An Ant Based Particle Swarm Optimization Algorithm for Maximum Clique Problem in Social Networks

Mohammad Soleimani-pouri, Alireza Rezvanian and Mohammad Reza Meybodi

Abstract In recent years, social network services provide a suitable platform for analyzing the activity of users in social networks. In online social networks, interaction between users plays a key role in social network analysis. One of the important types of social structure is a full connected relation between some users, which known as clique structure. Therefore finding a maximum clique is essential for analysis of certain groups and communities in social networks. This paper proposed a new hybrid method using ant colony optimization algorithm and particle swarm optimization algorithm for finding a maximum clique in social networks. In the proposed method, it is improved process of pheromone update by particle swarm optimization in order to attain better results. Simulation results on popular standard social network benchmarks in comparison standard ant colony optimization algorithm are shown a relative enhancement of proposed algorithm.

Keywords Social network analysis · Clique problem · ACO · PSO

1 Introduction

Today online social networks are formed a new type of life due to some facilities and services for a wide ranges of ages such as young to old. Besides, there is no doubt about either influence or growth of social networks. Therefore, several

M. Soleimani-pouri (✉)
Department of Electrical, Computer and Biomedical Engineering, Qazvin Branch, Islamic Azad University, Qazvin, Iran
e-mail: m.soleimani@qiau.ac.ir

A. Rezvanian · M. R. Meybodi
Soft Computing Laboratory, Computer and IT Engineering Department, Amirkabir University of Technology, Tehran, Iran
e-mail: a.rezvanian@aut.ac.ir

M. R. Meybodi
e-mail: mmeybodi@aut.ac.ir

F. Can et al. (eds.), *State of the Art Applications of Social Network Analysis*, Lecture Notes in Social Networks, DOI: 10.1007/978-3-319-05912-9_14,

many of researchers are focused on social network analysis aspects. It seems to be useful, studying certain structure of relation between users in social networks. One of the important user group structure associated with a full connected of some users, which known as clique structure [1, 2]. Several applications of finding clique are reported by researchers such as social networks analysis [3, 4], online shopping recommendation [5], evolution of social networks [6], scheduling [7], biochemistry [8], computer vision [9] and wireless sensor networks [10]. In literature, finding clique is categorized as NP-complete problems in graph theory [11].

In maximum clique problem, finding the largest complete subgraph is considered. It was introduced by Karp [12]. Various types of algorithms have been presented to solve clique problem, while evolutionary algorithms such as genetic algorithm (GA) and ant colony optimization (ACO) have been used more than others. Popular algorithm named *Ant-clique* algorithm, which make a maximal clique using sequential greedy heuristics based on ACO by adding vertices to the partial cliques iteratively [13]. Besides, another ACO based method hybridized by simulated annealing (SA) [14] and tabu search [15]. Although new hybrid algorithm obtained good results, they have a high complexity in practice.

In this study, Particle Swarm Optimization (PSO) algorithm has been applied as the heuristic to enhance the performance of ACO algorithm for finding the maximum clique in social network graph. Simulation results on social network benchmark are shown the better results in comparison with standard ACO algorithm. In the rest of this paper, Sects. 2 and 3 are consisted of ACO and PSO introduction respectively, in Sect. 4, proposed method is discussed. Simulation results on social networks datasets are reported in Sect. 5.

2 Ant Colony Optimization

Ant Colony optimization (ACO) algorithm works well for solving several discrete problems. The basic algorithm of ACO was proposed by Dorigo as a multi agent approach in order to solve traveling salesman problem (TSP) [16]. The main idea of ACO algorithm is inspired from the behavior of seeking out food by colonies of ants. Ants search their environment randomly to find food. They return some of the food to their nest once found a food and leave pheromone in their return path. The amount of pheromone left on their path depends on quality and size of the food source and it evaporates gradually. Remaining pheromones will persuade other ants to follow the path and just after a short time, majority of the ants will trace the shorter path which is marked with stronger pheromone. Procedure of ACO algorithm has been presented in Fig. 1.

