Discrete Group Search Optimizer for Community Detection in Social Networks

Moustafa Mahmoud Ahmed[2,3(✉)], Mohamed M. Elwakil[1,4], Aboul Ella Hassanien[1,3], and Ehab Hassanien[1]

[1] Faculty of Computers and Information, Cairo University, Giza, Egypt
{m.elwakil,e.ezat}@fci-cu.edu.eg, aboitcairo@gmail.com
[2] Faculty of Computer and Information, Minia University, Minya, Egypt
moustafa.ali@mu.edu.eg
[3] Scientific Research Group in Egypt (SRGE), Cairo, Egypt
[4] Software Engineering Lab, Innopolis University, Innopolis, Russia
m.elwakil@innopolis.ru
http://www.egyptscience.net

Abstract. Discovering community structure in complex networks has been intensively investigated in recent years. Community detection can be treated as an optimization problem in which an objective fitness function is optimized. Intuitively, the objective fitness function captures the subgraphs in the network that has densely connected nodes with sparse connections between subgraphs. In this paper, we propose Discrete Group Search Optimizer (DGSO) which is an efficient optimization algorithm to solve the community detection problem without any prior knowledge about the number of communities. The proposed DGSO algorithm adopts the locus-based adjacency representation and several discrete operators. Experiments in real life networks show the capability of the proposed algorithm to successfully detect the structure hidden within complex networks compared with other high performance algorithms in the literature.

Keywords: Social network · Community detection · Complex network · Unsupervised learning · Group search optimizer

1 Introduction

Discovering communities hidden within the structure of complex networks has a significant practical importance for many fields such as sociology, physics, and biology. Community detection in networks can be defined as dividing a network into a set of internally densely connected groups of nodes, that has sparse connections in-between. Over the last few years, the problem of community detection has received a lot of attention and many different approaches have been proposed in different fields of research: computer science, physics, sociology, and others. Results of a recent survey can be seen in [1].

Recently, He et al. proposed a swarm intelligence optimization algorithm, called group search optimizer (GSO) [2]. This algorithm mimics the searching

V. Flores et al. (Eds.): IJCRS 2016, LNAI 9920, pp. 439–448, 2016.
DOI: 10.1007/978-3-319-47160-0_40

behavior of animals. Considering the efficiency of GSO algorithm, we propose to extend it into a discrete group search optimizer (DGSO) algorithm for the community detection problem. We employ the optimization mechanism of the basic GSO algorithm with two modifications. First, we avoid the angle evolution strategy. Second, we propose new evolution operations in the producer, scrounger, and ranger phases. Experiments on real life networks show the ability of the DGSO algorithm to correctly detect communities with results comparable to the state-of the-art approaches.

The rest of the paper is organized as follows. In Sect. 2, we define the community detection problem and introduce the objective functions adapted in this paper as well as we describe the basic GSO algorithm. In Sect. 3, we describe our proposed algorithm. In Sect. 4, the results of the method on synthetic and real life networks are presented and discussed. In Sect. 5, we give concluding remarks.

2 Preliminaries

In this section, we will provide a brief background on the community detection problem and optimization problem, and the group search optimizer algorithm.

2.1 The Community Detection Problem

A network can be defined as a graph $G = (V, E)$, in which V is the set of nodes, and E is a set of ties that connect nodes. In the field of social networks, nodes represent persons or actors within the network, and ties represent the relationships or the interaction between those persons. A community structure S in a network is a set of groups of nodes such that each group is densely connected internally and sparsely connected with other groups. So this problem can be defined as dividing network's nodes into k disjoint communities, where the number k is unknown, that best satisfy a given quality measure of communities $F(S)$. Thus, we treated this problem as an optimization problem in which one usually wants to optimize the given quality measure $F(S)$. A single objective optimization problem $(\Omega; F)$ is formulated as in the Eq. 1.

$$min\ f(S),\ s.t\ S \in \Omega \tag{1}$$

Where $F(S)$ is an objective function that needs to be optimized, and $\Omega = \{S_1, S_2, .., S_r\}$ is the set of feasible community structures in a network.

