Applied Soft Computing xxx (2011) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Multi-objective community detection in complex networks

Chuan Shi [a,*], Zhenyu Yan [b], Yanan Cai [a], Bin Wu [b]

[a] Beijing University of Posts and Telecommunications, Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing 100876, China
[b] Research Department, Fair Isaac Corporation (FICO), San Rafael, CA 94903, USA

ARTICLE INFO

Article history:
Received 17 February 2010
Received in revised form 28 June 2011
Accepted 23 October 2011
Available online xxx

Keywords:
Community detection
Complex network
Evolutionary multi-objective algorithm
Modularity

ABSTRACT

Community detection in social network analysis is usually considered as a single objective optimization problem, in which different heuristics or approximate algorithms are employed to optimize a objective function that capture the notion of community. Due to the inadequacy of those single-objective solutions, this paper first formulates a multi-objective framework for community detection and proposes a multi-objective evolutionary algorithm for finding efficient solutions under the framework. After analyzing and comparing a variety of objective functions that have been used or can potentially be used for community detection, this paper exploits the concept of correlation between objective which charcterizes the relationship between any two objective functions. Through extensive experiments on both artifical and real networks, this paper demonstrates that a combination of two negatively correlated objectives under the multi-objective framework usually leads to remarkably better performance compared with either of the orignal single objectives, including even many popular algorithms..

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

In recent years, community detection in complex networks has attracted a lot of attention. The main reason is that communities are supposed to play special roles in the structure-function relationship, and thus detecting communities (or modules) can be a way to identify substructures which could correspond to important functions. For example, the communities in WWW are sets of web pages sharing the same topic [12]; the modular structure in biological networks are widely believed to play important roles in biological functions [11].

Loosely speaking, communities are groups of nodes that are densely interconnected but only sparely connected with the rest of the network [3,14]. In order to extract such groups of nodes, one typically chooses an objective function that captures the intuition of a community as a group of nodes with better internal connectivity than external connectivity. The objective is usually *NP*-hard to optimize, so one usually employs heuristics (e.g., betweenness [21]) or approximation algorithms (e.g., spectral method [25], genetic algorithm [23,28]) to find sets of nodes that approximately optimize the objective function and can be understood as communities. As a consequence, the conventional community detection can be considered as a single-objective optimization problem (SOP). That is, the community detection corresponds to discover a community structure that is optimal on one single-objective function. Most of conventional community detection algorithms are based on this SOP. Different algorithms have variations of the evaluation function and optimization method. These single-objective approaches have been successfully applied to both artificial and real problems. However, the single-objective community detection also faces some fundamental difficulties. These single-objective algorithms usually confine the solution to a particular community structure property because of only considering one objective function. When the optimization objective is inappropriate, these algorithms may fail, such as resolution limit problem [13]. In addition, one single fixed community partition returned by the single-objective algorithms may not be suitable for the networks with multiple potential structures (e.g., hierarchical and overlapping structures).

In this paper, we first formulate the community detection as a multi-objective optimization problem (MOP). That is, the community detection corresponds to discovering community structures that are optimal on multiple objective functions, instead of one single-objective function in the single-objective community detection. In order to effectively solve this problem, we propose a novel multi-objective community detection algorithm (called MOCD). The MOCD includes two phases. In the first community detection phase, MOCD simultaneously optimizes two conflicting objective functions with evolutionary algorithm (EA) and returns a set of solutions which are optimal in terms of optimization objectives. In order to assist decision makers (DMers) in selecting proper community partitions, in the second model selection phase, two model selection methods are proposed to select one recommendation solution from the solution set returned by the first phase. We perform extensive experiments to validate the advantages of

* Corresponding author.
E-mail address: shichuan@bupt.edu.cn (C. Shi).

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.10.005

the multi-objective community detection. Compared to one single solution returned by conventional single-objective community detection algorithms, MOCD returns a set of solutions. These solutions not only can reveal the implicit structure information (e.g., hierarchical and overlapping) from different perspectives but also can provide DMers with the flexibility to select the proper solution based on other subjective information. Moreover, the solution recommended by MOCD with the proposed model selection methods are more accurate than those generated by well-established community detection algorithms.

This paper is arranged as follows. Section 2 formally proposes the multi-objective community detection problem. Section 3 describes the MOCD in detail. Then the extensive experiments are done to validate the effectiveness of MOCD in Section 4. We compare MOCD with those most related work in Section 5. Finally, Section 6 concludes the paper.

2. Multi-objective community detection

Conventional community detection algorithms can be roughly classified into two categories. (1) The first type of algorithms convert community detection into an optimization problem, and detect the community structure through optimizing an evaluation criterion. For example, the spectral method [25] formulates community detection as a quadratic optimization and optimizes a "cut" function [16]. The modularity optimization methods employ the optimization technologies (e.g., genetic algorithm [23,28,29], simulated annealing [14] and extremal optimization [10]) to optimize a criterion, such as Q [21]. (2) The second type of algorithms convert community detection into the design of heuristic rules, such as the edge betweenness [21] and link clustering coefficient [26]. These heuristic methods usually need a criterion to stop the iteration process. For example, the maximum modularity Q is used as the stopping criterion in GN [21]. In both types of approaches, the community detection can be regarded as a single-objective optimization problem (Ω, P).

Definition 1. Single-objective community detection. It determines a partition C^* for which

$$P(C^*) = \min_{C\in\Omega} P(C) \tag{2.1}$$

where Ω is the set of feasible partitions, C is a community structure of a given network G and $P: \Omega \to R$ is a measure function. Without loss of generality, we assume P is to be minimized. Most of conventional community detection algorithms are based on this single-objective optimization problem. Different algorithms vary in the optimization criterion P and optimization technique, such as the modularity Q in GN [21], the "cut" function in spectral method [16] and the "description length" in the information-theoretic based method [18].

The single-objective community detection algorithms have been widely applied. However, they also have some crucial disadvantages. (1) These single-objective algorithms attempt to optimize just one of such criteria and this confines the solution to a particular community structure property. And thus it often causes a fundamental discrepancies that different algorithms may produce distinct solutions on the same networks. (2) The single-objective algorithms may fail when the optimization objectives are inappropriate. An example is the resolution limit problem existing in the modularity optimization: the modularity optimization may fail to identify modules smaller than a scale which is determined by the size of the network and the degree of interconnectedness of the modules, even in cases where modules are unambiguously defined [13]. Further researches also reveal that the optimization on other single-objective has similar resolution limit [17]. (3) Many single-objective algorithms require some priori information, such as the number of communities which is usually unknown for real networks. (4) A single fixed community partition returned by single-objective algorithms may not be suitable for the networks with multiple potential structures (e.g., hierarchical or overlapping). Taking the hierarchical network for example, a fixed community partition cannot reveal the hierarchical structure.

Although the difficulty in selecting an appropriate criterion in single-objective community detection can be alleviated through the application of a criterion with a tuning parameter (e.g., RB [27] and AFG [2]), a more natural approach may be to consider the community detection as a multi-objective optimization problem ($\Omega, P_1, P_2, \ldots, P_m$), which can defined as follows.

