A Memetic Algorithm for community detection by maximising the Connected Cohesion

1st Mohammad Nazmul Haque
School of Elect. Eng. and Computing
The University of Newcastle
Callaghan, NSW 2305, Australia
Mohammad.Haque@newcastle.edu.au

2nd Luke Mathieson
School of Software
University of Technology Sydney
Ultimo, NSW 2007, Australia
Luke.Mathieson@uts.edu.au

3rd Pablo Moscato
School of Elect. Eng. and Computing
The University of Newcastle
Callaghan, NSW 2305, Australia
Pablo.Moscato@newcastle.edu.au

***Abstract*—Community detection is an exciting field of research which has attracted the interest of many researchers during the last decade. While many algorithms and heuristics have been proposed to scale existing approaches a relatively smaller number of studies have looked at exploring different measures of quality of the detected community.**

Recently, a new score called 'cohesion' was introduced in the computing literature. The cohesion score is based comparing the number of triangles in a given group of vertices to the number of triangles only partly in that group. In this contribution, we propose a memetic algorithm that aims to find a subset of the vertices of an undirected graph that maximizes the cohesion score. The associated combinatorial optimisation problem is known to be NP-Hard and we also prove it to be W[1]-hard when parameterized by the score. We used a Local Search individual improvement heuristic to expand the putative solution. Then we removed all vertices from the group which are not a part of any triangle and expand the neighbourhood by adding triangles which have at least two nodes already in the group. Finally we compute the maximum connected component of this group.

The highest quality solutions of the memetic algorithm have been obtained for four real-world network scenarios and we compare our results with ground-truth information about the graphs. We also compare the results to those obtained with eight other community detection algorithms via interrater agreement measures.

Our results give a new lower bound on the parameterized complexity of this problem and give novel insights on its potential usefulness as a new natural score for community detection.

***Index Terms*—community detection, triangles, connected cohesion, memetic algorithm**

I. Introduction

The relationships among the components of a complex system can be easily represented by networks of different types and in some cases by undirected, simple graphs. In those cases, the individual components of these systems are represented by nodes and the relationships by edges. Undirected graphs are extensively used in social behaviour modelling, where the individuals are represented by vertices and the behaviour can be linked with edges of the graph [8], [12], [14]. We can potentially mine for structures of densely connected vertices to identify interesting relationships and behaviours in the complex system. One such graph structure is called a *community*, which represents a relativity tightly interconnected group of vertices with relatively fewer links to the rest of the graph. Community detection is regarded as one of the most effective approaches for uncovering the underlying relationships in complex networks. It has been used in various fields such as social science, biology, economics, communications, and scientific collaborations [5], [11], [18]–[20]. Many algorithms have been proposed in the field of community detection, from simple partitioning approaches to sophisticated optimisation techniques based on the maximisation of various objective functions.

The most widely used objective functions for community detection are *Modularity* [21] and the *Degree of Centrality* [2]. The modularity score is based on comparing the strength of inter- and intra-community connections with a null model in which edges are randomly connected. The higher the value of the modularity the higher the density of edges that connect the vertices of the community, but optimization using this score also uncovers the relatively sparse connections that these vertices have with other vertices in the graph. In the last few years, researchers have shown that modularity, as a score function, suffers from certain drawbacks, including a resolution limit [6], which prevents it from recognising smaller communities [22]. It also misses small communities in scale-free networks [13]. On the other hand, the structural connectivity density of a vertex is used to measure the degree of centrality [7]. The idea of measuring centrality is based on the number of neighbours a vertex has. Assuming the edges represent a friendship relationship, it is natural to expect that a densely connected group may have some sort of advantage in life by exploiting this social network. This leads to the idea of degree centrality, which refers to the degree of a given node in the graph representing the centrality measure in a social network. One possible common drawback of centrality measures is that they only take into account the direct connections of a vertex with its neighbours.

In this work, we consider the concept of *cohesion* [8] as a score for community detection. There are many ways to measure the cohesion, e.g. average degree of the network, graph density, fragmentation as the proportion of node pairs who are not located in the same component, cliques. Friggeri and Fleury [8] proposed a measure of cohesion based on the number of triangles in and partially in a set of vertices to help identify the communities in the complex network. We will use their cohesion score to identify the largest connected

978-1-5386-2726-6/17/$31.00 ©2017 IEEE

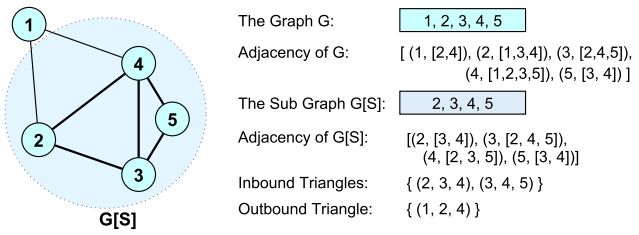


Fig. 1. An example of of a graph $G = (V, E)$ where $V(G) = \{1, 2, 3, 4, 5\}$, and $E(G) = \{(1, 2), (1, 4), (2, 3), (2, 4), (3, 4), (3, 5), (4, 5)\}$. The subgraph $G[S]$ induced by $S = \{2, 3, 4, 5\}$ is highlighted where and shows the mapping to corresponding adjacency list.

community.

