Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 18 (2008) 353–357
Progress in Natural Science

Special report

Modified PSO algorithm for solving planar graph coloring problem

Guangzhao Cui [a], Limin Qin [a,*], Sha Liu [a], Yanfeng Wang [a], Xuncai Zhang [b], Xianghong Cao [a]

[a] College of Electrical and Electronic Engineering, Zhengzhou University of Light Industry, Zhengzhou 450002, China
[b] Research Institute of Biomolecular Computer, Huazhong University of Science and Technology, Wuhan 430074, China

Received 29 October 2007; received in revised form 20 November 2007; accepted 20 November 2007

Abstract

The graph coloring is a classic NP-complete problem. Presently there is no effective method to solve this problem. Here we propose a modified particle swarm optimization (PSO) algorithm in which a disturbance factor is added to a particle swarm optimizer for improving its performance. When the current global best solution cannot be updated in a certain time period that is longer than the disturbance factor, a certain number of particles will be chosen according to probability and their velocities will be reset to force the particle swarm to get rid of local minimizers. It is found that this operation is helpful to improve the performance of particle swarm. Classic planar graph coloring problem is resolved by using modified particle swarm optimization algorithm. Numerical simulation results show that the performance of the modified PSO is superior to that of the classical PSO.

Keywords: Planar graph coloring problem; PSO; Optimization algorithm

1. Introduction

The graph coloring is a classical combination–optimization problem which has a favorable applied background both in theories and engineering applications, such as circuit layout problem, working procedure problem, timetable problem and storage problem. Therefore, many scholars have been attracted to carry on researches on this problem [1]. The graph coloring problem has been proved to be a classic NP-complete problem, which has not an effective strategy to get the best solution until now. For solving this kind of problem, both the exact algorithm and approximate algorithm have been used including genetic algorithm [2], swarm algorithm [3,4], DNA algorithm [5,6], tabu search algorithm, greedy algorithm, neural network algorithm [7], etc.

Particle swarm optimizer (PSO) originated from the observation of the flock foraging behavior, the research of artificial life and evolutionary computing theory, which is a new swarm intelligence algorithm proposed by Kennedy and Eberhart in 1995. Because of its fast convergence rate, simple computation and easy realization, PSO has extensive applications in optimization problem of continuous space [8]. In certain cases, it has an even higher efficiency than genetic algorithm. However, it does not effectively solve the discrete and combination–optimization problem. Kennedy [9] proposed a binary PSO algorithm in 1997, Fatih and Liang [10] applied this algorithm to solve the batch order problem. Aiming at the discrete space optimization problem like the graph coloring, we present a modified PSO algorithm by establishing its objective optimal mathematical model. A disturbance factor is added to a particle swarm optimizer for improving PSO algorithms' performance. It is found that this operation is helpful to improve the performance of particle swarm. Our numerical simulation results show that the performance of the modified PSO (MPSO) algorithm is superior to that of the classical PSO.

* Corresponding author. Tel.: +86 13783573060; fax: +86 371 63556791.
E-mail address: qinlimin2008@163.com (L. Qin).

