Soft happy colourings and community structure of networks

Mohammad H. Shekarriz [a],[*], Dhananjay Thiruvady [a], Asef Nazari [a], Rhyd Lewis [b]

[a] *School of Information Technology, Deakin University, Geelong, VIC, Australia*
[b] *School of Mathematics, Cardiff University, Cardiff, Wales, United Kingdom*

ARTICLE INFO

ABSTRACT

For $0 < \rho \leq 1$, a ρ-happy vertex v in a coloured graph G has at least $\rho \cdot \deg(v)$ same-colour neighbours, and a ρ-happy colouring (aka soft happy colouring) of G is a vertex colouring that makes all the vertices ρ-happy. A community is a subgraph whose vertices are more adjacent to themselves than the rest of the vertices. Graphs with community structures can be modelled by random graph models such as the stochastic block model (SBM). In this paper, we present several theorems showing that both of these notions are related, with numerous real-world applications. We show that, with high probability, communities of graphs in the stochastic block model induce ρ-happy colouring on all vertices if certain conditions on the model parameters are satisfied. Moreover, a probabilistic threshold on ρ is derived so that communities of a graph in the SBM induce a ρ-happy colouring. Furthermore, the asymptotic behaviour of ρ-happy colouring induced by the graph's communities is discussed when ρ is less than a threshold. We develop heuristic polynomial-time algorithms for soft happy colouring that often correlate with the graphs' community structure. Finally, we present an experimental evaluation to compare the performance of the proposed algorithms thereby demonstrating the validity of the theoretical results.

1. Introduction

Graph colouring is a powerful technique for solving complex network-related problems, with applications varying from engineering to quantum physics and social science to cybersecurity. Depending on the problem's nature, various types of graph colouring have already been introduced. From the conventional *proper vertex colouring* (Lewis, 2021) to the recently defined *happy colouring* (Zhang and Li, 2015) (which is the subject of this paper), each of these versions appeared to be seminal in both theoretical and practical research.

Happy colouring in graphs is proposed for detecting *homophily* (McPherson et al., 2001) in social networks. For a k-coloured graph, Zhang and Li (2015) defined a vertex to be *happy* if all its neighbours have the same colour as its colour, and an edge is *happy* if both of its ends have the same colour. It is well-known that homophily in social networks can be expressed by their *community structures* (see for example Fani and Bagheri, 2017; Solomon et al., 2019; Ye et al., 2020), a structure that real-world networks often possess. In other words, homophily can be translated to grouping vertices into some sets in which they are connected with many edges to themselves and substantially fewer edges to the rest of the vertices. Each group is called a *community*, aka a *cluster* (Condon and Karp, 1999). A community is then a subgraph whose vertices have most of their neighbours inside

the community, and the problem has some similarities with finding the *minimum cut* in the network (Newman, 2012). Hence, by a community, we mean a group of vertices having an ample number of vertices that are adjacent together more than they are adjacent to the vertices of other communities. This way, singletons are not considered as a community, while they are valid vertex cuts.

Zhang and Li in Zhang and Li (2015) devised algorithms for maximising the number of happy vertices (MHV or k-MHV) or maximising the number of happy edges (MHE or k-MHE) in a graph with some vertices already precoloured with k colours. Additionally, they proved that MHV and MHE problems are solvable in polynomial time if $k \leq 2$, and they are NP-hard otherwise. Moreover, for $0 < \rho \leq 1$, they introduced ρ-happy colouring as follows: a vertex v is ρ-happy if its colour is the same as the colour of at least $\rho \cdot \deg(v)$ of its neighbours. The problem of maximising the number of ρ-happy vertices is called *soft happy colouring*. Furthermore, they devised two algorithms for this problem, namely Greedy-SoftMHV and Growth-SoftMHV.

To the best of our knowledge, the problem of soft happy colourings has remained dormant. We focus on this problem and our motivation to revisit this problem is its relation to the analysis of community structure in networks. However, conventional happy colouring (which is a special case of soft happy colouring when $\rho = 1$) and its two related
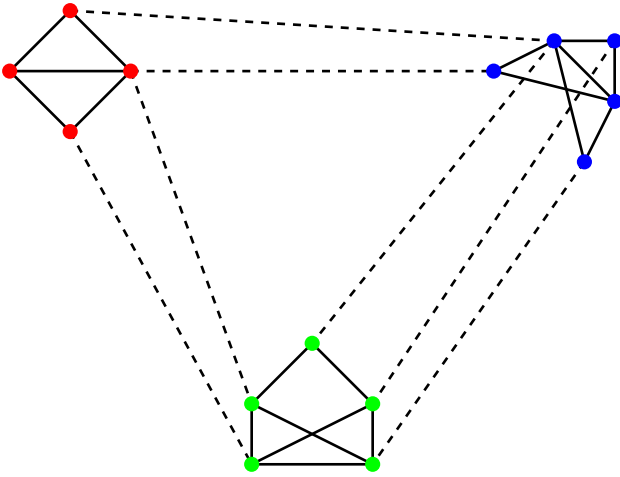

Fig. 1. A graph with 3 communities, represented by three colour classes. Dashed lines represent inter-community edges. Here the colouring is 0.5-happy.

problems, MHV and MHE, have already been the subject of several studies. For example, Aravind, Kalyanasundaram, and Kare (Aravind et al., 2016) proposed linear time algorithms for MHV and MHE for trees. Meanwhile, Agrawal et al. (2020) studied the complexity of (weighted) MHV and MHE based on the density measures of a graph.

A few studies have proposed exact, heuristic, metaheuristic and matheuristic approaches to tackle the (conventional) MHV problem (Lewis et al., 2019; Thiruvady et al., 2020; Thiruvady and Lewis, 2022). Lewis et al. (2019) proved bounds on the number of happy vertices and introduced a *construct, merge, solve, and adapt* algorithm — CMSA for short — for the MHV problem. Another study (Thiruvady et al., 2020), devised a *tabu search* metaheuristic which proves superior to other available algorithms, especially on very large networks. The later study by Lewis et al. (2021) explores a variant of the MHV problem, specifically the maximum induced happy subgraph in a network. They proposed some heuristic and metaheuristic algorithms and proved bounds on the maximum number of happy vertices. Thiruvady and Lewis (2022) continued this direction of study by developing approaches based on tabu search and evolutionary algorithms hybrid CMSA-tabu search — CMSA-TS for short — and showed that CMSA-TS is almost always the most effective approach for solving the MHV problem among the methods they tried. More recently, Ghirardi and Salassa (2022) presented an algorithm to improve the quality of a solution given by any another algorithm. Zhao et al. (2023) considered the relation of k-MHV and k-MHE with the *maximum k-uncut problem* and proposed an algorithm with the time complexity of $\mathcal{O}(kn^2)$.

Finding a connection between community structure and soft happy colouring is straightforward. If the communities are non-overlapping (i.e., communities partition the vertex set), we can express them by colour classes, which is the colouring *induced by the graph's communities*. Consequently, this paper is about how communities of a graph can be considered as the ρ-happy colour classes. While the conventional happy colouring (i.e. $\rho = 1$) with more than one colour can never be achieved for a connected graph (Lewis et al., 2019), we show that communities can make the entire graph ρ-happy when $0 < \rho < 1$. Fig. 1 illustrates this for a graph having 3 communities. As we see, here the communities in the illustrated graph entirely match the 0.5-happy colour classes. On the other hand, if a complete ρ-happy colouring is known for a graph G, then one may take its colour classes as an indication of the graph's communities.

Section 2 presents the required background and terminology for the theoretical contribution of this paper. Specifically, we review the stochastic block model, one of the simplest random graph models for graphs with community structure. It should also be emphasised

that although we build our theorems and experimental tests based on this model, the relation between community structure and soft happy colouring can be applied to any other community structure model.

The theoretical contribution of this paper is presented in Section 3. Theorem 3.1 gives a sufficient condition on the model parameters of a graph G in the SBM so that the probability of its communities inducing a ρ-happy colouring is greater than a function. This enables us to find a threshold $0 \leq \xi \leq 1$ such that, with high probability, communities of G induce a ρ-happy colouring for $\rho \leq \xi$. Moreover, the asymptotic property of the induced colouring is detected in Theorem 3.3 as the probability of ρ-happiness of this colouring approaches 1 when the number of vertices tends to infinity. The validity of these results is also discussed by experimental analysis in Section 3 as well as in Section 5.

In Section 4, four heuristic algorithms are discussed for finding ρ-happy colouring. One of them, Local Maximal Colouring, is new and has shown a high correlation with the graphs' community structure. We report experimental results considering communities inducing colour classes in Section 5 and carry out experiments on four algorithms for several randomly generated graphs.

2. Nomenclature

In this paper we use "graphs" and "networks" interchangeably, meaning simple unweighted finite graphs. Usually, a graph is denoted by its vertex set and its edge set, i.e. $G = (V(G), E(G))$. When two vertices u and v are adjacent, we write $u \sim v$ or $uv \in E(G)$. For $k \in \mathbb{N}$, a k-colouring of a graph G is a function $c : V(G) \longrightarrow \{1, \ldots, k\}$. It is well known that a k-colouring partitions the vertex set into k parts $V_1, \ldots, V_k$ while conversely, every k-partition of the vertex set induces a k-colouring which is unique up to permutation of colours. So, when colour orderings are unimportant for us, we write $c = \{V_1, \ldots, V_k\}$. A partial k-colouring is a function $c : S \longrightarrow \{1, \ldots, k\}$ where $S \subset V(G)$. By assigning another colour, say $k+1$, to the uncoloured vertices, every partial k-colouring becomes a $(k + 1)$-colouring.

For a vertex v, the set of its neighbours is denoted by $N(v)$. The degree of a vertex v, denoted by $\deg(v)$, is the number of edges incident on it, i.e. $\deg(v) = |N(v)|$. The minimum and maximum degrees are respectively denoted by δ and Δ. When we speak about a vertex v in a set, a colour class, or a community of vertices, we define the degree of v inside that set, which we denote here by $\deg_{in}(v)$. Further, standard graph theoretical notations and definitions can be found in Diestel (2017) by Diestel.

2.1. Happy colouring

We now formally define a ρ-happy colouring of a graph. It should be said that these definitions have their origins in the very first paper on the subject of happy colourings (Zhang and Li, 2015).

Definition 2.1. Let G be a graph, $0 \leq \rho \leq 1$ and $c : V(G) \longrightarrow \{1, \ldots, k\}$ for $k \in \mathbb{N}$ be a vertex colouring. A vertex $v \in V(G)$ is called ρ-*happy* if at least $\rho \cdot \deg(v)$ of its neighbours have the same colour as $c(v)$. The colouring c is called ρ-happy if every vertex of G is ρ-happy.

Immediately, we can see that 1-happy colouring is the conventional happy colouring, while every vertex colouring is a 0-happy colouring. For practical purposes, we may assume $\rho \neq 0$. Moreover, because the number of vertices having the same colour in any neighbourhood is integral, a vertex v being ρ-happy is equivalent to having at least $\lceil \rho \cdot \deg(v) \rceil$ same-colour neighbours, see Fig. 2.
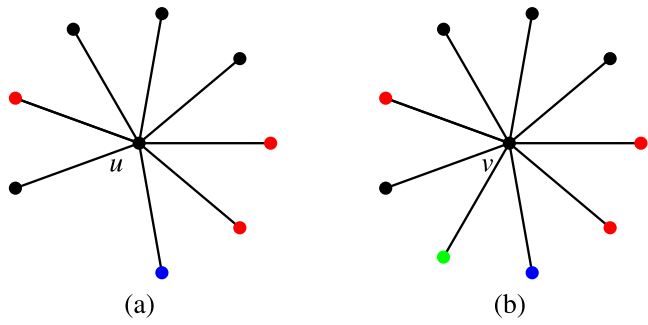
Fig. 2. An example of a ρ-happy colouring problem using $\rho = 0.5$. (a) The vertex u is 0.5-happy because its degree is 8 and at least $\lceil 0.5 \times 8 \rceil = 4$ of its neighbours have the same colour as its colour (black). (b) The vertex v is not 0.5-happy because its degree is 9 and $\lceil 0.5 \times 9 \rceil = 5$, but only 4 of its neighbours have the same colour as its colour (black).

2.2. Stochastic block model

For modelling general networks on n nodes, random graphs are often used. They are probability spaces consisting of n-vertex graphs. See Bollobás (2001) by Bollobás for further details on random graphs. There are several random graph models such as the *Erdős-Rényi model* (Erdős and Rényi, 1959), the *hierarchical network model* (Ravasz and Barabási, 2003), the *Watts–Strogatz model* (Watts and Strogatz, 1998), and the *stochastic block model* (SBM) (Jerrum and Sorkin, 1998). It should be noted that graphs with high probability in the Erdős-Rény model seldom have community structure. One reason is that in this model, the expected number of edges between any pair of vertices are equal. Consequently, this model is not suitable for studying graphs with community structure. In contrast, other mentioned models are designed so that their graphs, whose probabilities in their spaces are the highest, possess community structures.