During running of the algorithm, ants first produce different solutions randomly in the main loop after initialization. Afterwards, the solutions are improved by updating the pheromones and using a local search optionally. According to the problem and graph traverse, pheromones set on vertices or edges of graph. Traversing the edge

Fig. 1 Pseudo-code of ACO algorithm [17]

Algorithm 1 ACO algorithm

```
procedure ACO_MetaHeuristic
    while (termination_conditions)
        generateSolutions()
        daemonActions() {Optional}
        pheromoneUpdate()
    endwhile
end procedure
```

between vertices i and j depends on the probability of edge which is calculated as below:

$$p_{ij}^{k} = \frac{\left(\tau_{ij}^{\alpha}\right)}{\sum\left(\tau_{ij}^{\alpha}\right)} \tag{1}$$

where, ${p_{ij}}^{k}$ is probability of traversing the edge between vertices i and j, while ${\tau_{ij}}^{\alpha}$ is amount of pheromone present on the above mentioned edge. An optional local search can contribute to improvement of the results prior to updating the pheromones. However, method of updating the pheromones can be like this:

$$\tau_{ij}^{t+1} = (1-\rho)\,\tau_{ij}^{t} + \Delta\tau_{ij}^{t} \tag{2}$$

where, ρ is evaporation rate of pheromone, ${\tau_{ij}}^{t+1}$ is amount of new pheromone for edge between i and j, ${\tau_{ij}}^{t}$ is amount of current pheromone for edge between i and j, ${\Delta\tau_{ij}}^{t}$ is amount of reinforced pheromone for proper solutions which can be calculated from the following equation.

$$\Delta\tau_{ij}^{t} = \begin{cases} 1 & \text{if } \tau_{ij}^{t} \in \text{good solution} \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

3 Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based optimization technique developed by Eberhart and Kennedy in 1995, which inspired from social behavior of birds seeking for food. In this method, group of birds seek for food in a specified space randomly. Birds follow the bird with the shortest distance to food in order to find position of the food [18].

Every solution in PSO algorithm which is called a particle corresponds to a bird in their pattern of social movement. Each particle has a value of fitness calculated by a fitness function. A particle bears greater fitness once it is located in a closer position to the target (food in the model of moving birds) within search space. Moreover, any particle represents a speed which is responsible to direct the particle. A particle will keep going through the problem space by following the optimum particle at current state.

A group of particles (solutions) are created randomly at the beginning of particle swarm optimization algorithm and they try to find the optimum solution through being updated among generations. A particle is updated in each step using the best local and global solution. The best position a particle has ever succeeded to reach is called *pbest* and saved while the best position achieved by the population of particles is named *gbest*. Velocity and location of each particle will be updated using Eqs. (4) and (5) in each step of implementing the algorithm after finding the best local and global values.

$$\mathrm{v_i^{t+1}} = \mathrm{wv_i^t + c_1r_1\left(pbest_i^t - x_i^t\right) + c_2r_2\left(gbest_i^t - x_i^t\right)} \tag{4}$$

$$\mathrm{x_i^{t+1}} = \mathrm{x_i^t + v_i^t} \qquad \mathrm{i = 1, \ldots, m} \tag{5}$$

where, v_i is the velocity of ith particle and x_i is the current position of it. r_1 and r_2 are random values in the range of (0,1). c_1 and c_2 are learning parameters usually assumed equal ($c_1 = c_2$). w is inertia weight which is considered as a constant or variable coefficient as random, linear, nonlinear and adaptive [19]. PSO has been used in various applications [20–22] and this research utilizes it to improve the amount of pheromones.

4 Proposed Algorithm

High complexity was a major drawback of the previous heuristic methods for solving the clique problem since it significantly adds to the volume of calculations. All methods provided so far apply the following relation to calculate $\Delta\tau$ although the proposed algorithm take the advantage of PSO algorithm to improve results and reduce complexity.

$$\Delta\tau^{t+1} = \frac{1}{(1 + |G\text{-best}\,| - |\text{best-tour}|\,)} \tag{6}$$

This algorithm has used the hybrid of ACO and PSO algorithms in order to find the maximum clique in a graph. For this purpose, some ants are placed initially on the graph and follow paths to find the maximum clique. After evaporation of existing pheromones, proper path is updated by its amount on the edges using PSO algorithm. This procedure is repeated until the optimum clique is obtained on the desired graph. Determining the amount of pheromone through PSO is such that the total pheromone measured at this step and the total pheromone associated with the best answer up

Algorithm 2 proposed Algorithm

For each ant choose randomly a first vertex $v_f \in V$
$C \leftarrow \{v_f\}$
candidates$\leftarrow\{v_f/(v_i,v_i)\in E\}$
while candidates$\neq\emptyset$ do
  choose a vertex $v_i\in$ candidate with probability by equation (1)
  $C\leftarrow C\cup\{v_i\}$
  candidates$\leftarrow$candidates$\cap\{v_f/(v_i,v_f)\in E\}$
endwhile
evaporate()
Evaluation best current clique for all ant
Pheromone update $\Delta\tau$ based on PSO by equation (7)

