

Graph Clustering with Surprise: Complexity and Exact Solutions^{*}

Tobias Fleck, Andrea Kappes, and Dorothea Wagner

Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Germany

Abstract. Clustering graphs based on a comparison of the number of links within clusters and the expected value of this quantity in a random graph has gained a lot of attention and popularity in the last decade. Recently, Aldecoa and Marín proposed a related, but slightly different approach leading to the quality measure *surprise*, and reported good behavior in the context of synthetic and real world benchmarks. We show that the problem of finding a clustering with optimum surprise is \mathcal{NP} -hard. Moreover, a bicriterial view on the problem permits to compute optimum solutions for small instances by solving a small number of integer linear programs, and leads to a polynomial time algorithm on trees.

1 Introduction

Graph clustering, i.e., the partitioning of the entities of a network into densely connected groups, has received growing attention in the literature of the last decade, with applications ranging from the analysis of social networks to recommendation systems and bioinformatics [8]. Mathematical formulations thereof abound; for an extensive overview on different approaches see for example the reviews of Fortunato [8] and Schaeffer [16].

One line of research that recently gained a lot of popularity is based on *null models*, the most prominent objective function in this context being the *modularity* of a clustering [13]. Roughly speaking, the idea behind this approach is to compare the number of edges within the same cluster to its expected value in a random graph that inherits some properties of the graph given as input.

In a wider sense, the measure called *surprise* that has recently been suggested as an alternative to modularity is also based on a null model, although, compared to modularity and its modifications [8], it uses a different tradeoff between the observed and expected number of edges within clusters. Surprise is used as a quality function in the tools UVCLUSTER and Jerarca to analyze protein interaction data [5,1]. The authors' main arguments for using surprise instead of modularity is that it exhibits better behavior with respect to synthetic benchmarks and, empirically, it does not suffer to the same extent from the *resolution limit* of modularity [9], i.e. the tendency to merge small natural communities into larger ones [2,3,4]. However, these results are hard to assess, since a meta-heuristic is used instead of directly optimizing the measure. It chooses among a

^{*} This work was partially supported by the DFG under grant WA 654/19-1.

set of clusterings produced by general clustering algorithms the one that is best with respect to surprise.

In this work, we take first steps towards a theoretical analysis of surprise. We show that the problem of finding a clustering with optimal surprise is \mathcal{NP} -hard in general and polynomially solvable on trees. Moreover, we formulate surprise as a bicriterial problem, which allows to find provably optimal solutions for small instances by solving a small number of integer linear programs.

Notation. All graphs considered are unweighted, undirected and simple, i.e. they do not contain loops or parallel edges. A clustering ζ of a graph $G = (V, E)$ is a partitioning of V . Let $n := |V|$ and $m := |E|$ denote the number of vertices and edges of G , respectively. If C is a cluster in ζ , $i_e(C)$ denotes the number of *intracluster edges* in C , i.e., the number of edges having both endpoints in C . Similarly, $i_p(C) := \binom{|C|}{2}$ is the number of vertex pairs in C . Furthermore, let $p := \binom{n}{2}$ be the number of vertex pairs in G , $i_p(\zeta) := \sum_{C \in \zeta} i_p(C)$ be the total number of intracluster vertex pairs and $i_e(\zeta) := \sum_{C \in \zeta} i_e(C)$ the total number of intracluster edges. If the clustering is clear from the context, we will sometimes omit ζ and just write i_p and i_e . To ease notation, we will allow binomial coefficients $\binom{n}{k}$ for all n and $k \in \mathbb{N}$. If $k > n$, $\binom{n}{k} = 0$ by definition.

2 Definition and Basic Properties

Let ζ be a clustering of a graph $G = (V, E)$ with i_e intracluster edges. Among all graphs labeled with vertex set V and exactly m edges, we draw a graph \mathcal{G} uniformly at random. The surprise $S(\zeta)$ of this clustering is then the probability that \mathcal{G} has at least i_e intracluster edges with respect to ζ . The lower this probability, the more *surprising* it is to observe that many intracluster edges within G , and hence, the better the clustering. The above process corresponds to an urn model with $i_p(\zeta)$ white and $p - i_p(\zeta)$ black balls from which we draw m balls without replacement. The probability to draw at least i_e white balls then follows a hypergeometric distribution, which leads to the following definition¹; the lower $S(\zeta)$, the better the clustering:

$$S(\zeta) := \sum_{i=i_e}^m \frac{\binom{i_p}{i} \cdot \binom{p-i_p}{m-i}}{\binom{p}{m}}$$

Basic Properties. For a fixed graph, the value of S only depends on two variables, i_p and i_e . To ease notation, we will use the term $S(i_p, i_e)$ for the value of a clustering with i_p intracluster pairs and i_e intracluster edges. The urn model view yields some simple properties that lead to a better understanding of how surprise behaves, and that are heavily used in the \mathcal{NP} -hardness proof.

