Combining Bat Algorithm with angle modulation for Graph Coloring Problem

Halima Djelloul[1] ,Salim Chikhi[2]
Constantine 2 abdelhamid mehri University
Computer Science department
MISC Laboratory
Constantine,Algeria
[1] halima_djelloul@yahoo.fr
[2]chikhi@umc.edu.dz

Abstract—**Bat algorithm (BA) is one of the most recent bio-inspired algorithms. It is based on the echolocation behavior of microbats. The standard BA is proposed only for continuous optimization problems. In this paper, a binary bat algorithm has been developed and implemented to solve the graph coloring problem. To show the feasibility and the effectiveness of the algorithms, we have used the DIMACS benchmark, and the obtained results are very encouraging.**

Keywords-bat algorithm,bio-inspired algorithm, discrete bat algorithm, graph coloring problem, heuristic, discrete optimization problem.

I. INTRODUCTION

The Graph Coloring Problem (GCP) is one of the most interesting, studied and difficult combinatorial optimisation problems. The GCP consists in coloring each vertex of a given graph by using a minimum number of colors called chromatic number [1], so that no two adjacent vertices are colored with the same color. Unfortunately, GCP has been shown to be NP-hard [2], hence, several approaches were developed to handle this problem, that we can classify into three classes; exact approaches, heuristic approaches and metaheuristics approaches. There exist quite a few exact approaches for this problem; they are generally based on the implicit enumeration algorithms [3], as well as branch-and-bound algorithm and its variants [4, 5]. On the other hand, the constructive approaches have been widely proposed to solve the graph coloring problem. The constructive approaches progressively build the solution, in this category we can cite the algorithm developed by Welsh and Powell [6], the degree of saturation (DSATUR) [7], and the recursive largest first algorithm (RLF) [8]. Moreover, many kinds of metaheuristics and their hybridizations have been used to solve GCP like the tabu Search of Hertz and de Werra[9], Simulated annealing suggested by Johnson et al[10], genetic algorithms [11], ant colony [12],variable neighborhood search VNS [13], Variable Search Space VSP that is an extension of the VNS [14], a memetic algorithm by Zhipeng and Jin-Kao Hao [15], a hybrid Artificial Bee Colony Algorithm for Graph 3-Coloring[16], etc.

Unlike other evolutionary algorithms, the BA does not use any evolution operators (crossover or mutation) to generate new solutions through the search process. Instead, the BA is based on cooperation and communication between individuals of the population. In fact, each bat adjusts its flying in the search space according to its companions' flying experience and its proprieties. The BA has many advantages, and one of the main advantages is that it can balance between the diversification and intensification strategies by adjusting parameters when the global optimality is approaching.

Initially, the BA has been developed to optimize continuous non-linear functions [17,18]. Actually, the algorithm has become a popular bio-inspired algorithm. It has successfully been applied solve a large variety of problems [19]. Accordingly, a few good variants of bat algorithm have been attempted to enhance the performance of the original algorithm. [20] have presented a K-Means Bat Algorithm for efficient clustering.[21] have also introduced a Fuzzy Logic Bat Algorithm for clustering problems.[22] have achieved very good results with their modified bat algorithm to detect phishing web pages.[23] have incorporated differential operator and Lévy flights with the basic concepts of the BA to solve function optimisation problems.[24] have provided a new chaotic bat algorithm to carry out parameter estimation in dynamic biological systems and more other applications [19].

The conventional BA had been impossible to exploit it to solve discrete problems until 2012, when [25] introduced the first binary version of bat algorithm (BBA) to solve feature selection problems. In fact, BBA has been proposed as an heuristic algorithm to improve optimum path forest (OPF) and to deal with any classification technique. The authors have kept the same structure of the original algorithm. Since, they have transformed the bat position obtained by BA in a binary solution using the sigmoid function. This transformation is used in both global and local search strategies.

The present study was designed to investigate the use of the BA to deal with the GCP. The main features of the proposed approach consist in adopting a binary representation of the search space and using an angle modulation in order to generate binary values.