Definition 2. Multi-objective community detection. It determines partitions C^* for which

$$P(C^*) = \min_{C\in\Omega}(P_1(C), P_2(C), \ldots, P_m(C)) \tag{2.2}$$

where m is the number of criteria and P_i represents the ith criterion. With the introduce of multiple criteria, there is usually no single best solution for this optimization task, but instead, the notion of Pareto optimality should be embraced [8]. For two partitions $C_1, C_2 \in \Omega$, partition C_1 is said to dominate partition C_2 (denoted as $C_1 \preceq C_2$) if and only if

$$\begin{aligned} &\forall i \in \{1, \ldots, m\}\ P_i(C_1) \le P_i(C_2) \\ &\wedge \exists i \in \{1, \ldots, m\}\ P_i(C_1) < P_i(C_2) \end{aligned} \tag{2.3}$$

A partition $C \in \Omega$ is said to be Pareto optimal if and only if there is no other partition dominating C. The set of all Pareto optimal partitions is the Pareto optimal set and the corresponding set in the objective space is called the non-dominated set, or Pareto front. The objective function, $P_1, \ldots, P_m$, can be the existing criteria (e.g., the modularity Q [21], the "cut" function [16], and the "description length" [18]) or newly designed criteria.

Compared to the single-objective community detection, the multi-objective community detection has the following advantages.

- The Pareto optimal set obtained by the multi-objective community detection defined by $(\Omega, P_1, \ldots, P_m)$ always comprise the optimal solutions of the single-objective community detection defined by $(\Omega, P_1), \ldots, (\Omega, P_m)$. With the assumption that the single-objective and multi-objective community detection can respectively identify all globally optimal solutions and the entire Pareto optimal set, we can deduce that the multi-objective community detection can always find a partition, which is as good as or better than those of the single-objective community detection. In situations where the best partition corresponds to a tradeoff between different objectives, only the multi-objective community detection will be able to find it.
- The multiple objectives can measure characteristics of community structure from different perspectives, and thus avoid the risk that one single-objective may only be suitable to a certain kind of networks. Moreover, the multi-objective optimization process tradeoffs the balance of the multiple objectives, which can effectively avoid being trapped to local optima.
- The number of communities can be automatically determined during the multi-objective optimization process. The well-established optimization objectives have the potential to balance each other's tendency to increase or decrease the number of communities. The interaction of optimization objectives is able to keep the number of communities dynamic, avoiding the convergence to trivial solutions.
- The multi-objective community detection usually returns a set of community partitions according to the multiple optimization

Please cite this article in press as: C. Shi, et al., Multi-objective community detection in complex networks, Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.10.005

objectives. These community partitions reveal community structure from different angles, which help to discover complex and comprehensive community structures.

Some researchers have begun to be aware that enumerating the modules in a network is a tradeoff among multiple criteria. Fortunato and Barthelemy believed that finding the maximum modularity is to look for the ideal tradeoff between the number of modules and the value of each term [13]. Similarly, Brandes et al. proposed that the modularity Q is a combination of two conflicting components [4]. Rosvall and Bergstrom also thought that enumerating the modules in a network has an inevitable tradeoff between the amount of the structure information of a network and its description length [18]. Although these researchers have been aware of the intrinsic tradeoff, most algorithms still optimize one single-objective which often simply combines some conflicting components (e.g., the linear combination of the two components in modularity Q [21]). On the other hand, the research community in evolutionary computation has confirmed that evolutionary multi-objective optimization is a more intrinsic and effective approach to identify this kind of tradeoffs [8,15].

3. The MOCD algorithm

For the multi-objective community detection problem, both mathematical programming and heuristic approaches can be applied to solve it. This paper applies the Evolutionary Algorithm (EA), a type of heuristic approaches, to solve the problem. Compared to mathematical programming techniques, EA has many advantages [8], such as simultaneously generating a set of candidate solutions and easily dealing with the discontinuous and concave Pareto fronts. Thus, in recent years, there has been a growing effort in applying EA in multi-objective optimization, giving rise to many successful evolutionary multi-objective optimization (EMO) algorithms. Conventional EMO algorithms are designed for numerical optimization problems. When it is applied to the multi-objective community detection, many components of EA need to be redesigned. It is not a trivial task, because the design of these components directly determine the algorithm performance.

Concretely, the multi-objective community detection with EA faces following challenges: (1) Selection of optimization objectives. The objective functions should reflect the structural characteristics of communities from different aspects. Ideal objective functions had better contain intrinsic conflicts, such that the optimal community partitions can be obtained through the trade-off of multiple objectives. (2) Effective genetic representation. The genetic representation should be delicately designed according to the characteristics of community detection, since it decides the algorithm performance and scalability to a large extent. (3) Utilization of multiple community partitions. EMO methods usually return a set of solutions. How to make the best of those non-dominated solutions is a key issue to improve the algorithm performance.

This paper proposes a novel solution, multi-objective community detection algorithm (MOCD), which includes two phases: community detection and model selection. In the detection phase, MOCD simultaneously optimizes two conflicting yet complementary objectives and the locus-based adjacency schema is applied to effectively represent community partition. The output of this phase is a set of community partitions, which correspond to different tradeoffs among these two objectives and different numbers of communities. In the model selection phase, MOCD employs two methods to select the most preferable solution from the partition set, which provides an estimate of the qualities of the partition set and determines the recommendation solution. These elaborately designed components of MOCD effectively solve those challenges.

3.1. Community detection phase

In this phase, MOCD simultaneously optimizes multiple objectives with EA and returns a set of solutions.

3.1.1. Multi-objective optimization mechanism

A good EMO algorithm needs to generate a set of solutions that uniformly distributed along the Pareto front [30]. Many EMO algorithms have designed effective multi-objective optimization mechanism to realize the essence of EA: survival of the fittest. In this paper, we select an existing EMO algorithm, the Pareto Envelope-based Selection Algorithm version 2 (PESA-II) [6], as the multi-objective optimization mechanism of the MOCD. PESA-II follows the standard principles of an EA with the difference that two populations of solutions are maintained: an internal population (*IP*) of a fixed size, and an external population (*EP*). The *IP* explores new solutions through the standard EA process of reproduction and variation. The *EP* is to exploit good solutions by maintaining a large and diverse set of the non-dominated solutions discovered during search. Selection occurs at the interface between the two populations, primarily in the update of *EP*. The solutions in *EP* are stored in "niches", implemented as a hyper-grid in the objective space. A tally of the number of solutions that occupy each niche is kept and this is used to encourage solutions to cover the whole objective space, rather than bunch together in one region. The detailed implementation can be seen in Ref. [6]. There are five basic parameters in the algorithm and their meanings are illustrated here:

- *ipsize* and *epsize* are the size of *IP* and *EP*.
- p_c and p_m are the ratio of crossover and mutation.
- *gen* is the running generation.

3.1.2. Objective functions

Objective function (i.e., fitness function), which guides the search process, is one of the most important components in MOCD. The objective functions quantify the optimality of a solution, so we should select optimization objectives that reflect the fundamental aspects of a good community partition. Modularity is a foundational quality index for community detection. Given a graph $G=(V,E)$, we follow Ref. [21] and define the following equation:

$$Q(C)=\sum_{c\in C}\left[\frac{|E(c)|}{m}-\left(\frac{\sum_{v\in c}deg(v)}{2m}\right)^2\right] \tag{3.4}$$

where the sum is over the modules of the partition, $|E(c)|$ is the number of links inside module c, m is the total number of links in the network, C is a partition result, and $deg(v)$ is the degree of the node v in module c. Observing the equation, to maximize the modularity Q, we should maximize the first term ($\sum_{c\in C}(|E(c)|/m)$) and minimize the second term ($\sum_{c\in C}(\sum_{v\in c}deg(v)/2m)^2$). To maximize the first term, many edges should be contained in modules (i.e., "densely interconnected"). To minimize the second term, the graph should be split into many modules with small total degrees each (i.e., "sparely connected with the rest"). These two complementary terms reflect two fundamental aspects of a good partition, and the modularity Q is an intrinsic trade-off between these two objectives [4,13].