One of the most popular random graph models for generating and modelling networks with community structure is the *Stochastic block model* which is introduced by Holland et al. (1983). The model is also known as the *planted partition model* (Condon and Karp, 1999, 2001), and is considered in numerous research papers (Abbe, 2018; Abbe et al., 2016; Abbe and Sandon, 2015; Jerrum and Sorkin, 1998; Lee and Wilkinson, 2019; Nadakuditi and Newman, 2012; Zhang et al., 2012). The model itself has also been subject to further generalisation such as the *degree-corrected stochastic block model* (Karrer and Newman, 2011) and the *Ball et al. model* (Ball et al., 2011). In this paper, we limit ourselves to the conventional form of the stochastic block model for our theoretical results (see Section 3) and experimental tests (see Section 5). We now summarise the main concepts and terminology.

In the simplest form of the SBM, denoted by $\mathcal{G}(n, k, p, q)$, a graph has n vertices and k communities. Each vertex belongs to exactly one of the k communities, hence the communities are not overlapping. Moreover, communities are also assumed to be of nearly the same size $\approx \frac{n}{k}$, where if n is not divisible by k, some communities have only one additional vertex compared to the others.

In the general case of the SBM, the probability of the existence of an edge uv depends on community membership of u and v. If $C_1, \ldots, C_k$ are the communities and $u \in C_i$ while $v \in C_j$ for $i, j \in \{1, \ldots, k\}$, then the probability of $uv \in E(G)$ is P_{ij}. In the simplest form of $\mathcal{G}(n, k, p, q)$, however, each P_{ij} is considered to be either p or q for $0 < q < p < 1$. In other words, we assume

$$P_{ij} = \begin{cases} p & \text{if } i = j \\ q & \text{otherwise.} \end{cases} \tag{1}$$

As a result, the expected degree of every vertex $v \in C_i$ is

$$\bar{d} = \mathbb{E}[\text{DEG}(v)] = \left(\frac{n}{k} - 1\right) p + \frac{k-1}{k} nq, \tag{2}$$

where $\text{DEG}(v)$ represents the random variable giving $\deg(v)$ for each $v \in V(G)$. When n is sufficiently large, we can assume $\bar{d} \simeq \frac{n}{k} p + \frac{k-1}{k} nq$.

It is evident that when p and q are very close to each other, the SBM becomes indistinguishable from the Erdős-Rény model and we might be unable to identify any community structure in the graph. For $k = 2$, Decelle et al. (2011) have also conjectured that detecting communities in large networks in the SBM is possible if and only if $(a - b)^2 > 2(a + b)$ for $a = np$ and $b = nq$. It is shown afterwards that when $(a - b)^2 > C(a + b)$ for a large enough positive integer C, the communities can be revealed by the spectral clustering algorithm (Coja-Oghlan, 2010). The conjecture was finally proven in Mossel et al. (2015). For the general case (when $k \geq 2$), there is a — distinguishability of community structure — phase transition result worth noting (Abbe and Sandon, 2015). In any case, the larger the value for $\frac{p}{q}$, the more distinguishable the community structure. Therefore, we assume, especially in our tests, that $\frac{p}{q}$ is large enough so that the communities are distinguishable. Further information about recent developments on the SBM can be found in Abbe (2018) by Abbe.

3. Soft happy colouring of graphs in the SBM

For any pair of vertices u and v of a graph G in the SBM, the adjacency probability is known. If u and v belong to the same community, the probability of u being adjacent to v is p while if they belong to different communities, they are adjacent with the probability q. Using this fact, one might like to know that in the induced colouring by communities, what the probability ρ-happiness of a vertex is. In other words, given an $0 < \varepsilon < 1$, we are interested in finding a condition that the probability of ρ-happiness of a vertex is at least $1 - \varepsilon$. Knowing this condition enables us to increase the probability of ρ-happiness of the induced colouring by the communities. The following theorem gives this condition.

Theorem 3.1. *Suppose that G is a graph modelled by the SBM, i.e., $G \in \mathcal{G}(n, k, p, q)$, n is a sufficiently large integer, $2 \leq k$, $0 < q < p < 1$, and $0 < \rho \leq 1$. Then, for $0 < \varepsilon < 1$, at least with the probability of $(1 - \varepsilon)^n$ the communities of G induce a ρ-happy colouring on G if*

$$q(k-1)(e^\rho - 1) + p(e^\rho - e) < \frac{k}{n} \ln(\varepsilon). \tag{3}$$

Proof. Let the communities of G be $C_1, \ldots, C_k$ and let v be an arbitrary vertex of the jth community of G. For $i \in \{1, \ldots, k\}$, let X_i be the random variable which gives the number of adjacent vertices to v in C_i. In other words,

$$X_i = \sum_{u \in C_i} A_{uv}, \tag{4}$$

where $A_{uv} \in \{0, 1\}$ are independent random variables denoting the number of edges between u and v. Then X_i have a binomial distribution and

$$\mathbb{E}[X_i] = \begin{cases} p\frac{n}{k} & \text{if } i = j \\ q\frac{n}{k} & \text{otherwise.} \end{cases}$$

For an arbitrary $t > 0$, we can now easily find the moment generating function

$$\mathbb{E}\left[e^{t \cdot X_i}\right] = \begin{cases} \left(1 - p + p \cdot e^t\right)^{\frac{n}{k}} & \text{if } i = j \\ \left(1 - q + q \cdot e^t\right)^{\frac{n}{k}} & \text{otherwise.} \end{cases}$$

Note that

$$\text{DEG}(v) = \sum_{i=1}^{k} X_i$$

while

$$\text{DEG}_{in}(v) = X_j$$

are random variables giving $\deg(v)$ and $\deg_{in}(v)$, respectively.

Now, by the Chernoff bound, for $t > 0$ we have

$$\Pr(\rho\,\mathrm{DEG}(v) - \mathrm{DEG}_{in}(v) > 0) \leq \mathbb{E}\left[e^{t(\rho\,\mathrm{DEG}(v) - \mathrm{DEG}_{in}(v))}\right].$$

Therefore,

$$\Pr(\rho\,\mathrm{DEG}(v) - \mathrm{DEG}_{in}(v) > 0) \leq \left(1 - p + p \cdot e^{t\rho}\right)^{\frac{n}{k}} \cdot \left(1 - p + p \cdot e^{t}\right)^{-\frac{n}{k}}$$
$$\cdot \prod_{i=1}^{k-1} \left(1 - q + q \cdot e^{t\rho}\right)^{\frac{n}{k}},$$

which means that

$$\Pr(\rho\,\mathrm{DEG}(v) - \mathrm{DEG}_{in}(v) > 0) \leq \left(1 - p + p \cdot e^{t\rho}\right)^{\frac{n}{k}} \cdot \left(1 - p + p \cdot e^{t}\right)^{-\frac{n}{k}}$$
$$\cdot \left(1 - q + q \cdot e^{t\rho}\right)^{\frac{k-1}{k}n}$$
$$\leq e^{\ln\left(\left(1 - p + p \cdot e^{t\rho}\right)^{\frac{n}{k}} \cdot \left(1 - p + p \cdot e^{t}\right)^{-\frac{n}{k}} \cdot \left(1 - q + q \cdot e^{t\rho}\right)^{\frac{k-1}{k}n}\right)}$$
$$\leq e^{\frac{n}{k} \cdot \left(\ln\left(1 - p + p \cdot e^{t\rho}\right) - \ln\left(1 - p + p \cdot e^{t}\right) + (k-1) \cdot \ln\left(1 - q + q \cdot e^{t\rho}\right)\right)}.$$

Since $e^x \geq 1 + x$ for $x \in \mathbb{R}$, we can deduce that

$$1 - p + p \cdot e^{t\rho} = 1 + p \cdot \left(e^{t\rho} - 1\right) \leq e^{p \cdot \left(e^{t\rho} - 1\right)}.$$

Consequently, using similar inequalities, we have

$$\Pr(\rho\,\mathrm{DEG}(v) - \mathrm{DEG}_{in}(v) > 0) \leq e^{\frac{n}{k} \cdot \left(p \cdot \left(e^{t\rho} - 1\right) - p \cdot \left(e^{t} - 1\right) + (k-1)q \cdot \left(e^{t\rho} - 1\right)\right)}$$
$$\leq e^{\frac{n}{k} \cdot \left(p \cdot \left(e^{t\rho} - e^{t}\right) + (k-1)q \cdot \left(e^{t\rho} - 1\right)\right)}.$$

Hence, for $\Pr(\rho\,\mathrm{DEG}(v) - \mathrm{DEG}_{in}(v) > 0) < \varepsilon$, it is sufficient to have

$$e^{\frac{n}{k} \cdot \left(p \cdot \left(e^{t\rho} - e^{t}\right) + (k-1)q \cdot \left(e^{t\rho} - 1\right)\right)} < \varepsilon, \tag{5}$$

or

$$p \cdot \left(e^{t\rho} - e^{t}\right) + (k-1)q \cdot \left(e^{t\rho} - 1\right) < \frac{k}{n}\ln(\varepsilon).$$

The inequality also holds when $t = 1$, resulting in the inequality we wanted to prove.

On the other hand, when Inequality (3) holds, the probability of v being ρ-unhappy is less than ε, and therefore, with the probability $1 - \varepsilon$, the communities of graph G induce a colouring on vertices of G in which v is ρ-happy. Since the values of A_{uv} are independent for all pairs $u, v \in V(G)$, it can be inferred that $\mathrm{DEG}(v)$ and $\mathrm{DEG}_{in}(v)$ are independent from $\mathrm{DEG}(u)$ and $\mathrm{DEG}_{in}(u)$, respectively. Consequently, $\rho\,\mathrm{DEG}(v) - \mathrm{DEG}_{in}(v)$ and $\rho\,\mathrm{DEG}(u) - \mathrm{DEG}_{in}(u)$ are independent too, which means that when Inequality (3) holds, with the probability of at least $(1 - \varepsilon)^n$, the communities of G induce a ρ-happy colouring on G. □

We now need additional notation to consider the implications of Theorem 3.1. Suppose that G is a graph modelled by the SBM, v is a vertex, $0 \leq \rho \leq 1$, and c is the vertex colouring induced by the communities of G. If c makes v ρ-happy, we write $v \in H_\rho$ and, if it makes the entire graph ρ-happy, we write $G \in H_\rho$. By Theorem 3.1, we know that $\Pr(v \in H_\rho) \geq 1 - \varepsilon$, while $\Pr(G \in H_\rho) \geq (1 - \varepsilon)^n$. Since this is dependent on the choice of ε, using Eq. (5) we define

$$\tilde{\varepsilon} = e^{\frac{n}{k} \cdot \left(p \cdot \left(e^{\rho} - e\right) + q(k-1) \cdot \left(e^{\rho} - 1\right)\right)}. \tag{6}$$

Therefore, it can be seen that $\Pr(v \in H_\rho) \geq 1 - \tilde{\varepsilon}$ and $\Pr(G \in H_\rho) \geq (1 - \tilde{\varepsilon})^n$, where $\tilde{\varepsilon}$ can be determined by ρ and the model's parameters, as per Eq. (6).

For a graph G in the SBM, i.e. $G \in \mathcal{G}(n, k, p, q)$, we can expect to find a ρ such that the communities of G present ρ-happy colour-classes for G. This is not always possible in the general case. For instance, the star graph $K_{1,m}$ admits no ρ-happy colouring for $0 < \rho \leq 1$ with more than one colour. However, for a graph in the SBM, we can find a ξ such that for $0 < \rho \leq \xi$, the graph's communities induce a colouring that makes almost all vertices ρ-happy. Of course, to have a high probability for happiness of an arbitrary vertex v in the colouring induced by communities of G, we must have

$$\mathbb{E}\left[\rho\,\mathrm{DEG}(v) - \mathrm{DEG}_{in}(v)\right] = \frac{n}{k}\left(\rho(p + (k-1)q) - p\right) < 0,$$

because otherwise, more than half of all vertices are ρ-unhappy. Therefore,

$$\rho\,(p + (k-1)q) < p.$$

Consequently, we have

$$\xi \leq \frac{p}{p + (k-1)q}. \tag{7}$$

The following theorem gives another upper bound for ξ.

Theorem 3.2. *Given n, k, p, and q, there exist a $0 \leq \xi \leq 1$ such that, for $\rho \leq \xi$, with high probability the communities of $G \in \mathcal{G}(n, k, p, q)$ induce a ρ-happy colouring.*

Proof. We need a $0 < \rho \leq 1$ so that Eq. (3) holds. Then, by Theorem 3.1, the probability of the existence of a ρ-happy colouring of G is at least $1 - n\varepsilon \leq (1 - \varepsilon)^n$, which will be close to 1 if ε is small enough. Hence, we choose a small $\varepsilon > 0$ for example $\varepsilon \leq \frac{1}{n^2}$.