The remainder of this paper is organized as follows. In section 2, a formulation of the tackled problem is given. In section 3, the bat algorithm is presented. .In section 4,the proposed method is described . Experimental results are discussed in section 5. A brief discussion is presented in section 6. Finally, conclusions and future work are drawn.

II. PROBLEM FORMULATION

The GCP is a well-known combinatorial problem, and important task in solving many real problems such as the frequency assignment problem [26], crew scheduling [27], register allocation [28], etc. A graph is k-colorable if and only if it can be colored using k colors. Formally, a k-coloring will be represented by a set S = {C(v_1), C(v_2),…,C(v_n)} such as C(v_i) is the color assigned to the vertex v_i. If for all $\{u,v\} \in E$, $C(u) \neq C(v)$, then S is a legal k-coloring; otherwise, S is an unfeasible k-coloring. In the optimization version of the graph coloring problem, the principal objective is to minimize the total number of colors used to color a given graph.

Formally, the graph coloring problem can be formulated as follows:

Given a k-coloring S = {C(v_1),C(v_2),…,C(v_n)} with the set V = {v_1, ... , v_n} of vertices, the evaluation function f counts the number of conflicting vertices produced by S such that:

$$f(S) = \sum_{\{u,v\} \in E} \delta_{uv} \; . \qquad (1)$$

Where:

$$\delta_{uv} = \begin{cases} 1, & \text{if } C(u) = C(v) \\ 0, & \text{otherwise.} \end{cases} \qquad (2)$$

By consequent, a coloring S with f (S) = 0 corresponds to a feasible k-coloring.

III. BAT ALGORITHM

In order to solve complex problems, ideas gleaned from natural mechanisms have been exploited to develop heuristics. Nature inspired optimisation algorithms have been extensively investigated during the last decade paving the way for new computing paradigms. The ultimate goal is to develop systems that have ability to learn incrementally, to be adaptable to their environment and to be tolerant to noise. Several nature inspired optimisation algorithms were developed in the last decade such as Neural Networks [29], Particle Swarm Optimization [30], Ant Colony Optimization [31]. Artificial Plant Optimisation Algorithm[32], Gravitational Search Algorithm[33], etc. One of the recent developed bioinspired algorithms is the Bat Algorithm [17] which is based on the echolocation behaviour of microbats. BA is a new metaheuristic that simulates some echolocation characteristics of microbats to find prey; it is based on three idealized rules [17]:

1. All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers;
2. Bats fly randomly with velocity v_i at position with a fixed frequency $fr_{\min}$, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, where 0 means no pulses at all, and 1means the maximum rate of pulse emission.
3. The loudness can vary from a large (positive) A_0 to a minimum constant value $A_{\min}$.

The basic pseudo code of bat algorithm is summarized in Figure 1.

```
Objective function f(x), (x=x_1, .., x_d)^T
Initialize the bat population x_i ,(i=1,2,..,n) and v_i
Define pulse frequency fr_i at x_i
Initialize the rates r_i and the loudness A_i
While (t < Max number of iteration ) do
        Generate new solution s by adjusting frequency
        and updating velocities and locations/ solutions
        [(3) to(5)]
        If (rand > r_i) then
           Select a solution among the best solutions
           Generate a local solution around the selected
        best solution
        End
        Generate a new solution by flying randomly
        If (rand < A_i & fr(x_i) < fr(x_*)) then
        Accept the new solutions
        Increase r_i and reduce A_i
        End
        Rank the bats and find the current best x_*
        end
Post process results and visualization
```

Figure 1. Pseudo-code of the bat Algorithm.

The technique is initialized with a swarm of bats collaborate together during the process to find the best solution(s). Each individual bat x_i is determined by five properties: a current position in search space x_i^t, a current velocity v_i^t, a pulse rate r_i^t, a loudness A_i^t and a frequency fr_i.

During the research, all bats will move toward the best solution(s). Consequently, each bat updates its position and its velocity using the following equations:

$$fr_i = fr_{\min} + (fr_{\max} - fr_{\min})\,\beta \qquad (3)$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x_*) fr_i \qquad (4)$$

$$x_i^t = x_i^{t-1} + v_i^t \qquad (5)$$

Where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution and where x_* is the current global best solution which is located after comparing all the solutions among all the n bats.