Fig. 2 Pseudo-code of proposed algorithm based on ACO and PSO

to present step will be calculated taking into account the amount of pheromones at current step. Now, $\Delta\tau$ for the reinforced pheromone of the desired clique will be calculated with PSO algorithm using the following equation:

$$\Delta\tau^{t+1} = \Delta\tau^{t} + V^{t} \tag{7}$$

where, $\Delta\tau_{ij}{}^{t}$ is amount of reinforcement for current pheromone and $\Delta\tau_{ij}{}^{t+1}$ is amount of reinforcement for new pheromone, while V^t gives the amount of change which can be achieved from this equation:

$$\mathrm{V}^{t+1} = \mathrm{c}_1\mathrm{r}_1(\mathrm{p}\tau - \Delta\tau) + \mathrm{c}_2\mathrm{r}_2(\mathrm{g}\tau - \Delta\tau) + \mathrm{c}_3\mathrm{V}^{t} \tag{8}$$

where, V^{t+1} is the new value of V^t. r_1 and r_2 are two random values within range of (0,1), while c_1, c_2 and c_3 are learning parameters ($c_1 = c_2 = c_3$). $p\tau$ and $g\tau$ are considered as the pheromone correspondent with the best current clique and the best clique found so far, respectively. In this case, discovering the changed amount of pheromone will be implemented much more intelligent.

Taking into consideration the mentioned items, general routine of proposed the algorithm can be presented in Fig. 2.

5 Simulation Results

5.1 Comparison of the Proposed Algorithm

For evaluation of the proposed method, experiments applied on some popular social network datasets [19] and DIMACS graphs [2]. The descriptions of these networks are listed in the Table 1.

Table 1 Descriptions of social network datasets and DIMACS graphs for experiments

Name	Descriptions	Nodes	Edges
SN_I	Zachary's karate club	34	78
SN_II	Common adjective and nouns in "David Copperfield"	112	425
SN_III	Neural network of the nematode C. Elegans	297	8479
SN_IV	Social network of dolphins, Doubtful Sound, New Zealand	62	159
SN_V	Pajek network: Erdos collaboration network 971	472	1314
SN_VI	Pajek network: Erdos collaboration network 991	492	1417
SN_VII	Pajek network: World Soccer, Paris 1998	35	295
SN_VIII	Pajek network: graph and digraph glossary	72	118
SN_IX	Pajek network: Slovenian journals 1999–2000	124	823168
SN_X	Co-authoship of scientists in network theory and experiments	1589	1190
SN_XI	Pajek network: SmaGri citation network	1059	4919
SN_XII	Email interchange network, Univ. of Rovira i Virgili,Tarragona	1133	5451
DC_I	DIMACS graphs: C250.9	250	27984
DC_II	DIMACS graphs: gen200_p0.9_44	200	17910
DC_III	DIMACS graphs: keller4	171	9435
DC_IV	DIMACS graphs: MANN_a27	378	70551
DC_V	DIMACS graphs: p_hat700-2	700	121728
DC_VI	DIMACS graphs: brock400_2	400	59786
DC_VII	DIMACS graphs: C1000_9	1000	450079
DC_VIII	DIMACS graphs: hamming10_4	125	6963

Topologies of some social networks and DIMACS are presented in Fig. 3 as the most popular of the social network datasets.

The setting of parameters for experiment is listed in below. It is noted that choosing different values for improving the results is also possible.

In this chapter has used 30 ants with $\rho = 0.95, \Phi = 0.0002, \Delta\tau_{initial} = 0, \tau_{\min} = 0.01, \tau_{\max} = 6$.