Lemma 1. *Let i_e , i_p , p and m be given by a clustering, i.e. $0 \leq i_e \leq i_p \leq p$, $i_e \leq m$ and $m - i_e \leq p - i_p$. Then, the following statements hold:*

¹ This is the definition used in the original version [5]; later on, it was replaced by maximizing $-\log_{10} S(\zeta)$, which is equivalent with respect to optimum solutions.

- (i) $S(i_p, i_e + 1) < S(i_p, i_e)$.
- (ii) If $i_e > 0$, then $S(i_p - 1, i_e) < S(i_p, i_e)$
- (iii) If $p - i_p > m - i_e$, then $S(i_p + 1, i_e + 1) < S(i_p, i_e)$.

Proof. Statement (i) is obvious. Similarly, statement (ii) is not hard to see if we recall that $S(i_p - 1, i_e)$ corresponds to the probability to draw at least i_e white balls after replacing one white ball with a black one.

For statement (iii), we show that the number k_1 of m -element subsets of the set of all balls containing at least i_e white balls is larger than the number k_2 of m -element subsets containing at least $i_e + 1$ white balls after painting one black ball b white. Any subset A that contributes to k_2 also contributes to k_1 , as at most one ball in A got painted white. On the other hand, every m -element subset not containing b that contains exactly i_e white balls contributes to k_1 , but not to k_2 . As there are at least i_e white balls, and $p - i_p > m - i_e$ implies that there are at least $m - i_e + 1$ black balls, there is at least one subset with these properties. Hence $k_1 > k_2$, which is equivalent to $S(i_p + 1, i_e + 1) < S(i_p, i_e)$. \square

In other words, the value of surprise improves the more edges and the less vertex pairs within clusters exist. Moreover, part (iii) shows that if we increase the number of intracluster edges such that the number of *intracluster non-edges*, i.e., vertex pairs within clusters that are not linked by an edge, does not increase, this leads to a clustering with strictly smaller surprise. This immediately yields some basic properties of optimal clusterings with respect to surprise. Part (i) of the following proposition is interesting as it shows that optimal clusterings always fulfill the assumptions of Lemma 1(ii)-(iii).

Proposition 2. *Let $G = (V, E)$ be a graph that has at least one edge and that is not a clique and ζ be an optimal clustering of G with respect to surprise. Then,*

- (i) $i_e(\zeta) > 0$ and $p - i_p(\zeta) > m - i_e(\zeta)$
- (ii) $1 < |\zeta| < |V|$
- (iii) ζ contains at least as many intracluster edges as any clustering ζ' of G into cliques.
- (iv) Any cluster in ζ induces a connected subgraph.

Proof. (i): If $i_e(\zeta) = 0$ or $p - i_p(\zeta) = m - i_e(\zeta)$, it can be easily seen that $S(\zeta) = 1$. On the other hand, let us consider a clustering ζ' where each cluster contains one vertex, except for one cluster that contains two vertices linked by an edge e . As $m < p$, there is at least one labeled graph on V with m edges that does not contain e .

(ii): If $|\zeta| = 1$, $p - i_p(\zeta) = 0 = m - i_e(\zeta)$ and if $|\zeta| = |V|$, $i_e(\zeta) = 0$. The statement now follows from (i).

(iii): Let us assume that $i_e(\zeta) < i_e(\zeta')$. Lemma 1(ii) can be used to show that $S(\zeta) = S(i_p(\zeta), i_e(\zeta)) \geq S(i_e(\zeta), i_e(\zeta))$ and from Lemma 1(iii), it follows that $S(i_e(\zeta), i_e(\zeta)) > S(i_e(\zeta'), i_e(\zeta')) = S(\zeta')$.

(iv): Follows from Lemma 1(ii) and the fact that splitting a disconnected cluster into its connected components decreases the number of intracluster pairs and does not affect the number of intracluster edges. \square

Bicriterial View. From Lemma 1, it follows that an optimal solution with respect to surprise is *pareto optimal* with respect to (maximizing) i_e and (minimizing) i_p . Interestingly, this also holds for a simplification of modularity whose null model does not take vertex degrees into account and that was briefly considered by Reichardt and Bornholdt [15,14], although the tradeoff between the two objectives is different. Hence, an optimal clustering can be found by solving the following optimization problem for all $0 \leq k \leq m$ and choosing the solution that optimizes surprise.

Problem 3 (minIP). Given a graph G and an integer $k > 0$, find a clustering ζ with $i_e(\zeta) = k$, if there exists one, such that $i_p(\zeta)$ is minimal.