The update of the velocities and positions of bats have some similarity to the procedure in the standard particle swarm optimization (PSO) [34] as fr_i essentially controls the pace and range of the movement of the swarming particles .To a degree, BA can be considered as a balanced combination of the

standard PSO and the intensive local search controlled by the loudness and pulse rate.
For the local search part, once a solution is selected among the current best solutions, a new one for each bat is generated locally using a random walk:

$$x_{new}=x_{old}+ \epsilon\ A^t \tag{6}$$

Where $\epsilon \in [-1, 1]$ is a random number, x_{old} is one of the best current position,
$A^t =\langle A_i^t\rangle$ is the average loudness of all the bats at this time step.

During the flying process, each bat i updates its loudness value and its emission pulse rate r_i, whenever it finds a new best solution. The updating process is performed as follow:

$$A_i^{t+1}=\alpha\ A_i^t \tag{7}$$

$$r_i^{t+1}=r_i^0[1-exp(-\gamma t)] \tag{8}$$

Where A_i^{t+1} and r_i^{t+1} are the new parameters of the bat i in the next generation (t+1), α and γ are constants, generally $\alpha= \gamma$ and the initial values A_i^0 and r_i^0 are chosen randomly.

IV. The binary bat Algorithm for graph coloring problem

The original BA algorithm is based on the echolocation behavior of microbats, and it operates in continuous search space. Consequently, BA algorithm gives a set of real numbers as a solution of the handled problem. However, a binary optimisation problem needs a binary solution and the real solutions are not acceptable, because they are considered as illegal solutions. Therefore, the solutions must be converted from real values to binary values. In the aim to extend the BA algorithm to discrete binary areas, several binary versions of the Bat algorithm have been introduced we cite: binary bat algorithm for feature selection [35], discrete binary bat algorithm for multidimensional knapsack problem [36].

In this paper, the graph coloring problem is modeled as a binary problem Fig.2 and solved by using a binary bat algorithm which limits the new bat's position to only binary values using the angle modulation.

A. Binary Representation of Graph Coloring Problem

The main objective of the BA is to deal with the binary optimisation problems. Therefore, we need to map the graph coloring solution into a binary representation that could be manipulated by BA. So, the graph coloring solution is represented as binary Fig.2 satisfying the following criteria:

- For a graph of n nodes and k colors, the size of the binary matrix is k*n.
  The columns represent the nodes and the rows represent the colors.
- The presence of 1 in the position *(i, j)* indicates that the node *j* colored with the color *i.*
- In each column there is a single 1, i.e. the node is colored with one color.

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 2. Binary representation of the solution

The main feature of binary bat algorithm is to transform the matrix x which represents the solution from real search area to binary search area, and consequently obtain the binary solution representation. To meet this need, the angle modulation is used to convert the real solution produced by the continuous bat algorithm to binary solution $\boldsymbol{x'}$ as follows:

$$S(x_{id}^t)= \sin(2\pi x_{id}^t \cos 2\pi x_{id}^t)) \tag{9}$$

Where $S(x_{id}^t)$ represents the bit string generated by the function.
To obtain the binary position x_{id}^t, we have replaced (5) by (9 and 10) .In these equations, d=1.. D where D is the problem dimension.

$$x_{id}^t = \begin{cases} 1 & if \quad S(x_{id}^t) > 0 \\ 0 & otherwise \end{cases} \tag{10}$$

For the local search part, we have replaced (6) by (11, 12)

$$S(\epsilon A^t)=\sin 2\pi\epsilon A^t \cos 2\pi\epsilon A^t \tag{11}$$

$$x_{new\,i}^t = \begin{cases} 1 & if \quad S(\varepsilon A^t) > 0 \\ 0 & otherwise \end{cases} \tag{12}$$

B. Constructive Heuristic for Generating the Initial Population

For generating the initial population, we have used a modified version of the constructive RLF algorithm proposed in [38], which colors the vertices one node at a time, in the following greedy way:
Let *Ncl* to be the next color to be assigned, *Y* the set of uncolored vertices that can be assigned to the color Ncl, and *B* the set of uncolored vertices that cannot be assigned to the color Ncl.

