared to other algorithm its performance is not competitive. In 1995, Costa et al. [12] were the first who published results of an experiment with genetic local search (GLS). Author tested this evolutionary descent method on random graph and found that it perform better than Tabucol. In 1996, Fleurent and Ferland [6] also experimented with GLS but they used tabu search as a local search operator, instead of a descent method used by Costa et al [15]. Experiment shows that GLS outperforms Tabucol. In 2010, a new evolutionary algorithm known as memetic algorithm (MACOL) is proposed by Zhipeng et al. [42] which uses an adaptive multiparent cross

operator (AMPaX) and a distance and quality base replacement criterion for pool updation. Experiment shows that MACOL obtains very competitive results on many of the benchmark graphs.

2.2.2.2 Ant Colony Optimization

In 1997, Costa and Hertz [9] were the first to apply an ant colony optimization (ACO) to the graph coloring problem. In their work they introduced ANTCOL which embed two graphs coloring constructive heuristics RLF and DSATUR called as ANTRLF and ANTDSATUR respectively. The experiment showed that the result obtained of ANTCOL based on RLF outperforms those based on DSATUR. In 2008, Salari and Eshghi [15] proposed a modification of ANTCOL, a Max-Min ant system algorithm for graph coloring (MMGC) to improve the performance of ANTCOL. Experiment shows that the result achieved by MMGC is quite better than ANTCOL. In 2010, Plumettaz et al. [32] proposed ant local search coloring method (ALS-COL). In most ACO based algorithms, the role of each ant is to create a solution in a constructive way while in this paper the authors proposed the concept of a local search. In the proposed scheme, each ant performs local search and at each step updates the current solution by the use of greedy force. Experiment results show that ALS-COL outperforms PARTIALCOL and other ant colony optimization heuristics for graph coloring problem.

2.2.2.3 Particle Swarm Optimization

In 2008, Cui et al. [8] applied particle swarm optimization (PSO) to solve the graph coloring problem. In this a modified particle swarm optimization was added to the disturbance factor to improve the performance of the algorithm. The experiment showed that the performance of the modified PSO is better than that of classical PSO. In 2010, Mostafa et al. [16] proposed hybrid algorithm which uses a recombination operator. Author proposed a modified PSO with fuzzy logic to obtain a high performance algorithm for solving planer graph coloring problem. Experiment shows better result and less complexity of time and storage. In 2011, Hsu et al. [22] added a modified turbulence to previous PSO, to solve the graph coloring problem. The proposed model consists of walking one strategy, assignment strategy and turbulent strategy. It solves the planer graph coloring problem using four colors more effectively and accurately.

2.3. Hybrid Methods

In 1999, Hao and Galinier [35] proposed a hybrid coloring algorithm (HCA) that uses tabu search and genetic algorithm. HCA uses the greedy partition crossover (GPX) operator which combines color classes instead of specific color assignments. Hybrid approach proves to be very powerful. In 2004, Lim and Wang [2] applied various metaheuristic to solve robust graph coloring problem (RGCP). Metaheuristic algorithms used were genetic algorithm, simulated annealing and tabu search. Experimental results on various sizes of input graph provide the performance of these meta-heuristics in terms of accuracy and run time. In 2005, Sivanandam et al. [37] proposed a new permutation based representation of the graph coloring problem. In this a migration model of parallelism for genetic algorithm (PGA) is used with Message Passing Interface (MPI). In addition, three crossover operators are used to namely : Greedy partition crossover (GPX), Uniform independent set crossover (UISX) and Permutation based crossover (PX) are used .The experiment showed that GPX works well in context of convergence and PX in context of execution time. In 2010 Ray et al. [3] proposed a combination of evolutionary algorithm known as genetic algorithm with multi point Guided Mutation for the graph coloring problem. In this, a new operator called double point guided mutation operator is used to increase the performance level of the simple genetic algorithm dramatically. In 2011, David [36] proposed two new metaheuristic algorithms for graph coloring algorithm. One is population based multiagent evolutionary algorithm (MEA) using a multiagent system where an agent represents a tabu search procedure. The second is a pseudo reactive tabu search (PRTS) where a new online learning strategy is introduced. Both algorithms empirically outperform basic tabu search algorithm Tabucol on the well established DIMACS instances. In 2011, Qin et al. [26] proposed a hybrid discrete particle swarm algorithm (HPSO) to solve the graph coloring problem. In this, initially a general discrete PSO algorithm is proposed. Then a hybrid discrete PSO algorithm is proposed by combining a local search known as Tabucol. Experiment with a set of eight DIMACS benchmarks was conducted and the computational results show that HPSO is feasible and competitive with other well-known algorithms. In 2011, Titiloye et al. [39] proposed an effective quantum annealing algorithm to solve the k-coloring. The work has been inspired from the quantum mechanics. Compared with simulated annealing, it includes an additional parameter (with the classical temperature parameter). While simulated annealing evolves in a neighborhood of constant radius, this additional parameter is used here to modify the radius of the neighborhood (to control diversity) and to reinforce the evaluation function with a kinetic energy relying on interactions between replicas (roughly speaking, it quantifies the similarity of replicas). This kinetic energy aims to help escaping local optima. Experiments show remarkable results on some tested DIMACS graphs.