II. Preliminaries and Definitions

Let $G = (V, E)$ be a graph with vertex set V and edge set E where $n = |V| \geq 4$. The subgraph of G induced by vertex set S is denoted $G[S] = (S, E_s)$. Fig. 1 shows an example of a graph and an induced subgraph with their properties.

Definition: II.1 (Triangle)**.** A *Triangle* (Δ) in a Graph G is a triplet $(u, v, w) \in V^3$ of pairwise connected vertices, such that $(uv, vw, uw) \in E^3$.

Definition: II.2 (Inbound Triangle Count)**.** The *Inbound Triangle Count* for $G[S]$ is denoted as $\Delta_i(S) = |\{(u, v, w) \in S^3 : (uv, vw, uw) \in {E_s}^3\}|$, that is, it is the number of triangles in G whose vertices are all in S.

Definition: II.3 (Outbound Triangle Count)**.** The *Outbound Triangle Count* for $G[S]$ is denoted as $\Delta_o(S) = \{(u, v, w), (u, v) \in S^2, w \in V \setminus S : (uv, vw, uw) \in E^3|\}$, that is the count of those triangles in G, which have *exactly two* vertices in S.

Definition: II.4 (Triangle Neighbours)**.** The Δ*Neighbours(u)* are the neighbouring vertices $\{v_1, v_2, \cdots, v_m\} \subset V$ of a vertex u, such that for each i, there exists a j where u, v_i and v_j form a triangle in G.

A. The Connected-Cohesive Problem

Let S be a set of vertices in a graph, $\Delta_i(S)$ be the number of triangles with all vertices in S and $\Delta_o(S)$ the number of triangles with exactly two vertices in S, then the cohesion of S, written $C(S)$, is defined as:

$$C(S) = \frac{\Delta_i(S)^2}{\binom{|S|}{3} \cdot (\Delta_i(S) + \Delta_o(S))} \tag{1}$$

This give the Connected-Cohesive problem:

Connected-Cohesive
Input: A graph $G = (V, E)$, $\lambda \in \mathbb{Q}$ with $0 \leq \lambda \leq 1$.
Question: Is there a subset $S \subseteq V$ with $C(S) \geq \lambda$?

Theorem 1 (Friggeri and Fleury [9])**.** Connected-Cohesive *is* NP*-complete.*

We can then extend this result into the parameterized world:

Theorem 2. Connected-Cohesive *is* W[1]*-hard when parameterized by* λ*.*

Proof. A *parameterized reduction* is a many-one reduction that is computable in time $f(k) \cdot n^{O(1)}$, for any f, where k is the parameter of the input instance, and the parameter of the target instance is bounded by a function of k. In particular normal Karp reductions are a special case of parameterized reductions when the parameter mapping is also preserved. Thus in this case we may use the same construction as [9] to build a reduction from Clique to Connected-Cohesive and observe that the reduction already a parameterized reduction with:

$$\lambda = \frac{\binom{k}{3}}{\binom{k}{3} + \binom{k}{2}(n - k)} \leq 1 \leq k$$

Note that we can exclude the case where $k = 0$ as a trivial case without affecting the correctness of the reduction. □

This reduction also immediately gives the following results:

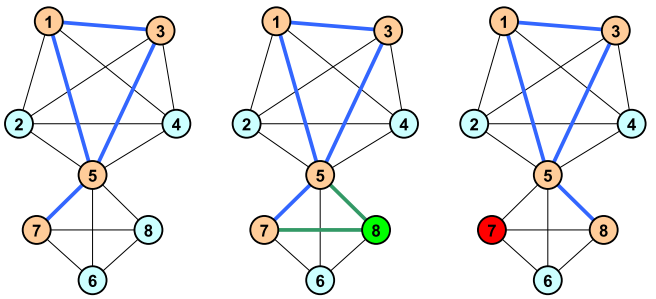
Theorem 3. Connected-Cohesive *is* W[1]*-hard under the following parameterizations:* $|S|$*,* $\Delta_i(S)$*, and* $|S| + \Delta_i(S) + \lambda$*.*

The class W[1]-hard also includes problems like Perfect Code and Planar Capacitated Dominating Set. Approaches based on Memetic Algorithms have been used fruitfully for some problems in this class. Other classical combinatorial optimization problems that are W[1]-Complete include Subset Sum (when parameterized by the size of the subset), Maximum Independent Set and Maximum Clique (when parameterized by the number of vertices). In particular, on the Max Clique problem Wei and Dinneen have investigated the effects of a fitness function on the run-time performance of an Adaptive Memetic Algorithm [24]. Together these results are part of the background motivation to evaluate first the results of optimizing with a memetic algorithm using the cohesion score and to compare with the results of other algorithms that employ different techniques.

III. Algorithmic Implementation

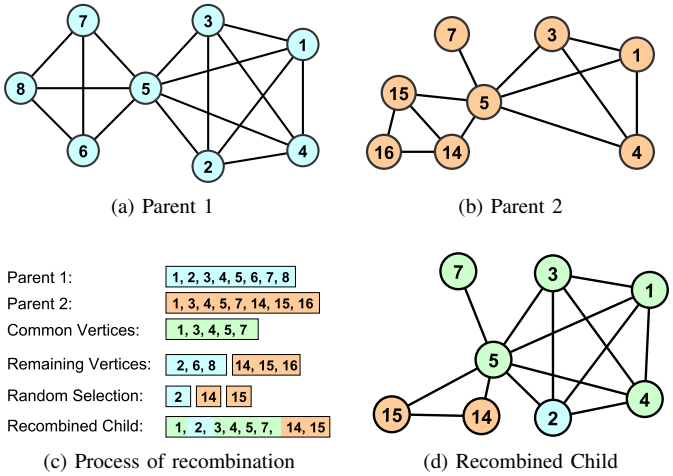
Our memetic algorithm (MA) was coded using the Python Programming Language, version 2.7.6. We represent the input graph using a lightweight graph data structure wrapping a Python list which stores the vertex labels and a Python dictionary which constitutes an adjacency list representation of the edges.