- If Ncl > k then the remaining nodes will be colored randomly using the set of colors {1, 2,.., k}.
- Otherwise, choose the first vertex v $\in$ Y randomly, color v with the color Ncl, and move its neighbors included in Y from Y to B.

C. Outlines of The proposed Algorithm

Now, we describe how the angle modulation function has been embedded within a bat algorithm to obtain a binary solution to the graph coloring problem.

Like any other population-based meta-heuristics, the first step in the developed algorithm called AMBCOL (Angle Modulated Bat algorithm for graph COLoring problem) involves setting the parameters for the algorithm. In fact, this is one of the main challenges in getting the algorithm to run in optimum way. In AMBCOL, all parameters are initialized randomly in their intervals shown in Fig.3 except the bat positions which are initialized by using the RLF algorithm [38] for the aim of generating feasible solutions with certain quality and diversity.

AMBCOL progresses through a number of generations according to the BA dynamics. At each iteration, the following main tasks are performed. A new population is generated using the formulas (3), (4), (9) and (10). The next step is to evaluate the new solutions by using the number of edge violated as objective function.

For each bat *i*, one solution among the best ones is chosen randomly x^t_{old} according to the pulse emission rate r^t_i . The selected individual is transformed locally using (11) and (12) for the aim of finding a new improved solution x^t_{new} to replace the current position x^t_i and to replace the best global solution found so far.

Subsequently, a new random solution x^t_{rand} is generated.

If the new solution is better than the best global solution then it can replace it according to the acceptance rate measured by the loudness value A^t_i. Whenever a new solution is accepted, the rate of pulse emission and the loudness of the current bat are updated using the formula (7) and (8).

Eventually, the current best global solution is kept for the next generation and the evolutionary process continues. If a feasible *k*-colouring is found the number of colours k will be reduced by one, and the procedure will be iterated until a legal *k*-colouring cannot be found after a stopping criterion is met. In more details, the proposed AMBCOL can be described as in Fig. 3.

V. IMPLEMENTATION AND VALIDATION

BA for graph coloring problem (AMBCOL) is implemented in Matlab 7.9 with Intel core i3 processor and 4 GB of memory. To assess the efficiency of our approach, a set of standard DIMACS benchmark have been used. The results are given in Table1. The first column is the name of the instance, the second column contains the number of vertices, the third contains the number of edges, the fourth column contains the chromatic number, the fifth column contains the results of our approach (AMBCOL), and the sixth column contains the results of the heuristic used for generating the initial population [38], and the final columns contain the results of two genetic algorithms proposed in [39] and [40]. Finally, Friedman test is used to compare statistically the results found. According to table 1, the AMBCOL was able to reach the optimum in 11 cases out of 13. For the graphs queen6_6 and miles500 the difference was only one.

The statistical Friedman test of Fig. 4 represents a comparison of Exact, BBcol,initPop,GA[39],PGA[40]. First, there are clear differences between the proposed approach and the best solutions given by the randomized constructive heuristic InitPop, Moreover, compared to GA and PGA,AMBCOL has the same performances to the PGA and comparable results to GA algorithm which has managed to obtain the optimal results through using a more sophisticated specific operators.

Input: problem data.

Output: problem solution.

Objective function f(x), $x = (x_1,.., x_d)^T$

- Initialize the bat population x_i , (i = 1, 2,.., n) $\in \{0,1\}^T$ and $v_i \in [0.0, 1.0]$
- Define Pulse frequency f_i at x_i , $f_i = [f_{min}, f_{max}]$, $f_{min}, f_{max} \in [0, 1]$
- Initialize the rates $r_i \in [0.0, 1.0]$ and the loudness $A_i \in [1.0, 2.0]$

while (t < Max number of iterations)

- Generate new solutions by adjusting frequency, and updating velocities and locations [(3),(4),(9) and (10)]

if (rand > r_i)

- Generate a local solution around one of the best solutions [(11) and (12)]

end if

- Generate a new solution by flying randomly

if (rand < A_i & f(x_i) < f(x_*))

- Accept the new solution
- Increase r_i and reduce A_i

end if

- Rank the bats and find the current best x_*

end while

- Post process results and visualization

Figure 3. Pseudo-code of the binary bat algorithm.