Unfortunately, the decision variant of minIP is \mathcal{NP} -complete even on bipartite graphs, as it is equivalent to the unweighted Minimum Average Contamination problem [12]. However, the formulation of minIP does not involve binomial coefficients and is thus in some aspects easier to handle. For example, in contrast to surprise, it can be easily cast into an integer linear program. We will use this in Sect. 4 to compute optimal solutions for small instances.

One might guess from the \mathcal{NP} -completeness of minIP that surprise minimization is also \mathcal{NP} -complete. However, there is no immediate reduction from minIP to the decision variant of surprise optimization, as the number of intra-cluster edges in an optimal clustering with respect to surprise is not fixed. In the following section, we will therefore give a proof for the hardness of finding a clustering with optimal surprise.

3 Complexity

We show \mathcal{NP} -completeness of the corresponding decision problem:

Problem 4 (SURPRISE DECISION (SD)). Given a graph G and a parameter $k > 0$, decide whether there exists a clustering ζ of G with $S(\zeta) \leq k$.

As S can be clearly evaluated in polynomial time, SD is in \mathcal{NP} . To show \mathcal{NP} -completeness, we use a reduction from EXACT COVER BY 3-SETS [10]:

Problem 5 (EXACT COVER BY 3SETS (X3S)). Given a set \mathcal{X} of elements and a collection \mathcal{M} of 3-element subsets of \mathcal{X} , decide whether there is a subcollection \mathcal{R} of \mathcal{M} such that each element in \mathcal{X} is contained in exactly one member of \mathcal{R} .

Let $I = (\mathcal{X}, \mathcal{M})$ be an instance of X3S. The reduction is based on the idea of implanting large disjoint cliques in the transformed instance that correspond to the subsets in \mathcal{M} . The size of these cliques is polynomial in $|\mathcal{M}|$, but large enough to ensure that they can neither be split nor merged in a clustering with low surprise. Hence, each of these cliques induces a cluster. The transformed instance further contains a vertex for each element in \mathcal{X}

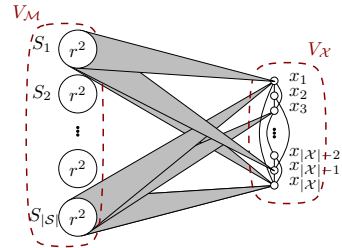


Fig. 1. Illustration for reduction

that is linked with the cliques corresponding to the subsets it is contained in. The idea is to show that in a clustering ζ with low surprise, each of these vertices is contained in a cluster induced by exactly one subset, and each cluster contains either three “element vertices” or none, which induces an exact cover of \mathcal{X} .

In the following, we will assume without loss of generality² that each element of \mathcal{X} belongs to at least one set in \mathcal{M} , hence $|\mathcal{X}| \leq 3|\mathcal{M}|$. We construct an instance $I' = (G, k)$ of SD in the following way. Let $r := 3|\mathcal{M}|$. First, we map each set M in \mathcal{M} to an r^2 -clique $C(M)$ in G . Furthermore, we introduce an $|\mathcal{X}|$ -clique to G , where each of the vertices $v(x)$ in it is associated with an element x in \mathcal{X} . We link $v(x)$ with each vertex in $C(M)$, if and only if x is contained in M . Let $V_{\mathcal{X}}$ be the set containing all vertices corresponding to elements in \mathcal{X} , and $V_{\mathcal{M}}$ the set of vertices corresponding to subsets. Fig. 1 illustrates the reduction, clearly, it is polynomial. In the proof, we will frequently use the notion *for large r , statement $A(r)$ holds*. Formally, this is an abbreviation for the statement that there exists a constant $c > 0$ such that for all $r \geq c$, $A(r)$ is true. Consequently, the reduction only works for instances that are larger than the maximum of all these constants, which suffices to show that SD is \mathcal{NP} -complete³.

Lemma 6. *Let ζ be an optimal clustering of G with respect to S . Then, $i_e(\zeta) \geq |\mathcal{M}| \cdot \binom{r^2}{2}$.*

Proof. Follows from Proposition 2(iii) and the fact that the clustering whose clusters are the cliques in $V_{\mathcal{M}}$ and the singletons in $V_{\mathcal{X}}$ is a clustering into cliques with $|\mathcal{M}| \cdot \binom{r^2}{2}$ intracluster edges. \square

Next, we give an upper bound on the number of *intracluster non edges*, i.e., vertex pairs within clusters that are not linked by an edge, in an optimal clustering of G . Its (rather technical) proof makes use of the asymptotic behavior of binomial coefficients and can be found in our technical report [7].