The overall design follows the typical basic MA template. Populations of individual solutions are generated randomly then repeatedly mutated and crossbred with the "fittest" solutions being selected for mutation and crossover. Periodically the evolutionary cycle is interrupted by an iterative improvement phase where every individual is subjected to deterministic optimisation (in our case via an individual search procedure). The details of the number of generations, the frequency of the deterministic improvement and the size and number of the populations are all parameters which can be tuned to improve



(a) A feasible solution (b) added ΔVertex 8 (c) vertex 7 is removed

Fig. 2. An example of *Mutation* operation of an Individual. (a) A solution $[1, 3, 5, 7]$. (b) The mutation operation randomly adds ΔVertex 8, hence the mutated individual become $[1, 3, 5, 7, 8]$. (c) The mutation operation randomly removes Vertex 7 and the mutated individual turns into $[1, 3, 5, 8]$. These alteration operations occurred with a user-defined probability (μ_R).



(a) Parent 1 (b) Parent 2

(c) Process of recombination (d) Recombined Child

Fig. 3. An example of *Recombination* operation for a pair of parents (a) $[1, 2, 3, 4, 5, 6, 7, 8]$, (b) $[1, 3, 4, 5, 7, 14, 15, 16]$. The recombination operation (c) keeps common vertices from both parents, then adds some random vertices form the remaining vertices of parents and (d) produces a recombined child $[1, 2, 3, 4, 5, 7, 14, 15]$.

performance, either from an optimality standpoint or from a resource-use perspective.

A. Representation of Solutions

A feasible solution is represented by a list, which contains the name of vertices it consists of. In our Python implementation of the mapping of a vertex to corresponding adjacency, for an edge (u, v), there will be an entry of vertex v in the adjacency dictionary of vertex u, and vice versa. An illustration of about the individual representation is shown in Fig. 1 where $[2, 3, 4, 5]$ represents $G(S)$ in the graph G. The solution has two Δ_i and one Δ_o.

B. Genetic Algorithm Operators

a) Mutation of solutions: We randomly alter a solution at a given probability. First we add a random vertex from the ΔNeighbourhood of the solution, then remove a random vertex from the solution, if and only if the solution already has more than four elements. The probability of these alterations is defined by the user as a parameter (we used 0.02). An example of the mutation process is described in Fig. 2.

b) Recombination of solutions: To recombine two solutions, we first include all common vertices of the two parent solutions in the new one to be created. If the two parents were actually identical, we select uniformly at random a neighbour vertex to the ones in the solution. If there were not, we uniformly at random select a subset of vertices (of those which were not common) and include them in the newly created soluiton. The probability of selecting a vertex from this group is also user-defined (we used 0.05). Fig. 3 shows an example of the recombination process.

C. Fitness Function

We initially found that an MA that explicitly uses *Cohesion Score* (defined in (1)) as a guiding function was producing very small communities (sometimes even reducing to a single triangle in some preliminary experiments). Consequently, our fitness function for the MA is a slightly more complex evaluation than the *Cohesion Score*. The fitness function $fitness(S)$ is defined as:

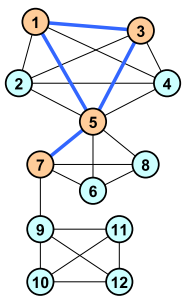
$$fitness(S) = \frac{|S|}{|V|} * C(S). \tag{2}$$

where $S \subseteq V$ in G. The numerator ($|S|$) in the first part of fitness function multiplied with the cohesion score gives preference to the CONNECTED-COHESIVE groups with more vertices. This helps the MA to find larger communities in preference to finding small communities (such as triangles or other small cliques) with the maximum cohesion score.

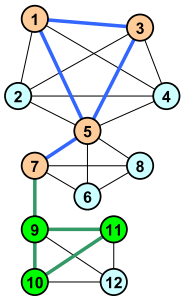
D. Individual Optimisation of Solutions

Individual optimisation has always helped to explore the fitness landscape [17] it is a critical component in MAs. In this case, the process to optimise a solution is illustrated in Fig. 4.

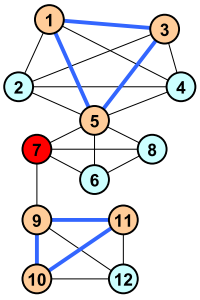
The local search optimisation process inspired tries to find a new Δ to add into the group which is a neighbour of the individual. The Search outcome of a given scenario is shown in Fig. 4b. The individual is then pruned by removing any vertices which contributes no Δ in the individual (see Fig. 4c). Now, for this individual, we expand the neighbourhood by adding more Δs which has at least 2 neighbours already are in the individual. The outcome of the process is shown in Fig. 4d. The final step of the optimisation process is to compute the maximum connected component (see Lemma 1.1 in [9]). The process of computing the maximum connected component is illustrated in Fig. 4e. This maximum connected component is the optimised individual.