1002-0071/$ - see front matter
doi:10.1016/j.pnsc.2007.11.009

2. Description of the planar graph coloring problem

2.1. Definition of the planar graph

The planar graph coloring problem is one of the three recent mathematical difficulties. As early as 100 years ago there were a large number of people interested in this issue, and many mathematicians had proven that any planar graph could be colored by four kinds of colors, which is called the four-color problem, also known as the four-color conjecture. The planar map coloring is a vivid particular case of the planar graph coloring problem in that adjacent areas are colored by different colors so that the region can be easily distinguished. It is demonstrated by mathematical language that the plane is arbitrarily subdivided into different areas which do not overlap and every area can always be marked with one of the four numbers 0, 1, 2 and 3, so it will not obtain the same numbers in two adjacent areas. Though "the four-color conjecture" had been proposed nearly 160 years ago, it has not found a rigorous mathematical proof yet. In 1879 Kempe presented the first proof of the conjecture, but in 1890 Hewood found it was not true. It was not until 1976 Appel and Haken had proven and solved the 4-color graph coloring problem with computer-assisted method based on the order approximation. However, it cost much longer time when they solved the large-scale problems, and the computing time was proportional to the $O(n^2)$ (n is the number of areas on the map). In the terms of the knowledge of graph theory, this conclusion can be transformed into another explanation that any planar graph may color normally to the node with not more than four different colors. In this paper, we do not consider the four-color theorem itself, but consider how to elicit a coloring plan to color the node normally with four different colors to any planar graph, and how to use modified PSO heuristic algorithm to solve the map and planar graph coloring problem.

2.2. Mathematical description of the planar graph coloring problem

Given an undirected connected planar graph $V(G) = \{v_1, v_2, \ldots, v_n\}$ which is a node set consisting of n nodes, the incidence matrix of each node is $R = (r_{ij})_{n\times n}$, where

$$r_{ij} = \begin{cases} 1, & \text{node } i \text{ and } j \text{ connected and } i \neq j; \\ 0, & \text{node } i \text{ and } j \text{ unconnected, or } i = j \ (1 \leqslant i, j \leqslant n) \end{cases}$$

Four different colors are applied to the coloring of all the nodes to make any two adjacent nodes colored in difference. We use 0, 1, 2 and 3 to represent four different colors, and accordingly a reasonable coloring program corresponds to a string with a length of n.

Fig. 1(a) illustrates a kind of coloring program of seven areas of planar graph; Fig. 1(b) is the incidence matrix of the planar graph. According to our coding assumption above, this coloring program can be expressed by

$$V = v_1, v_2, v_3, v_4, v_5, v_6, v_7 \tag{1}$$

The corresponding coloring sequence can be expressed as

$$S = 0, 1, 2, 3, 2, 1, 0 \tag{2}$$

To facilitate the description, a certain coloring program of the incidence matrix $R = (r_{ij})_{n\times n}$ of planar graph with n nodes $V_n = v_1, v_2, \ldots, v_n$ can be indicated as $S = q_1, q_2, \ldots, q_n$, where $q_i \in \{0, 1, 2, 3\}$. To determine whether the coloring sequence S satisfies the conditions of the program, we define the following fitness function $f(s)$ as the following.

To any $v_i \in V = \{v_1, v_2, \ldots, v_n\}$, given $S = q_1, q_2, \ldots, q_n$, make $x_i = q_i$, $g(x_i)$ is the colored node number with x_i, which adjoin to the node of v_i, ($i = 1, 2, \ldots, n$); $g(x_i)$ will be computed by inquiring incidence matrix

$$f(s) = \sum_{i=1}^{n} g(x_i) \tag{3}$$

It is obvious that $f(s) \geqslant 0$ and $f(s)$ is an even number. It corresponds to a reasonable program when $f(s) = 0$. From the above definition we can see that $g(x_i)$ represents the aggregate of coloring conflict node of node v_i in the coloring program s, while $f(s)$ represents the aggregate of node coloring in coloring program s. For a convenient description, the fitness function $f(s)$ can be considered as the conflict's degree of coloring program s in this paper.

3. Modified PSO

PSO is an efficient, robust and simple optimization algorithm for solving many continuous optimization problems. Aiming at the particularity of the discrete space optimization problem such as the graph coloring, we adapt the quaternary PSO to our problem and introduce the disturbance strategy.

3.1. Original PSO

PSO is a stochastic optimization approach which maintains a swarm of candidate solutions, referred to as particles. The method is inspired by the movement of particles and their interactions with their neighbors in the group. Every particle in the swarm begins with a randomized position (x_i) and (possibly) randomized velocity (v_i) in the

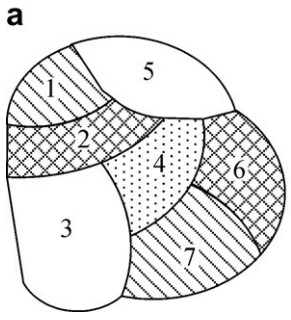

Fig. 1. Four coloring program of seven areas map and its incidence matrix.