Lemma 7. *Let ζ be an optimal clustering of G with respect to surprise. Then, for large r , $i_p(\zeta) - i_e(\zeta) \leq \frac{r^4}{2}$.*

This can now be used to show that an optimal clustering of G is a clustering into cliques. We start by showing that the cliques in $V_{\mathcal{M}}$ cannot be split by an optimal clustering.

Lemma 8. *Let r be large and ζ be an optimal clustering of G with respect to S . Then, the cliques $C(M)$ in $V_{\mathcal{M}}$ are not split by ζ .*

Proof. Assume that there is at least one clique that is split by ζ . ζ induces a partition of each clique that it splits. We call the subsets of this partition the *parts* of the clique.

Claim 1: Every clique $C(M)$ contains a part with at least $r^2 - 6$ vertices.

² Otherwise, the instance is trivially non-solvable.

³ Smaller instances have constant size and can therefore be trivially solved by a brute-force algorithm.

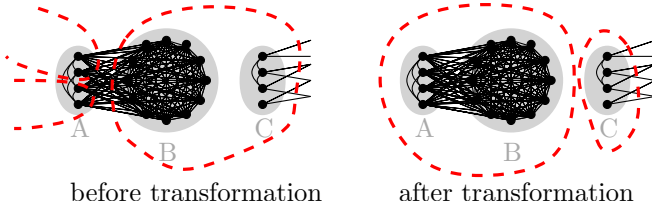


Fig. 2. Illustration for proof of Lemma 8

Proof of Claim 1: Assume that there is a clique K where each part has at most $r^2 - 7$ vertices. We can now greedily group the parts in two roughly equal sized regions, such that the smaller region contains at least 7 vertices and the larger region at least $r^2/2$ vertices. Let us look at the clustering we get by removing the vertices in K from their clusters and cluster them together. The vertices in K have in total $3r^2$ edges to vertices outside K and we gain at least $7/2 \cdot r^2$ new intracluster edges between the regions. Hence, the number of intracluster edges increases and the number of intracluster non-edges can only decrease. By Lemma 1(iii) and Lemma 1(i), it can be seen that this operation leads to a clustering with better surprise, which contradicts the optimality of ζ .

Let us now call the parts with size at least $r^2 - 6$ *large parts* and the other parts *small parts*.

Claim 2: No two large parts are clustered together.

Proof of Claim 2: Assume that there is a cluster that contains more than one large part. This cluster induces at least $(r^2 - 6)^2$ intracluster non-edges. For large r , this is larger than $r^4/2$ and Lemma 7 tells us that ζ was not optimal.

A simple counting argument now yields the following corollary.

Corollary: There must exist a large part B contained in a split clique whose cluster contains at most $|B| + 6$ vertices in $V_{\mathcal{M}}$.

Let B as in the corollary and A be the set of the vertices that are in the same clique as B but not in B and C be the set of vertices that are in the same cluster as B but not in B . Fig. 2 illustrates this case. We consider the clustering that we get by removing the vertices in A and B from their cluster and cluster them together. The number of vertices in A and C , respectively, is at most 6, and each of these vertices has at most 3 neighbors in $V_{\mathcal{X}}$. Hence, we lose at most 36 intracluster edges by this operation. On the other hand, we gain at least $r^2 - 6$ intracluster edges between A and B , thus, for large r , the number of intracluster edges increases. Again, the number of intracluster non-edges can only decrease and by Lemma 1(iii) and Lemma 1(i), we get that this operation leads to a clustering with better surprise, which contradicts the optimality of ζ . \square

Lemma 9. Let r be large and ζ be an optimal clustering of G with respect to S . Then, no two of the cliques in $V_{\mathcal{M}}$ are contained in the same cluster.

Proof. A cluster that contains two cliques in $V_{\mathcal{M}}$ induces at least r^4 intracluster non-edges. The statement now follows from Lemma 7. \square

Lemma 10. *Let r be large and ζ an optimal clustering of G with respect to S . Then, each $v(x)$ in $V_{\mathcal{X}}$ shares a cluster with a clique $C(M)$ such that $x \in M$.*

Proof. From Lemma 8 and Lemma 9 we know that ζ clusters the vertices in $V_{\mathcal{M}}$ according to the cliques we constructed. Assume that there is a vertex $v(x)$ in $V_{\mathcal{X}}$ that is not contained in any of the clusters induced by the sets containing x . Since each element in \mathcal{X} is contained in at least one set in \mathcal{M} , there exists a clique K in $V_{\mathcal{M}}$ that contains r^2 neighbors of $v(x)$. As $v(x)$ has at most $|\mathcal{X}| - 1$ neighbors in its own cluster, removing it from its cluster and moving it to the cluster of K increases the number of intracluster edges. On the other hand, x is linked with all vertices in its new cluster and thus, the number of intracluster non-edges cannot increase. Hence, this operation leads to a clustering with better surprise, which contradicts the optimality of ζ . \square