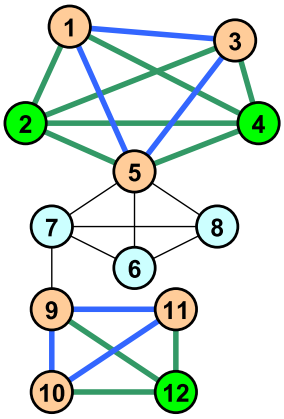
(a) Individual in the Graph



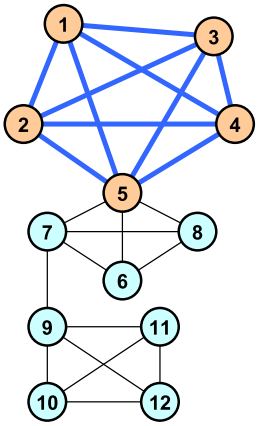
(b) Adding a Triangle



(c) Pruning



(d) Expanding Neighbourhood



(e) Maximum Connected Component

Fig. 4. An example of the individual improvement process using local search. (a) The individual $\langle [1,3,5,7], 0.0185 \rangle$ to improve. (b) Local search adds a neighbouring triangle $(9,10,11)$, now the individual is $\langle [1,3,5,7,9,10,11], 0.0102 \rangle$. (c) Pruning vertices which contribute no triangles to the group, now the individual is $\langle [1,3,5,9,10,11], 0.2058 \rangle$. (d) Expanding the neighbourhood by adding triangular vertices that have at least 2 neighbours already in the individual, now the individual is $\langle [1,2,3,4,5,9,10,11], 0.2058 \rangle$. (e) Computing the maximum connected component of the individual and the improved individual become $\langle [1,2,3,4,5], 0.8333 \rangle$.

E. The Framework of the Connected Cohesion with Memetic Algorithm (CCMA)

In our memetic algorithm, we first randomly generate the initial population. Then an objective function consisting of the cohesion score is defined which maximises the size of connected cohesive group from the network. After that, the recombination followed by mutation is performed. Then our local search operation is performed after each 100 generations to improve the population. We chose the top SZ_{pop} (here the population size = 20) individuals from the population for next generation of the memetic algorithm. The memetic algorithm repeats for $N = 500$ generations. The detail of the *CCMA* framework is outlined in Algorithm 1. The parameters used in our implementation of CCMA were hand tuned at a coarse level, guided by empirical experiments on randomly generated Watts-Strogatz graphs and prior experience.

Algorithm 1: The CCMA Algorithm

Input: The Graph G, Maximum number of iteration N, size of the population SZ_{pop}, mutation rate μ_R, recombination rate χ_R
Output: The best individual I_{best}

```
/* Initialisation of population                          */
P ← RandInitPop(SZ_pop)
/* The Memetic Algorithm Loop                            */
for genIdx = 1 : N do
    /* Offspring generation                              */
    P_χ ← φ
    for recoCntr = 1 : SZ_pop/2 do
        I_p1 ← randomChoice(P)
        I_p2 ← randomChoice(P)
        I_χ ← Recombination(I_p1, I_p2, χ_R)
        P_χ.append(I_χ)
    end
    P ← P ∪ P_χ
    /* mutation of the individuals                       */
    P_μ ← φ
    for idx = 1 : size(P) do
        if random() ≥ μ_R then
            P_μ [idx] ← Mutation(P [idx], μ_R)
        end
    end
    /* selection of best individuals                     */
    P_new ← Selection(P_μ, SZ_pop)
    /* local search optimisation                         */
    if mod(genIdx, 100) = 0 then
        P_opt ← φ
        for idx = 1 : SZ_pop do
            P_opt [idx] ← localSearch(P_new [idx])
        end
        P_new ← P_opt
    end
    /* update population for next generation             */
    P ← P_new
end
I_best ← getBest(P)
return I_best
```

IV. Results on Cohesion on Real-World Datasets

We evaluated the connected cohesive groups obtained in the following way:

- The MA was first applied to find the large connected cohesive groups in the networks of four datasets.
- For each different domain we used the ground-truth knowledge to interpret the results.
- The interrater agreement between what was found by our algorithm and what can be obtained with other state-of-the-art community detection algorithms was evaluated using the Kappa score.

We use Gephi software's version 0.9.1 [1] to produce the visuals of the graphs used in the experiments.

A. Experiment 1 - Customer Behaviour

The *Customer Behaviour* network dataset was obtained as a result of a modelling process that originated in a study of online consumer behaviour [4]. The original survey contained 69 questions which address several areas of interest to digital marketers. The questions pertain to specific *constructs* of interest. In [4], the authors used symbolic regression modelling

to identify if the given answers of one question can be predicted by a model involving all the other answers. The network is then constructed from these 69 models; each question is a vertex and an arc between two vertices exists if the source vertex/question is part of a model for the target vertex/question. Full details of the process to formulate the networks from the raw data can be found in [4]. We will ignore edge directions, so our undirected graph has 69 vertices, corresponding to the original 69 questions and 250 edges.