2.2 Group Search Optimizer

The *GSO* algorithm was proposed by He [2]. This algorithm simulates animal searching (foraging) behavior. The basic variant of the *GSO* algorithm works by having a population (called a group) of candidate solutions (called members). Each member in the group has its own position, search angle, and search direction. In the *GSO* algorithm, a group contains three kinds of members:

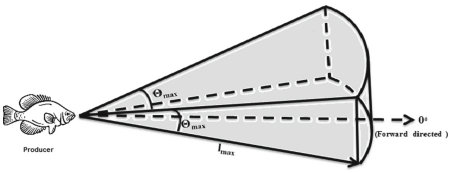


Fig. 1. Scanning field in 3-D space.

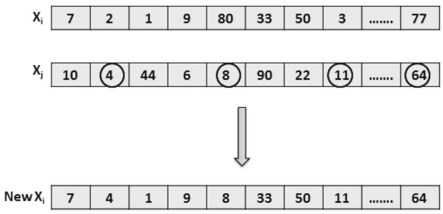


Fig. 2. The movement from x_i to x_j with Step =4.

producers and scroungers whose behaviors are based on the *producer-scrounger* (PS) model [3], and rangers who perform random transitions in the search space. At each iteration, the producers perform producing strategy to search for the positions containing the best resources. The producer's scanning field of vision is generalized to an n-dimensional space, which is specified by maximum pursuit angle $\theta_{max} \in R^1$ and maximum pursuit distance $l_{max} \in R^1$ as illustrated in a 3D space [4] in Fig. 1. The scroungers perform a following strategy to join resources found by the producers: the remaining members are the rangers that walk randomly in the searching space to stay in new positions. In the *GSO* algorithm, a position of the individual represents a solution of the optimization problem, and the fitness of the position represents the fitness of the solution. The basic *GSO* algorithm is discussed in [4].

3 The Proposed Discrete Group Search Optimizer *DGSO* for the Community Detection Problem

Owing to the continuous nature of the *GSO* algorithm, this algorithm doesn't directly fit for the community detection problem. So it's necessary to develop a suitable mapping which can efficiently convert individuals to solutions. In this paper we propose a discrete version of the *GSO* algorithm for the community detection problem. A detailed description of the proposed algorithm is introduced below.

3.1 Individual Representation

The DGSO algorithm used locus-based adjacency representation proposed in [5] to encode group members, a detailed description of this representation strategy can be found in [6]. To detect community structure, a decoding step is necessary to discover connected components. Each of these components corresponds to community, So the number of these components equals the number of communities in the discovered structure. Thus, there is no need to know in advance the number of communities.

3.2 Initialization

Randomly initializing group members could generates components that are disconnected in the original network, for example, gene g_i could be assigned to value j, but no connection between nodes i and j exists in the original network, this means that assigning both nodes i and j to the same group is a wrong choice. In order to avoid such this case, we proposed to use the initialization process proposed in [7] (safe initialization), which takes in account the effective connections of nodes in the social network. Using safe initialization such this case is avoided by substituting value j with one of the neighbors of i.

3.3 Producer

Group members that obtain the best fitness values are chosen as the producers. A producer tries to guide other group members to the food sources (optima). In nature, animals use vision or other senses, to realize the concentration of food in the environment, to determine the direction of the next movement. In our algorithm, the scanning field of vision is simplified and limited by maximum pursuit distance l_{max}, which is a selected constant number $\in [0, 1]$. In our algorithm, the producer behaves as follows:

1. A producer scans the search space by randomly selecting three points in the scanning field, let x_p is the producer's current state and x_1, x_2, x_3 are the randomly selected states in the x_p's visual, where distance $(x_p, x_i) < l_{max}$ and $i \in \{1, 2, 3\}$.
2. Then, the producer selects the fittest point with the best resource. If this point has a better resource than producer's current position, then it will move a step to this point $Move(x_p, x_i)$. Otherwise it will stay in its current position.