In this paper, we select these two terms as the objective functions. In order to formulate the problem as a minimum optimization problem, we revise the first term. The first objective function minimizes 1 minus the intra-link strength of a partition, and it is called *intra* objective.

$$intra(C)=1-\sum_{c\in C}\frac{|E(c)|}{m} \tag{3.5}$$

Please cite this article in press as: C. Shi, et al., Multi-objective community detection in complex networks, Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.10.005

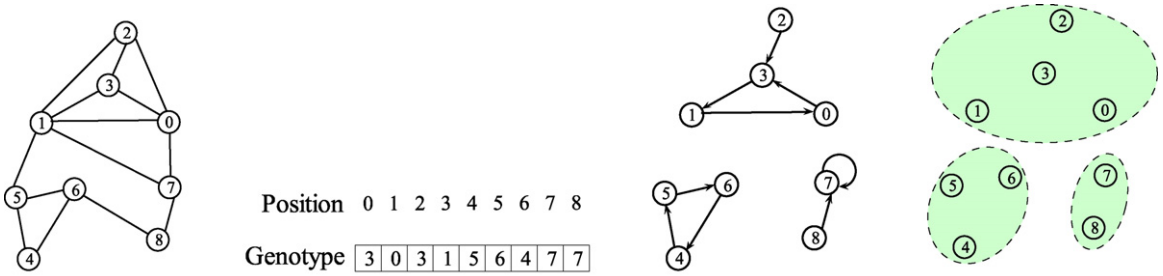


Fig. 1. Illustration of the locus-based adjacency genetic representation. (a) The topology of the graph representing a complex network. (b) One possible genotype. (c) How (b) is translated into the graph structure, for example node 0 links to node 3, since gene g_0 is 3. (d) The community structure.

The second objective function minimizes the inter-link strength of a partition, and it is called *inter* objective.

$$inter(C) = \sum_{c \in C} \left(\frac{\sum_{v \in c} deg(v)}{2m} \right)^2 \tag{3.6}$$

According to the two definitions, we can deduce that

$$Q(C) = 1 - intra(C) - inter(C) \tag{3.7}$$

The motivation of selecting two components of the modularity Q as the objective functions, rather than other criteria, are stated as follows. (1) These two functions have the potential to balance each other's tendency to increase or decrease the number of communities, which enables the use of a representation that does not fix the number of communities. With an increasing number of communities, the fewer edges fall in communities (i.e., $E(c)$ becomes smaller), and thus the *intra* objective value tends to increase. The opposite trend happens to the *inter* objective. The interaction of the two objective functions is crucially important in order to keep the number of communities dynamic, avoiding the convergence to trivial solutions. (2) Modularity is a widely used objective in community detection and many single-objective algorithms are based on the modularity optimization. It is easy and fair to compare MOCD optimizing over two components of the modularity with those prevalent single-objective algorithms. The differences on their performances are more likely caused by the multi-objective optimization. (3) After many experimentations, we find these two objective functions are more empirically suitable. The conflict between two objectives reflects and balances two distinct but important aspects of community structure. More objective functions will not necessarily lead to better solutions, but may result in some practical difficulties, such as larger search space and more candidate solutions.

3.1.3. Genetic representation and operators

When EA is applied to community detection, a community partition should usually be encoded in a character string (i.e., genotype) with a genetic representation, and inversely a genotype can also be decoded into a community partition. This paper employs the locus-based adjacency representation [22] as illustrated in Fig. 1. In this graph-based representation, each genotype g consists of n genes $g_1, g_2, \ldots, g_n$ and each g_i can take one of the adjacent nodes of node i (including node i itself). Thus, a value of j assigned to the ith gene, is then interpreted as a link between node i and j. In the resulting solution, they will be in the same community. The decoding of this representation requires the identification of all connected components. All nodes belonging to the same connected component are then assigned to one community. Using a simple backtracking scheme, this decoding step can be performed in a linear time [5,28].

Since the encoding schema cannot represent all partitions of nodes (e.g., a community including disconnected sub-graphs), someone may doubt that the solutions with a good community structure may be not in the solution space constructed by the genetic representation. Fortunately, Brandes et al. have analyzed the basic structural properties of the clustering with maximum modularity and proposed the following theoretical results [4].

Property 1. There is always a clustering with maximum modularity, in which each cluster consists of a connected sub-graph.

Property 2. A clustering of maximum modularity does not include disconnected clusters.

Although the modularity optimization has the resolution limit [13], the community partition with a large modularity is usually a good solution. Since the genetic representation contains all possibility of connected sub-graphs, it guarantees that the good community partitions can be represented with the genetic representation.

The locus-based adjacency encoding scheme has three major advantages for community detection. (1) There is no need to fix the number of communities in advance, as it is automatically determined in the decoding step. (2) The representation is well-suited for the design of genetic operators. The standard crossover and mutation operations will not generate any invalid genotypes. (3) More importantly, the search space constructed by the representation is reduced distinctly. The popular cluster center genetic representation [29], which uses a number ranging from 1 to n to represent the community a node belonging to, has the n^n space complexity. Brandes et al. [4] cast the modularity optimization into an integer linear program with the complexity 2^{n^2} in the search space. In contrast, the space complexity of the locus-based adjacency representation is $\prod_{i=1}^{n} d_i$ (n is the number of nodes, d_i is the degree of node i). For most real problems, $d_i \ll n$.

We choose the uniform two-point crossover because it is unbiased with respect to the ordering of genes and is able to generate any combination of alleles from the two parents. Thus the crossover operator will not generate any invalid solutions. In the mutation operation, we randomly select some genes and assign them with other randomly selected adjacent nodes.

In the initialization process, we randomly generate some individuals. For each individual, each gene g_i randomly takes one of the adjacent nodes of node i.

3.2. Model selection phase

As noted previously, MOCD does not return a single solution, but a set of solutions. These solutions correspond to different tradeoffs between the two objectives and consist of the communities with different sizes. This provides the DMers with the opportunities to incorporate their domain knowledge into the community detection process. It is crucial for the problem with unknown structure. In addition, the DMers may desire that the set of candidate solutions could be narrowed down to those of most interest. In this section, we therefore propose two automated methods for assessing

Please cite this article in press as: C. Shi, et al., Multi-objective community detection in complex networks, Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.10.005

the quality of solutions. These methods can identify one promising solution in the candidate set, and then automatically deliver an estimate of the actual number of communities in the network. In addition, these two model selection methods provide an convenient way to compare MOCD with single-objective algorithms, since those algorithms return only one solution.

Max Q. The criterion selects the solution with the maximum modularity *Q*. Because of the relationship between *Q* and these two objective functions (see Eq. (3.7)), we can directly select the solution with maximum *Q* by aggregating the two objective values.

$$S_{Max-Q} = \underset{C \in SF}{argmax}\{1 - intra(C) - inter(C)\} \quad (3.8)$$

where *SF* is the candidate solution set (i.e., the Pareto front).

Max–Min Distance. We know that the physical meaning of *Q* is the fraction of edges that falls within communities, minus the expected value of the same quantity if the edges fall at random without regard for the community structure, so the *Q* evaluates the extent to which the community structure deviates from randomness [21]. The similar principle can also be used in our model selection. Concretely, MOCD first runs on the real network and a randomly generated network with the same scale. Thus, the real candidate solution set (called real Pareto front) and the random control solution set (called random Pareto front) can be obtained, respectively. And then from the real Pareto front we select the solution that mostly deviates from the solutions in the random Pareto front as the recommendation solution. Because there are multiple solutions in both the real and random Pareto fronts, we propose a max–min rule to quantitatively evaluate the deviation.