3. Particle Swarm Optimization

Particle swarm optimization [18] is a swarm intelligence approach which is inspired by the intelligence, experience-sharing, social behavior of bird flocking and fish schooling. In 1995 Kennedy and Eberhart developed PSO to solve continuous-valued

space but later on it was modified for binary/discrete optimization problems. PSO is a searching technique in which a collection of particles find the global minimum that move through the search space. In PSO, a group of a particle's position and particle velocity is initialized randomly. Each particle position represents a possible solution and particle velocity represent is the rate of changes of the next position with respect to the current position. In PSO, there are local and global extremes. The fitness value is computed. If this fitness value has better-quality than the best fitness value then the current fitness value is set as new local best. Now the particle with the best fitness value of all particles is chosen as the global best. Finally for each particle velocity and particle position is calculated and updated using equations (1) and (2) respectively. It has been used across a wide range of applications, such as image and video analysis, design and restructuring of electricity networks, control, antenna design, electronics and electromagnetic. The advantages of particle swarm optimization are there is no centralized controller in the system. Hence, failure of any particle does not affect the search process. PSO is a simple, easy and only few parameters needed to be adjusted.

$$\begin{aligned} \text{Vid}(t+1) &= w\text{Vid}(t) + c_1 R_1(\text{Pid}(t) - \text{Xid}(t)) + c_2 R_2 (\text{Pgd}(t) - \text{Xid}(t)) & (1) \\ \text{Xid}(t+1) &= \text{Xid}(t) + \text{Vid}(t+1) & (2) \end{aligned}$$

where,

Vid - Velocity of the *i*th particle of dimension *d*

Pid - Best previous position of the *i*th Particle of dimension *d*

Pgd - Best position of the neighbours of dimension *d*

Xid - Current position of the *i*th particle of dimension *d*

R1, R2 - Random function in the range [0, 1]

w - Inertia weight that forces the particle to move in the same direction of the previous iteration

*c*₁ & *c*₂ - Acceleration constants. The first coefficient “*c*₁” controls the impact of the cognitive component on the particle trajectory and the second coefficient, “*c*₂” controls the impact of the social component. While *c*₁ component is responsible to maintain the diversity, *c*₂ component is responsible to ensure convergence.

If *c*₁ << *c*₂, then particles may result into premature convergence as particles are attracted more towards the global best position then to their personal best positions. If *c*₂ << *c*₁, then particles may result in slow convergence or may not converge at all because each particle is more attracted to its personal best positions then to global best. Since the PSO relies on a combination of both personal and social knowledge of the given search space, so the coefficients *c*₁ and *c*₂ are chosen such that *c*₁ ≈ *c*₂. For too large values of *c*₁ and *c*₂, particle velocities accelerate too fast, leading in swarm divergence. On the other hand for too small values of *c*₁ and *c*₂, swarm convergence time increases as particles move too slowly. In order to find the optimum solution efficiently proper control on global exploration and local exploration is crucial. Generally high diversity is necessary during the early part of the search to allow the use of the full range of the search space on the other hand, during the latter part of the search, when the algorithm is converging to the optimal solution, fine tuning of the solution is important to find the global optima efficiently. Therefore a proper control of these two components is very important to find the optimum solution accurately and efficiently. Particles draw their strength from their cooperative nature, and are most effective when *c*₁ and *c*₂ are adaptive to the particle value to facilitate exploitation and exploration of the search area. Hence to enhance the performance of PSO, Acceleration based PSO (APSO) is introduced. In the next section, APSO is explained.

4. Acceleration Based Particle Swarm Optimization (APSO)

In this research work, an Acceleration based Particle Swarm Optimization (APSO) algorithm is developed to solve premature convergence problem of the standard PSO. In APSO, the acceleration coefficients are selected based on the fitness value which will increase the accuracy of the results. The selection of acceleration coefficient values in APSO algorithm is described as follows:

$$nc_1 = c_1 \times (1 - \lambda) \tag{3}$$

$$nc_2 = c_2 \times (1 + \lambda) \tag{4}$$

In Equ. (3) and (4), the λ value is computed based on the fitness values is calculated as,

$$\lambda = \frac{\chi (1 + \varphi (f_{\max} - f_{\min})^\omega - (f_{\text{avg}})^\omega)}{\delta (f_{\max} - f_{\min})^\omega - f_{\text{avg}}^\omega} \tag{5}$$

$$\delta = \left(\frac{f_{\max} - f_{\min}}{f_{\text{avg}}} \right)^\omega \tag{6}$$