As we have the ground-truth data for the responses of leading to the customer behaviour network, we use those for further analysis. A total of 371 responders scored their answer in a range of 1 to 7 for those questions in the connected-cohesive group. Six questions have appeared in a connected cohesive group, a clique involving two questions regarding Usage Intensity and two regarding Subjective Knowledge. They are singled out as a cohesive community which is part of a larger community described in [4] that also had questions related to Social Value (see Fig. 3 in [4]). These kinds of highly connected questions may indicate the presence of *functional constructs*. In this case, a strong association seems to exists between the feeling of being knowledgeable about using a social media platform and the subjective perception of the frequency of use. In this case cohesion seems to have been able to "zoom in" in a particular community previously discovered and helped to extract a strong association. Remarkably, in this case at least, this could have also been done perhaps by mining for maximal cliques in polynomial-time.

B. Experiment 2 - Dolphins Social Network

The *Dolphins Social Network* contains an undirected social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand, as compiled by Lusseau et al. (2003) [14]. The most cohesive largest community we found from our algorithm consists of six dolphins (see the group of green coloured vertices in Fig. 6). The connected-cohesive group consists of **Patchback, Trigger, MN83, MN105, Jonah** and **Topless** dolphins. This concurs with Lusseau *et al.*'s analysis of the community structures within the Doubtful Sound dolphin community. In this case we do not have a clique as there are 14 edges between the 6 nodes.

C. Experiment 3 - Storm of Swords

We now search for a connected cohesive group of the characters of the novel "A Storm of Swords" [15] of the book series "A Song of Ice and Fire"[1] written by George R. R. Martin adapted by HBO as the famous series *Game of Thrones*[2]. The network consists of 107 vertices (characters) and 353 edges. There is an edge between two characters if they appear in the book separated by maximum 15 words, which is suggestive of character interaction and thus the existence of a connection. The outcome of the proposed method identified a group of 12 characters as the most cohesive group in the network highlighted in Fig. 7. Once again, we do not have a clique since only 50 edges are present. The characters in the most cohesive group are **Arya Stark, Cersei Baratheon née Lannister, Eddard Stark, Gregor Clegane, Ilyn Payne, Jaime Lannister, Joffrey Baratheon, Meryn Trant, Robert Baratheon, Sandor Clegane, Sansa Stark** and **Tyrion Lannister**. In terms of relationships in the novel, we found some characters centred on the Stark and Lannister-Baratheon families with the strong cohesive connection. Without recounting the events of the novel, the relationships can be summarised from a narrative standpoint: *Sansa*, is the daughter of *Eddard Stark*, and was betrothed to Prince, later King *Joffrey Baratheon*, who himself is the putative son of King *Robert Baratheon* (Eddard's best friend) and *Cersei Baratheon née Lannister*, however he is, in fact, the son of *Cersei* and her brother *Jaime Lannister*. *Jaime* also serves as the King's bodyguard, alongside *Meryn Trant*. Later, she married *Tyrion Lannister*, the uncle of Joffrey Baratheon and brother of *Cersei* and *Jaime*, following *Joffrey's* execution of her father at the hand of *Ilyn Payne* the court executioner. The other character, *Arya*, sister of *Sansa*, is the youngest daughter of Lord *Eddard* who spends a significant portion of the novel travelling with *Joffrey's* retainer *Sandor*. *Sandor* is the younger brother of *Gregor*, who himself is a vassal of the Lannister family, serving as a retainer to *Cersei*. It can be noted that a significant portion of the narrative involving these characters takes place in King's Landing (a city in the novels), giving the characters a spatio-temporal relationship as well as familial. In particular, other members of the Stark household do not spend much (if any) time in King's Landing throughout the novel. This brief summary of the relationships gives some indication of the complex intertwining of these characters in the narrative, supporting the community suggested by our algorithm.

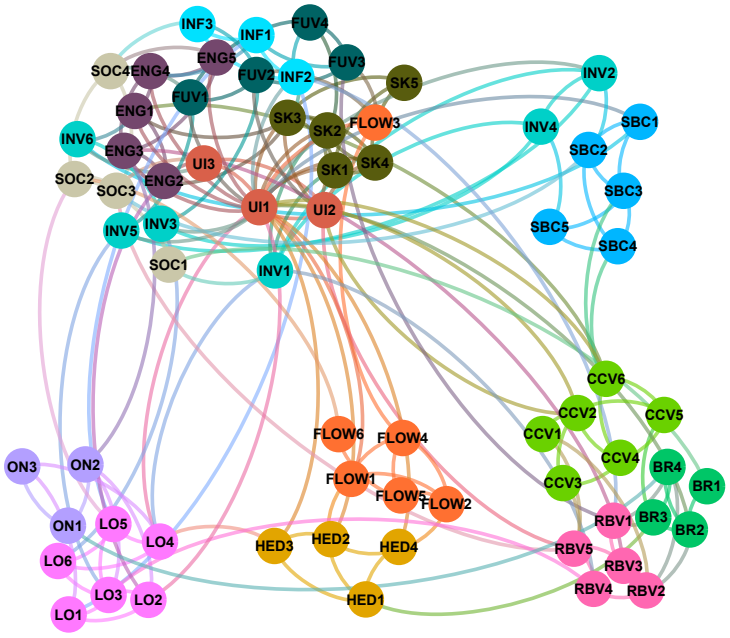
D. Experiment 4 - Kodak's Co-purchasing Network

In [10], the authors analyze Kodak's co-purchasing network on Amazon. Their intent was to identify *"Where does a brand end ?"*. We refer to that publication for the generation of a graph that is Kodak-centric (i.e. contains co-purchasing information in which at least one of the products belongs to the brand Kodak). Products are of different types, not restricted to cameras. This gives a view of the "mind" of customers. A (different) memetic algorithm identifies a set of overlapping partitions of the set of products. We then identify "the core" of the graph (the set of products which are members of at least two communities in [10]). This core graph is shown in Fig. 8a and we them proceed to run the CCMA to identify a large connected cohesive network.