Theorem 11. *For large r , $I = (\mathcal{X}, \mathcal{M})$ has a solution if and only if there exists a clustering ζ of G with $S(\zeta) \leq k := \binom{p}{m}^{-1} \cdot \left(\frac{(|\mathcal{M}| \cdot \binom{r^2 + |\mathcal{X}|}{2}) - |\mathcal{M}| \cdot \binom{r^2}{2} - |\mathcal{X}| \cdot r^2 - |\mathcal{X}|}{(3|\mathcal{M}| - |\mathcal{X}|) \cdot r^2 + \binom{|\mathcal{X}|}{2} - |\mathcal{X}|} \right)$.*

Proof. \Rightarrow : Let R be a solution of I . R induces a clustering of G in the following way: For each $M \in \mathcal{M} \setminus R$ we introduce a cluster $C_M = C(M)$ and for each $M' \in R$ a cluster $C_{M'} = C(M') \cup \{v(x) \mid x \in M'\}$. As R is an exact cover, this is a partition ζ of the vertex set. It is $p = \binom{|\mathcal{M}| \cdot r^2 + |\mathcal{X}|}{2}$, $m = |\mathcal{M}| \cdot \binom{r^2}{2} + 3 \cdot |\mathcal{M}| \cdot r^2 + \binom{|\mathcal{X}|}{2}$ and $i_p(\zeta) = i_e(\zeta) = |\mathcal{M}| \cdot \binom{r^2}{2} + |\mathcal{X}| \cdot r^2 + |\mathcal{X}|$. It can be easily verified that $S(\zeta) = k$.

\Leftarrow : Let ζ be an optimal clustering of G with respect to surprise and assume that $S(\zeta) \leq k$. From Lemma 8, Lemma 9 and Lemma 10, we know that, for large r , we have one cluster for each set M in \mathcal{M} that contains $C(M)$ and each vertex $v(x)$ in $V_{\mathcal{X}}$ shares a cluster with a clique $C(M)$ such that $x \in M$. In particular, all clusters in ζ are cliques and hence $\binom{i_p(\zeta)}{i_e(\zeta)} = 1$. It follows that $\binom{p}{m} \cdot k \geq \binom{p}{m} \cdot S(\zeta) = \binom{p - i_e(\zeta)}{m - i_e(\zeta)}$. This term is strictly decreasing with $i_e(\zeta)$ and the above bound is tight for $i_e(\zeta) = |\mathcal{M}| \cdot \binom{r^2}{2} + |\mathcal{X}| \cdot r^2 + |\mathcal{X}| := t$. Hence, ζ contains at least t intracluster edges. The number of intracluster edges within $V_{\mathcal{M}}$ is exactly $|\mathcal{M}| \cdot \binom{r^2}{2}$ and the number of intracluster edges linking $V_{\mathcal{M}}$ with $V_{\mathcal{X}}$ is exactly $|\mathcal{X}| \cdot r^2$. The only quantity we do not know is the number of intracluster edges within $V_{\mathcal{X}}$, which we denote by $i_e(V_{\mathcal{X}})$. As $i_e(\zeta) \geq t$, it follows that $i_e(V_{\mathcal{X}}) \geq |\mathcal{X}|$. Thus, every vertex in $V_{\mathcal{X}}$ has in average two neighbors in $V_{\mathcal{X}}$ that are in the same cluster. On the other hand, vertices in $V_{\mathcal{X}}$ can only share a cluster if they are “assigned” to the same clique $C(M)$. As the sets in \mathcal{M} only contain three elements, vertices in $V_{\mathcal{X}}$ can only have at most two neighbors in $V_{\mathcal{X}}$ in their cluster. It follows that ζ partitions $V_{\mathcal{X}}$ into triangles. Hence, the set of subsets R corresponding to cliques $C(M)$ whose clusters contain vertices in $V_{\mathcal{X}}$ form an exact cover of \mathcal{X} . \square

We now have a reduction from X3S to SD that works for all instances that are larger than a constant $c > 0$. Hence, we get the following corollary.

Corollary 12. SURPRISE DECISION is \mathcal{NP} -complete.

To show that an optimal clustering with respect to surprise can be found in polynomial time if G is a tree, we consider the following problem MACP [12]:

Problem 13 (MACP). Given a graph $G = (V, E)$ together with a weight function $w : V \rightarrow \mathbb{Q}_{\geq 0}$ on V and a parameter k . Find a clustering ζ of G such that $m - i_e(\zeta) = k$ and $\sum_{C \in \zeta} (\sum_{v \in C} w(v))^2$ is minimal.

