Graph coloring: a novel heuristic based on trailing path—properties, perspective and applications in structured networks

Abhirup Bandyopadhyay, Amit kumar Dhar & Sankar Basu

Soft Computing
A Fusion of Foundations,
Methodologies and Applications

ISSN 1432-7643
Volume 24
Number 1

Soft Comput (2020) 24:603-625
DOI 10.1007/s00500-019-04278-8

Springer

Soft Computing (2020) 24:603–625
https://doi.org/10.1007/s00500-019-04278-8

METHODOLOGIES AND APPLICATION

Graph coloring: a novel heuristic based on trailing path—properties, perspective and applications in structured networks

Abhirup Bandyopadhyay[1] · Amit kumar Dhar[2,3] · Sankar Basu[4,5,6]

Published online: 14 August 2019

Abstract

Graph coloring is a manifestation of graph partitioning, wherein a graph is partitioned based on the adjacency of its elements. The fact that there is no general efficient solution to this problem that may work unequivocally for all graphs opens up the realistic scope for combinatorial optimization algorithms to be invoked. The algorithmic complexity of graph coloring is non-deterministic in polynomial time and hard. To the best of our knowledge, there is no algorithm as yet that procures an exact solution of the chromatic number comprehensively for any and all graphs within the polynomial (P) time domain. Here, we present a novel heuristic, namely the 'trailing path', which returns an approximate solution of the chromatic number within P time, and with a better accuracy than most existing algorithms. The 'trailing path' algorithm is effectively a subtle combination of the search patterns of two existing heuristics (DSATUR and largest first) and operates along a trailing path of consecutively connected nodes (and thereby effectively maps to the problem of finding spanning tree(s) of the graph) during the entire course of coloring, where essentially lies both the novelty and the apt of the current approach. The study also suggests that the judicious implementation of randomness is one of the keys toward rendering an improved accuracy in such combinatorial optimization algorithms. Apart from the algorithmic attributes, essential properties of graph partitioning in random and different structured networks have also been surveyed, followed by a comparative study. The study reveals the remarkable stability and absorptive property of chromatic number across a wide array of graphs. Finally, a case study is presented to demonstrate the potential use of graph coloring in protein design—yet another hard problem in structural and evolutionary biology.

Keywords Chromatic number · Graph partitioning · NP to P · Motif identifier · Protein design

1 Introduction

In graph theory, graph coloring (Jensen and Toft 2011) is a special case of graph labeling (Díaz et al. 2002). It is an assignment of labels (Gallian 2015) traditionally known as 'colors' to edges and/or vertices of a graph subject to certain constraints. In trivial formalism, it is a way of coloring the vertices (nodes) of an undirected graph such that no two adjacent vertices could be assigned the same

Communicated by V. Loia.

Abhirup Bandyopadhyay and Sankar Basu have contributed equally to this work.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00500-019-04278-8) contains supplementary material, which is available to authorized users.

✉ Sankar Basu
nemo8130@gmail.com

[1] Department of Mathematics, National Institute of Technology, Durgapur, Mahatma Gandhi Avenue, Durgapur, West Bengal 713209, India

[2] Department of IT, IIIT Alahabad, Jhalwa, Alahabad 211012, India

[3] Present Address: Department of EECS, IIT Bhilai, Raypur 492015, India

[4] Department of Physics and Astronomy, Clemson University, Clemson, SC, USA

[5] Present Address: 3BIO, ULB, 1050 Brussels, Belgium

[6] Department of Microbiology, Asutosh College, Kolkata 700026, India

color. This is called vertex coloring (MacDougall et al. 2002). Similarly, an edge coloring (Wallis et al. 2000) assigns a color to each edge so that no two adjacent edges share the same color, and a face coloring of a planar graph (Sanders and Zhao 2001) assigns colors to each face or region so that no two faces which share a common boundary share the same color. Given all this, vertex coloring remains the first chapter of the subject, and other coloring problems are transformable into a vertex version. For example, an edge coloring of a graph is actually a vertex coloring of the corresponding line graph, and a face coloring of a planar graph is a vertex coloring of its dual graph. However, non-vertex coloring problems are often stated and studied independently. That is partly because of perspective, and partly because some problems when naturally extended could be best studied in the form of edges.