Meanwhile, parameters of PSO were initialized as $V = 0$, $c_1 = c_3 = 0.3$, and $c_2 = 1 - c_1$. α and ρ were given the following values based on the experiments done and t is the number of iterations.

$$\alpha(t) = \begin{Bmatrix} 1 & t \le 100 \\ 2 & 100 < t \le 400 \\ 3 & 400 < t \le 800 \\ 4 & t > 800 \end{Bmatrix} \tag{9}$$

$$\rho(\text{t+1}) = \begin{cases} (1-\varphi)\rho(\text{t}) & \\ 0.95 & \text{if } \rho(\text{t}) > 0.95 \end{cases} \tag{10}$$

Average of 10 independent runs with 1,000 iterations for each Social network dataset implementation have been listed in Table 2, and 30 independent runs with 1,500 iteration for DIMACS have been presented in Table 3 for proposed method

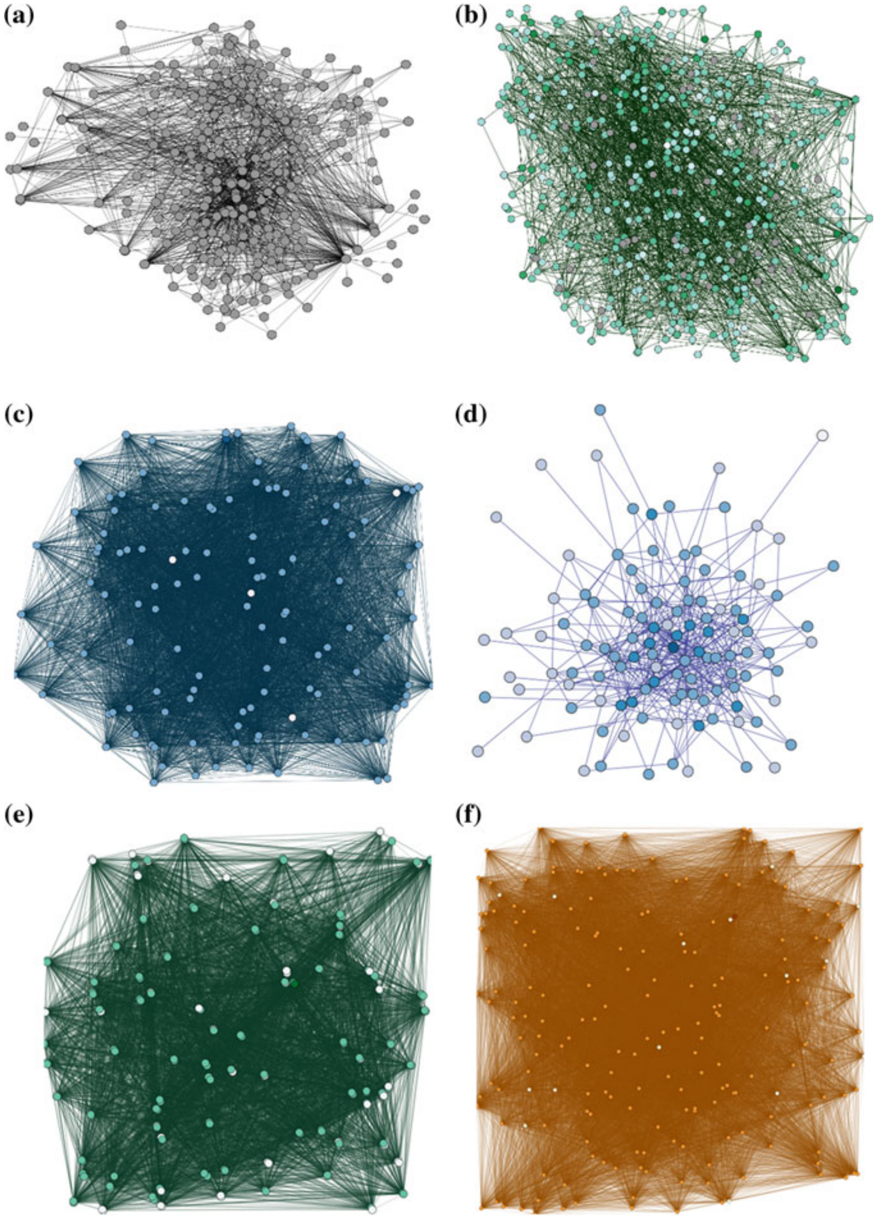

Fig. 3 Visulization of some datasets for experiments. **a** SN_III. **b** SN_VI. **c** SN_IX. **d** SN_II. **e** DC_III. **f** DC_II

Table 2 Simulation results for finding a maximum clique in social network datasets