We can find a bound on ρ based on Eq. (3) as follows. Since we want

$$p \cdot (e^{\rho} - e) + (k-1)q \cdot (e^{\rho} - 1) < \frac{k}{n}\ln(\varepsilon),$$

we can separate terms that contain e^{ρ} from the left-hand side of the above inequality to have

$$e^{\rho} \cdot (p + (k-1)q) < \frac{k}{n}\ln(\varepsilon) + pe + (k-1)q.$$

Therefore,

$$e^{\rho} < \frac{\frac{k}{n}\ln(\varepsilon) + pe + (k-1)q}{p + (k-1)q},$$

which means that

$$\rho < \ln\left(\frac{\frac{k}{n}\ln(\varepsilon) + pe + (k-1)q}{p + (k-1)q}\right). \tag{8}$$

Using Eqs. (7) and (8), we now get

$$\xi = \max\left\{\min\left\{\ln\left(\frac{\frac{k}{n}\ln(\varepsilon) + pe + (k-1)q}{p + (k-1)q}\right), \frac{p}{p + (k-1)q}\right\}, 0\right\}, \tag{9}$$

which has the desired property. □

Therefore, ξ is a threshold so that communities of the graph induce a ρ-happy colouring for $0 \leq \rho \leq \xi$. Note that we did not exclude the 0-happy colouring in Definition 2.1 since ξ can sometimes be 0 and therefore 0-happy colouring must be defined. To increase the probability that communities of the graph induce a ρ-happy colouring, we may consider $\rho \leq \frac{\xi}{2}$.

The threshold ξ (Eq. (9)) has shown to be useful in our experiments. We investigated whether ξ is a meaningful threshold for communities to induce ρ-happy colour classes, and for this purpose, we examined 100,000 graphs in the SBM with 500, 1000, 2000, 3000 and 5000 vertices with randomly chosen $k \in \{2, 3, \ldots, 20\}$, $0 < p \leq 1$, $0 < q \leq \frac{p}{2}$ and $0 < \rho \leq 1$, When $\rho \leq \frac{\xi}{2}$, the number of ρ-happy vertices — in the colouring induced by the graph's communities — is always high, evidence of this is provided in Fig. 3. Moreover, as it can be seen in Fig. 4, the minimum number of ρ-happy vertices in the same set of graphs are almost always very high when $\xi > \rho$ (yellow dots) while low numbers of ρ-happy vertices appear only when $\xi < \rho$ (purple dots). This tells us that ξ presents a useful threshold so that we can expect to have a complete ρ-happy colouring when $\rho < \xi$.

Moreover, when we know that the probability of a vertex being ρ-happy by the community-induced colouring is at least $1 - \varepsilon$, then the minimum probability of such colouring being ρ-happy depends not only on n but also on ε. If $n \to \infty$, then this probability goes to zero when ε remains constant. If ε decreases harmonically, the probability goes to a positive number between 0 and 1. For example, if $\varepsilon = \frac{1}{n}$, then $(1 - \frac{1}{n})^n \to e^{-1} \simeq 0.36787\ldots$ Meanwhile, since $n \to \infty$ results in $(1 - \frac{1}{n^t})^n \to 1$ for $t > 1$, the probability approaches 1 when n becomes sufficiently large when ε drops sharply, for example when $\varepsilon = n^{-2}$.
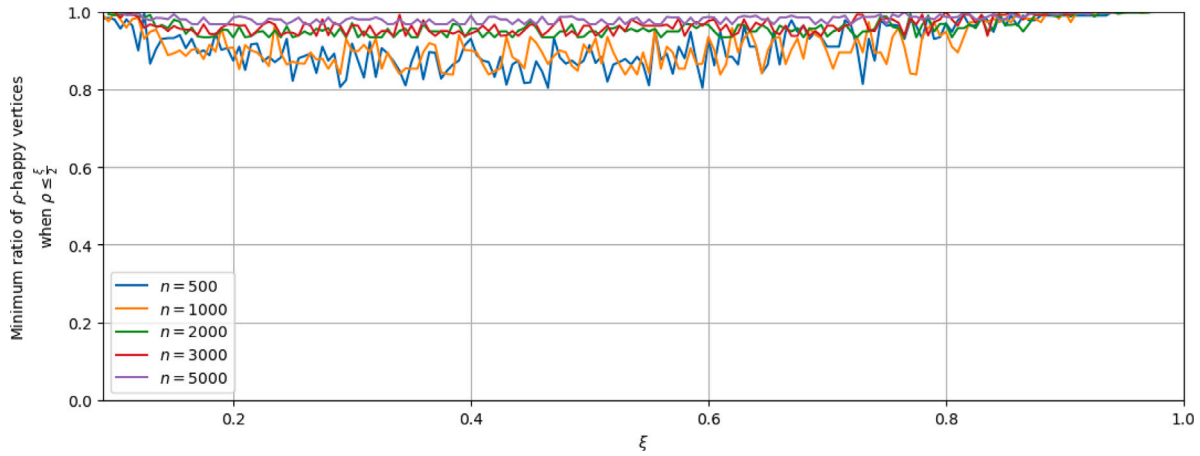
Fig. 3. This graph shows the minimum ratio of ρ-happy vertices when $\rho \leq \frac{\tilde{\xi}}{2}$ in the induced colouring by graphs communities. Test were performed over 20,000 randomly generated graphs with $n \in \{500, 1000, 2000, 3000, 5000\}$ (100,000 graphs in total) and parameters randomly selected from $2 \leq k \leq 20$, $0 < p \leq 1$, and $0 < q \leq \frac{p}{2}$.
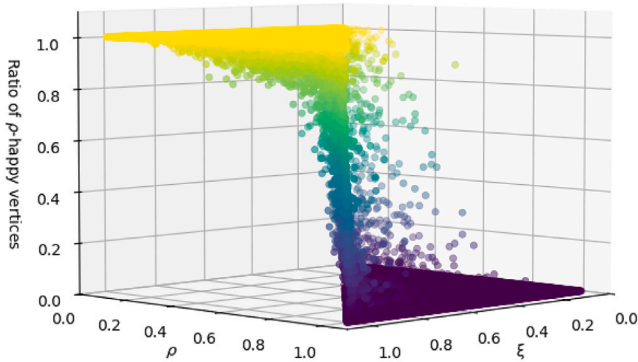

Fig. 4. The minimum ratio of ρ-happy vertices of the induced colouring by communities of 100,000 randomly generated precoloured graphs with $n \in \{500, 1,000, 2,000, 3,000, 5,000\}$ (20,000 graphs for each of these numbers) while other parameters are randomly chosen from $2 \leq k \leq 20$, $0 < p \leq 1$, and $0 < q \leq \frac{p}{2}$. It can be seen when $\rho < \xi$, the probability of having a large number of ρ-happy vertices is high (yellow areas).

To simplify the notations in Eq. (5), define

$$\varphi = \varphi(k, p, q, \rho) = \frac{1}{k} \cdot \left(p \cdot (e^\rho - e) + q(k-1)(e^\rho - 1) \right). \quad (10)$$

Hence,

$$\tilde{\varepsilon} = e^{n\varphi}.$$

It is evident that the lower the value of $n \cdot \varphi$, the higher the following probability

$$\Pr(G \in H_\rho) \geq (1 - \tilde{\varepsilon})^n. \quad (11)$$

This probability has a direct relationship with p and inverse relationships with k, q and ρ. Its relation with n is not as straightforward because sometimes increasing the fraction $\frac{n}{k}$ results in a higher probability of a vertex being ρ-happy, but reduces the probability that all the vertices remain ρ-happy. Our intention here is to consider the asymptotic interpretation of the relation of the number of vertices with ρ-happiness of the induced colouring of a graph in the SBM when other parameters remain constant. This is expressed in the following theorem.

Theorem 3.3. Let $0 < q < p < 1$, $k \in \mathbb{N} \setminus \{1\}$ be constants, and

$$\tilde{\xi} = \min \left\{ \ln \left(\frac{pe + (k-1)q}{p + (k-1)q} \right), \frac{p}{p + (k-1)q} \right\}.$$

Then, for $0 \leq \rho < \tilde{\xi}$ and $G \in \mathcal{G}(n, k, p, q)$, we have $\Pr(G \in H_\rho) \to 1$ when $n \to \infty$. In other words, the probability that the communities of G induce a ρ-happy colouring on its vertex set approaches 1 when n becomes large enough.

Proof. For any $0 < \varepsilon < 1$ we have

$$\lim_{n\to\infty} \ln \left(\frac{\frac{k}{n}\ln(\varepsilon) + pe + (k-1)q}{p + (k-1)q} \right) = \ln \left(\frac{pe + (k-1)q}{p + (k-1)q} \right).$$

As a result, $\lim_{n\to\infty} \xi = \tilde{\xi}$. And for $\rho < \tilde{\xi}$, we not only have $\mathbb{E}\left[\rho \operatorname{DEG}(v) - \operatorname{DEG}_{in}(v) \right] < 0$, but also

$$e^\rho < \frac{pe + (k-1)q}{p + (k-1)q}$$

$$\implies e^\rho(p + (k-1)q) - pe - (k-1)q < 0$$

$$\implies \varphi(k, p, q, \rho) < 0.$$

Therefore, as n grows, $1 - \tilde{\varepsilon}$ also increases (see Eq. (6)). In particular, since

$$\lim_{n\to\infty} n \cdot \varphi(k, p, q, \rho) = -\infty,$$

by Eq. (11), we have

$$\lim_{n\to\infty} \Pr(G \in H_\rho) \geq \lim_{n\to\infty} (1 - \tilde{\varepsilon})^n$$

$$= \lim_{n\to\infty} (1 - e^{n\varphi})^n$$

$$= \lim_{n\to\infty} e^{-ne^{n\varphi}}$$

$$= \operatorname{EXP} \left(\lim_{n\to\infty} -ne^{n\varphi} \right)$$

$$= \operatorname{EXP} \left(-\lim_{n\to\infty} \frac{n}{e^{-n\varphi}} \right).$$

Therefore, by l'Hospital's rule, we have

$$\lim_{n\to\infty} \Pr(G \in H_\rho) \geq \operatorname{EXP} \left(-\lim_{n\to\infty} \frac{1}{-\varphi e^{-n\varphi}} \right) = e^0 = 1.$$

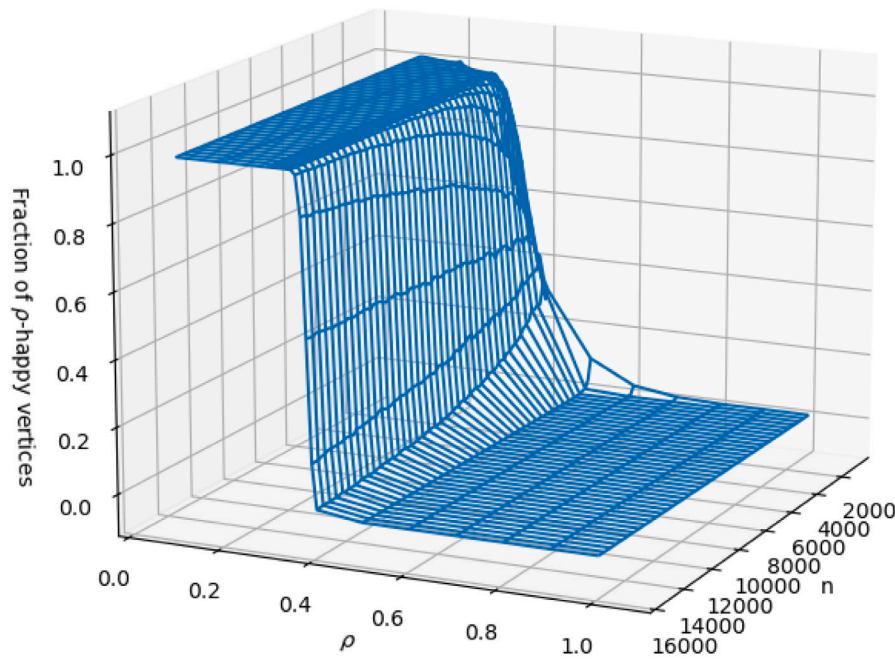Consequently, when $0 \leq \rho < \tilde{\xi}$, we have $\lim_{n\to\infty} \Pr(G \in H_\rho) = 1$. $\quad\square$

Fig. 5 illustrates how increasing the number of vertices n affects the number of ρ-happy vertices of the colouring induced by the communities of graphs in $\mathcal{G}(n, 20, 0.7, 0.06)$ (i.e, $k = 20$, $p = 0.7$ and $q = 0.06$). For $\rho < \tilde{\xi}$, where

$$\tilde{\xi} = \min \left\{ \ln \left(\frac{pe + (k-1)q}{p + (k-1)q} \right), \frac{p}{p + (k-1)q} \right\}$$

$$= \min\{\ln(1.6547), 0.3804\} = 0.3804,$$

we must have $\lim_{n\to\infty} \Pr(G \in H_\rho) = 1$. The phase transition of trends is obvious when $\rho \approx 0.38$ (see the almost flat trend in Fig. 5(a)).
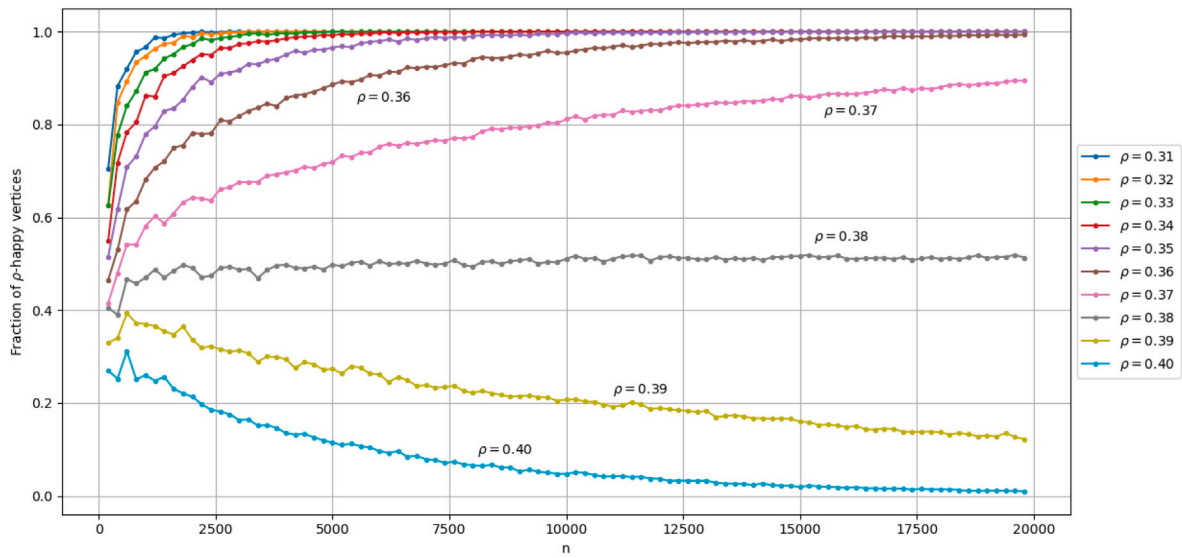
(a)