The convention of using colors was originated from coloring the countries of a geographical map, where each face is literally colored. This particular problem was formalized as coloring the faces of a planar graph. By implementing 'planar duality', a characteristic feature of planar graphs, the problem reduces to coloring of their vertices. As a generalization the face coloring problem could be viewed as vertex coloring problem of its dual graph. For the sake of simplicity and computational efficiency, first few positive or nonnegative integers are used as the 'colors' (Zhang 2015) without the loss of generality, so that one can use any finite set of colors as the 'color set'.

Graph coloring could be viewed as the problem of assigning colors to a graph subject to number of constraints. Different constraints could range from constraints on a subgraph to those on the full graph or even on the color itself. The face coloring problem even attained popularity among common people in the form of the popular number puzzle Sudoku, the traveling salesman problem and the Chinese postman problem. One of the major applications of graph coloring is the register allocation in compilers. The range of applications grows even further ranging from coding theory to X-ray Crystallography (Blum et al. 1987), from radar and astronomy (Zarrazola et al. 2011) to circuit design and communication networks. Day-to-day real-life problems like guarding an art gallery, physical layout segmentation, round robin sports scheduling, aircraft scheduling (Marx 2003), etc., should potentially be benefited by an elegant algorithmic solution of the problem. Graph coloring is still a very active area of research with a bunch of unsolved problems, e.g., the chromatic number of the plane is unknown where two points are adjacent if they have unit distance. Other open problems concerning the chromatic number of graphs include the Hadwiger's conjecture (Bollobás et al. 1980) stating that every graph with chromatic number k has a k-complete subgraph with k vertices; the Erdős–Faber–Lovász conjecture bounding the chromatic number of unions of complete graphs that have exactly one vertex in common to each pair, and the Albertson conjecture (Albertson et al. 2010) that among k-chromatic graphs, the complete graphs are the ones with the smallest crossing number.

The first results about graph coloring deal almost exclusively with planar graphs in the form of the coloring of *maps* (Stiebitz and Škrekovski 2006). While working on the map coloring problem of the counties of England, Francis Guthrie postulated the four color conjecture, noting that four colors were sufficient to color a map so that no regions sharing a common border receives the same color. In 1879, Alfred Kempe published a paper (Kempe 1879) that claimed to establish the result which was controversial, followed by much debate. In fact it took close to a century until the four-color theorem was finally proved in 1976 by Kenneth Appel and Wolfgang Haken (Appel and Haken 1977). The proof was the first major computer-aided proof in this problem which went back to the ideas of Heawood and Kempe while largely disregarding the intervening developments. From that time onwards, active research is ongoing on the algorithmic attributes of graph coloring. The chromatic number problem falls in the list of Karp's 21 NP-complete problems (Karp 1972) and remains computationally NP-hard (Garey et al. 1974). That is to say that it is NP-complete to decide whether a given graph admits a k-coloring for any given k except for the trivial cases $k \in \{0,1,2\}$. In other words, the 3-coloring problem remains NP-complete even on 4-regular planar graphs (Dailey 1980), and the approximation algorithm (Halldórsson 1993), the most established one in the field, computes a coloring of graph size n at most within a factor of $O(n(\log n)^{-}3(\log (\log (n)))^{2})$ of the chromatic number.

The relatively recent concept of chromatic polynomial (Dong et al. 2005) has provided another alternative approach toward solving the graph coloring problem, serving important fundamental structures in algebraic graph theory. However, nowadays, the most celebrated conjecture is perhaps the '*strong perfect graph conjecture*', which was first brought about by Claude Berge, originally motivated by an information-theoretic concept called the 'zero error capacity' (Lovasz 2006) of a graph introduced by Claude E. Shannon.