D-dimensional search space, where x_{id} represents the location of particle index i in the dth dimension of the search space. Candidate solutions are optimized by flying the particles through the virtual space, with attraction to positions in the space that yielded the best results. Each particle remembers the position in which it achieves its highest performance (p_{id}) and this position is marked as pbest. Every particle is also a member of some neighborhood of particles, and remembers the best overall position in which it achieves that neighborhood (given by the index g) and this position is marked as gbest.

$$\begin{cases} v_{id}^{t+1} = w \times v_{id}^{t} + c_1 \times rand() \times (p_{id} - x_{id}^{t}) \\ \qquad + c_2 \times rand() \times (p_{gd} - x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \end{cases} \tag{4}$$

for $i = 1, \ldots, M$ and $d = 1, \ldots, D$, where w is the inertia coefficient which controls the momentum of the particle by weighing the contribution of the previous velocity—basically controlling how much memory of the previous flight direction will influence the new velocity; c_1 is the weight given to the attraction to the previous best location of the current particle and c_2 is the weight given to the attraction to the previous best location of the particle neighborhood; *rand*() is a sampling of a uniformly-distributed random variable in $[0, 1]$.

3.2. Quaternary PSO

The original PSO algorithm can only optimize problems in which the elements of the solution are continuous real numbers, while it cannot be used to discrete problem directly. Aiming at this large type of problems, Kennedy and Eberhart proposed a discrete binary version PSO method. Though the discrete-value modification for PSO has been shown to be able to optimize various discrete-valued problems, it is limited to discrete problems with binary-valued solution elements. Here, we developed a quaternary-valued PSO method by defining the particles' trajectories and velocities in terms of changes of probabilities of solution elements' assumed values. It has improved the model in the literature [9] and combined the solution requirement of the graph coloring problem to demonstrate the efficiency of PSO in combination–optimization problem. Thus a particle moves in a state space restricted to zero, one, two, and three on each dimension, where each v_{id} represents the probability of x_{id} taking the assumed value. The velocity is updated in the same way as in standard PSO. For assigning a new particle value for x_{id}, the velocity term is transformed to the range $[0, 1]$, and according to the relations of size among *rand*(), $r(r = 0.5)$ and $S(v_{ij})$, the particle element is randomly set. The equations for the new algorithm are given below:

$$\begin{cases} v_{id}^{t+1} = w \times v_{id}^{t} + c_1 \times rand() \times (p_{id} - x_{id}^{t}) \\ \qquad + c_2 \times rand() \times (p_{gd} - x_{id}^{t}) \\ x_{id}^{t+1} = Mod((x_{id}^{t} + f(v_{id}^{t+1})), 4) \end{cases} \tag{5}$$

where the *Mod*(number, divisor) function returns the remainder after a number is divided by a divisor, $f(v)$ is given by Eq. (6) and $S(v)$ is the sigmoid function given by $S(v) = 1/(1 + e^{-v})$,

$$f(v) = \begin{cases} 0, rand() > r \& rand() < S(v) \\ 1, rand() < r \& rand() < S(v) \\ 2, rand() \leqslant r \& rand() \geqslant S(v) \\ 3, rand() \geqslant r \& rand() \geqslant S(v) \end{cases} \tag{6}$$