Distance: Since there is no straightforward method to measure distance between two group members (solutions), we adopted the distance measure proposed in [8]. This measure uses Normalized Mutual Information (NMI) [9] to valuate the degree to which two solutions are close to each other as calculated in Eq. 2. NMI is a similarity measure proved to be robust and accurate by Danon et al. [9].

$$dis(x_i, x_j) = 1 - NMI(C(x_i), C(x_j)) \tag{2}$$

where $C(x)$ is the decode functions used to interpret group member state back to a community structure and $NMI(C(x_i), C(x_j))$ calculates the NMI similarity between the two community structures x_i and x_j.

Step: Represents the number of nodes copied from a solution x_p to a solution x_i to move a solution x_i a step in the direction of a solution x_p as illustrated in Fig. 2, where $Step \in [1..n]$.

Movement: We use a crossover operator used previously in genetic algorithms [6] where the mixing ratio is the step size of the move. So in order to move a group member x_i to group member x_j $Move(x_i, x_j)$, the two group member in the crossover operator are considered as the parents of the new offspring

(new member state). The new group member state has randomly chosen Steps optimizing variables from x_j and the rest are from x_i as illustrated in Fig. 2.

Recently, Couzin et al. [10] found that, for large groups, only a very small proportion of informed individuals is needed to guide the group to achieve a high accuracy. So, for accuracy and simplicity, there is only one producer in the *DGSO* algorithm, which means that the best member is the producer and the remaining members in the group are scroungers or rangers.

3.4 Scrounger

After selecting members that will perform producing behavior, the remaining members are distributed into scroungers and rangers, with the probability of P and $(1-P)$, respectively. The scroungers will continue searching for opportunities to join the resources found by the producer. In our algorithm each scrounger x_s randomly selects a producer x_p to move a step towards $Move(x_s, x_p)$.

3.5 Ranger

Rangers are the group members that randomly search in the search space, seeking to find other promising solutions that are yet to be refined. The purpose of this operation is to diverse the search in order to avoid getting trapped in a local optimum. Here each ranger x_r randomly selects a point x_i in the total search space to move a step towards $Move(x_r, x_i)$. If the rangers cannot find a better area after A iterations, a percent RP of the rangers are randomly selected to be mutated with a mutation rate MR, $Mutate(x_r)$. Regardless of whether the movement or the mutation process leads rangers to a better position (fitness value) than the original one, the rangers will do enhance the global search ability.

Mutation: Randomly changing values of a randomly chosen member's genes might causes a useless exploration of the search space. So, as in the initialization step, we propose to randomly select a percentage of the genes and for each selected gene i we randomly change its value to j such that node i and j are neighbors.

Similar to the course of evolution, once the fitness of new member generated by scroungers or rangers is better than the fitness of the producer, the producer will be updated.

3.6 Fitness Function

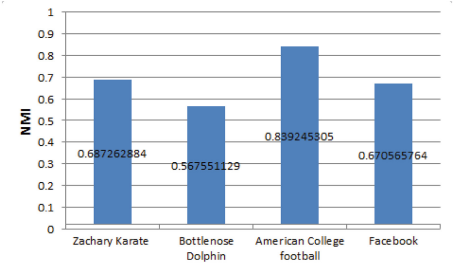
We decided to use Modularity [11], which is an effective quality function, to quantify and measure how "good" the discovered community structure is. Studies in the literature proved that modularity is effective in many kinds of complex networks [11].

The pseudo code of the DGSO algorithm processes are shown in Algorithm 1.

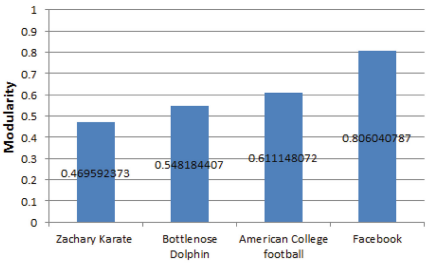
Data: A Network G =(V, E)
Result: Community membership assignment for each node in the network G