Firstly, the distance between a solution in the real Pareto front and one in the random Pareto front is defined in Eq. 3.9.

$$dis(C, C') = \sqrt{(intra(C) - intra(C'))^2 + (inter(C) - inter(C'))^2} \quad (3.9)$$

where C and C' represent the solutions in the real and random Pareto fronts, respectively. Then the deviation of a solution in the real Pareto front from the whole random Pareto front is quantified by the minimum distance between this solution and any solutions in the random Pareto front. The deviation is defined in Eq. 3.10.

$$dev(C, CF) = min\{dis(C, C') | C' \in CF\} \quad (3.10)$$

where *CF* represents the random Pareto front. Finally we select the solution in the real Pareto front with the maximum deviation. The model selection process is formulated in Eq. (3.11).

$$S_{Max-Min} = \underset{C \in SF}{maxarg}\{dev(C, CF)\} \quad (3.11)$$

where *SF* represents the real Pareto front.

Fig. 2 illustrates an example of the *Max–Min Distance* model selection method. From the figure, we can observe that the number of communities corresponds to the tradeoff between the *intra* and *inter* objectives and the solution with the correct community structure has the largest deviation from random solutions. In fact, the purpose of the random control solution set is to obtain an estimate of the values of *intra* and *inter* that would be expected for unstructured networks and the *Max–Min Distance* criterion evaluates the difference between the real objective values and the expected ones from unstructured networks. So the solution selected by the *Max–Min Distance* method has the most obvious community structure compared to random networks. In order to avoid the random factors, multiple random networks are applied (three random Pareto fronts are generated in the following experiments).

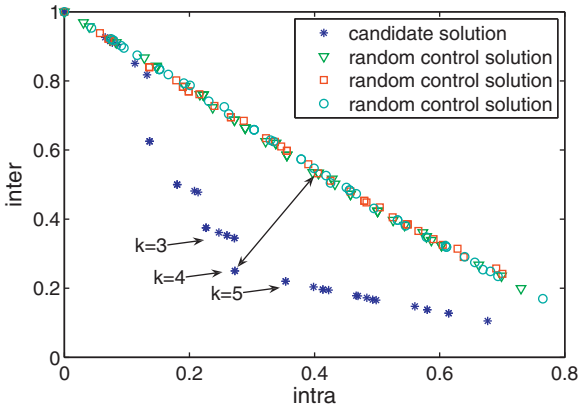


Fig. 2. Illustration of the *Max–Min Distance* model selection method. The candidate solutions and random control solutions are obtained by runs of MOCD on the symmetric artificial network with $z_{out} = 6$ in Section 4.2.1. The solution with the *Max-Min Distance* to the random control fronts is indicated, which correctly identifies 4 communities. *k* is the number of communities.

4. Experiments

This section will validate the effectiveness of MOCD through experiments on artificial and real networks. The first experiments will illustrate the advantages of multiple solutions returned by MOCD, and the second experiments will validate the quality of the solution recommended by the model selection methods. The experiments are carried out on a 3GHz and 1G RAM computer running Windows XP.

4.1. Analysis of multiple candidate solutions

This section demonstrates the advantages of multiple solutions returned by MOCD in revealing the implicit and comprehensive community structures through both artificial hierarchical and overlapping networks.

4.1.1. Hierarchical network

The artificial network example is the H13-4 network with 256 nodes [1,2]. Each node in the network has 13 links with the most internal community (formed by 16 nodes), 4 links with the most external community (formed by 64 nodes), and 1 more link with any other node at random in the network. Fig. 3(b) and (c) represent the network's modular structures at two different scales: Fig. 3(b) shows the first hierarchical level consisting of 4 communities, each with 64 nodes and Fig. 3(c) shows the second level consisting of 16 communities, each with 16 nodes (each community in the second level is called a unit community).

This experiments run MOCD with the following parameters: the *ipsize* and *epsize* both are 100, the *gen* is 800, p_c and p_m are 0.6 and 0.4, respectively. The average running time is 223 seconds. All the 100 solutions in *EP* are illustrated in Fig. 3(a) (note that some solutions are same). Observing the community structure of solutions, we find that the solution with the maximum *Q* (labeled I in Fig. 3(a)) represents the hierarchical structure in the first level, and another solution labeled II reveals the second level. The solutions between the solutions I and II are called the "strong" community solutions [26]. That is, in each community of these solutions, each node has more connections within the community than the rest of the network. These strong community solutions usually combine two or more connected unit communities into one community. According to the structure of the network, these community partitions all are correct. As for the other solutions (i.e., the solutions on the right side of solution II), although most communities in those solutions are correctly partitioned, a small portion of unit communities are

Please cite this article in press as: C. Shi, et al., Multi-objective community detection in complex networks, Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.10.005

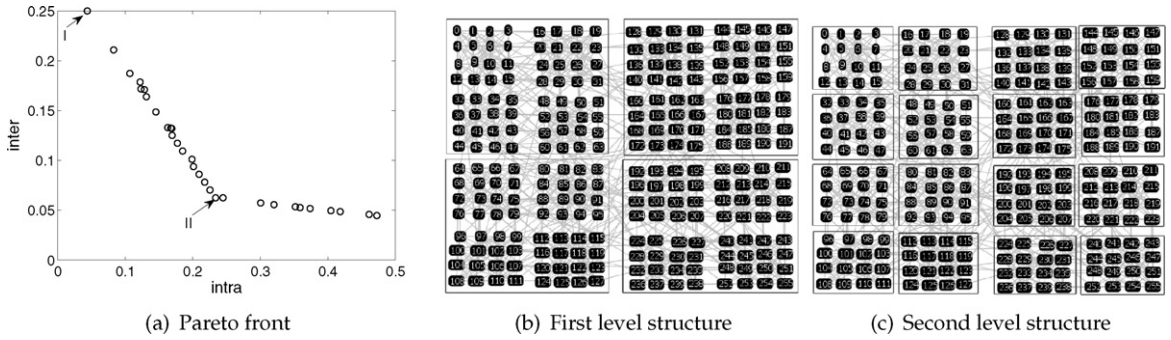


Fig. 3. Multiple resolution of modular structure in the artificial hierarchical network. (a) The Pareto front found by MOCD. (b) The first level real community structure which is the result of the solution labeled I in (a). (c) The second level real community structure which is the result of the solution labeled II in (a).

broken. This results in some solutions with the number of communities being larger than 16.

Using the experimental data, we further analyze the relationship of the objective values and the number of communities (NC) as shown in Fig. 4(a). It is obvious that with the increase of NC, the *intra* values increase, whereas the *inter* values decrease. It confirms that the modularity Q is a trade-off between these two objectives. The Q values are close to 0.7 when NC ranges from 4 to 16. When NC is larger than 16, the Q value rapidly decreases, and it seems to decrease with the increase of NC. It is reasonable to confine NC to be between 4 and 16, and thus their corresponding community partitions all are correct. So their Q values are large. When NC is larger than 16, some unit communities may be broken. In this situation, some nodes will be mis-partitioned, thus the Q values become small. This is also the reason why the solution labeled II, whose NC is 16, is the inflexion in the Pareto front in Fig. 3(a).

We then show the relationship between NC and the Q values of those strong community solutions (i.e., their NC are between 4 and 16) in Fig. 4(b). Their Q values are very close in general. But it still can be observed that, with the increase of NC, the Q value tends to become small. When the modularity Q is used as the optimization objective in a single-objective community detection algorithm, the algorithm tends to the solution labeled I, since it has the largest Q value. Thus it not only fails to reveal the hierarchical structure, but also cannot discover a finer community structure (e.g., the solution labeled II). The experiment further illustrates that the single-objective community detection has the resolution limit problem [13,17]: methods based on optimizing a modularity measure or other single criterion may fail to identify modules smaller than some thresholds. Compared to single-objective algorithms which may fail to identify small modules, the MOCD, in one run, can find a set of valuable community partitions which include both large and small modules. This trait is especially useful for hierarchical networks, where multiple levels of partitions exist.