(b)

Fig. 5. The charts show the effects of increasing the number of vertices on the average number of ρ-happy vertices in the colouring induced by communities for graphs in the SBM. Here, $k = 20$, $p = 0.7$ and $q = 0.06$. The chart (a) shows an image of these effects when $200 \leq n \leq 20{,}000$ and $0.31 \leq \rho \leq 0.4$. Part (b) shows these effects when $0.1 \leq \rho \leq 1$. When $\rho \leq 0.37$ and n grows, the average number of ρ-happy vertices in the colouring induced by communities also grows, while the graphs are entire ρ-happily coloured by their communities when $\rho \leq 0.35$ and $n \geq 10{,}000$. Interestingly, this number decreases when $\rho > 0.38$. When $\rho = 0.38$, the chart is still increasing, but at a very slow rate.

4. Algorithms

In this section, we propose four heuristic algorithms to tackle the problem of maximising the number of ρ-happy vertices and examining the correlation of this problem and theoretical results from the previous section. In the proposed algorithms, we denote the number of ρ-happy vertices in a colouring c of a graph G by $H(G, c, \rho)$.

4.1. Greedy-SoftMHV

Algorithm 1 is a straightforward update of GreedyMHV from Zhang and Li (2015), the only difference being the change from "happiness" to "ρ-happiness". It turns a partial colouring into a k-colouring, where

only one colour is assigned to all the uncoloured vertices, ensuring the chosen colour makes as many vertices ρ-happy as possible. The algorithm is polynomial, i.e., $\mathcal{O}(km)$ (Carpentier et al., 2023) where m is the number of edges and k is the number of permissible colours.

The algorithm works as follows. As input, it takes a graph G, the parameter that determines the proportion of happiness ρ and a partial colouring c. The initialisation takes place in Lines 1–2, including assigning precolours to vertices in V, the set of uncoloured vertices U and $Max \leftarrow 0$ which records the maximum number of happy vertices. Between Lines 3–11, each colour is selected and assigned to the uncoloured vertices, and the total number of happy vertices is computed (Line 6). If an improvement is found, it is recorded via the

variables Max and i_{Max}. The output of the algorithm is a complete colouring $\tilde{c}$.

Algorithm 1 (Greedy-SoftMHV) Approximating the optimal solution of ρ-happy colouring (Zhang and Li, 2015)

    Input: $G, \rho, c : V' \longrightarrow \{1, \ldots, k\}$ $\triangleright V' \subsetneq V(G)$
    Output: $\tilde{c} : V(G) \longrightarrow \{1, \ldots, k\}$ $\triangleright$ a complete colouring of G

1: $\forall i \in \{1, \ldots, k\}, V_i \leftarrow \{v : c(v) = i\}$
2: $U \leftarrow V(G) - V', Max \leftarrow 0$ and $i_{Max} \leftarrow 0$ $\triangleright U =$ uncoloured vertices

3: **for** $i = 1, \ldots, k$ **do** $\triangleright$ try all colours
4:     **Append** members of U to V_i
5:     $\tilde{c} \leftarrow \{V_1, \ldots, V_k\}$
6:     **Calculate** $H(G, \tilde{c}, \rho)$ $\triangleright$ the number of ρ-happy vertices in $\tilde{c}$
7:     **if** $Max < H(G, \tilde{c}, \rho)$ **then**
8:         $Max \leftarrow H(G, \tilde{c}, \rho)$ and $i_{Max} \leftarrow i$
9:     **end if**
10:     **Remove** members of U from V_i
11: **end for**
12: **Append** members of U to $V_{i_{Max}}$
13: **Return** $\tilde{c} = \{V_1, \ldots, V_k\}$

4.2. Neighbour Greedy Colouring (NGC)

At the cost of increased computational time, we propose NGC, in which, unlike the previous approach, all uncoloured vertices do not always receive the same colour. Its initialisation and colour assignment are exactly like Greedy-SoftMHV. The only difference is that at each step, it only colours neighbours of already coloured vertices.

Like the Greedy-SoftMHV, the NGC takes a graph G, $0 \le \rho \le 1$, and a partial colouring c. Then, it initialises variables U as the set of uncoloured vertices, Max as the maximum number of ρ-happy vertices until now, and i_{Max} as the colour that makes this maximum. In Lines 3–12, it colours all the uncoloured vertices to see for which colour, say i_{Max}, the number of ρ-happy vertices is the largest. Afterwards in Lines 13–14, only the uncoloured neighbours of already coloured vertices by i_{Max} receive the colour i_{Max}. The procedure runs until all uncoloured vertices receive a colour, and then reports the complete colouring $\tilde{c}$.

The time complexity of the NGC is $d = \mathrm{diam}(G)$ times the time complexity of Greedy-SoftMHV, or $\mathcal{O}(dkm)$. This is because in the worst case, it has to repeat d times the processes of Greedy-SoftMHV in Lines 4–12 to colour all the vertices.

Moreover, the solution quality of the NGC is at least as good as that of Greedy-SoftMHV. This is because if the NGC ends with assigning only one colour to all non-precoloured vertices, then it gives the exact output as Greedy-SoftMHV. If NGC ends with assigning more than two colours to non-precoloured vertices, the number of ρ-happy vertices is higher than that of the output of the Greedy-SoftMHV.

4.3. Local Maximal Colouring

Our third proposed heuristic algorithm, Local Maximal Colouring (LMC), finds ρ-happy colour classes by examining local neighbourhoods of the uncoloured vertices, identifying the most frequently appearing colour, and assigning this colour to all uncoloured vertices in this neighbourhood.

The LMC's inputs are a graph G and a partial colouring c. Then, it employs two variables in Lines 2–3, namely U and C, as the set of uncoloured vertices and the set of coloured ones, respectively. Afterwards, an uncoloured vertex v, adjacent to at least one coloured vertex, is chosen. Then in Lines 6–7, q is the most frequent colour in $N(v)$, and v receives the colour q. The vertex v is now coloured, which is added

---

Algorithm 2 (Neighbour Greedy Colouring) Approximating the optimal solution of ρ-happy colouring

---

    Input: $G, \rho, c : V' \longrightarrow \{1, \ldots, k\}$ $\triangleright V' \subsetneq V(G)$
    Output: $\tilde{c} : V(G) \longrightarrow \{1, \ldots, k\}$ $\triangleright$ a complete colouring of G

1: $\forall i \in \{1, \ldots, k\}, V_i \leftarrow \{v : c(v) = i\}$
2: $U \leftarrow V(G) - V', Temp \leftarrow \emptyset, Max \leftarrow 0$ and $i_{Max} \leftarrow 0$ $\triangleright U =$ uncoloured vertices

3: **while** $U \ne \emptyset$ **do**
4:     **for** $i = 1, \ldots, k$ **do** $\triangleright$ try all colours
5:         **Append** members of U to V_i
6:         $\tilde{c} \leftarrow \{V_1, \ldots, V_k\}$
7:         **Calculate** $H(G, \tilde{c}, \rho)$ $\triangleright$ the number of ρ-happy vertices in $\tilde{c}$
8:         **if** $Max < H(G, \tilde{c}, \rho)$ **then**
9:             $Max \leftarrow H(G, \tilde{c}, \rho)$ and $i_{Max} \leftarrow i$
10:         **end if**
11:         **Remove** members of U from V_i
12:     **end for**
13:     $Temp \leftarrow U \cap N(V_{i_{Max}})$
14:     **Remove** members of $Temp$ from U and **Append** members of $Temp$ to $V_{i_{Max}}$
15: **end while**
16: **Return** $\tilde{c} = \{V_1, \ldots, V_k\}$

---

to the set of coloured vertices C and removed from the uncoloured vertices. This procedure continues until no uncoloured vertex remains.

Algorithm 3 (Local Maximal Colouring) Approximating the optimal solution of ρ-happy colouring.

    Input: $G, c : V' \longrightarrow \{1, \ldots, k\}$ $\triangleright V' \subsetneq V(G)$
    Output: $\tilde{c} : V(G) \longrightarrow \{1, \ldots, k\}$ $\triangleright$ a complete colouring of G

1: $\forall i \in \{1, \ldots, k\}, V_i \leftarrow \{v : c(v) = i\}$
2: $U \leftarrow V(G) - V'$ $\triangleright U =$ uncoloured vertices
3: $C \leftarrow V_1 \cup \ldots \cup V_k$ $\triangleright C =$ coloured vertices

4: **while** $U \ne \emptyset$ **do**
5:     **Choose** $v \in U \cap N(C)$
6:     $q \leftarrow$ the colour which appears the most in $N(v)$
7:     **Append** v to V_q, **Append** v to C and **Remove** v from U
8: **end while**
9: **Return** $\tilde{c} = \{V_1, \ldots, V_k\}$

The LMC has a time complexity of $\mathcal{O}(m)$, owing to the main loop (Lines 4–10), where the colours of neighbours of v are examined. This happens at most $2m$ times, i.e., twice for each edge. Experiments (Section 5) show that LMC can detect community structures close to the graphs' true community structures. Moreover, its time complexity, near-linear, is the lowest among the algorithms investigated. However, since it does not rely on ρ, its output can occasionally have fewer ρ-happy vertices than other algorithms.

4.4. Growth-SoftMHV

Algorithm 4 has been adapted from the Growth-SoftMHV algorithm of Zhang and Li (2015) to solve Soft happy colouring. The main process of this algorithm is determining the vertex classifications introduced in the following definition.

Definition 4.1 (*Zhang and Li (2015, altered from Definitions 8 and 9)*). Let G be a graph whose vertices are partially coloured by $c : V' \longrightarrow$

$\{1, \dots, k\}$, $V' \subsetneq V(G)$, $V_1, \dots, V_k$ are colour classes, U is the set of uncoloured vertices, $v \in V(G)$ and $0 < \rho \le 1$.

1. v is an H-vertex if it is coloured and ρ-happy.

2. v is a U-vertex if

    2.a. v is coloured, and
    2.b. v is destined to be ρ-unhappy, (i.e., $|N(v) \cap V_{c(v)}| + |N(v) \cap U| < \rho \cdot \deg(v)$)

3. v is a P-vertex if

    3.a. v is coloured,
    3.b. v has not been ρ-happy (i.e., $|N(v) \cap V_{c(v)}| < \rho \cdot \deg(v)$), and
    3.c. v can become an H-vertex (i.e., $|N(v) \cap V_{c(v)}| + |N(v) \cap U| \ge \rho \cdot \deg(v)$)

4. v is an L-vertex if it has not been coloured.

    4.1. v is an L_p-vertex if it is adjacent to a P-vertex,
    4.2. v is an L_h-vertex if

        4.2.a. v is not adjacent to any P-vertex,
        4.2.b. v is adjacent to an H-vertex or a U-vertex, and
        4.2.c. v can become ρ-happy, that is,

$$|N(v) \cap U| + \max\{|N(v) \cap V_i| : 1 \le i \le k\} \ge \rho \cdot \deg(v).$$

    4.3. v is an L_u-vertex if

        4.3.a. v is not adjacent to any P-vertex,
        4.3.b. v is adjacent to an H-vertex or a U-vertex, and
        4.3.c. v is destined to be ρ-unhappy, that is,

$$|N(v) \cap U| + \max\{|N(v) \cap V_i| : 1 \le i \le k\} < \rho \cdot \deg(v).$$

    4.4. v is an L_f-vertex if it is not adjacent to a coloured vertex.

The inputs of the Growth-SoftMHV algorithm are a graph G, ρ, and a partial k-colouring c. It first generates the colour classes (Line 1) and identifies all P, L_h, and L_u-vertices in Line 2, where it assigns them to sets $L = P \cup L_h \cup L_u$. Following this in Lines 4–8, it checks if there is a P-vertex v, it then chooses just enough neighbours of v and colours them by the colour of v so that v becomes ρ-happy. Then it recalculates the vertex classes P, L_h, L_u, and L. The above steps may generate other P-vertices, hence, the procedure repeats until no P-vertex remains.