Graph	ACO				ACO-PSO			
	Best	Avg	Std	Run-time	Best	Avg	Std	Run-time
SN_I	5	4.991	0.080	64.03	5	4.995	0.070	11.84
SN_II	5	4.709	0.495	355.81	5	4.783	0.412	23.37
SN_III	7	5.521	1.568	1121.71	7	5.543	1.482	49.47
SN_IV	5	4.991	0.094	132.12	5	4.998	0.044	15.69
SN_V	7	8.753	0.861	5853.59	7	5.848	0.667	105.11
SN_VI	7	5.961	0.924	6281.35	7	6.011	0.819	111.89
SN_VII	5	3.910	0.293	55.15	5	4.118	0.322	10.742
SN_VIII	4	3.943	0.232	153.66	4	3.985	0.121	12.94
SN_IX	4	3.017	0.316	438.87	4	3.185	0.299	22.07
SN_X	3	2.175	0.434	2142.03	3	2.253	0.371	118.59
SN_XI	8	6.025	0.815	9153.65	8	6.409	0.765	481.34
SN_XII	12	7.562	2.370	11109.17	12	7.997	2.240	529.04

Table 3 Simulation results for finding a maximum clique in DIMACS datasets

Graph	ACO				ACO-PSO			
	Best	Avg	Std	Run-time	Best	Avg	Std	Run-time
DC_I	44	43.20	0.761	43343.98	44	43.83	0.461	27978.04
DC_II	44	41.50	2.270	23992.76	44	42.90	1.863	19485.80
DC_III	11	11.00	0.000	8294.87	11	11.00	0.000	8417.40
DC_IV	126	125.60	0.498	194254.88	126	126.00	0.000	200678.54
DC_V	44	43.60	0.621	89776.41	44	44.00	0.000	89539.63
DC_VI	25	23.53	0.628	24730.51	29	24.30	1.087	29580.20
DC_VII	63	59.86	1.569	265326.84	67	64.06	1.172	276632.40
DC_VIII	40	36.30	1.556	151290.21	40	38.60	1.404	177169.91

(ACO-PSO) and ACO algorithm respectively, including the maximum (Best), average (Avg), standard deviation (Std) and run-time of algorithm for finding a maximum clique in each graph datasets.

Tables 2 and 3 indicate that the proposed method (ACO-PSO) produces better results in comparison with ACO method since the proposed approach is an appropriate method to update pheromones of the traversed paths for ants in calculating the optimum clique.

5.2 Comparison of Convergence of the Proposed Algorithm

Convergence behavior of the proposed algorithm in comparison standard ACO have been presented along running for some graphs. Figure 4 shows an average of 30

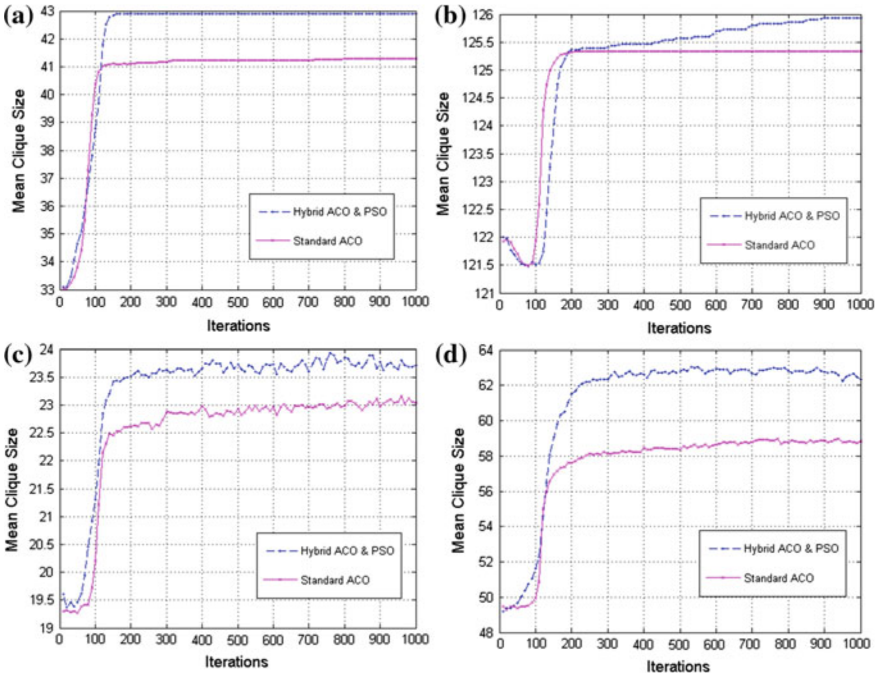

Fig. 4 Convergence behavior of the proposed algorithm in comparison standard ACO. **a** DC_II. **b** DC_IV. **c** DC_VI. **d** DC_VII

independent runs for 1,000 periods within intervals of 10 periods for some network graphs listed in table 1.