Where,

χ - Alteration probability

ω, φ - Coefficient factors

$F_{max}, F_{min}, F_{avg}$ - Maximum, minimum and average fitness of the particles

By exploiting Equ. (3) and (4), the acceleration coefficients values, the velocity formula which is given in Equ. (1) is updated by equation 7. However the position update equation remains the same

$$Vid(t+1) = wVid(t) + nc1 R1(Pid(t) - Xid(t)) + nc2 R2 (Pgd(t) - Xid(t)) \quad (7)$$

5. Experiment and Result

The values of parameters in the APSO algorithm for the graph coloring problem are presented in Table 1. These values are selected based on some preliminary trials. To validate the performance of the proposed APSO algorithm, an extensive experiments on some benchmarks are conducted. Benchmark graphs are derived from the well known DIMACS challenge benchmarks [22] illustrated in Table 2. These instances cover a variety of types and sizes of graphs.

For each instance in table 2, 10 independent runs of the algorithm are carried out. Table 3 shows the experimental results. Column 2 records the graph name, column 3 records the expected chromatic number. Column 4 is reported for standard PSO which is divided in 2 sub columns in which first column represent the total steps and other column represent PSO result. Column 5 is recorded for APSO which is divided into two parts in which one part represent the total steps and second part represent the APSO result. We run APSO independently on every graph and found that in case of Mycielski graphs such as myciel3.col, myciel4.col, myciel5.col, APSO found the optimal solution in each run. Then Stanford GraphBase (SGB) such as huck.col, jean.col, david.col. games120.col, anna.col is tested independently and APSO found an optimal solution in each run. We then test the miles graphs and found that for miles250.col APSO is able to produce an optimal solution in each run while miles1000.col produces an optimal solution in 9 runs. At last we tested our algorithm on queen graph i.e on queen 5_5.col we were unable to get an optimal solution with the predefined parameter. Hence we tuned the parameter value of population size to 50 and maximum step size to 500 then we got the optimal solution 7 runs using APSO.

We compared standard PSO with the proposed APSO algorithm. Experimental results show that the APSO algorithm for graph coloring problem is feasible and robust. The algorithm was able to scale across the different graphs and produce optimum solutions in each case. APSO is capable of finding an optimal solution for all the graphs with a very high probability. APSO finds an optimal solution in each run for all graphs except miles1000.col and queen 5_5.col in 9 and 7 runs respectively while standard PSO was unable to find an optimal solution in 100 steps. The standard PSO algorithm failed to produce the optimum solution for all graphs except the two graphs named myciel3.col and myciel4.col. From the simulation result, it has been observe that APSO is competitive with the standard PSO algorithm.

Table 1: Parameter setting

Parameter	Population size	Maximum step size	Inertia weight	Maximum velocity
Value	100	100	0.9	10

Table 2 : Benchmark Graphs

S No	Graph Name	Vertices	Edges
1	myciel3.col	11	20
2	myciel4.col	23	71
3	myciel5.col	47	236
4	huck.col	74	301
5	jean.col	80	254
6	david.col	87	406
7	games120.col	120	638
8	miles250.col	128	387
9	miles1000.col	128	3216
10	anna.col	138	493
11	queen5_5.col	25	160

Table 3: Experimental Result

S N	Graph Name	Expected $\chi(G)$	Standard PSO		APSO	
			Total steps	PSO result	Total steps	APSO result

1	myciel3.col	4	14	4	1	4
2	myciel4.col	5	59	5	1	5
3	myciel5.col	6	100	Unable to optimize	1	6
4	huck.col	11	100	Unable to optimize	1	11
5	jean.col	10	100	Unable to optimize	1	10
6	david.col	11	100	Unable to optimize	1	11
7	games120.col	9	100	Unable to optimize	1	9
8	miles250.col	8	100	Unable to optimize	1	8
9	miles1000.col	42	100	Unable to optimize	2	42
10	anna.col	11	100	Unable to optimize	1	11
11	queen5_5.col	5	100	Unable to optimize	32	5