For the special case that $w(v)$ equals the degree of v and G is a tree, Dinh and Thai give a dynamic program that solves MACP for all $0 \leq k \leq m$ simultaneously [6]. This yields an $O(n^5)$ algorithm for modularity maximization in (unweighted) trees. In the context of surprise, we are interested in the special case that $w(v) = 1$ for all $v \in V$. The following conversion shows that this is equivalent to minIP with respect to optimal solutions:

$$i_p(\mathcal{C}) = \sum_{C \in \mathcal{C}} \frac{|C|(|C| - 1)}{2} = \frac{1}{2} \sum_{C \in \mathcal{C}} |C|^2 - \underbrace{\frac{1}{2} |V|}_{=\text{const.}} \quad (1)$$

The dynamic program of Dinh and Thai has a straightforward generalization to general vertex weights, which is polynomial in the case that each vertex has weight 1. For completeness, our technical report [7] contains a description of the dynamic program in this special case, together with a runtime analysis.

Theorem 14. *Let $T = (V, E)$ with $n := |V|$ be an unweighted tree. Then, a surprise optimal clustering of T can be calculated in $O(n^5)$ time.*

4 Exact Solutions

In this section, we give an integer linear program for minIP and discuss some variants of how to use this to get optimal clusterings with respect to surprise.

Linear Program for minIP. The following ILP is very similar to a number of linear programs used for other objectives in the context of graph clustering and partitioning, in particular, to one used for modularity maximization [6]. It uses a set of $\binom{n}{2}$ binary variables \mathcal{X}_{uv} corresponding to vertex pairs, with the interpretation that $\mathcal{X}_{uv} = 1$ iff u and v are in the same cluster. Let $\text{Sep}(u, v)$ be a minimum u - v vertex separator in G if $\{u, v\} \notin E$ or in $G' = (V, E \setminus \{u, v\})$, otherwise. The objective is to

$$\text{minimize} \quad \sum_{\{u, v\} \in \binom{V}{2}} \mathcal{X}_{uv} \quad (2)$$

such that

$$\mathcal{X}_{uv} \in \{0, 1\}, \quad \{u, v\} \in \binom{V}{2} \quad (3)$$

$$\mathcal{X}_{uw} + \mathcal{X}_{vw} - \mathcal{X}_{uv} \leq 1, \quad \{u, v\} \in \binom{V}{2}, w \in \text{Sep}(u, v) \quad (4)$$

$$\sum_{\{u, v\} \in E} \mathcal{X}_{uv} = k \quad (5)$$

Dinh and Thai consider the symmetric and reflexive relation induced by \mathcal{X} and show that Constraint (4) suffices to enforce transitivity in the context of modularity maximization [6]. Their proof solely relies on the following argument. For an assignment of the variables \mathcal{X}_{uv} that does not violate any constraints, let us consider the graph G' induced by the vertex pairs $\{u, v\}$ with $\mathcal{X}_{uv} = 1$. Now assume that there exists a connected component in G' that can be partitioned into two subsets A and B such that there are no edges in the original graph G between them. Setting $\mathcal{X}_{ab} := 0$ for all $a \in A, b \in B$ never violates any constraints and strictly improves the objective function. It can be verified that this argument also works in our scenario. Hence, a solution of the above ILP induces an equivalence relation and therefore a partition of the vertex set. As $\text{Sep}(u, v)$ is not larger than the minimum of the degrees of u and v , we have $O(nm)$ constraints over $O(n^2)$ variables.

Variants. We tested several variants of the approach described in Sect. 1 to decrease the number of ILPs we have to solve.

- *Exact(E)*: Solve m times the above ILP and choose among the resulting clusterings the one optimizing surprise.
- *Relaxed(R)*: We relax Constraint (5), more specifically we replace it by

$$\sum_{\{u,v\} \in E} \mathcal{X}_{uv} \geq k \quad (6)$$

Lemma 1(i) tells us that the surprise of the resulting clustering is at least as good as the surprise of any clustering with exactly k intracluster edges. Moreover, by Lemma 1(ii), if i_p is the value of a solution to the modified ILP, $S(i_p, k')$ is a valid lower bound for the surprise of any clustering with $k' \geq k$ intracluster edges. In order to profit from this, we consider all possible values for the number of intracluster edges in increasing order and only solve an ILP if the lower bound is better than the best solution found so far.

- *Gap(G)*: Similarly to the relaxed variant, we replace Constraint (5) by (6) and modify (2) to

$$\text{minimize} \quad \sum_{\{u,v\} \in \binom{V}{2}} \mathcal{X}_{uv} - \sum_{\{u,v\} \in E} \mathcal{X}_{uv} \quad (7)$$

By Lemma 1(ii), if g is the objective value and i_e the number of intracluster edges in a solution to the modified ILP, $S(k' + g, k')$ is a valid lower bound for the surprise of any clustering with $k' \geq k$ intracluster edges. Moreover, by Lemma 1(iii), we know that $S(i_e + g, i_e)$ is not larger than the surprise of any clustering with exactly k intracluster edges. Again, we consider all k in increasing order and try to prune ILP computations with the lower bound.