4.1.2. Overlapping network

We further explore the advantages of a set of solutions returned by MOCD through an overlapping network. The network consists of two large communities A and B, each containing 128 nodes. Each node has on average 12 internal links. In addition, communities A and B each contains a subgroup of 32 nodes, which are denoted by a and b, respectively. Each node within a subgroup has six of its 12 intra community links with the rest 31 nodes of the subgroup. The two subgroups a and b have on average three links per node connected with each other within the subgroups. Additionally, each node has one link with a randomly chosen node from the whole network. It is clear that, besides two large communities (i.e., A and B), the overlapping part between A and B results in one overlapping community which is denoted by $a \& b$. A similar network has been used by Reichardt and Bornholdt to discover the overlapping structure [27].

MOCD settles the following parameters: the *ipsize* and *epsize* both are 200, the *gen* is 500, p_c and p_m are 0.6 and 0.4, respectively. We also run two representative algorithms on the network: the betweenness-based heuristic algorithm proposed by Newman and Girvan [21] (called GN) and the EA-based modularity optimization algorithm [28] (called GACD). GN and GACD both reveal two large communities A and B accurately. However, they fail to discover the overlapping structure. MOCD obtains 200 solutions which are illustrated in Fig. 5(a). Two special partition models are selected from these solutions. As shown in Fig. 5(b), the solution labeled I in Fig. 5(a) reveals the same communities partition (i.e., two large communities A and B) as that obtained with GN and GACD. However, in Fig. 5(c), we can see that the solution labeled II consists of three communities: two large communities and an overlapping community that is constituted by the nodes in a and b. The result shows that MOCD not only finds the obvious large community structure as what the single-objective algorithms can do, but also reveals the implicit overlapping community in one run.

4.2. Analysis of model selection

This section will evaluate the quality of the solutions recommended by the proposed model selection methods through both artificial and real networks.

Six algorithms are tested in the experiments. Besides GN and GACD that have been described in Section 4.1.2, the information-theoretic framework based algorithm [18] (called INFO) is included. We also include a multi-objective community detection algorithm MOGA-Net [24]. In order to obtain one single recommendation solution, MOGA-Net also employs the *Max–Min Distance* model selection method to select a partition from the Pareto front. As for MOCD, we consider both the two proposed model selection methods, i.e., MOCD with $Max - Q$ model selection (called MOCD-Q) and MOCD with *Max–Min Distance* model selection (called MOCD-D).

4.2.1. Artificial networks

The artificial networks with a known community structure have 128 nodes grouped in four communities of 32 nodes [18,21]. Each node has on average z_{in} edges to nodes in the same community and z_{out} edges to nodes in other communities, keeping an average degree $z_{in} + z_{out} = 16$. Since the multiple communities within the network have the same properties, the network is called the symmetric network in the paper. In order to validate the effectiveness of the algorithms for different kinds of networks, the experiments vary the network structures in the following ways. The first variation, called the node asymmetric network, merges three of the four groups in the benchmark test to form a series of test networks each with one large group of 96 nodes and one small group with

Please cite this article in press as: C. Shi, et al., Multi-objective community detection in complex networks, Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.10.005

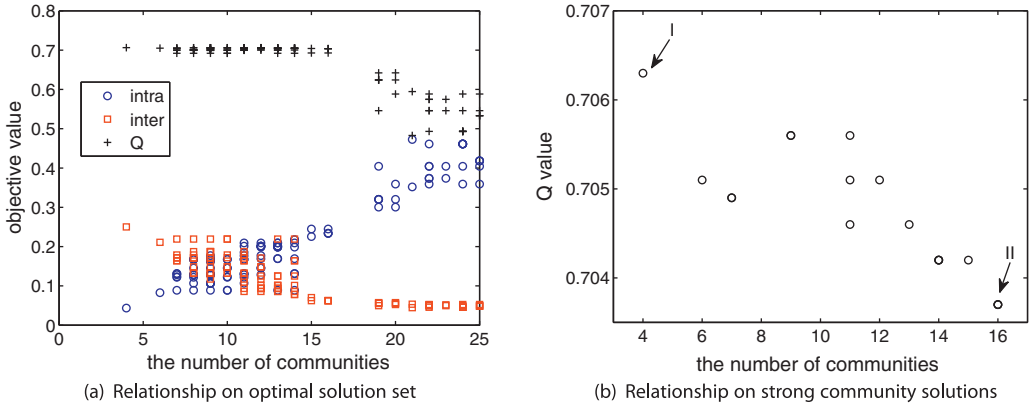


Fig. 4. Illustration of the relationship of the number of communities and the objective values. (a) Relationship on the optimal solution set. (b) Relationship on the strong community solutions. The solutions labeled I and II in (b) correspond to the solutions labeled I and II in Fig. 3(a), respectively.

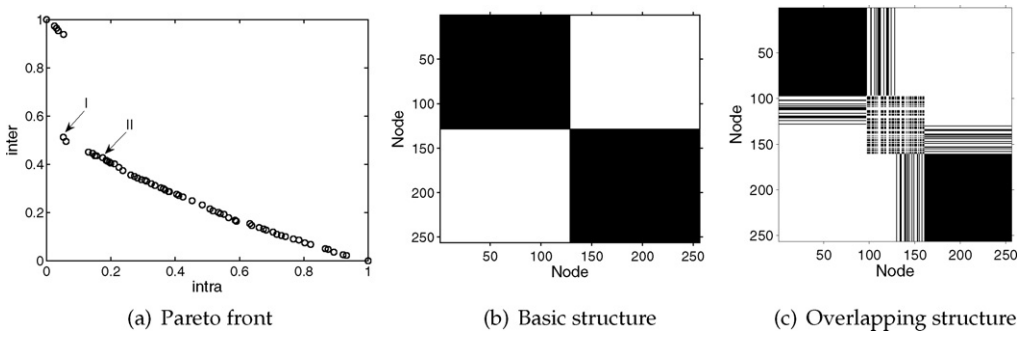


Fig. 5. Multiple resolutions of modular structure in the overlapping network. (a) Pareto front found by MOCD. (b) The partition result of the solution labeled I with a gray graph. (c) The partition result of the solution labeled II. The gray graph records whether two nodes are in the same community. The rows and columns of the matrix both correspond to the indices of the nodes and a black color means two nodes are in the same community.

32 nodes. In the second variation, the benchmark network, called the edge asymmetric network, composes two groups each with 64 nodes, but with different average degrees of edges (8 and 24) per node. As the average number of edges z_{out} increases, it becomes harder and harder to identify the community structure. According to the scale of the problem, four EA-based algorithms (i.e., GACD, MOGA-Net, MOCD-Q, and MOCD-D) set the uniform parameters: the population size is 100 (for MOCD, $ipsize = epsize = 100$), gen is 200, and p_c and p_m are 0.6 and 0.4, respectively. Note that, in the comparison experiments, the EA-based algorithms evaluate the same number of individuals.