6. Conclusion

In this paper, we attempted an adaptation of PSO to solve the graph coloring problem. We proposed an Acceleration based Particle Swarm Optimization method (APSO) to solve the graph coloring problem. With a view to achieve a highly accurate optimal outcome, the defects inherent in the PSO technique such as premature convergence and loss of diversity. These have to be properly tackled and surmounted by initiating effective alteration or augmentation in the procedure of PSO. Therefore, to attain a further precise outcome and to steer clear of the PSO defects, rather than fixing the value of acceleration coefficient we introduce an Acceleration base Particle swarm optimization (APSO) technique in which acceleration coefficient values are updated on the basis of evaluation function. The proposed algorithm is then applied to solve the graph coloring problem. The algorithm is tested on a set of ten DIMACS benchmark test while limiting the number of usable colors to the expected optimal coloring number. We compared APSO with the standard PSO and found that the proposed algorithm succeeded in solving the sample data set and even outperformed the standard PSO algorithm in terms of the minimum number of colors and minimum number of steps. APSO finds an optimal solution in one step for all graphs, while standard PSO is unable to find an optimal solution in 100 steps. The standard PSO algorithm failed to produce the optimum solution for all graphs except the two graphs named myciel3.col and myciel4.col. Hence computational results show that APSO is feasible and competitive with the standard PSO algorithm. Based on the above conclusions, the following recommendations are being made here with future work. We have tested our algorithms on a few instances. In the future more tests will be carried out to check the performance of the algorithm. Also, the proposed algorithm can be applied to other combinatorial optimization problems.

References

1. M. Baghel, S. Agrawal, S. Silakari, "Survey of Metaheuristic Algorithms for Combinatorial Optimization". International Journal of Compute Application, Vol. 58 No. 19, 2012, pp. 21-31.
2. C. H. Papadimitriou, K. Steiglitz, "Combinatorial Optimization: Algorithms and Complexity". Dover Publications, Inc., New York, 1998.
3. R. Garey, D.S. Johnson, "A guide to the theory of NP-completeness". Computers and intractability. W. H. Freeman, New York, 1979.
4. D.J. Welsh, M.B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problem", The Computer Journal, Vol.10, 1967, pp. 85-86.
5. D. W. Matula, G. Marble, D. Isaacson, "Graph coloring algorithms", Graph Theory and Computing Academic Press, New York, 1972, pp. 109-122.
6. D.S. Johnson, "Approximation algorithms for combinatorial problems". Journal of Computer System Science. Vol. 9, 1974, pp. 256-278.
7. D. Brelaz, "New methods to color the vertices of a graph". Communications of the ACM, Vol. 22, No. 4, 1979, pp. 251-256.
8. F. T. Leighton, "A graph coloring algorithm for large scheduling problems". Journal of research of the National Bureau of Standards, Vol. 84, No. 6, 1979, pp. 489-505.
9. D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon, "Optimization by simulated annealing : an experimental evaluation". Part II, graph coloring and number partitioning, Journal of operation research, Vol. 39, No. 3, 1991, pp. 378-406.
10. M. Chams, A. Hertz, D. Werra, "Some experiments with simulated annealing for coloring graphs". European Journal of operation research, Vol. 32, No. 2, 1987, pp. 260-266.
11. A. Hertz, D. de Werra, "Using tabu search techniques for graph coloring". Computing, Vol. 39, No. 4, 1987, pp. 345-351.
12. C. Avanthay, A. Hertz, N. Zufferey, "A variable neighborhood search for Graph coloring". European Journal of operation research, Vol.151, No. 2, 2003, pp.379-388.
13. A. Hertz, M. Plumettaz, N. Zufferey, "Variable space search for graph coloring". Discrete Applied Mathematics, Vol. 156, No.13, 2008, pp. 2551-2560.
14. L. Davis, "Order-based genetic algorithms and the graph coloring problem", Handbook of Genetic Algorithms, 1991, pp. 72-90.
15. D. Costa, A. Hertz, "Ants Can Color Graphs". Journal of operation research, Vol. 48, 1997, pp. 295-305.
16. G. Cui, L. Qin, S. Liu, Y. Wang, X. Zhang, X. Cao, "Modified PSO algorithm for solving planar graph coloring problem". Progress in Natural. Science, Vol. 18, 2008, pp.353-357.
17. M. Baghel, S. Agrawal, S. Silakari, "Recent trends and development in graph coloring", in Proceedings of the International Conference on Frontiers in Intelligent Computing: Theory and Applications, 2012.

18. J. Kennedy, R. C. Eberhart, "Particle swarm optimization", in Proceedings of the IEEE International Conference on Neural Network, Piscataway, NJ, USA, 1995, pp. 1942-1948.
19. M. R. Hestenes, E. Stiefel, "Methods of conjugate gradients for solving linear systems". Journal of research of the National Bureau of Standards, 1952, pp. 409-436.
20. R. Fletcher, C.M. Reeves, "Function minimization by conjugate gradient methods for optimization". SIAM Journal of optimization, Vol 7, 1964, pp. 149-154..
21. A. Ravindran, K. M. Ragsdell, G. V. Reklaitis, Engineering Optimization, Methods and Applications, John Wiley and Sons Inc, Second Edition, 2006.
22. <http://mat.gsia.cmu.edu/COLOR/instances.html>.