Case Study. Table 1 shows an overview of running times and the number of solved ILPs of the different strategies on some small instances. **karate**($n =$

Table 1. Number of linear programs solved and running times in seconds of successive ILP approach, different strategies.

	karate		lesmis		grid6		dolphins	
variant	ILP	t(s)	ILP	t(s)	ILP	t(s)	ILP	t(s)
Exact	79	51	255	1192	61	470	160	494
Relaxed	49	21	176	282	42	449	107	163
Gap	39	15	112	205	37	401	91	147

34, $m = 78$), **dolphins**($n = 62, m = 159$) and **lesmis**($n = 77, m = 254$) are real world networks from the website of the 10th DIMACS implementation Challenge⁴ that have been previously used to evaluate and compare clusterings, whereas **grid6**($n = 36, m = 60$) is a 2 dimensional grid graph. We used the C++-interface of **gurobi5.1** [11] and computed the surprise of the resulting clusterings with the help of the GNU Multiple Precision Arithmetic Library, in order to guarantee optimality. The tests were executed on one core of an AMD Opteron Processor 2218. The machine is clocked at 2.1 GHz and has 16 GB of RAM. Running times are averaged over 5 runs.

It can be seen that the gap variant, and, to a smaller extent, the relaxed variant, are able to prune a large percentage of ILP computations and thus lead to less overall running time. These running times can be slightly improved by using some heuristic modifications described and evaluated in [7].

Properties of Optimal Clusterings. Fig. 3 illustrates optimal clusterings with respect to surprise and modularity on the test instances, Table 2 summarizes some of their properties. We also included one slightly larger graph, **football**($n = 115, m = 613$), as it has a known, well-motivated *ground truth clustering* and has been evaluated in [2]. The surprise based clusterings contain significantly more and smaller clusters than the modularity based ones, being *refinements* of the latter in the case of **karate** and **lesmis**. Another striking observation is that the surprise based clusterings contain far more *singletons*, i.e. clusters containing only one vertex with usually low degree; this can be explained by the fact that surprise does not take vertex degrees into account and hence, merging low degree vertices into larger clusters causes larger penalties. It reconstructs the ground-truth clustering of the **football** graph quite well. This confirms the observations of Aldecoa and Marín based on heuristically found clusterings [2]; in fact, we can show that for **karate**, this clustering was already optimal.

5 Conclusion

We showed that the problem of finding a clustering of a graph that is optimal with respect to the measure surprise is \mathcal{NP} -hard. The observation that surprise