The algorithm of [10] assigns community labels to the vertices in a brand-oblivious manner (totally unsupervised by brand information). Since different products are sold for the same brand, we initially expected that "the core" would be mainly composed by Kodak products (perhaps of different types). However, the result was somewhat surprising; Fig. 8b shows the word cloud based on the frequency of the brands in the core Kodak network. The MA revealed a large cohesive

[1]https://en.wikipedia.org/wiki/A_Song_of_Ice_and_Fire

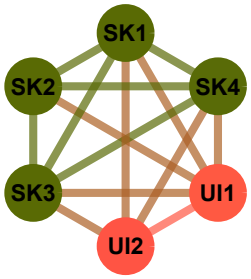
[2]http://www.hbo.com/game-of-thrones



(a) Customer Behaviour Network

Vertex	Question
UI1	I frequently use the Facebook brand page
UI2	I often use the Facebook brand page
SK1	I know a lot about using Facebook
SK2	I feel knowledgeable about using Facebook
SK3	Among my circle of friends, I'm one of the "experts" on using Facebook
SK4	Compared to most other people, I know more about using Facebook

(b) Value of Nodes in Connected Cohesive Group



(c) Connected Cohesive Group

Fig. 5. The Connected Cohesion outcome for the Customer Behaviour Network. (a) The customer behaviour network (b) The questions represented by the nodes appeared in the connected cohesive community (c) The connected cohesive community found in the customer behaviour network.

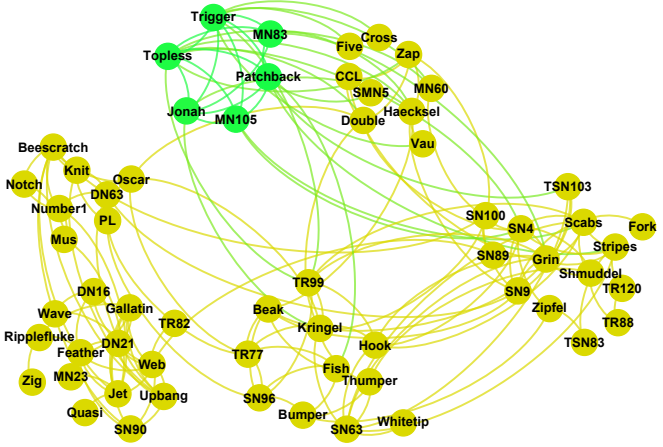


Fig. 6. The best group found (six nodes, and 14 edges) by the algorithm for the *Dolphins Social Network* showing the interaction with other dolphins.

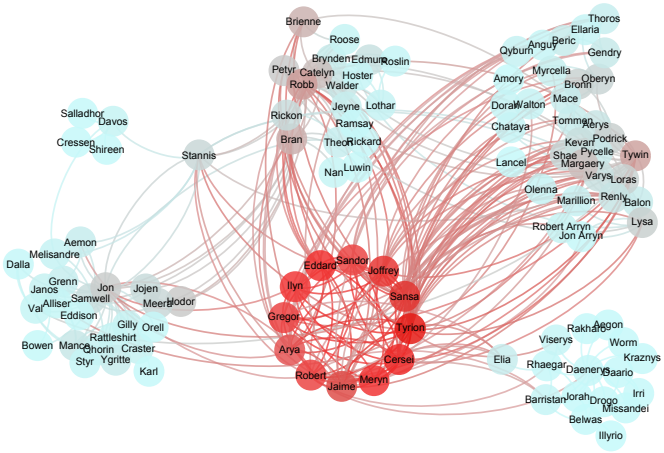


Fig. 7. The best group found by the memetic algorithm for the *Storm of Swords* network and the interconnection with other characters.