3.3. Synergistic optimization algorithm and disturbance strategy of particle swarm

From Eq. (4) it is not difficult to find that the particles of classical PSO algorithm are always chasing the current global best solution and the partial best solution of a single particle is its own that they searched for up to now. Therefore, the velocity of the particles will be quickly dropped close to 0 and they are likely to be trapped into the local minimizers that will not allow them to escape. This phenomenon is known as the PSO's "convergence". This "convergence" restricts the search scope of particles. It is necessary, for expanding the search scope, to increase the number of the particles of particle swarm or decrease the chasing of particles to the current global best optimization of the entire particle swarm. Increasing the number of particles will lead to the increase of the computational complexity of the algorithm; however, decreasing the chasing of particles to the global best solution will also lead to the disadvantage of misconvergence of the algorithm. Because PSO is likely to fall into the local minimizers, the velocity of particles is close to 0 leading to the unchanged position of particles. We increase the disturbance factor in classical PSO: If the continuous step u iterations of the global best adaptive value that we have looked for so far have been not been updated, we randomly select a certain number of particles according to probability r and reset the velocity of them. u is a natural number which is called disturbance factor, r is a random number between 0 and 1. The disturbance strategy can be described as,

$$\text{If } t - t_u > u, \text{ Then reset } v \tag{7}$$

where t_u is the iterative step of the global best adaptive value that has been updated and searched recently. The thought of disturbance strategy is when the PSO falls into local minimizers, we should randomly choose a certain amount of particles whose velocities will be reset in order to force swarms getting out of local minimizers to trigger a new search process.

4. Experimental design and result

4.1. Fitness function

The planar graph coloring is a complicated NP-complete problem. However, it is considerably simple to determine

whether a coloring project satisfies the conditions, as long as the adjacent nodes do not color the same. Based on the above mathematical description of planar graph coloring problem, a coloring scheme $S = q_1, q_2, \ldots, q_n$ for the nodes of the known planar graph $V = (G)$, function $f(s) = \sum_1^n g(x_i)$ is the adaptive value of scheme S, when $f(s) = 0$, S is a reasonable coloring scheme of the planar graph.

4.2. MPSO algorithm of the planar graph coloring problem

According to the thought and improved strategy of the above algorithms [11,12], the main steps of MSPO's realization are as follows:

Step 1: Input the incidence matrix R, the initialized maximal iterations M, the number of particles N, w, c_1, c_2 and the step of disturbance factor u; randomly initialize the initial locations and velocities of all the particles and guarantee them in a feasible space (that is to satisfy the given constraint); the current position of each particle is as individual extreme value P_i, the swarm optimal value of all the particles as global extreme value P_g.

Step 2: Decide whether $f(s) = 0$ or $f(s)$ reaches the maximal iterations, if satisfied, turn to step 6, or perform step 3.

Step 3: Adjust and update each particle: ① compute the velocity v_i of each particle according to Eq. (5) and update each position x_i; ② amend the acquired position, the velocities should not exceed the feasible space, if exceed, reset the velocity as a random number between $-V_{max}$ and V_{max}; ③ judge whether each particle satisfies the given constraint, if not, correct the value of w and re-update the local velocity and location of the particle, rectify the particle until it goes into the feasible space; ④ calculate the adaptive value of each particle with Eq. (3); ⑤ update P_i, and P_g according to the adaptive value of each particle.

Step 4: If $t - t_u > u$, randomly select a certain number of particles according to probability r and reset the velocities v_i of them, then turn to step 3, or perform step 5.

Step 5: Examine the exit conditions of the loop: return to step 2 when it achieves the maximal iterations or $f(s) = 0$.

The flow chart is described in Fig. 2.

4.3. Experimental simulation result

It is necessary to validate the results of the introduced disturbance factor to compare the performance of the improved MPSO algorithm with the original PSO algorithm. Under the setting conditions of the same parameters in this paper, we randomly generate planar graph using C++ to program and perform numerical simulation to actualize the two algorithms. Taking weight coefficient for example, the larger inertia weight is favorable to the global search of PSO, and the smaller one is inclined to the local search. The inertia weight is decreased by iterations, i.e., $w = w_{min} - l \times (w_{max} - w_{min})/l_{max}$, l is iterations, l_{max} is maximal iterations. PSO is able to search in the entire solution space with a greater probability in the initial search stage, and it will rapidly converge in a local area of the best solution, then PSO realizes fine tuning in the area with the descending of the inertia weight. Set the number of PSO as 200 and initial weight w as 2 which will descend to 0.8 in linear iteration. To avoid too small setting of the experience value which will results in the too small changes of particle position after taking the module, $c_1 = 2$, $c_2 = 1.8$ are used here and the maximal iterations of algorithm are 10,000. We timely adjust the corresponding disturbance factor in accordance with the scale of the problem.