After running out of P-vertices, the algorithm checks in Lines 9–15 if there is an L_h vertex v and chooses a colour i that appears the most among the neighbours of v. Then it colours v and some of its neighbours by the colour i so that v becomes a ρ-happy vertex. Next, because these colour assignments might generate new P or L_h-vertices, the vertex classification of P, L_h, L_u and L is updated. The algorithm repeats Lines 4–8 and 9–15 until no P or L_h-vertex is left.

If a complete colouring has not yet been obtained, at least one L_u-vertex must exist because the graph is assumed to be connected and L_f-vertices have no coloured neighbours. The algorithm in Lines 16–22 chooses an L_u-vertex v and a colour i that appears the most among neighbours of v, then the colour of v and some of its uncoloured vertices turns into the colour i. After recalculating the vertex classifications P, L_h, L_u, and L, the algorithm repeats Lines 4–22 until no P, L_h, or L_u vertex is left. Now, Every vertex must have a colour, so the algorithm reports the complete colouring. The time complexity of Growth-SoftMHV has been calculated by Carpentier et al. (2023) as $\mathcal{O}(mn)$.

5. Experimental evaluation

We conducted experiments on various graphs based on the SBM to investigate the performance of the proposed algorithms and validate

---

Algorithm 4 (Growth-SoftMHV) Approximating the optimal solution of soft happy colouring

> **Input:** G, ρ, $c : V' \longrightarrow \{1, \dots, k\}$ $\triangleright V' \subsetneq V(G)$
> **Output:** $\tilde{c} : V(G) \longrightarrow \{1, \dots, k\}$ $\triangleright$ a complete colouring of G

1: $\forall i \in \{1, \dots, k\}, V_i \leftarrow \{v : c(v) = i\}$
2: **Calculate** P, L_h, L_u and L $\triangleright$ Definition 4.1

3: **while** $L \ne \emptyset$ **do**
4:      **while** $P \ne \emptyset$ **do**
5:          **Choose** $v \in P$
6:          **Choose** $\lceil \rho \cdot \deg(v) \rceil - |N(v) \cap V_{c(v)}|$ from $N(v) \cap L$ and **Append** them to $V_{c(v)}$
7:          **ReCalculate** P, L_h, L_u and L $\triangleright$ Definition 4.1
8:      **end while**
9:      **while** ($P = \emptyset$ and $L_h \ne \emptyset$) **do**
10:         **Choose** $v \in L_h$
11:         **Choose** $i \in \{1, \dots, k\}$ such that $\forall t, |V_i \cap N(v)| \ge |V_t \cap N(v)|$
12:         **Append** v to V_i
13:         **Choose** $\lceil \rho \cdot \deg(v) \rceil - |N(v) \cap V_i|$ from $N(v) \cap L$ and **Append** them to V_i
14:         **ReCalculate** P, L_h, L_u and L $\triangleright$ Definition 4.1
15:      **end while**
16:      **while** ($P = \emptyset$ and $L_h = \emptyset$ and $L_u \ne \emptyset$) **do**
17:         **Choose** $v \in L_u$
18:         **Choose** $i \in \{1, \dots, k\}$ such that $\forall t, |V_i \cap N(v)| \ge |V_t \cap N(v)|$
19:         **Append** v to V_i
20:         **Choose** $\lceil \rho \cdot \deg(v) \rceil - |N(v) \cap V_i|$ from $N(v) \cap L$ and **Append** them to V_i
21:         **ReCalculate** P, L_h, L_u and L $\triangleright$ Definition 4.1
22:      **end while**
23: **end while**
24: **Return** $\tilde{c} = \{V_1, \dots, V_k\}$

---

our theoretical results. We developed a problem generator, from which we generated our sample graphs.[1] The problem generator, the source code for the algorithms and the graphs tested (stored in DIMACS format) are available online.[2] Here, we used a computer running 12×3.60 GHz Intel™Xeon™CPUs, 32 GB of RAM, and 512 GB memory to generate the problem instances and run all algorithms.

To test the algorithms, we generated partially coloured sample graphs considering the following parameters: number of vertices n, number of partitions (communities or colour classes) k, the edge probability inside communities p, the probability of inter-community edges q, the number of precoloured vertices per community pcc, a random seed for generating the graphs and the proportion of happiness ρ. Because increasing n impacts the running time of the algorithms, we split the test into two parts. First, we check the algorithms on 19,000 partially coloured graphs with 1000 vertices on several combinations of other parameters. This enables us to see the effects of changing other parameters more clearly. Afterwards, 28,000 partially coloured graphs on $200 \le n < 3000$ vertices are tested (10 graphs for each n), while other parameters are chosen at random. In both tests, we make sure that no two vertices in a community are precoloured with different colours. This is to prevent the induced precolouring from contradicting the community structure of the graphs. Otherwise, we cannot expect that the induced colouring by communities makes the entire vertex set ρ-happy. The settings for the two tests are as follows.
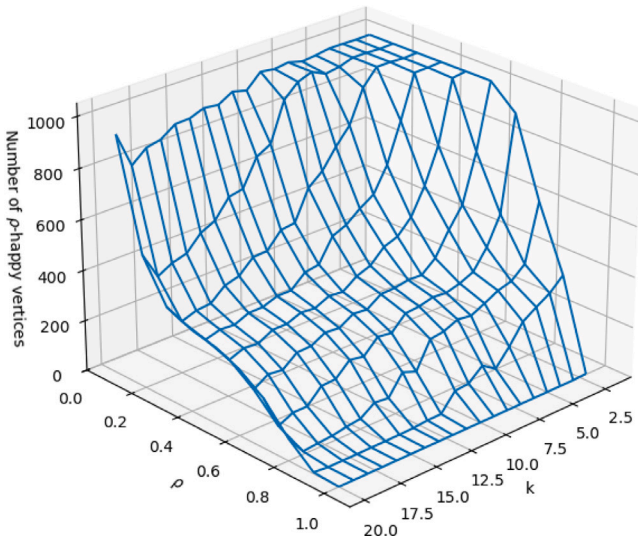

Fig. 6. The average number of ρ-happy vertices of the induced colouring by communities with respect to ρ and the number of colours.

Fig. 7. The average number of ρ-happy vertices of colourings induced by communities with respect to $\frac{p}{q}$. Here, the average is taken for all tested ρs, including the case $\rho = 1$ for which we almost always have a lot of unhappy vertices.

Test 1. Graphs for this test were generated with the settings $n = 1{,}000$, $k = 2, 3, \ldots, 20$, $p = 0.1, 0.2, \ldots, 0.9$, $q = 0.01, 0.11, 0.21, \ldots, \leq \frac{p}{2}$. For each combination of these values, four instances were generated. The number of precoloured vertices in each community varied from 1 to 10. Each of the 19,000 graphs is then tested for $\rho = 0.1, 0.2, \ldots, 1$. The time limit is set to 40 s. The results of running each of the algorithms 190,000 times in total are summarised in Section 5.1.

Test 2. For this test graphs are randomly generated with $200 \leq n < 3000$ vertices. For each n, 10 instances are generated, which makes it a set of 28,000 randomly generated graphs. For each graph, other parameters are randomly chosen over the same interval of the previous test i.e. $k \in \{2, 3, \ldots, 20\}$, $p \in (0, 1]$, $q \in (0, \frac{p}{2}]$ and $\rho \in (0, 1]$. The time limit for this test is 120 s. The results from this test are summarised in Section 5.2.

5.1. Test 1: graphs on 1000 vertices

Perhaps one of the most important test results for us, as anticipated by Theorem 3.1, is verifying that if k and ρ are small enough, then communities of networks in the SBM induce a ρ-happy colouring. The results of the first test fully supports this, see Fig. 6. On the other hand, when k or ρ increases, the induced colouring by graph communities cannot make all the vertices ρ-happy. Especially when $\rho = 1$, the induced colouring makes almost no vertex happy.

Fig. 7 illustrates how the average number of ρ-happy vertices in the colouring induced by the tested graphs' communities can be affected by the fraction $\frac{p}{q}$. As expected, when $\frac{p}{q}$ is large enough, the communities of the graphs are more straightforward to detect. Consequently, the average number of ρ-happy vertices in the colouring induced by the communities is directly related to $\frac{p}{q}$.

Fig. 8 demonstrates how changes in parameters affect the average number of ρ-happy vertices in the colouring induced by the graphs' communities. Fig. 8(a) indicates that this number increases when p increases. Fig. 8(b) emphasises how increasing q makes the average number of ρ-happy vertices drop. These two figures, and Figs. 6 and 8(f) as well, demonstrate the negative effect of increasing the number of communities, k, on the average number of ρ-happy vertices induced by these communities. Figs. 8(c) and 8(d), and Figs. 6 and 8(e) as well, illustrate a similar trend for ρ. The insignificant relation of the average number of ρ-happy vertices induced by graphs' communities and the
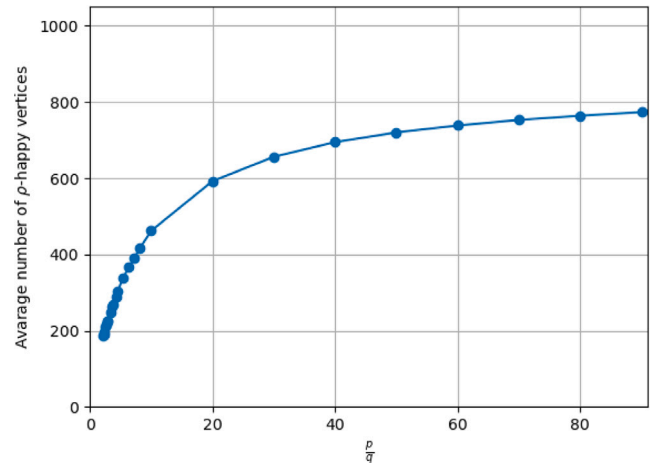
number of precoloured vertices per community can be seen in Figs. 8(e) and 8(f). All these trends agree with the theoretical results in Section 3.

Figs. 9–11, respectively, show the average CPU runtime, the average quality of performance and the average quality of community detection of Algorithms 1 to 4. The measure of performance is the number of ρ-happy vertices in their outputs, and their measure of community detection is the fraction of vertices whose colours in the output graph agree with their communities in the generated graph. Detecting communities is not explicitly an objective of these algorithms, nonetheless, all algorithms (especially LMC) can effectively detect communities.

Increasing the number of precoloured vertices per community does not significantly affect the run times, see Fig. 9(e). It negatively impacts the number of ρ-happy vertices detected by the algorithms (see Fig. 10(e)), while detecting communities improves (see Fig. 11(e)).

Interestingly, both algorithms Greedy-SoftMHV and NGC detect a similar number of ρ-happy vertices (see Fig. 10, especially Fig. 10(a)). However, they do not detect communities effectively (see Fig. 11, especially Fig. 11(a)). This is unsurprising since they assign only the one colour to all uncoloured vertices. The run times for Greedy-SoftMHV is substantially lower than NGC, and hence, using NGC cannot be justified for real-world applications.

Growth-SoftMHV requires large run times, but to make fair comparisons, we limit its CPU run time to 40 s (see Fig. 9). In this setting, all other algorithms find a higher number of ρ-happy colourings and typically detect communities more accurately (see Figs. 10 and 11, especially Figs. 10(d) and 11(c)).

LMC has the lowest run times. It does not detect ρ-happy vertices as effectively as Greedy-SoftMHV and NGC, but detects communities more accurately than the other algorithms (Figs. 10 and 11). For nearly 10% of the tests, it found a colouring where all the vertices are ρ-happy, far greater than 1.41% of the Growth-SoftMHV algorithm (see Fig. 12(a) and Table 1).[3] For Greedy-SoftMHV and NGC, the percentage of complete ρ-happy colouring of their outputs are inconsiderable, although NGC outperforms Greedy-SoftMHV. For certain instances, only LMC detects communities accurately (Fig. 12(b) and Table 1), despite not being an explicit objective.