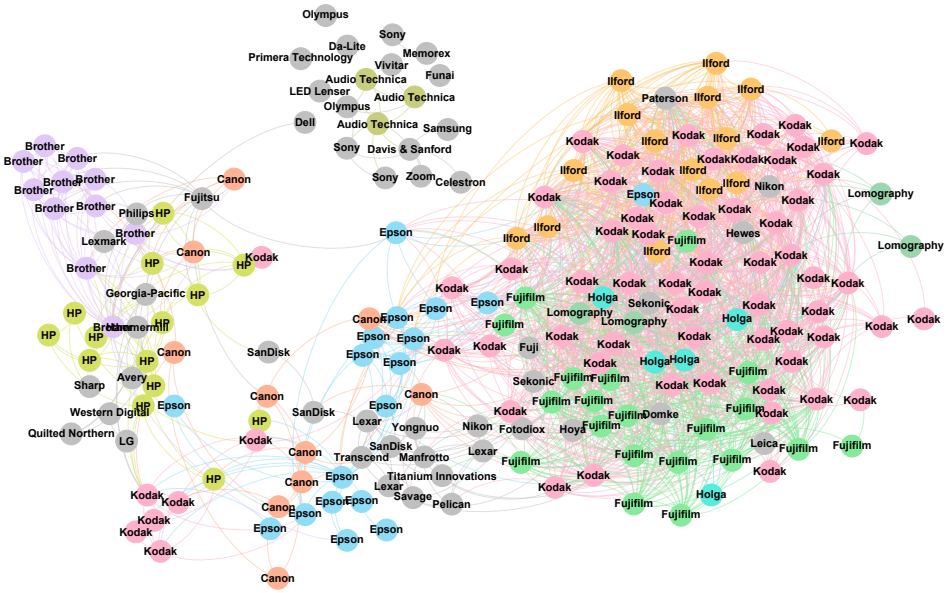
group consists of 61 nodes. The group consists of **Kodak** (33 products), **Fujifilm** (15 products), **Illford** (10 products), **Domke** (1 product), **Paterson** (1 product) and **Sekonic** (1 product) brands. The word cloud of the cohesive group is shown in Fig. 8c. This result revealed a strong co-purchasing tendency of Fujifilm and Ilford products with the most frequently purchased Kodak products, even in the most cohesive connected subgraph we could identify. The word cloud of Fig. 8 indicates the two major rivals are "at close range" the in the Amazon co-purchasing scenario. Given the fact that Kodak invented the digital camara but "failed to adapt by asking the wrong marketing question"[3] we can only conjecture if they could have advert disaster[4] with modern analytic techniques.

V. Comparison with other algorithms

We will compare the connected-cohesive groups found with other available algorithms. We quantify the performance in

[3]https://www.forbes.com/sites/avidan/2012/01/23/kodak-failed-by-asking-the-wrong-marketing-question/

[4]https://www.forbes.com/sites/chunkamui/2012/01/18/how-kodak-failed



(a) The Core Kodak Co-purchasing Network

(b) Word cloud of the core Kodak network

(c) Connected Cohesive Group

Fig. 8. Analysis of Kodak co-purchasing behaviour on Amazon. (a) The Core Kodak Co-purchasing Network (b) Word cloud of the Core Kodak network (c) Word cloud of the connected cohesive group indicating predominantly Fujifilm and Ilford as close competitors.

terms of the inter-rater agreement [16] between the ground-truth community detected by state-of-the-art algorithms and the identified connected-cohesive group. We calculate the Cohen's kappa coefficient statistics [3] to quantify the agreement of nodes between detected communities by a pair of different algorithms.

To compare the performance of our proposed algorithm with the community detection algorithms, we considered eight state-of-the-art algorithms implemented in the igraph[5] package: Edge betweenness (eb), Fastgreedy (fg), Infomap (im), Label propagation (lp), Leading eigenvector (lev), Multilevel (ml), Spinglass (sg), Walktrap (wt). We compare the agreement of our connected cohesive group with the communities detected by those algorithms. We calculated the Kappa score of agreement between the proposed algorithm and each of the other algorithm. The comparison of the inter-rater agreement is shown in Fig. 9.

From the inter-rater agreement of Kappa score, we found that the memetic algorithm of this contribution exhibited perfect agreement (Kappa=1.0) with Edge betweenness (eb) algorithm for all experiments. It also showed at least moderate agreement (Kappa = 0.41 to 0.60) with Fastgreedy (fg), Leading eigenvector (lev), Multilevel (ml) and Spinglass (sg) algorithms. There exists at least good agreement with Infomap (im), Label propagation (lp) and Walktrap (wt) algorithms. Clearly the community identified in the Dolphins network is perhaps notorious enough that the different algorithms all considered it of interest, which brings some doubts as to the use of further comparison between different techniques.

[5]http://igraph.org/

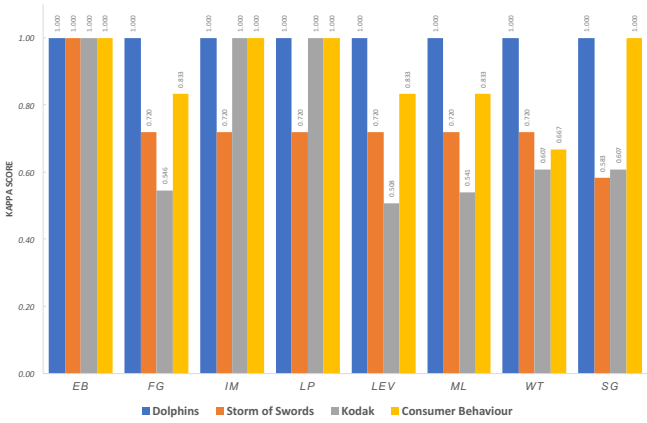


Fig. 9. Inter-rater agreement between proposed method and state-of-the-art community detection algorithms.