By randomly generating a given scaled planar graph, and separately calculating 100 times to MPSO algorithm and classical PSO, the experimental results we obtained are shown in Table 1. It can be seen that under the same parameters MSPO has a faster convergent velocity and a better global search capability to solve the same problem. In order to demonstrate the performance of MPSO algorithm for solving the planar graph coloring problem further, we take the coloring problem of the map of China as an example. Fig. 3 is the map of China, including 31 provinces, municipalities and autonomous regions, of which incidence matrix is a symmetric matrix of 31×31. We, respectively, use PSO and the improved MPSO to perform the experiment and simulation. The results demonstrate that the improved MPSO method is able to present

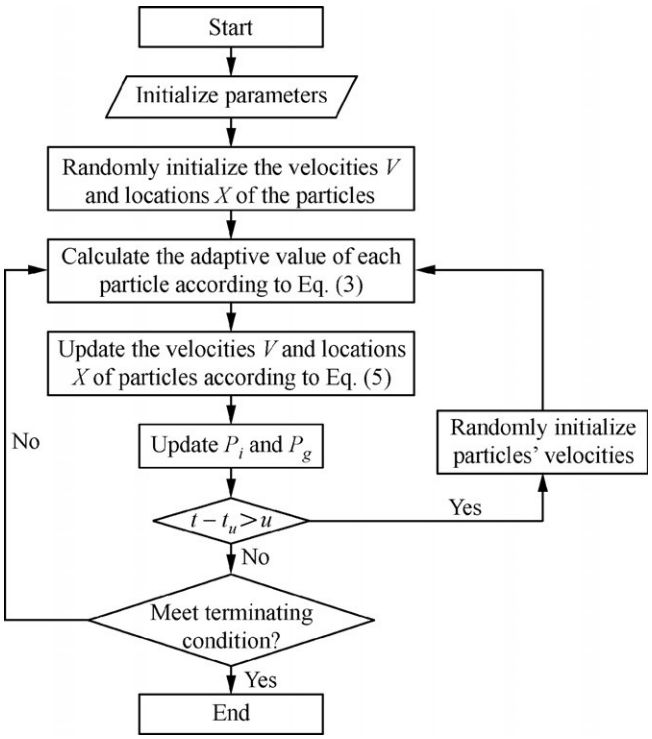

Fig. 2. Algorithm's flow diagram.

Table 1
Comparison of the experimental results

Nodes n	Algorithms	Maximal iterations	Minimal iterations	Average iterations	Correct coloring rate
7	MPSO	7	0	5	100
	PSO	8	0	6	100
10	MPSO	47	5	26	100
	PSO	66	6	32	100
20	MPSO	4637	36	1569	59.7
	PSO	5895	116	2418	42.3
30	MPSO	10,000	1786	5439	16.4
	PSO	10,000	3426	6432	10.9