Table 2 shows that LMC finds the highest number of ρ-happy vertices, 987 out of 1000, when communities induce ρ-happy colouring, i.e. when $\rho < \xi$. The best average number of ρ-happy vertices

---

[3] It should be noted that Growth-SoftMHV occasionally reached its time limit of 40 s, and only reports its partial solutions. Hence, we may observe improved performance if the time limit is not restricted.
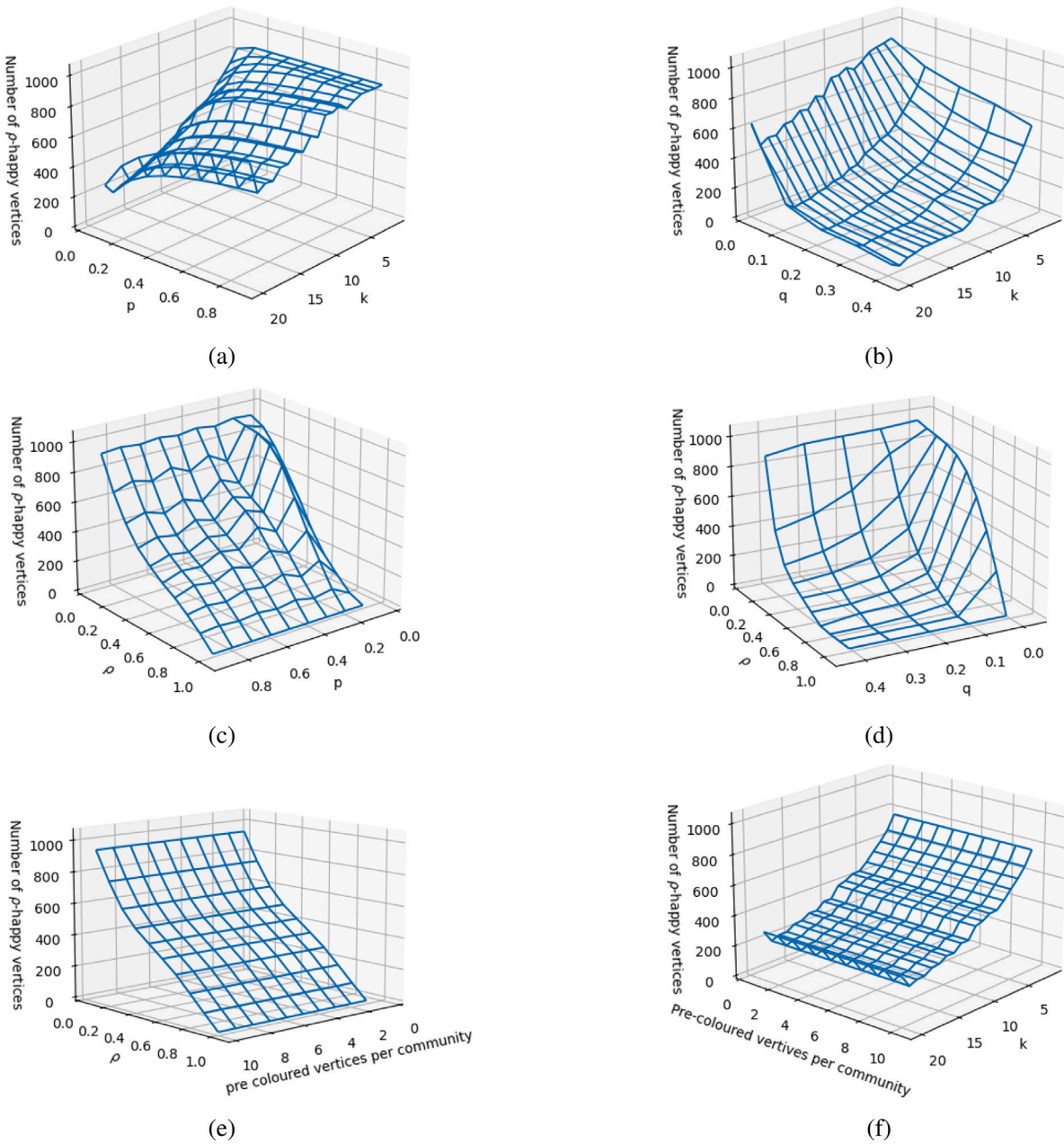
Fig. 8. These 3-dimensional charts show the average number of ρ-happy vertices when communities induce colour classes compared for (a) the parameters k and p, (b) for k and q, (c) for ρ and p, (d) for ρ and q, (e) for ρ and the number of precoloured vertices per community and (f) for k and the number of precoloured vertices per community.

across all the 190,000 runs belong to the NGC with 845 ρ-happy vertices. The LMC has the best accuracy of community detection whether communities induce ρ-happy colouring or not.

5.2. Test 2: graphs on different numbers of vertices

The four algorithms in Section 4, especially Growth-SoftMHV, are not designed to be run for large graphs. Therefore, if we are interested to see their scalability, we need to limit the number of total runs. For this, another test was necessary. We remind the reader again that we ran the algorithms over randomly generated graphs on $200 \leq n < 3000$ vertices for this test. Other parameters are randomly chosen over the same interval of the previous test (with an insist on $q \leq \frac{p}{2}$ to make

the communities distinguishable). For each n we have generated 10 instances. The results of 28,000 runs of the algorithms are summarised in the following paragraphs and figures.

First, we note that the data for this test agrees with the former test. For example, the average ratio of ρ-happy vertices in the induced colouring by the communities is given in Fig. 13, which is consistent with Fig. 6, especially when ρ and k are small enough, the induced colouring by the communities can make the entire vertex set ρ-happy.

The time limit for this part is 120 s. The only algorithm that occasionally reports incomplete solutions due to this time limit is the Growth-SoftMHV, which experiences the sharpest increase in its running time. As expected, increasing the number of vertices directly relates to the running time of the algorithms, which is obvious from Fig. 14(a). At the same time, LMC shows a small increase in its running time in comparison to the other algorithms.
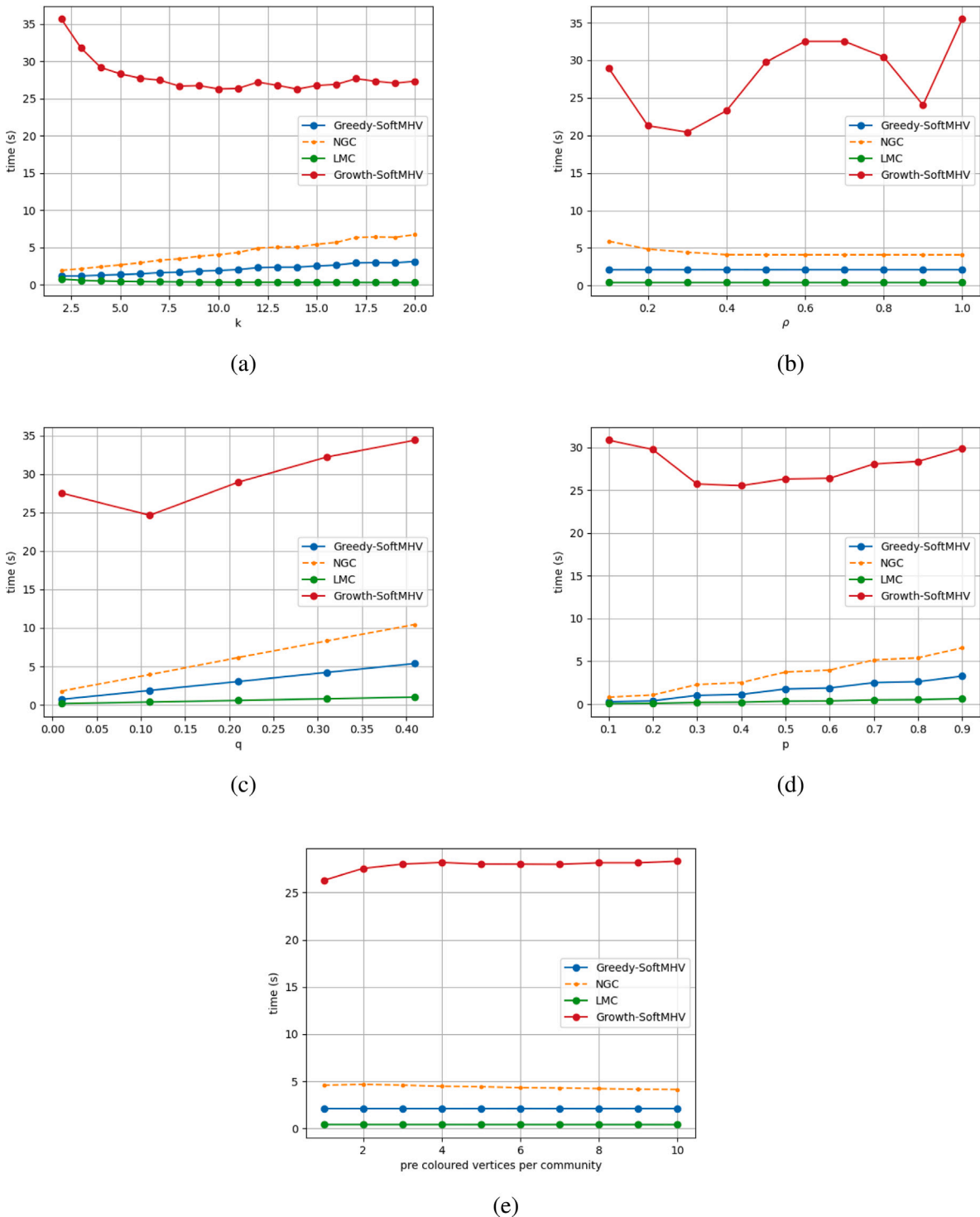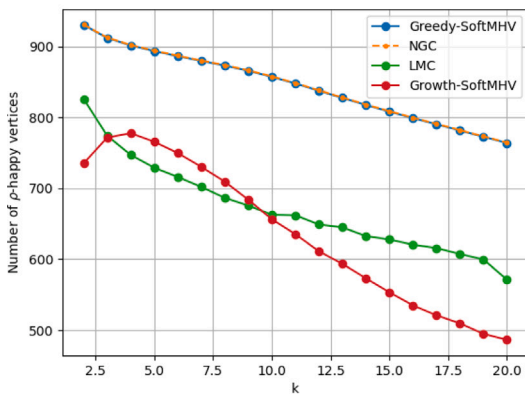
Fig. 9. Comparing the average running time of the four algorithms concerning the parameters (a) k, (b) ρ, (c) q, (d) p and (e) the number of precoloured vertices per community.
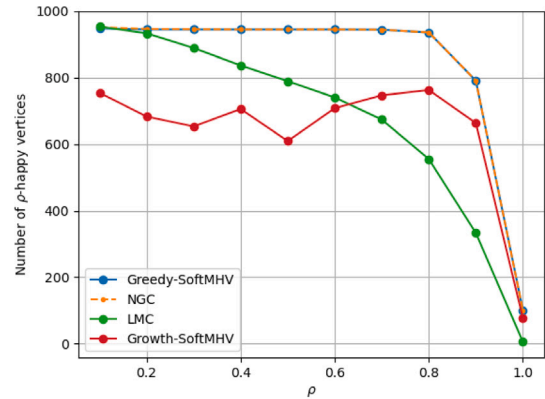
Like the former test for graphs on 1000 vertices, Greedy-SoftMHV and NGC perform almost similarly in finding ρ-happy vertices, see Fig. 14(b). The average number of ρ-happy vertices found by the Growth-SoftMHV divided by the total number of vertices drops as the number of vertices increases. This trend for LMC, which has the highest correlation with communities, remains increasing as Fig. 14(b) affirms. This complies with Theorem 3.3 that communities can find more ρ-happy vertices when n becomes larger.

Accuracy of community detection of Greedy-SoftMHV, NGC and Growth-SoftMHV drop sharply, according to Fig. 14(c). However, LMC maintains its community detection accuracy between 0.4 and 0.5.
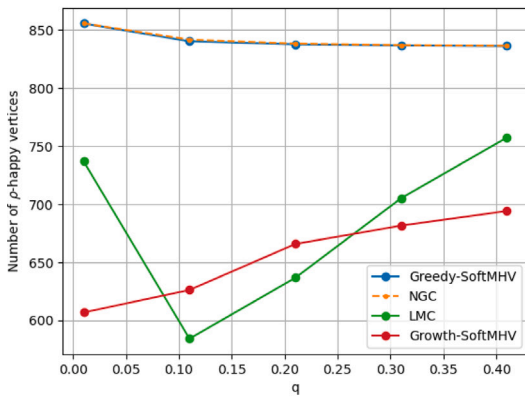
In general, according to Fig. 14, LMC is the most affordable algorithm for finding ρ-happy vertices, especially when another objective is community detection. In contrast, Growth-SoftMHV is costly and may perform poorly when the number of vertices is large.
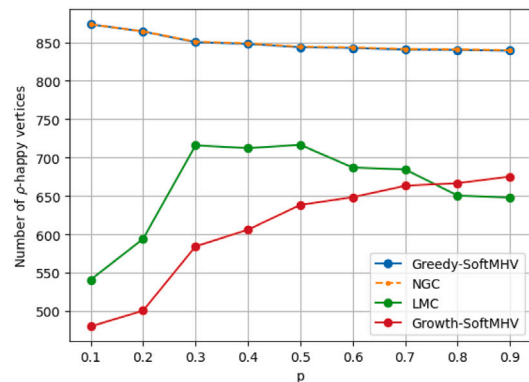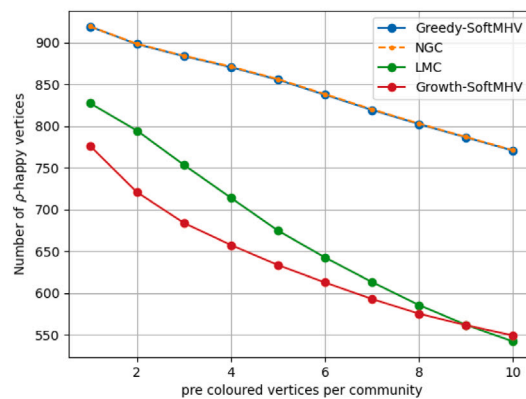
Fig. 10. Comparing the average number of ρ-happy vertices in the output of the four algorithms to the parameters (a) k, (b) ρ, (c) q, (d) p and (e) the number of precoloured vertices per community.

5.3. Performance of LMC

In this section, we briefly investigate the performance of LMC. As it can be seen in Line 5 of Algorithm 3, an uncoloured vertex is randomly chosen from the neighbourhood of already coloured vertices.