initialization Population size *popsize*, Randomly initialize group members, Maximum pursuit distance l_{max}, *Step*, Scrounging percent P, Mutation percent MP, Mutation rate MR, Ranging trials A, Maximum number of iterations $Max\_Iterations$
Calculate the fitness values of initial group members.
while *(Iteration number* $\leq Max\_Iterations$*)* **do**
    `/* Perform producing. */`
    Find the producer x_p of the group(the fittest member).
    The producer randomly sample three points in the scanning field using (2).
    The fittest point with the best resource is chosen. If this point has a better resource than producer's current position, then it will move a *Step* to this point. Otherwise it will stay in its current position.
    `/* Perform scrounging. */`
    Randomly select P percent from the rest of the members to perform scrounging, by moving a *Step* to words the producer.
    `/* Perform ranging. */`
    The remainder members leave their current position to perform ranging, by randomly selecting a point x_i in the total search space to move a step towards.
    If the rangers can not find a better area after A iterations, a percent RP of the rangers are randomly selected to be mutated with a mutation rate MR.
end

Algorithm 1. DGSO Algorithm.



(a) NMI values.



(b) Modularity values.

Fig. 3. Average NMI and Modularity values of the result community structure on each social network.

4 Experimental Results and Discussion

We tested our algorithm on four real life social networks: The Zachary Karate Club [12], The Bottlenose Dolphin network [13], American College football network [14], and Facebook Dataset [15]. The ground truth communities partitions for these networks are known. To compare the accuracy of the resulting community structures, we used Normalized Mutual Information (NMI) [9] to calculate the similarity between the true community structures and the discovered ones.

For each dataset, we applied the algorithm ten times. In each trial we calculated the NMI and Modularity values of the best solution. Then, we calculated the average NMI and average Modularity over the ten trials. The *DGSO* algorithm was applied with the following parameters values; $L_{max} = 0.8$, $Step = 0.2 * n$, population size $popsize = 200$, Scrounging percent $P = 80\,\%$ of popsize (Ranging percent= $20\,\%$ of popsize), the maximum number of iterations $Max\_Iterations = 200$, and the number of ranging trials $A = 5$. Figure 3a and b show the average NMI value and the average Modularity values, respectively, for the community structures detected in each dataset. We can observe that our algorithm achieves high NMI values for all social networks. The Modularity value of the community structure detected by our algorithm is higher than the corresponding Modularity value of the ground truth division of those networks as shown in Fig. 7a. This means that, according to Modularity measure, our algorithm detects more modular community structures than the original ones.

To understand the results produced by the algorithm we visualized the community structure detected on the small size dataset. Figure 4 shows a visualization of the discovered structure for the Zachary network. The original structure of the network is indicated by the black thick line and the structure detected by our algorithm is indicated by nodes'colors. From this figure we can observe in the top level the result is similar to the original division of the network, however in the result structure each group is farther subdivided into two groups.

Figure 5 visualizes the result for the Dolphin network. The original structure of the network is indicated by the black thick line and the detected structure is indicated by nodes'colors. From this figure we can observe in the top level the result is similar to the original division of the network, however in the result structure, the right group is further subdivided into four groups.

Figure 6 visualizes the result obtained for the College football network. The original division of the network is visualized in Fig. 6a; where nodes' labels refer to the groups they assigned to. From Fig. 6a; we can observe that some groups such as 5,10 are sparsely connected internally, however they densely connected with other groups. This problem disappears in the community structure detected by our algorithm. From Fig. 6b; we can observe that our algorithm discovered a community structure with 9 groups which assigns nodes from the smaller groups, such as 5,10 into a larger groups leading to a more modular community structure.

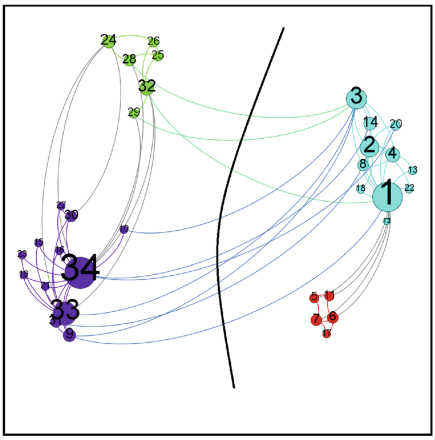


Fig. 4. Visualization of the result for the Zachary network.