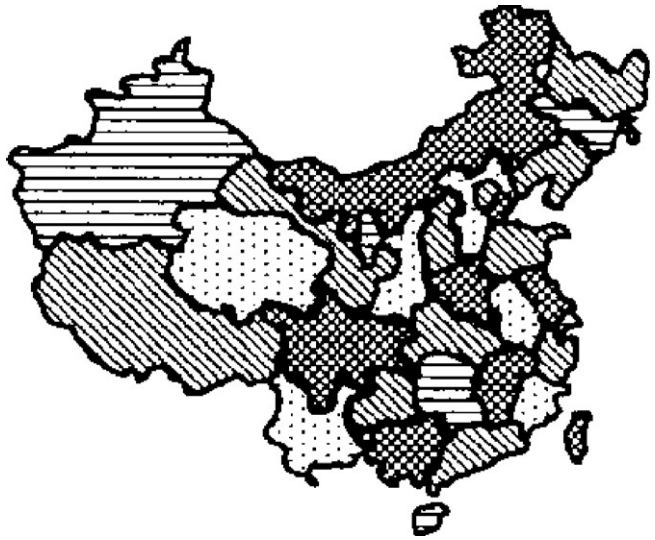

Fig. 3. A specific description of the map of China for 4-coloring problem.

a coloring scheme to this problem in a feasible time. Fig. 3 is also a specific description of one solution for solving its 4-coloring problem using MPSO algorithm.

5. Conclusion

In terms of the discrete space optimization problem such as the graph coloring, we propose a modified PSO algorithm by establishing its objective optimal mathematical model. A disturbance factor is added to a particle swarm optimizer for improving PSO algorithms' performance, which is helpful to avoid the disadvantage of particle swarm easy to be locally minimizers. The experimental results show that this algorithm is considerably effective to the graph coloring problem with moderate size, however, with the increasing scale of the problem, the particles' iterations need increase and they are more and more difficult to jump out of the approximate best solution. Generally speaking, the larger scale of the particle swarms needs more evolutionary time so the more quantities of the best solutions are required, and the computational complexity will subsequently increase greatly. Therefore, it is necessary to continually find a more effective coloring algorithm to the larger scale coloring problem. We believe that in combination with other optimization algorithms, such as the crossover and mutation operator of genetic algorithm, the global search capability of the swarms can be enhanced, or with simulated annealing algorithms and ant swarm algorithms, the convergence of the algorithm can be improved. This will be the direction in the future research.

Acknowledgment

This work was supported by National Natural Science Foundation of China (Grant Nos. 30370356, 60773122).

References

[1] Philippe G, Alain H. A survey of local search methods for graph coloring. Comput Oper Res 2006;33:2547–62.
[2] Hong B. Generic algorithm of color planar graph. J Guizhou Univ (Nat Sci) 1999;11(16):232–97.
[3] Wang XH, Zhao SM. Ant algorithms for solving graph coloring. J Inner Mongolia Agric Univ 2005;9(26):79–82.
[4] Salari E, Eshghi K. An ACO algorithm for graph coloring problem. IEEE Service Center 2005;1:4244-0020-1.
[5] Gao L, Xu J. A DNA algorithm for graph vertex coloring problem. Acta Electronic Sinica 2003;4:494–7.
[6] Wang SD, Liu WB, Xu J. DNA sticker algorithm for vertex-coloring problems of graph. Syst Eng Electron 2005;27(3):568–72.
[7] Wang XH, Wang ZO, Qiao QL. Artificial neural network with transient chaos for four-coloring map problems and k-colorability problems. Syst Eng Theory Practice 2002;05:92–6.
[8] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the IEEE international conference on neural networks, vol. 5, Piscataway, NJ; 1995. p. 1942–8.
[9] Kennedy J, Eberhard R. A discrete binary version of the particle swarm optimization. In: Proceedings of the conference on system, man, and cybernetics. USA: IEEE Service Center; 1997. p. 4104–8.
[10] Fatih M, Liang Y. A binary particle swarm optimization algorithm for lot sizing problem. J Econ Soc Res 2003;5(2):1–20.
[11] Cui GZ, Zhang XC, Wang YF. Design of encoding sequences for DNA computing based on PSO. Dyn Continuous Discrete Impulsive Syst 2007;14:40–6.
[12] Cui GZ, Niu YY, Wang YF. A new approach based on PSO algorithm to find good computational encoding sequences. Prog Nat Sci 2007;17(6):712–6.