VI. DISCUSSION

The bat algorithm is a new bio-inspired algorithm based on cooperation between individuals (called bats).

As stated above, the BA has been first devised to solve continuous optimisation problems. Aiming of introducing a new variant of the bat algorithm for solving any discrete binary problems, we have adapted some parameters and formulations used by the conventional BA to deal with binary solutions. In fact, we have based on the angle modulation to ensure that the bat results belong in the range zero-one. First, we have used the angle modulation of the velocity as a probability to determine the next position of the bat in the global search space. Secondly, we have utilized the angle modulation function of the local random walk as a probability to modify the state of some solution bits in order to obtain an improved solution. Finally, we have adjusted some parameter ranges to fit with the new equations.

TABLE I. RESULTS ON DIMACS GRAPHS

Instance	|V|	|E|	χ	AMBCOL	InitPop [45]	GA [30]	PGA [46]
myciel3	11	20	4	4	4	4	4
myciel4	23	71	5	5	5	5	5
queen5_5	25	160	5	5	7	5	5
queen6_6	36	290	7	8	9	7	8
myciel5	47	236	6	6	6	6	6
huck	74	301	11	11	11	11	11
jean	80	254	10	10	11	10	10
david	87	406	11	11	12	11	11
games120	120	638	9	9	9	9	9
miles250	128	387	8	8	11	8	8
miles500	128	1170	20	21	22	-	-
anna	138	493	11	11	11	11	11
fpsol2.i.1	496	11654	65	65	66	65	65

Figure 4. Friedman test compares the proposed algorithm against genetic algorithms.

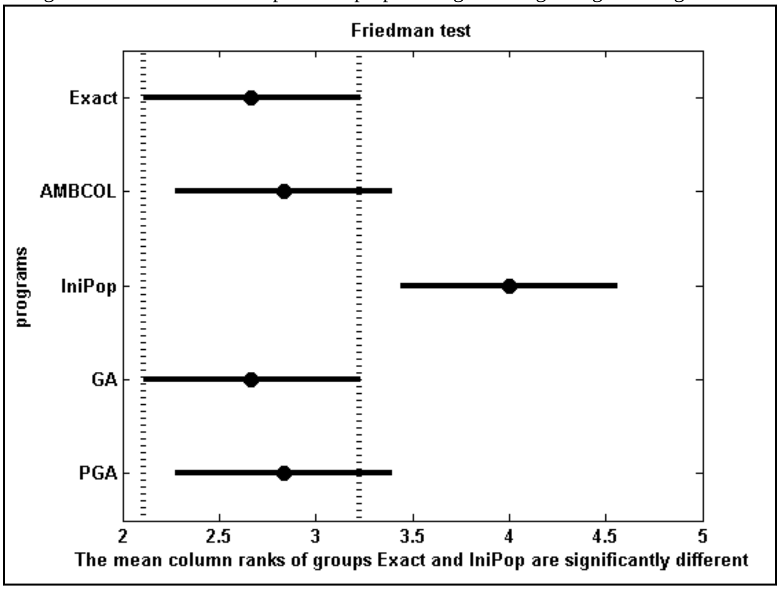

Figure 4. Friedman test compares the proposed algorithm against genetic algorithms.

To validate its performances, the proposed algorithm has been tested on different instances of the GCP. Experimental results have shown the effectiveness of our approach. This effectiveness can be explained as follows.

The bat algorithm is a powerful algorithm that based on three search strategies: the global search, the local search and the random search strategies. These latters are executed sequentially with a given probabilities controlled by the bat parameters (the emission pulse rate and the loudness). The global and the random strategies allow the algorithm to explore the different regions of the search space while the local strategy helps it to find the best solutions in the current regions. Moreover, the dynamic change of the bat parameters

during the flying search allows algorithm to balance between intensification and diversification proprieties.