To compare the quality of solutions, the experiments use the fraction of vertices identified correctly (FVIC) criterion, which has been used in many researches [7,10,21]. The larger the FVIC is the better the partition is. The FVIC can be calculated as follows.

$$\begin{aligned} olSet(c, c') &= \{v | v \in c \wedge v \in c'\} \\ maxOlSet(c, C_K) &= \max_{c' \in C_K} \{|olSet(c, c')|\} \\ FVIC &= \sum_{c \in C_F} \frac{maxOlSet(c, C_K)}{N} \end{aligned} \tag{4.12}$$

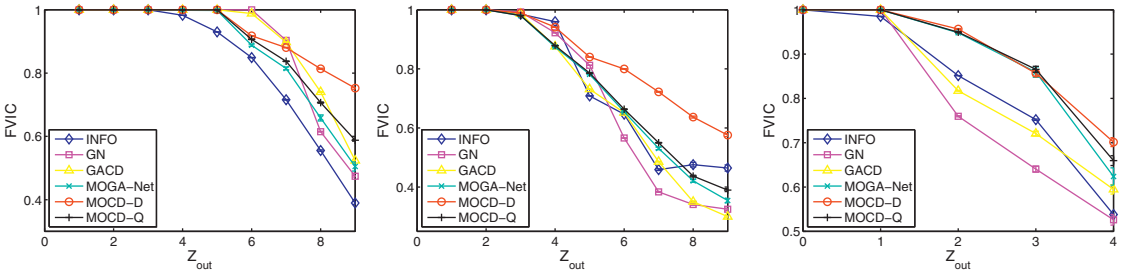
where C_F and C_K represent the found and known community partition, respectively; c and c' are a community in C_F and C_K, respectively. N is the number of nodes in the network.

Fig. 6 presents the distribution of the FVIC results of the six algorithms over 100 graph realizations. When z_{out} is small, all algorithms find the correct community partition. As z_{out} increases, these six algorithms have different performances, and their differences become more distinct. We can observe that two versions of MOCD (i.e., MOCD-Q and MOCD-D) have the highest FVIC in most situation, compared to not only those single-objective algorithms (i.e., INFO, GN and GACD) but also the multi-objective algorithm MOGA-Net. Comparing the results in the symmetric networks with those in the asymmetric networks, we can find that it is more difficult for all algorithms to discover the community structure in the asymmetric networks, especially for GN. However, the asymmetric networks have less impact on the MOCD. In other words, the MOCD still can obtain fairly good solutions for the asymmetric networks. The NCs found by these six algorithms are illustrated in Fig. 7. Similar to the results of FVIC, the NCs obtained with different algorithms are correct in all problems when z_{out} is small. The NCs deviate more and more from the correct values as z_{out} increases. It can be observed that the NCs found by MOCD-D are the closest ones to the correct values in most cases. When z_{out} becomes large, GN divides the network into so many communities that its results are not included in the figures. It also explains why GN's FVIC values decline rapidly as shown in Fig. 6. In both Figs. 6 and 7, we can observe that, with the increase of z_{out}, the Confidence Intervals (CIs) become broader, whereas they are still very tight for all algorithms. It reveals that the solution distributions of all algorithms have very small variances and the algorithms are all stable in this example. The experiments also show that GN and GACD are more effective for symmetric networks, whereas INFO is more effective for asymmetric networks. The results are consistent with those in Ref. [18].

4.2.2. Real networks

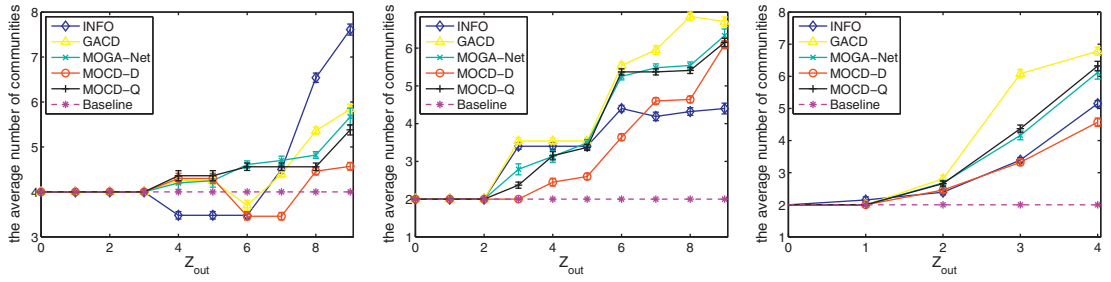
In order to further compare the performance of different algorithms, we use 10 real social networks which all are from Ref. [20]. These test problems are widely used as benchmarks in community detection [4,13,14,18], and they have different scales with the

Please cite this article in press as: C. Shi, et al., Multi-objective community detection in complex networks, Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.10.005



(a) Symmetric networks (b) Node asymmetric networks (c) Edge asymmetric networks

Fig. 6. Performances of different algorithms on the artificial networks measured by the fraction of vertices identified correctly (FVIC). Both the means and 90% confidence intervals are calculated based on the results of 100 graph realizations.



(a) Symmetric networks (b) Node asymmetric networks (c) Edge asymmetric networks

Fig. 7. The number of communities found by different algorithms for the artificial networks. Both the means and 90% confidence intervals are calculated based on the results of 100 graph realizations. The broken lines are the correct number of communities.

Table 1
Test problems and parameters settings in GACD, MOGA-Net, and MOCD.

		Karate (P1)	Lesmis (P2)	Polbooks (P3)	Adjnoun (P4)	Football (P5)	Celegansnearal (P6)	Celegansmetabolic (P7)	Netscience (P8)	Power (P9)	Hepth (P10)
Number of nodes		34	77	105	112	115	297	453	1589	4941	8361
Number of edges		78	254	441	425	613	2345	2025	2742	6594	15,751
GACD	*pop*	50	50	50	50	50	100	100	200	300	400
MOGA-Net	*gen*	50	50	100	100	100	100	100	200	300	400
MOCD	*ep*	50	50	50	50	50	100	100	100	100	100
	ip	50	50	50	50	50	100	100	200	300	400
	gen	50	50	100	100	100	100	100	200	300	400

number of nodes ranging from 34 to 8361. These test problems and the corresponding parameters of GACD, MOGA-Net, and MOCDs are illustrated in Table 1 (p_c and p_m are 0.6 and 0.4 for these four EA based algorithms, respectively). Note that we do not make any effort in setting good parameters for MOCD and the appropriate parameters are settled according to the scale of problems. Only one random Pareto front is generated in the *Max–Min Distance* model selection in the experiments. Moreover, the same numbers of individuals are evaluated in those EA based algorithms.

Since the actual community structures of most networks are unknown, we can only evaluate the qualities of solutions from the structural characteristics. Here we use two popular criteria to measure the qualities. According to the strong and weak community definition given by Radicchi et al. [26], each community c is validated based on whether satisfying the strong (or weak) community definition. The *ratio of strong (or weak) communities* is the fraction of communities in a partition C that satisfy the strong (or weak) community definition.

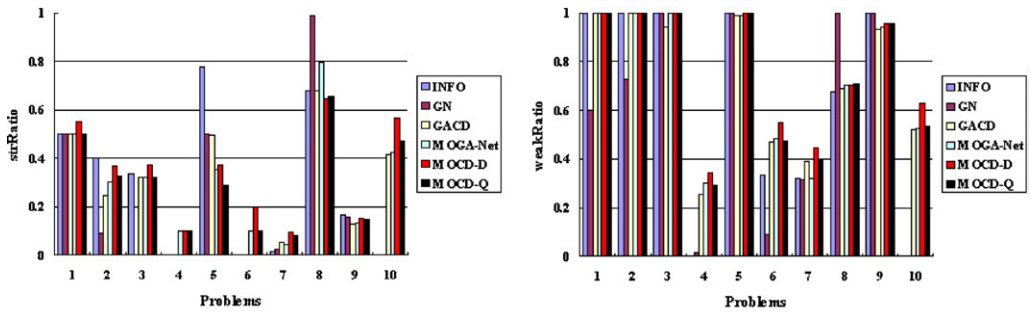
$$strRatio(C) = \frac{|\{c|k_i^{in}(c) > k_i^{out}(c)\ \ \forall i \in c \wedge \forall c \in C\}|}{|C|}$$
$$weakRatio(C) = \frac{|\{c|\sum_{i\in c} k_i^{in}(c) > \sum_{i\in c} k_i^{out}(c) \forall c \in C\}|}{|C|} \quad (4.13)$$

where c is a community in the partition C; $k_i^{in}(c)$ is the number of edges connecting node i to the other nodes belonging to c, and $k_i^{out}(c)$ is the number of edges connecting node i to the nodes in the rest of the network. These two criteria quantitatively evaluate how obvious the community structure is. The larger the value, the better the partition. According to the definitions, a strong community is also a weak community, whereas the reverse is not correct. So the *weakRatio*(C) is usually larger than the *strRatio*(C) for a given partition C. The results are the average of 10 runs.