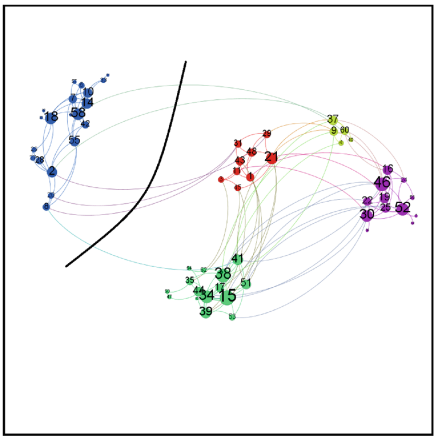
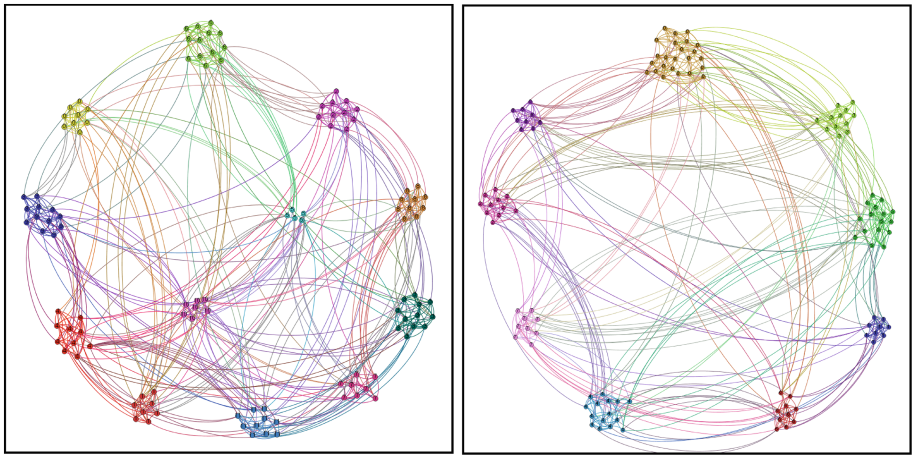


Fig. 5. Visualization of the result for the Dolphin network.

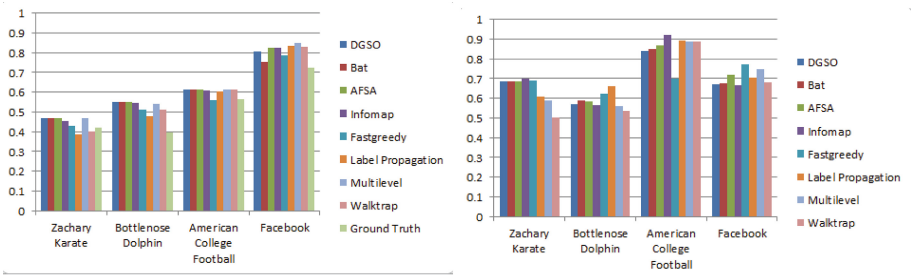


(a) Original Division. (b) Result Devision.

Fig. 6. Visualizations of the result for the American College football network.

4.1 Comparison Analysis

Here, we can show practical comparison between the results obtained by DGSO algorithm and other seven well-known methods proposed in the literature, which are Infomap [16], Fast greedy [17], Label propagation [18], Maulilevel [19], Walktrap [20], leading Eigenvector [11], and Artificial fish swarm algorithm [8]. We applied each method 10 times on each dataset and the average NMI and the average Modularity of the best community structure is reported. Figure 7 summarizes the NMI and Modularity values for all methods. In terms of Modularity,



(a) Modularity values.

(b) NMI values.

Fig. 7. NMI and Modularity values for each dataset reported by each method.

DGSO is very competitive with other methods as shown in Fig. 7a. For the small size datasets we can observe that DGSO detects a community structure with a high Modularity value compared to all other methods. Regarding the Facebook datasets, DGSO competes with the seven methods with a very small difference. In terms of NMI, DGSO produces results seems to be bad compared to other methods as shown in Fig. 7b. However we could return this to the different high modular community structures our algorithm produced compared to the ground truth divisions.