VII. CONCLUSION

In this work, we have presented a discrete binary version of bat algorithm for the graph coloring problem called AMBCOL. The main contributions of our approach are: the definition of a new binary representation for the graph coloring solution, the use of a modified Binary Solution Representation operation to avoid the infeasible solution, and the use of a modified RLF to create the initial solution. The approach has been thoroughly assessed with different instance types and problem sizes taken from the DIMACS benchmark. The proposed algorithm reduces efficiently the population size, and the number of iterations to have the optimal solution. However there are several issues to improve our algorithm, it is better to integrate a local search method like tabu seach in the core of the algorithm.

References

[1] Matula, D.M., Marble, G., Isaacson, J.D.:Graph coloring algorithms ,In R.C. read.editor,Graph theory and computing,Academic press, pp.109-122, New york (1972).

[2] M.R Garey, D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," , W.H. Freeman & Co, New York 1979.

[3] M. Kubale, B. Jackowski,"A generalized implicit enumeration algorithm for graph coloring," Communication of the ACM, , 1985,28(4):pp.412-418.

[4] I. Méndez-Diaz , P. Zabala, "A branch-and-cut algorithm for graph coloring," Disc. Appl. Math. 154, ,2006,pp.826-847.

[5] A. Mehrotra, M.A. Trick," A column generation approach for graph coloring," INFORMS Journal on Computing 8, ,1996,pp. 344–354.

[6] D.J.A. Welsh, M.B. Powell," An upper bound for the chromatic number of a graph and its application to timetabling problems," Computer Journal, ,1967, Vol. 10, No. 1, pp.85–86.

[7] D. Brélaz,"New methods to color the vertices of a graph,"Communications of the ACM, ,1979,Vol. 22, No. 4, pp.251–256.

[8] F.T. Leighton,"A graph coloring algorithm for large scheduling problems," Journal of Re-search of the National Bureau of Standards, 1979,Vol. 84, No. 6, pp.489–506.

[9] A. Hertz, D. de Werra, "Using tabu search techniques for graph coloring," Computing, 1987,Vol. 39, No. 4, pp.345–351.

[10] D.S. Johnson, C.R. Aragon, L.A. McGeoch, C. Schevon," Optimization by simulated an-nealing: An experimental evaluation", Part II, graph coloring and number partitioning, Operations Research 39, 1991,3, 378–406.

[11] C. Fleurent, J.A. Ferland,"Genetic and hybrid algorithms for graph coloring", Annals of Operations Research 63, 1996,pp. 437–464.

[12] K.A. Dowsland, , J.M. Thompson,"An improved ant colony optimization heuristic for graph colouring,discrete applied mathematics," , 2008,Volume 156,issue 3,Pages 313-324.

[13] C. Avanthay, A. hertz, N. Zufferey," A variable neighborhood search for graph coloring," European Journal of operational Research, 2003, 151, 379-388.

[14] A. Hertz, , M. Plumettaz, N. Zufferey," Variable space search for graph coloring," Discrete applied Mathematics, 2008, 156,2551-2560.

[15] L. Zhipeng, H. Jin-Kao,"A memetic algorithm for graph coloring, ",European Journal of Operational Research,2010, 203, 241–250.

[16] I.Jr. Fister, I. Fister and J. Brest," A Hybrid Artificial Bee Colony Algorithm for Graph 3-Coloring," In Swarm and Evolutionary Computation, Lecture Notes in Computer Science, 7269, Springer Berlin, Heidelberg, 2012, 66-74.

[17] Yang, X.S.:A New Metaheuristic Bat-Inspired Algorithm, In Nature Inspired Cooperative Strategies for Optimization (NICSO), Studies in Computational Intelligence, Springer Berlin, Vol. 284,pp.65--74(2010).

[18] Yang, X.S., Gandomi, A.H.:Bat algorithm: a novel approach for global engineering optimization, Engineering Computations, Vol. 29, No. 5,pp.464--483 (2012).

[19] Yang, X.S.:Bat algorithm: literature review and applications, Int. J. Bio-Inspired Computation, Vol. 5, No. 3,pp.141–-149 (2013).

[20] Komarasamy, G., Wahi, A.:An optimized K-means clustering technique using bat algorithm,European J. Scientific Research, Vol. 84, No. 2,pp.263--273 (2012).

[21] Khan, K., Nikov, A., Sahai A.:A fuzzy bat clustering method for ergonomic screening of office workplaces',In Third International Conference on Software, Services and Semantic Technologies S3T,Springer Berlin Heidelberg (pp.59-66).