The experimental results are shown in Fig. 8. MOCD-D discovers the community structures with the highest accuracy for most networks (e.g., P1, P3, P4, P6, P7, and P10), and INFO achieves the highest accuracy for three networks (e.g., P2, P5, and P9). MOCD-Q performs better than GACD for most problems (e.g., P1, P2, P3, P4, P6, P7, P9, and P10). Furthermore, MOCD-D performs better than MOCD-Q for almost all problems. In all, the performance comparison of the algorithms in the real networks is consistent with that in the artificial networks.

The running time of six algorithms are shown in Fig. 9, we can find that although the running time of the EA based algorithms (i.e., GACD, MOGA-Net, MOCD-D, and MOCD-Q) are generally longer than those of GN and INFO for small-scale problems, such as P1–P5, it is not the case for large-scale problems (e.g., P6–P10), especially for P10 where GN and INFO could not return any results in the

Please cite this article in press as: C. Shi, et al., Multi-objective community detection in complex networks, Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.10.005



(a) Performance on *strRatio* (b) Performance on *weakRatio*

Fig. 8. The performances of six algorithms for ten real networks. The numbers 1–10 represent the problems P1–P10, respectively. For P4, INFO always partitions the network into one community, and thus its *strRat* and *weakRat* both are settled with 0. For P10, INFO and GN cannot solve the problem in 20 h, so their *strRat* and *weakRat* all are settled with 0.

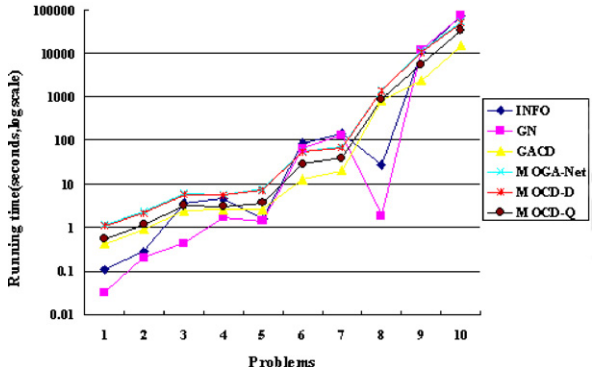


Fig. 9. The running time of six algorithms for ten real networks. The numbers 1–10 represent the problems P1–P10, respectively. For P10, INFO and GN cannot solve the problem in 20 h, so their running time both are settled with 20 h (i.e., 72,000 s) to conveniently compare with other algorithms.

given time (20 h). An exception is P8 which is a co-authorship network of scientists working on network theory and experiment [19]. The network has an obvious community structure and it is especially suitable for GN to detect the structure. The experiments show that the efficiency of MOCDs is not only acceptable for small-scale problems but also especially suitable for large-scale networks. Although GACD and MOCD evaluate the same number of individuals, the multi-objective algorithms are more complicated than the single-objective algorithms, so MOCDs have longer running time than GACD. Because MOCD-D needs to run twice to obtain the real and random Pareto fronts, the running time of MOCD-D is nearly the twice of that of MOCD-Q. With the *Max–Min Distance* model selection, MOCD-D achieves a better performance than MOCD-Q at the cost of a longer running time.

4.3. Discussion

Through the experiments on artificial and real networks, we find that two versions of MOCD (especially MOCD-D) have the best performance on most networks. To study the reason behind the superior performance of the MOCDs, we compare GACD with the MOCDs (especially MOCD-Q). A number of components of these two algorithms are the same (e.g., the EA framework, the genetic representation and operators, and the same number of individuals evaluated), except that GACD uses Q as the single function, whereas MOCDs treat the two components of Q as two functions. Therefore, the superior performances of MOCDs should be driven by the multiple optimization functions in MOCDs. We consider two reasons may account for the benefits of multiple optimization functions. (1) The multiple objectives can measure the community structure comprehensively and avoid the risk that one single-objective may only be suitable to a certain kind of networks (e.g., GN is only suitable for symmetric networks). (2) The multi-objective optimization process tradeoffs the multiple conflicting objectives, which can effectively avoid being trapped to local optima. MOCD and MOGA-Net both are based on multi-objective optimization. An important difference between MOCD and MOGA-Net lies in the objective functions. We think these two conflicting yet complementary objective functions in MOCD contributes to its better performances. In addition, when we compare MOCD-Q with MOCD-D, it is clear that MOCD-D has better performances. This shows that the *Max–Min Distance* may be a better model selection method than *Max Q* for this problem. We think the reason is that *Max–Min Distance* selects the model with the largest deviation from the random network with the same scale. This may indicate that the selected model represents the most significant community structure.

The fitness evaluation function (i.e., calculating the objective value) is the most time-consuming component in the algorithm. The calculation of objective functions (i.e., *intra* and *inter*) has the complexity $O(m)$, and the decoding of the genetic representation has the complexity $O(n)$ (m and n are the number of edges and nodes, respectively). As a consequence, the fitness evaluation of an individual has the complexity $O(m+n)$. Note that the multi-objective optimization process of MOCD (i.e., PESA-II) has the complexity $O(gs^2)$ [6]. So the whole complexity of MOCD is $O(gs^2(m+n))$ which grows linearly with the scale of the network. (g is the running generation, and s is the population size. For simplicity, *ipsize* and *epsize* both are s). The running generation and population size can affect the algorithm performance. However, increasing the population size or running generation does not necessarily yield better results after some points. Therefore, by settling the appropriate population size and running generation, MOCD can solve large-scale problems with an acceptable performance and a comparatively small time cost. As we known, many community detection algorithms have large time complexities [7], and thus they are not suitable for large-scale problems. For this reason, MOCD is a good solution for large-scale problems. Our experiments on real networks also confirm this point. As for model selection methods, *Max Q* has only the complexity $O(1)$, and *Max–Min Distance* has the complexity $O(gs^2(m+n))$. Compared to MOCD-Q, the better performances of MOCD-D are at cost of longer running time.

All MOCD's parameters are from those of PESA-II. These parameter settings follow the general rules of parameter settings in PESA-II. Generally, the large population size (i.e., *ipsize* and *epsize*) and running generations (i.e., *gen*) help to improve the accuracy of the algorithm and solve large-scale problems. The *epsize* controls the size of Pareto optimal set. The large *epsize* provides more candidate solutions. Simultaneously, it also becomes more difficult to choose a proper one. The large *ipsize* helps MOCD to extensively explore the

Please cite this article in press as: C. Shi, et al., Multi-objective community detection in complex networks, Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.10.005

solution space, whereas it will cost more running time. Large ratio of crossover (i.e., p_c) is helpful for quick convergence, but it may lead to premature. The ratio of mutation (i.e., p_m) has an opposite effect. In our experiments, we select the rational *ipsize*, *epsize*, and *gen* according to the scale of problems. The parameters p_c and p_m are fixed for all experiments, because we found that these settings are suitable for most problems.