The recent literature of combinatorial optimization problems in general consists of a wide variety of related yet distinct approaches ranging from adaptive and evolutionary algorithms [including the implementation of multi-variant strategies (Deng et al. 2015] like swarm intelligence algorithms (Deng et al. 2012b, b), bee and ant colony optimizations (Deng et al. 2015), strategies involving self-adaptive differential evolution (Deng et al. 2013), parallel hybrid intelligence optimization (Deng et al.

2012a, 2017c, 2019), machine learning approaches like optimal least square—support vector machines (Deng et al. 2017a) to empirical wavelet transform coupled with fuzzy entropy methods (Zhao et al. 2017b, 2018; Deng et al. 2018) and extending even to the regime of order control strategies (Zhao et al. 2017a) as in mechanical engineering. Consequently, different combinatorial optimization-based algorithms were used to address the graph coloring problem. However, in spite of the explosion of all these algorithms, the graph coloring problem (in particular) still remains hard and unsolved, due to the fact that an analytical solution of the chromatic number is yet not procurable and also that the chromatic number of an arbitrarily large graph is yet unknown. This has kept the field of graph coloring quite open, and a detailed survey of the current state of the art of existing graph coloring methods appears to reveal that there is a definite room for improvement in at least two of its major aspects: (1) minimizing the number of colors or the chromatic number for any given arbitrary large graph and (2) reducing the computational complexity.

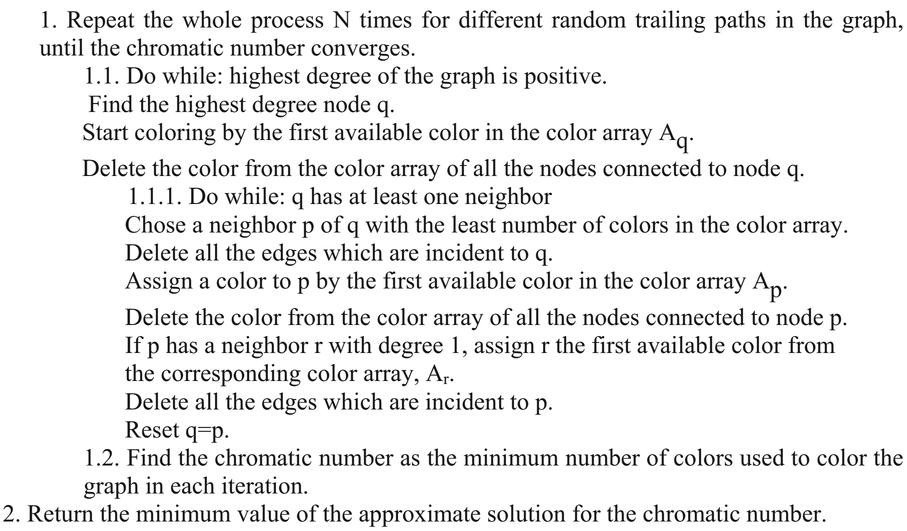
The current study presents the 'trailing path', a compound heuristic which finds (an approximate solution for) the chromatic number for any given graph unequivocally within the polynomial time domain (with respect to the input graph size) and with a better accuracy than most existing algorithms. The novelty of the current approach lies in (1) the meticulous combination of the search patterns of two existing heuristics (LF: based on the degree of nodes and DSATUR: based on the color availability of nodes to be colored) and in (2) following a trailing path of consecutively connected nodes while coloring, simultaneously. The algorithm has been tested on a large plethora of graphs of diverse size and connectivity and has resulted in running time(s) which are at most polynomial with respect to the input (i.e., the graph size), consistently throughout without hitting a single exception, and this is true even when running time is compromised at the cost of attaining the best possible accuracy. The approximate solution of the chromatic number of different structured networks, viz. small-world, regular, random, scale-free and modular networks have individually been surveyed. The effect of network parameters such as the average degree, rewiring probability, link density on the distribution of chromatic number has been vividly investigated. This should help to understand the distribution of chromatic number in real-world networks and facilitate their directed design. Pivotal graph coloring attributes as revealed from the analyses (viz., stability and absorptive properties) are critically introspected. Special and interesting cases are exemplified in the context of map coloring. Finally, to put into perspective the stability of graph partitioning as a critical and discerning feature in compartmentalization, a case study is presented, demonstrating its potential use in protein design—which is an active field of research in experimental and computational structural biology and molecular evolution.