It is possible that choosing another vertex at a step affects the quality of the final solution. To investigate this, we performed LMC 10 times over each of the 28,000 graphs of the second test with $200 \leq n < 3000$ vertices, introduced in Section 5.2. The result of running LMC is reported in Fig. 15.
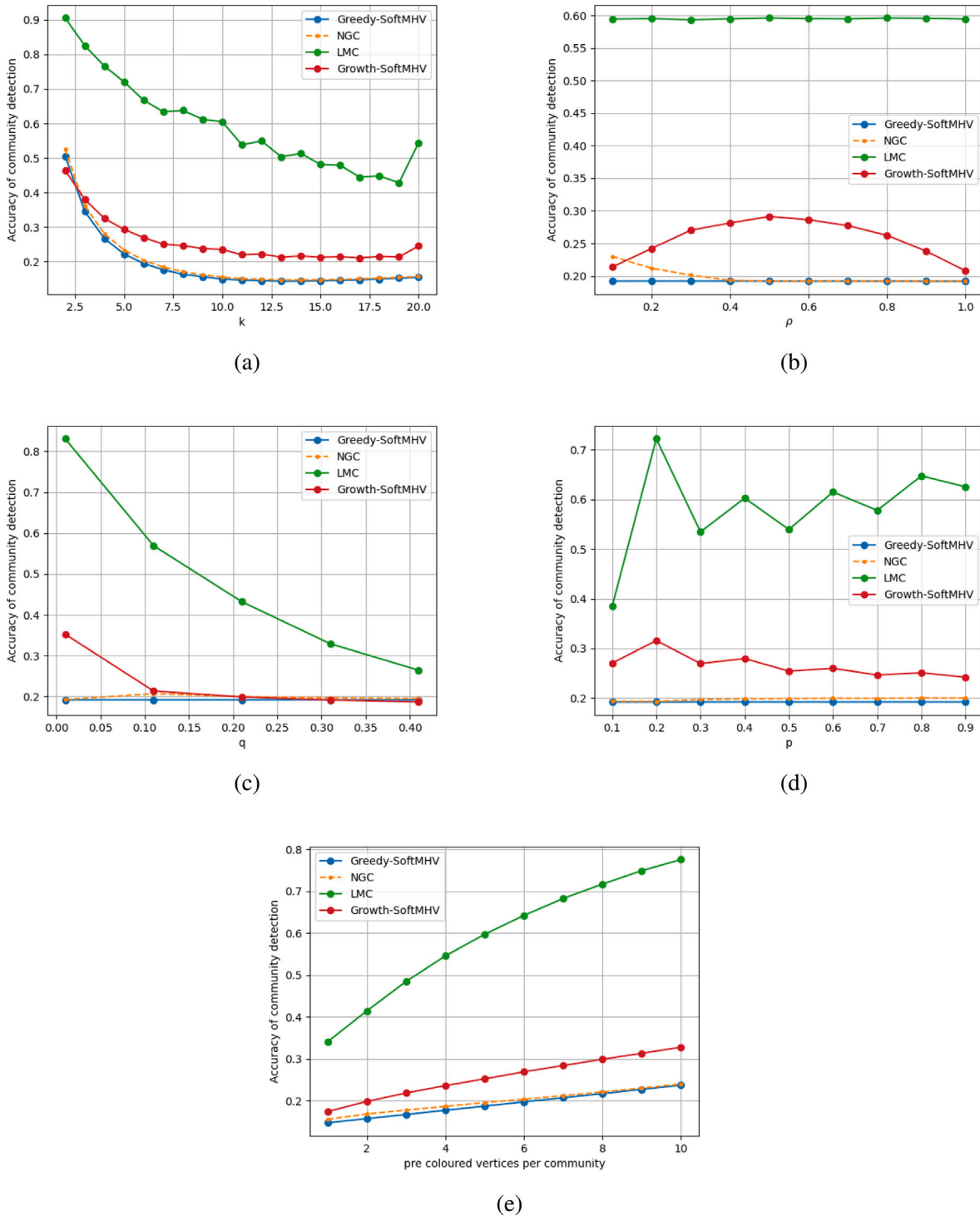
(a)

(b)

(c)

(d)

(e)

Fig. 11. Comparing the average quality of community detection of the four algorithms to changes of the parameters (a) k, (b) ρ, (c) q, (d) p and (e) the number of precoloured vertices per community. The quality of community detection is the ratio of vertices whose colours in the output graph agree with their communities in the generated graph.

Fig. 15(a) shows the minimum, mean and maximum of the number of ρ-happy vertices divided by the number of vertices. In this case, the minimum is almost always 0 while the maximum is almost always 1. The mean, however, is between 0.5 and 0.8, and as n increases, the mean also shows a tendency to rise. Figs. 15(b)–15(d) are for the same fraction with additional restrictions of $\rho > \xi$, $\rho < \xi$ and $\rho < \frac{\xi}{2}$,

respectively. When $\rho > \xi$, the maximum and minimum do not change, but the mean falls between 0.4 and 0.7. When $\rho < \xi$ (and when $\rho < \frac{\xi}{2}$), the maximum is again 1, but the minimum increases to around 0.4 (resp. 0.75) and the mean increases to around 0.95 (resp. 0.98), which again tends to be higher as n increases.
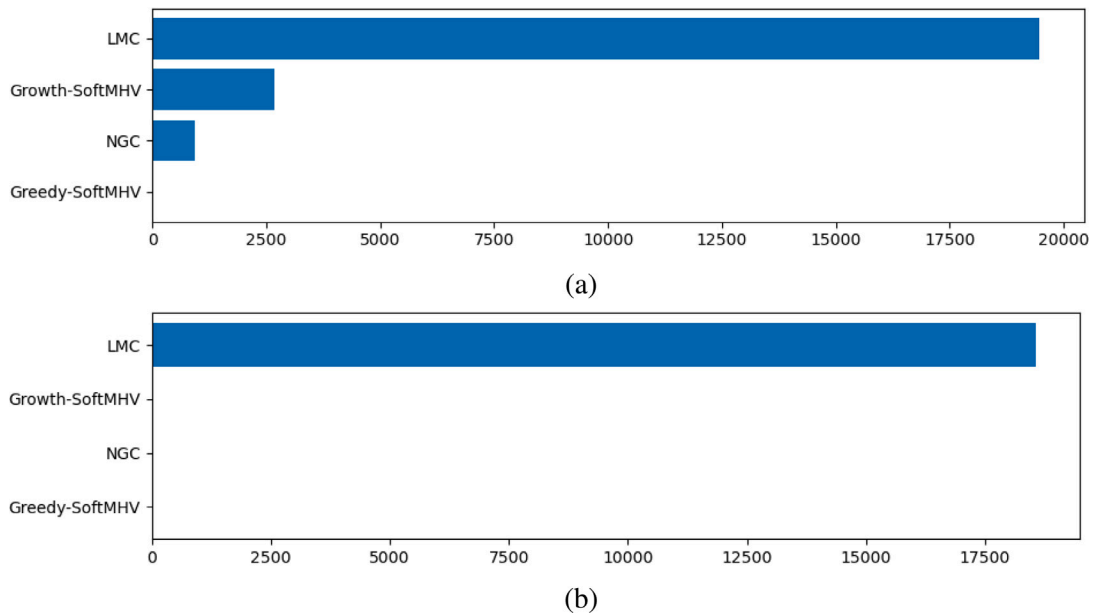
Fig. 12. (a) The number of times each algorithm generated a colouring that is ρ-happy out of 190,000 runs. It can be seen that LMC outperforms other algorithms in this regard. (b) The number of times that each algorithm detects communities accurately.

Table 1
The number of times and their percentage that the algorithms find colourings that make all the vertices ρ-happy versus the number of times and their percentage that they accurately detect communities, out of 190,000 runs.

Algorithm	# CH[a]	% CH[b]	# ACD[c]	% ACD[d]
Greedy-SoftMHV	6	0.00	0	0.00
Neighbour Greedy Colouring	921	0.48	0	0.00
Local Maximal Colouring	19,475	10.25	18,562	9.77
Growth-SoftMHV	2688	1.41	3	0.00
Communities induced colouring	27,510	14.48	190,000	100.00

[a] # CH = "the number of times it finds a complete ρ-happy colouring".
[b] % CH = "the percentage of # CH across 190,000 runs".
[c] # ACD = "the number of times it accurately detects communities".
[d] % ACD = "the percentage of # ACD across 190,000 runs".

Table 2
Average number of ρ-happy vertices and average accuracy of community detection for each algorithm are reported, in the entire test, when communities induce a ρ-happy colouring and when they do not.

Algorithm	H[a]	AC[b]	NAC[c]	C[d]	ACD[e]	NACD[f]
Greedy-SoftMHV	844	973	822	0.192	0.289	0.175
Neighbour Greedy Colouring	845	975	823	0.199	0.313	0.179
Local Maximal Colouring	671	987	617	0.594	0.840	0.553
Growth-SoftMHV	636	794	610	0.257	0.370	0.238

[a] H = "Average number of ρ-happy vertices across 190,000 runs".
[b] AC = "Average number of ρ-happy vertices when communities induce a complete ρ-happy colouring ($\rho < \xi$)".
[c] NAC = "Average number of ρ-happy vertices when communities do not induce a complete ρ-happy colouring ($\rho \geq \xi$)".
[d] C = "Average accuracy of community detection across 190,000 runs".
[e] ACD = "Average accuracy of community detection when communities induce a complete ρ-happy colouring ($\rho < \xi$)".
[f] NACD = "Average accuracy of community detection when communities do not induce a complete ρ-happy colouring ($\rho \geq \xi$)".

6. Conclusion

This paper theoretically and experimentally establishes a connection between the soft happy colouring of graphs and their community structure. Since conventional happy colouring is a special case of soft happy
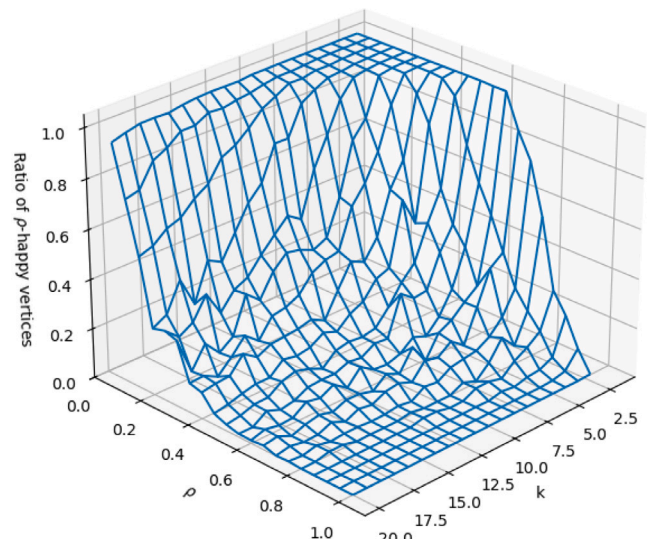

Fig. 13. The 3D chart of the average ratio of ρ-happy vertices in the induced colouring by communities, with ρ and k as variables. The data is from the second test, where the number of vertices is between 200 and 3000.

colouring, and the main intention of defining these sorts of colouring was homophily (which is a property of social networks explainable by community structures), soft happy colouring is a more powerful tool to achieve the intended goal.