⁴ <http://www.cc.gatech.edu/dimacs10/>

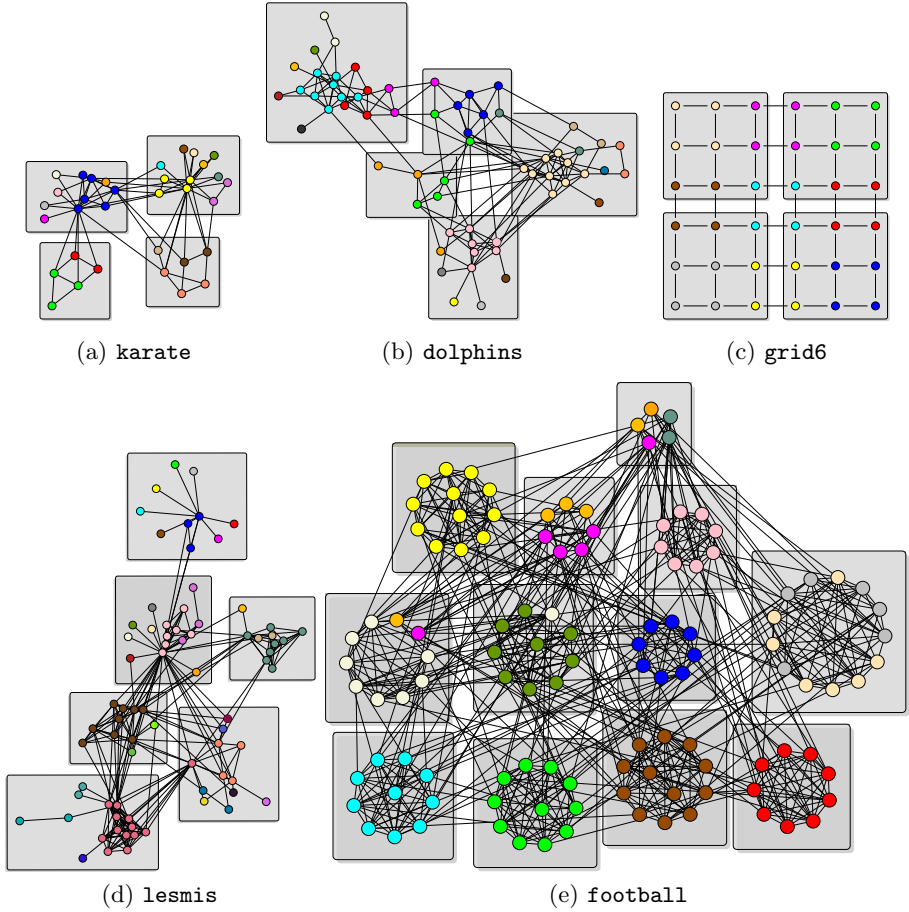


Fig. 3. Optimal clusterings with respect to surprise(colors) and, for (a) to (d), modularity(grouping). The grouping in (e) represents the *ground-truth clustering*, i.e. the mapping of teams to conferences.

Table 2. Properties of optimal clusterings with respect to surprise. S' denotes the surprise as defined by Aldecoa and Marín [2], i.e. $S'(\zeta) = -\log_{10} S(\zeta)$. S_o denotes the clustering with optimum surprise, S_h the heuristically found clusterings from [2], if this information was available, and M_o the modularity optimal clustering.

instance	i_e	i_p	$S(S_o)$	$S'(S_o)$	$S'(S_h)$	$ S_o $	$ S_h $	M_o
karate	29	30	$2,02 \cdot 10^{-26}$	25.69	25.69	19	19	4
grid6	36	54	$2,90 \cdot 10^{-29}$	28.54	-	9	-	4
dolphins	87	121	$9,93 \cdot 10^{-77}$	76.00	-	22	-	5
lesmis	165	179	$1,54 \cdot 10^{-184}$	183.81	-	33	-	6
football	399	458	$5,65 \cdot 10^{-407}$	406,25	-	15	15	10

is pareto optimal with respect to (maximizing) the number of edges and (minimizing) the number of vertex pairs within clusters yields a (polynomial time) dynamic program on trees. Furthermore, it helps to find exact solutions in small, general graphs via a sequence of ILP computations. The latter can be used to gain insights into the behavior of surprise, independent of any artifacts stemming from a particular heuristic. Moreover, optimal solutions are helpful to assess and validate the outcome of heuristics.

References

1. Aldecoa, R., Marín, I.: Jerarca: Efficient Analysis of Complex Networks Using Hierarchical Clustering. *PLoS ONE* 5, e11585 (2010)
2. Aldecoa, R., Marín, I.: Deciphering Network Community Structure by Surprise. *PLoS ONE* 6, e24195 (2011)
3. Aldecoa, R., Marín, I.: Exploring the limits of community detection strategies in complex networks. *Nature Scientific Reports* 3, 2216 (2013)
4. Aldecoa, R., Marín, I.: Surprise maximization reveals the community structure of complex networks. *Nature Scientific Reports* 3, 1060 (2013)
5. Arnau, V., Mars, S., Marín, I.: Iterative Cluster Analysis of Protein Interaction Data. *Bioinformatics* 21(3), 364–378 (2005)
6. Dinh, T.N., Thai, M.T.: Towards Optimal Community Detection: From Trees to General Weighted Networks. *Internet Mathematics* (accepted pending revision)
7. Fleck, T., Kappes, A., Wagner, D.: Graph Clustering with Surprise: Complexity and Exact Solutions. *ArXiv e-prints* (October 2013)
8. Fortunato, S.: Community detection in graphs. *Physics Reports* 486(3-5), 75–174 (2010)
9. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. *Proceedings of the National Academy of Science of the United States of America* 104(1), 36–41 (2007)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability. A Guide to the Theory of \mathcal{NP} -Completeness. W.H. Freeman and Company (1979)
11. I. Gurobi Optimization. Gurobi optimizer reference manual (2013)
12. Li, A., Tang, L.: The Complexity and Approximability of Minimum Contamination Problems. In: Ogihara, M., Tarui, J. (eds.) *TAMC 2011*. LNCS, vol. 6648, pp. 298–307. Springer, Heidelberg (2011)
13. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Physical Review E* 69(026113), 1–16 (2004)
14. Reichardt, J., Bornholdt, S.: Detecting Fuzzy Community Structures in Complex Networks with a Potts Model. *Physical Review Letters* 93(21), 218701 (2004)
15. Reichardt, J., Bornholdt, S.: Statistical Mechanics of Community Detection. *Physical Review E* 74(016110), 1–16 (2006)
16. Schaeffer, S.E.: Graph Clustering. *Computer Science Review* 1(1), 27–64 (2007)