[22] Damodaram, R., Valarmathi, M.L.:Phishing website detection and optimization using modified bat algorithm,Int. J. Engineering Research and Applications, Vol. 2, No. 1,pp.870-–876(2012).

[23] Xie, J., Zhou, Y. Q., Chen,H. `A novel bat algorithm based on differential operator and Lévy flights trajectory',Computational Intelligence and Neuroscience, Vol. 2013.

[24] Lin, J. H., Chou,C. W., Yang, C. H.,Tsai, H. L.:A chaotic Levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems, J.Computer and Information Technology, Vol. 2, No. 2,pp.56–-63 (2012).

[25] Nakamura, R.Y.M., Pereira, L.A.M., Costa, K.A., Rodrigues, D., Papa, J.P.,Yang, X.S.:BBA: A binary bat algorithm for feature selection,in: 25th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), IEEE Publication,pp.291--297(2012).

[26] Smith, D.H., Hurley, S., Thiel, S.U.: Improving heuristics for the frequency assignment problem, European Journal of Operational Research 107 (1) 76–86(1998)

[27] Gamache, M., Hertz, A. , Ouellet, J.O. : A graph coloring model for a feasibility problem in monthly crew scheduling with preferential bidding, Computers and Operations Research 34 (8) 2384–2395(2007)

[28] de Werra, D., Eisenbeis, C., Lelait, S. and Marmol, B.:On a graph-theoretical model for cyclic register allocation, Discrete Applied Mathematics, Vol. 93, Nos. 2–3, pp.191–203(1999)

[29] Haykin, S. S.: Neural networks: a comprehensive foundation. Englewood Cliffs, NJ: Pren-tice Hall(2007)

[30] García-Gonzalo, E., Fernández-Martínez, J. L.: A Brief Historical Review of Particle Swarm Optimization (PSO), Journal of Bioinformatics and Intelligent Control, 1(1): 3-16 (2012)

[31] Abdelaziz, A. Y., Osama, R. A., Elkhodary, S. M.: Application of ant colony optimization and harmony search algorithms to reconfiguration of radial distribution networks with dis-tributed generations. Journal of Bioinformatics and Intelligent Control, 1(1), 86-94(2012)

[32] Yu, B., Cui, Z., Zhang, G.: Artificial Plant Optimization Algorithm with Correlation Branches. Journal of Bioinformatics and Intelligent Control, 2(2), 146-155(2013)

[33] Sahu, R. K., Panda, S., Rout, U. K., Raul, P.:Application of Gravitational Search Algorithm for Load Frequency Control of Multi Area Power System. Journal of Bioinformatics and Intelligent Control, 2(3), 200-210(2013)

[34] Kennedy, J. and Eberhart, R.C.:Particle Swarm Optimization, IEEE Int. Conf. On Neural Networks, WA, Australia,pp.1942–-1948(1995)

[35] Nakamura, R.Y.M., Pereira, L.A.M., Costa, K.A., Rodrigues, D., Papa, J.P., Yang, X.S. : BBA: A binary bat algorithm for feature selection, in: 25th SIBGRAPI Conference on Graphics, Patterns and Images, IEEE Publication, pp.291-297(2012)

[36] Sabba, S., Chikhi, S.: A discrete binary version of bat algorithm for multidimensional knapsack problem, Int. j. Bio-Inspired Computation , Vol. 6, No. 2, 2014

[37] Mabrouk,B.B., Hasni, H., Mahjoub, Z., On a parallel genetic–tabu search based algorithm for solving the graph colouring problem, European Journal of Operational Research 197 1192–1201, 2009.

[38] Musa M. Hindi, Roman V. Yampolskiy. Genetic Algorithm Applied to the Graph Coloring Problem, the Twenty-third Midwest Artificial Intelligence and Cognitive Science Conference. 2012.

[39] Abbasian,R. and Mouhoub, M., An efficient hierarchical parallel genetic algorithm for graph coloring problem, In Proceedings of The 13th annual conference on Genetic and evolutionary computation, 521-528. Dublin, Ireland: ACM, 2011.