One of the achievements of this paper is to theoretically show that for the induced colouring by communities of a graph in the SBM, almost surely all the vertices are ρ-happy if ρ is small enough. Another is finding the threshold ξ which indicates how small this ρ should be. The other is showing that if $\rho < \lim_{n\to\infty} \xi$, then the communities almost certainly induce a ρ-happy colouring on the vertices. Experimental analysis of over large number of randomly generated graphs perfectly verified the theoretical expectations.
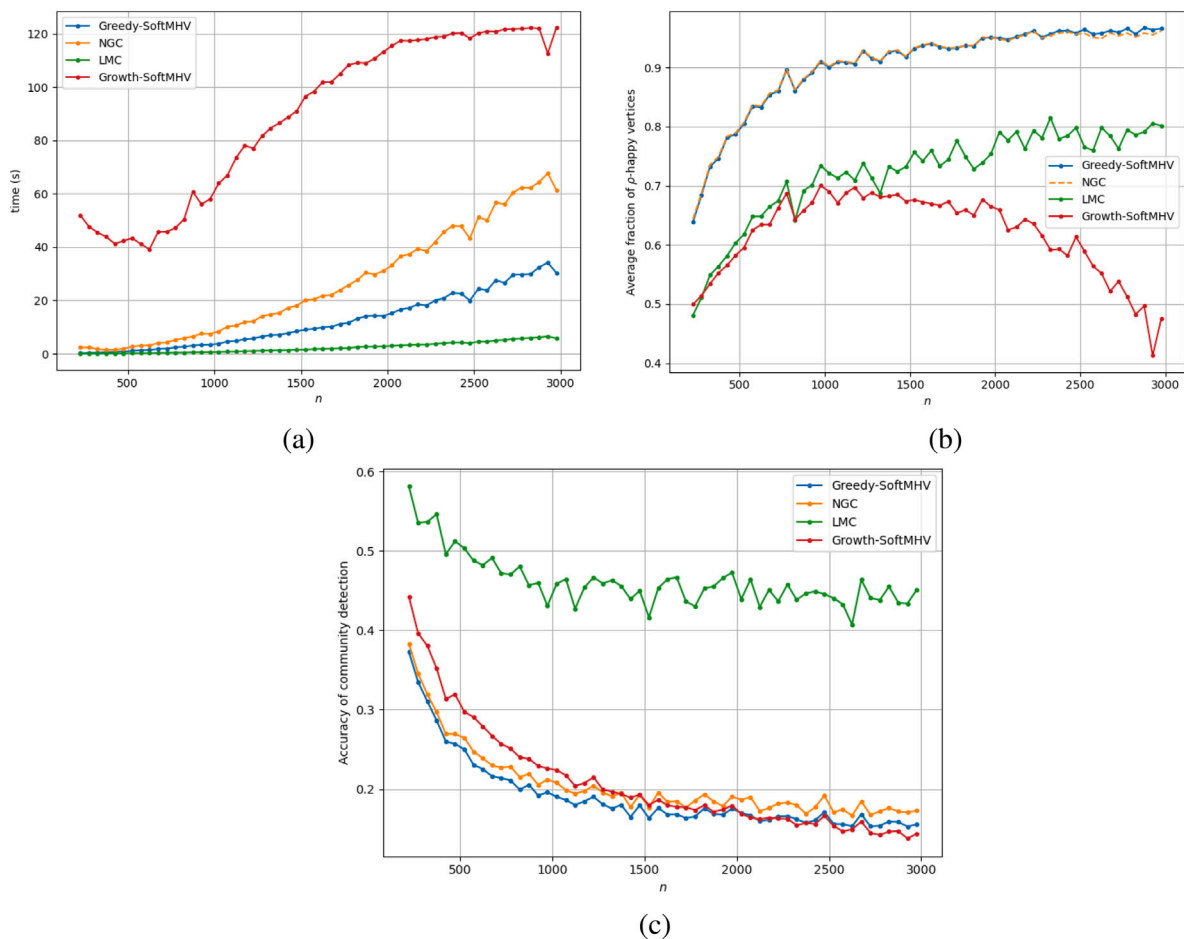
Fig. 14. Comparing (a) running time, (b) average number of ρ-happy vertices and (c) average accuracy of community detection of the four algorithms. The data is from the second test whose graphs are on $200 \leq n < 3000$ vertices.

Additionally, we developed algorithms to tackle the soft happy colouring problem. We adopt two existing algorithms from Zhang and Li (2015) and present a novel algorithm, Local Maximal Colouring. The algorithm has no reliance on ρ, however, its output shows considerable correlation with the communities of the tested networks while it has an inexpensive running cost. Therefore, for practical purposes, multiple runs of LMC are conducted to find the best possible result for the both objectives of soft happy colouring and community detection.

There are numerous possibilities for future work. Further studies can include considering the relations of (soft) happy colouring with communities of random graph models other than the simplified stochastic block model $\mathcal{G}(n, k, p, q)$. Looking at the induced colouring of detected communities by algorithms, such as spectral clustering, for their number of ρ-happy vertices can also be interesting on its own because these algorithms may find communities different from what graph generators produce. Another area for future work can be maximising the number of ρ-happy coloured vertices using mathematical/constraint programming, metaheuristics (tabu search and evolutionary algorithms), and matheuristics (CMSA and CMSA-TS) for ρ-happy colouring similar to Thiruvady and Lewis (2022) by Thiruvady and Lewis. Improving known algorithms for soft happy colouring, similar to what Ghirardi and Salassa (2022) suggest, is another possible study in future. Especially, one can think of improvements to the LMC because of its low time cost and its output's correlation with the community structure of its input.

CRediT authorship contribution statement

Mohammad H. Shekarriz: Writing – original draft, Visualization, Software, Methodology. **Dhananjay Thiruvady:** Writing – original draft, Supervision. **Asef Nazari:** Writing – review & editing, Supervision. **Rhyd Lewis:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

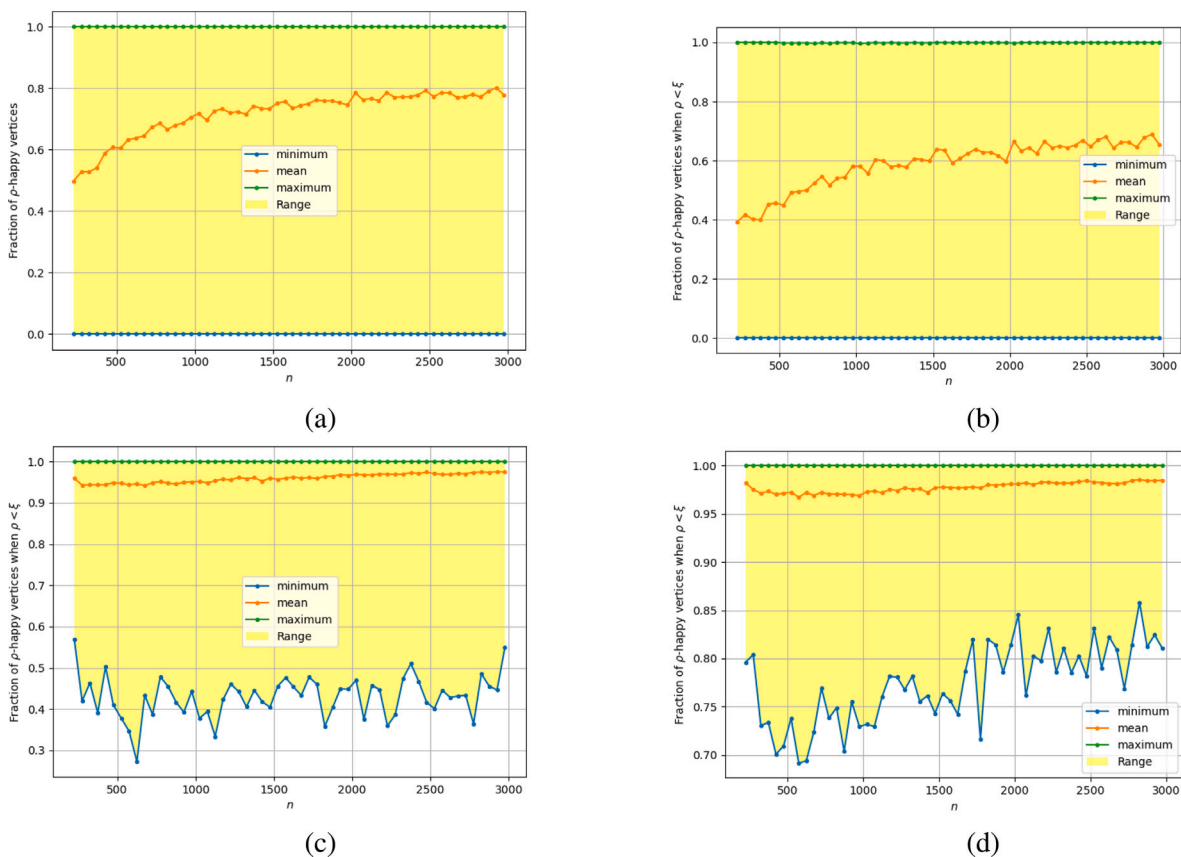https://github.com/mhshekarriz/HappyColouring_SBM.

Fig. 15. The performance of LMC (a) when there is no constraints on ρ, (b) when $\rho > \xi$, (c) when $\rho < \xi$ and (d) when $\rho < \frac{\xi}{2}$. The data were collected by running LMC 10 times for each of 28,000 randomly generated graphs with $200 \leq n < 3000$ vertices of the second test.

References

Abbe, E., 2018. Community detection and stochastic block models: Recent developments. J. Mach. Learn. Res. 18 (177), 1–86.

Abbe, E., Bandeira, A.S., Hall, G., 2016. Exact recovery in the stochastic block model. IEEE Trans. Inform. Theory 62 (1), 471–487.

Abbe, E., Sandon, C., 2015. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In: 2015 IEEE 56th Annual Symposium on Foundations of Computer Science. pp. 670–688.

Agrawal, A., Aravind, N., Kalyanasundaram, S., Kare, A.S., Lauri, J., Misra, N., Reddy, I.V., 2020. Parameterized complexity of happy coloring problems. Theoret. Comput. Sci. 835, 58–81.

Aravind, N.R., Kalyanasundaram, S., Kare, A.S., 2016. Linear time algorithms for happy vertex coloring problems for trees. In: Mäkinen, V., Puglisi, S.J., Salmela, L. (Eds.), Combinatorial Algorithms. Springer International Publishing, Cham, pp. 281–292.

Ball, B., Karrer, B., Newman, M.E.J., 2011. Efficient and principled method for detecting communities in networks. Phys. Rev. E 84, 036103.

Bollobás, B., 2001. Random Graphs, second ed. In: Cambridge Studies in Advanced Mathematics, Cambridge University Press.

Carpentier, L., Jooken, J., Goedgebeur, J., 2023. A heuristic algorithm using tree decompositions for the maximum happy vertices problem. J. Heuristics.

Coja-Oghlan, A., 2010. Graph partitioning via adaptive spectral techniques. Electron. Colloq. Comput. Complex. 19 (02), 227–284.

Condon, A., Karp, R.M., 1999. Algorithms for graph partitioning on the planted partition model. In: Hochbaum, D.S., Jansen, K., Rolim, J.D.P., Sinclair, A. (Eds.), Randomization, Approximation, and Combinatorial Optimization. Algorithms and Techniques. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 221–232.

Condon, A., Karp, R.M., 2001. Algorithms for graph partitioning on the planted partition model. Random Structures Algorithms 18 (2), 116–140.

Decelle, A., Krzakala, F., Moore, C., Zdeborová, L., 2011. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. Phys. Rev. E 84, 066106.

Diestel, R., 2017. Graph Theory, fifth ed. In: Graduate Texts in Mathematics, Vol. 173, Springer.

Erdős, P., Rényi, A., 1959. On random graphs I. Publ. Math. Debrecen 6, 290.

Fani, H., Bagheri, E., 2017. Community detection in social networks. Encyclopedia Semant. Comput. Robot. Intell. 01 (01), 1630001.

Ghirardi, M., Salassa, F., 2022. A simple and effective algorithm for the maximum happy vertices problem. TOP 30 (1), 181–193.

Holland, P.W., Laskey, K.B., Leinhardt, S., 1983. Stochastic blockmodels: First steps. Social Networks 5 (2), 109–137.

Jerrum, M., Sorkin, G.B., 1998. The Metropolis algorithm for graph bisection. Discrete Appl. Math. 82 (1), 155–175.

Karrer, B., Newman, M.E.J., 2011. Stochastic blockmodels and community structure in networks. Phys. Rev. E 83, 016107.

Lee, C., Wilkinson, D.J., 2019. A review of stochastic block models and extensions for graph clustering. Appl. Netw. Sci. 4 (1), 122.

Lewis, R., 2021. Guide to Graph Colouring: Algorithms and Applications, second ed. Springer Cham.

Lewis, R., Thiruvady, D., Morgan, K., 2019. Finding happiness: An analysis of the maximum happy vertices problem. Comput. Oper. Res. 103, 265–276.

Lewis, R., Thiruvady, D., Morgan, K., 2021. The maximum happy induced subgraph problem: Bounds and algorithms. Comput. Oper. Res. 126, 105114.

McPherson, M., Smith-Lovin, L., Cook, J.M., 2001. Birds of a Feather: Homophily in social networks. Annu. Rev. Sociol. 27 (1), 415–444.

Mossel, E., Neeman, J., Sly, A., 2015. Reconstruction and estimation in the planted partition model. Probab. Theory Related Fields 162 (3), 431–461.

Nadakuditi, R.R., Newman, M.E.J., 2012. Graph spectra and the detectability of community structure in networks. Phys. Rev. Lett. 108, 188701.

Newman, M.E.J., 2012. Communities, modules and large-scale structure in networks. Nat. Phys. 8 (1), 25–31.

Ravasz, E., Barabási, A.-L., 2003. Hierarchical organization in complex networks. Phys. Rev. E 67, 026112.

Solomon, R.S., PYKL, S., Das, A., Gambäck, B., Chakraborty, T., 2019. Understanding the psycho-sociological facets of homophily in social network communities. IEEE Comput. Intell. Mag. 14 (2), 28–40.

Thiruvady, D., Lewis, R., 2022. Recombinative approaches for the maximum happy vertices problem. Swarm Evol. Comput. 75, 101188.

Thiruvady, D., Lewis, R., Morgan, K., 2020. Tackling the Maximum Happy Vertices Problem in Large Networks. 4OR 18 (4), 507–527.

Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of 'small-world' networks. Nature 393 (6684), 440–442.

Ye, F., Chen, C., Wen, Z., Zheng, Z., Chen, W., Zhou, Y., 2020. Homophily preserving community detection. IEEE Trans. Neural Netw. Learn. Syst. 31 (8), 2903–2915.

Zhang, P., Krzakala, F., Reichardt, J., Zdeborová, L., 2012. Comparative study for inference of hidden classes in stochastic block models. J. Stat. Mech. Theory Exp. 2012 (12), P12021.

Zhang, P., Li, A., 2015. Algorithmic aspects of homophyly of networks. Theoret. Comput. Sci. 593, 117–131.

Zhao, X., Yan, B., Zhang, P., 2023. New algorithms for a simple measure of network partitioning. Theoret. Comput. Sci. 957, 113846.