VI. Numerical Estimation of Running Times

We now present empirical runtime behaviour of the algorithm against increasing $|V|$ (the number of vertices in the graph). In theory, the $O(|V|^3)$ asymptotic running times of the fitness function [9] and the individual optimisation step dominate the running time of a given generation. To examine the empirical performance, we randomly generate graphs with varying increasing numbers of vertices (10, 50, 100, 500 and 1000) using the Watts-Strogatz model for small world graphs [23]. The running time of the algorithm for 10 runs

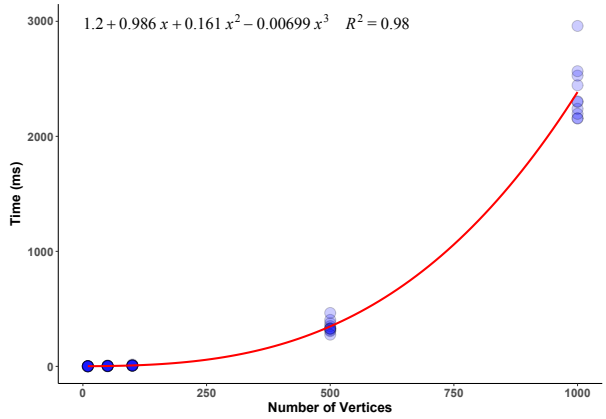


Fig. 10. Our MA running time exhibits a quadratic running timer behaviour with the number of vertices in the graph using the Watts-Strogatz model. In each case the MA was run for 500 generations (as in the real-world instances studied before), with an individual optimisation round performed every 100 generations. The fitted curve gives an approximation of the runtime constants in the complexity.

on each of the different graph size is shown in Fig. 10. From Fig. 10 we can see that the running time performance of the algorithm increases quadratically in terms of the number of vertices in the graph.

VII. Conclusion

We have presented the first memetic algorithm for community detection using the cohesion score. The cohesion score quantifies the natural community behaviour in the complex network. The memetic algorithm optimises the score to capture the largest and most cohesive single community. We also give theoretical results on the parameterized hardness of the connected cohesion approach. The memetic algorithm, along with the cohesion score, identifies the most cohesive group of vertices in the network and the outcome exhibited at least moderate agreement with state-of-the-art community detection algorithms. It showed the perfect agreement with the Edge betweenness algorithm. The approach can be extended further for identifying the largest community where it contains lightly connected multiple groups of the connected-cohesive community. However, the performance of the proposed community detection algorithm using triangle structure of the network relation showed good comparative performance and was able to found interesting semantic relations in the complex networks.

Acknowledgment

P. M. acknowledges funding of his research by the Australian Research Council (ARC, http://www.arc.gov.au/) grants Future Fellowship FT120100060 and Discovery Project DP140104183. The authors wish to thank Ademir C. Gabardo in connection to the Kodak's Co-purchasing Network dataset.

References

[1] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks, 2009.

[2] Alex Bavelas. A mathematical model for group structures. *Human organization*, 7(3):16–30, 1948.

[3] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

[4] Natalie de Vries, Jamie Carlson, and Pablo Moscato. A data-driven approach to reverse engineering customer engagement models: Towards functional constructs. *PLoS ONE*, 9(7), 2014.

[5] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[6] Santo Fortunato and Marc Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.

[7] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215 – 239, 1978.

[8] A. Friggeri, G. Chelius, and E. Fleury. Triangles to capture social cohesion. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 258–265, Oct 2011.

[9] Adrien Friggeri and Eric Fleury. Maximizing the Cohesion is NP-hard. October 2011.

[10] Ademir C. Gabardo, Regina Berretta, Natalie J. de Vries, and Pablo Moscato. *Where Does My Brand End? An Overlapping Community Approach*, pages 133–148. Springer International Publishing, Cham, 2017.

[11] Maoguo Gong, Qing Cai, Yangyang Li, and Jingjing Ma. An improved memetic algorithm for community detection in complex networks. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE, 2012.

[12] F. Huang, X. Li, S. Zhang, J. Zhang, J. Chen, and Z. Zhai. Overlapping community detection for multimedia social networks. *IEEE Transactions on Multimedia*, 19(8):1881–1893, Aug 2017.

[13] Sorn Jarukasemratana, Tsuyoshi Murata, and Xin Liu. Community detection algorithm based on centrality and node closeness in scale-free networks. *Transactions of the Japanese Society for Artificial Intelligence*, 29(2):234–244, 2014.

[14] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54:396–405, 2003.

[15] George RR Martin. *A storm of swords*. Bantam, 2000.

[16] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.

[17] Pablo Moscato. An introduction to population approaches for optimization and hierarchical objective functions: A discussion on the role of tabu search. *Annals OR*, 41(2):85–121, 1993.

[18] Leila M. Naeni, Regina Berretta, and Pablo Moscato. MA-Net: A reliable memetic algorithm for community detection by modularity optimization. In *Proceedings of the 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems, Volume 1*, pages 311–323. Springer, 2015.

[19] Leila M. Naeni, Hugh Craig, Regina Berretta, and Pablo Moscato. A novel clustering methodology based on modularity optimisation for detecting authorship affinities in shakespearean era plays. *PLOS ONE*, 11(8):1–27, 08 2016.

[20] Leila M. Naeni, Natalie Jane de Vries, Rodrigo Reis, Ahmed Shamsul Arefin, Regina Berretta, and Pablo Moscato. Identifying communities of trust and confidence in the charity and not-for-profit sector: A memetic algorithm approach. In *Big Data and Cloud Computing (BdCloud), 2014 IEEE Fourth International Conference on*, pages 500–507. IEEE, 2014.

[21] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.

[22] Stanislav Sobolevsky, Riccardo Campari, Alexander Belyi, and Carlo Ratti. General optimization technique for high-quality community detection in complex networks. *Physical Review E*, 90(1):012811, 2014.

[23] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1997.

[24] Kuai Wei and Michael J. Dinneen. Runtime analysis comparison of two fitness functions on a memetic algorithm for the clique problem. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, Beijing, China, July 6-11, 2014*, pages 133–140. IEEE, 2014.