In these figures can be observed that proposed algorithm has excellent result compared with standard ACO.

6 Conclusion

A new hybrid algorithm has been presented in this paper using ACO and PSO (ACO-PSO) for finding the maximum clique problem. Traditional algorithms suffered high complexity while the hybrid proposed algorithm just change the process of update pheromone. It has been shown that the new algorithm was able to improve the basic ACO algorithm as simply and quickly. Simulation results on popular social network datasets indicate the suitable results for proposed algorithm in comparison with the ACO algorithm.

References

1. Easley D, Kleinberg J (2010) Networks, crowds, and markets: reasoning about a highly connected world. Cambridge University Press, Cambridge
2. Soleimani-Pouri M, Rezvanian A, Meybodi MR (2012) Solving maximum clique problem in stochastic graphs using learning automata. In: 2012 Fourth international conference on computational aspects of social networks (CASoN), pp 115–119
3. Sadi S, Öğüdücü S, Uyar A (2010) An efficient community detection method using parallel clique-finding ants. In: 2010 IEEE congress on evolutionary computation (CEC), pp 1–7
4. Fadigas IS, Pereira HBB (2013) A network approach based on cliques. Physica A 392(10):2576–2587
5. Yang Q, Zhou P, Zhang H, Zhang J (2011) Clique discovery based on user similarity for online shopping recommendation. Inf Technol J 10:1587–1593
6. Yan F, Cai S, Zhang M, Liu G, Deng Z (2012) A clique-superposition model for social networks. Science China Information Sciences, pp 1–19
7. Liua Y, Zhanga D, China FYL (2011) A clique-based algorithm for constructing feasible timetables. Optim Methods Softw 26(2):281–294
8. Butenko S, Wilhelm WE (2006) Clique-detection models in computational biochemistry and genomics. Eur J Oper Res 173(1):1–17
9. Liu J, Lee YT (2001) Graph-based method for face identification from a single 2D line drawing. IEEE Trans Pattern Anal Mach Intell 23(10):1106–1119
10. Wang L, Wei R, Lin Y, Wang B (2010) A clique base node scheduling method for wireless sensor networks. J Netw Comput Appl 33(4):383–396
11. Kleinberg J, Tardos E (2005) Algorithm design. Addison Wesley
12. Karp RM (1972) Reducibility among combinatorial problems. Complex Comput Comput 40(4):85–103
13. Fenet S, Solnon C (2003) Searching for maximum cliques with ant colony optimization. In: Cagnoni S, Johnson CG, Cardalda JJR, Marchiori E, Corne DW, Meyer J, Gottlieb J, Middendorf M, Guillot G, Raidl R, Hart E (eds) EvoWorkshops. LNCS, vol 2611. Springer, Heidelberg, pp 236–245
14. Xu X, Ma J, Lei J (2007) An improved ant colony optimization for the maximum clique problem. In: 3rd International conference on natural computation (ICNC), pp 766–770
15. Al-Fayoumi M, Banerjee S, Mahanti PK (2009) Analysis of social network using clever ant colony metaphor. World Acad Sci Eng Technol 53:970–974
16. Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence. Oxford University Press, Oxford
17. Socha K, Dorigo M (2008) Ant colony optimization for continuous domains. Eur J Oper Res 185(3):1155–1173
18. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: IEEE international conference on neural networks, pp 1942–1948
19. Amiri F, Yazdani N, Faili H, Rezvanian A (2012) A novel community detection algorithm for privacy preservation in social networks. In: Abraham A, Thampi SM (eds) ISI 2012. AISC, vol 182. Springer, Heidelberg, pp 443–450
20. Nabizadeh S, Faez K, Tavassoli S, Rezvanian A (2010) A novel method for multi-level image thresholding using particle swarm Optimization algorithms. In: 2010 2nd International conference on computer engineering and technology (ICCET), pp v4-271–275
21. Nabizadeh S, Rezvanian A, Meybodi MR (2012) A multi-swarm cellular PSO based on clonal selection algorithm in dynamic environments. In: 2012 International conference on informatics, electronics and vision (ICIEV), pp 482–486
22. Nabizadeh S, Rezvanian A, Meybodi MR (2012) Tracking extrema in dynamic environment using multi-swarm cellular pso with local search. Int J Electron Inform 1(1):29–37