5. Related work

Many algorithms have been designed to analyze the community structure in complex networks. The algorithms use methods and principles of physics, artificial intelligence, graph theory and even electrical circuits [7]. Most of these algorithms are based on the single-objective community detection. However, our MOCD is based on multi-objective community detection. Here, we briefly compare our MOCD with those most related work.

Genetic algorithm (GA) has been applied for community detection. The GAs in Refs. [28,29] optimize the modularity *Q*. Pizzuti proposes GA-Net to optimize the "community score" criterion [23]. Similar to these GA-based community detection methods, MOCD is also a heuristic search algorithm based on GA but different from them in its multi-objective nature. A multi-objective community detection algorithm MOGA-Net [24] has been proposed, which simultaneously optimizes the *CommunityScore* and *CommunityFitness* with NSGA-II. Besides differences in the components of EA (e.g., multi-objective optimization mechanism and objective functions), MOCD proposes two effective model selection methods. Handle and Knowles have applied EMO to clustering (MOCK) [15]. The different characteristics of the two problems (i.e., community detection and clustering) make MOCD and MOCK have many differences in objective functions, genetic operators and model selection methods. Many effective EMO algorithms have been proposed, such as NSGA-II [9] and PESA-II [6]. However, those EMO algorithms are designed for numerical optimization problems. Our MOCD is designed for community detection problem.

Some multi-solutions methods have been proposed. The Hamiltonian-based method introduced by Reichardt and Bornholdt (RB) [27] considers the community indices of nodes as spins in a *q*-state Potts model. Arenas, Fernandez and Gomez (AFG) [2] propose a multiple resolution procedure that allows the modularity optimization to go deep into the structure. These two methods investigate the community structure at various resolutions through tuning a parameter in their criteria. Essentially, they both are the single-objective community detection algorithm. In order to obtain multiple solutions, they need to run many times by tuning the parameter. However, our MOCD only requires one run to obtain a set of solutions.

6. Conclusion

In this paper, we first study the multi-objective community detection problem and propose an effective solution MOCD. MOCD simultaneously optimizes over two conflicting yet complementary objective functions with evolutionary algorithm and returns a set of community partitions. To help the DMers select proper partitions from those candidate partitions, we further propose two model selection methods: *Max Q* and *Max–Min Distance*. Through extensive experiments on both artificial and real networks, we demonstrate the advantages of the multi-objective community detection. The multiple solutions returned by MOCD can reveal community structures at different resolution levels in just one run, which can alleviate the bias existing in the single-objective community detection. With the proposed model selection methods, MOCD can discover more the accurate and comprehensive community structure compared to those well-established community detection algorithms.

Acknowledgements

This work is supported by in part by the National Science Foundation of China (Nos. 60905025, 61074128, 61035003). It is also supported in part by the National High-tech R&D Program of China (No. 2009AA04Z136).

References

[1] A. Arenas, A. Diaz-Guiler, C.J. Perez-Vicente, Synchronization reveals topolotical scales in complex networks, Physics Review Letter 96 (11) (2006) 114102.
[2] A. Arenas, A. Fernández, S. Gómez, Analysis of the structure of complex networks at different resolution levels, New Journal of Physics 10 (053039) (2008 May).
[3] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.U. Hwang, Complex networks: Structure and dynamics, Physics Report 424 (4-5) (2006) 175–308.
[4] U. Brandes, D. Delling, M. Gaetler, On modularity clustering, IEEE Transactions on Knowledge and Data Engineering 20 (2) (2008) 172–188.
[5] T. Cormen, C. Leiserson, R. Rivest, C. Stein, Introduction to Algorithms, MIT Press, USA, 2001.
[6] D. Corne, N. Jerram, J. Knowles, M. Oates, Pesa-ii: region-based selection in evolutionary multiobjective optimization., in: GECCO01, 2001, pp. 283–290.
[7] L. Danon, A. Diaaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, Journal of Statistical Mechanics: Theory and Experiments (2005 Sep).
[8] K. Deb, Multiobjective Optimization using Evolutionary Algorithms, Wiley, UK, 2001.
[9] K. Deb, A. Pratab, S. Agarwal, T. MeyArivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE Transaction on Evolutionary Computation 6 (2) (2002) 182–197.
[10] J. Duch, A. Arenas, Community detection in complex networks using extremal optimization, Physical Review E 72 (2) (2005) 027104.
[11] E. Ravasz, A.L. Somera, A. Mongru, Z.N. Oltvai, A.L. Barabasi, Hierachical organization of modularity in metabolic networks, Science 297 (5586) (2002) 1551–1555.
[12] G.W. Flake, S. Lawrence, C.L. Giles, F.M. Coetzee, Self-organization and identification of web communities, IEEE Computer 35 (3) (2002) 66–71.
[13] S. Fortunato, M. Barthelemy, Resolution limit in community detection, Proceedings of the National Academy of Sciences 104 (1) (2007) 36–41.
[14] R. Guimera, L.A.N. Amaral, Functional cartography of complex metabolic networks, Nature 433 (2005) 895–900.
[15] J. Handle, J. Knowles, An evolutionary approach to multiobjective clustering, Transaction on Evolutionary Computation 11 (1) (2007) 56–76.
[16] R. Kannan, S. Vempala, A. Vetta, On clusterings: good, bad and spectral, Journal of the ACM 51 (3) (2004) 497–515.
[17] J. Kumpula, J. Saramäki, K. Kaski, J. Kertész, Limited resolution and multiresolution methods in complex network community detection, Fluctuation and Noise Letter (2007).
[18] R. Martin, T.B. Carl, An information-theoretic framework for resolving community structure in complex networks, Proceedings of the National Academy of Sciences 104 (18) (2007) 7327–7331.
[19] M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Physics Review Letter 74 (2006) 036104.
[20] M. Newman, Netdata. http://www-personal.umich.edu/mejn/netdata/, 2009.
[21] M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Physics Review E 69 (026113) (2004).
[22] Y. Park, M. Song, A genetic algorithm for clustering problem, in: ACGP98, 1998, pp. 568–575.
[23] C. Pizzuti, Ga-net: a genetic algorithm for community detection in social networks, in: PPSN2008, 2008, pp. 1081–1090.
[24] C. Pizzuti, A multi-objective genetic algorithm for community detection in networks, in: ICTAI09, 2009, pp. 379–386.
[25] A. Pothen, H. Sinmon, K.-P. Liou, Partitioning sparse matrices with eigenvectors of graphs, SIAM Journal on Matrix Analysis and Applications l (11) (1990) 430–452.
[26] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, Proceedings of the National Academy of Sciences 101 (9) (2004) 2658–2663.
[27] J. Reichardt, S. Bornholdt, Statistical mechanics of community detection, Physics Review E 74 (1) (2006) 016110.
[28] C. Shi, Z.Y. Yan, Y. Wang, Y.N. Cai, B. Wu, A genetic algorithm for detecting communities in large-scale complex networks, Advance in Complex System 13 (1) (2010) 3–17.
[29] M. Tasgin, H. Bingol, Community detection in complex networks using genetic algorithm. arXiv:cond-mat/0604419, 2006.
[30] D.A.V. Veldhuizen, G.B. Lamont, Multiobjective evolutionary algorithms: analyzing the state-of-the-art, Evolutionary Computation 18 (2) (2000) 125–147.

Please cite this article in press as: C. Shi, et al., Multi-objective community detection in complex networks, Appl. Soft Comput. J. (2011), doi:10.1016/j.asoc.2011.10.005