5 Conclusion and Future Work

DGSO is an optimization technique that suits the community detection problem. Experiments with real world networks showed the ability of this method to correctly detect community structures based on the quality function used (modularity). DGSO has the advantage that, number of communities is not required to be specified as a prior setting. A comparison with other recently proposed methods shows that DGSO is very competitive with such methods. Enhancing the capabilities of this algorithm to discover communities in multi-dimensional social networks is a necessary task that can be investigated in future work.

References

1. Khatoon, M., Banu, W.A.: A survey on community detection methods in social networks. Int. J. Educ. Manage. Eng. (IJEME) **5**(1), 8 (2015)
2. He, S., Wu, Q., Saunders, J.: A novel group search optimizer inspired by animal behavioural ecology. In: Evolutionary Computation, 2006, CEC 2006, IEEE Congress, pp. 1272–1278. IEEE (2006)
3. Barnard, C.J., Sibly, R.M.: Producers and scroungers: a general model and its application to captive flocks of house sparrows. Anim. Behav. **29**(2), 543–550 (1981)
4. He, S., Wu, Q.H., Saunders, J.: Group search optimizer: an optimization algorithm inspired by animal searching behavior. IEEE Trans. Evol. Comput. **13**(5), 973–990 (2009)

5. Park, Y., Song, M.: A genetic algorithm for clustering problems. In: Proceedings of the Third Annual Conference on Genetic Programming, pp. 568–575 (1998)
6. Ahmed, M.M., Hafez, A.I., Elwakil, M.M., Hassanien, A.E., Hassanien, E.: A multi-objective genetic algorithm for community detection in multidimensional social network. In: Gaber, T., Hassanien, A.E., El-Bendary, N., Dey, N. (eds.) The 1st International Conference on Advanced Intelligent System and Informatics (AISI), November 28–30, 2015, Beni Suef, Egypt. AISC, pp. 129–139. Springer, Heidelberg (2016)
7. Pizzuti, C.: GA-Net: a genetic algorithm for community detection in social networks. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 1081–1090. Springer, Heidelberg (2008)
8. Hassan, E.A., Hafez, A.I., Hassanien, A.E., Fahmy, A.A.: Community detection algorithm based on artificial fish swarm optimization. In: Filev, D., et al. (eds.) Intelligent Systems'2014. AISC, vol. 323, pp. 509–521. Springer, Heidelberg (2015). doi:10.1007/978-3-319-11310-4_44
9. Danon, L., Diaz-Guilera, A., Duch, J., Arenas, A.: Comparing community structure identification. J. Stat. Mech. Theory Exp. **2005**(9), P09008 (2005)
10. Couzin, I.D., Krause, J., Franks, N.R., Levin, S.A.: Effective leadership and decision-making in animal groups on the move. Nature **433**(7025), 513–516 (2005)
11. Newman, M.E.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**(3), 036104 (2006)
12. Zachary, W.W.: An information flow model for conflict and fission in small groups. J. Anthropol. Res. **33**, 452–473 (1977)
13. Lusseau, D.: The emergent properties of a dolphin social network. Proc. R. Soc. Lond. B Biol. Sci. **270**(Suppl 2), S186–S188 (2003)
14. Girvan, M., Newman, M.E.: Community structure in social and biological networks. Proc. Nat. Acad. Sci. **99**(12), 7821–7826 (2002)
15. Leskovec, J., Lang, K.J., Mahoney, M.: Empirical comparison of algorithms for network community detection. In: Proceedings of the 19th International Conference on World Wide Web, pp. 631–640. ACM (2010)
16. Rosvall, M., Axelsson, D., Bergstrom, C.T.: The map equation. Eur. Phys. J. Spec. Top. **178**(1), 13–23 (2010)
17. Clauset, A., Newman, M.E., Moore, C.: Finding community structure in very large networks. Phys. Rev. E **70**(6), 66–111 (2004)
18. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**(3), 036106 (2007)
19. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exp. **2008**(10), P10008 (2008)
20. Pons, P., Latapy, M.: Computing communities in large networks using random walks. In: Yolum, I., Güngör, T., Gürgen, F., Özturan, C. (eds.) ISCIS 2005. LNCS, vol. 3733, pp. 284–293. Springer, Heidelberg (2005)