It is to be noted that in order to avoid repetitive use of the long jargon 'approximate solution for the chromatic number', we simply use 'chromatic number' wherever applicable in the paper, which practically refers to its closely approximate solution rendered by the algorithm. This is particularly followed while discussing the general graph-partitioning properties and while relating the parameter to the more trivial graph parameters (e.g., degree, link density, etc.). In contexts where the 'approximation' itself needs to be emphasized specifically and elaborated (e.g., in describing the algorithm, discussing its complexity, convergence and comparing it with other heuristics, etc.), we do spell out the whole phrase.

2 Materials and methods

2.1 The 'trailing path' algorithm

Let $G = (V, e)$ be a graph with N nodes. Let A_i represent an array, referred as the 'color array' assigned for the ith node, which stores all available colors that can be used to color it, i.e., all those colors by which any neighbor of the ith node is not yet labeled. The initial length of the color array, A_i, is then set to N, as the minimum number of colors to label N nodes must be lesser than or equal to N. Thus, initially A_i is set to an ordered array of N colors for each (ith) node in the network. The algorithm has two hierarchical levels in its structure. In the first level, the algorithm starts coloring nodes from the *highest degree node*. In cases of degenerate paths where there exist more than one node with the highest degree, it starts coloring from an arbitrarily chosen *highest degree node* and assigns colors to nodes along different random paths from this *highest degree node*. In each iteration the algorithm trails through different random paths and labels each ith node on the path sequentially by the first available color in the corresponding color array A_i. At each iteration, subsequent to coloring the ith node, all edges incident to that (ith) node are deleted, along with the node itself. Thus, at each step, the algorithm encounters a new and unique induced subgraph of the original graph. In case of disjoint subgraphs, the algorithm colors them separately, independent of each other. In the next (second) level, the algorithm starts its trailing path from the node which has the least number of colors available for labeling it. Likewise to the earlier level, the algorithm arbitrarily chooses a node in cases of degenerate paths, i.e., when there exist more than one node with the same least number of available colors. By this way, the algorithm keeps assigning labels (i.e., colors) to a graph iteratively by the trailing path and keeps track of the minimum number of colors required to label all nodes till convergence. Finally, this updated minimum number of colors is returned as the value of the chromatic number. The structure of the trailing path algorithm could be written as follows.



```
1. Repeat the whole process N times for different random trailing paths in the graph,
until the chromatic number converges.
    1.1. Do while: highest degree of the graph is positive.
     Find the highest degree node q.
    Start coloring by the first available color in the color array $A_q$.
    Delete the color from the color array of all the nodes connected to node q.
        1.1.1. Do while: q has at least one neighbor
        Chose a neighbor p of q with the least number of colors in the color array.
        Delete all the edges which are incident to q.
        Assign a color to p by the first available color in the color array $A_p$.
        Delete the color from the color array of all the nodes connected to node p.
        If p has a neighbor r with degree 1, assign r the first available color from
        the corresponding color array, $A_r$.
        Delete all the edges which are incident to p.
        Reset q=p.
    1.2. Find the chromatic number as the minimum number of colors used to color the
    graph in each iteration.
2. Return the minimum value of the approximate solution for the chromatic number.
```

To note is that the algorithm involves randomness from its very first step. It starts coloring nodes from any *highest degree node* taken at random (in case of more than one *highest degree nodes*). It then chooses a path from that *highest degree node* to a neighboring node which has the minimum number of available colors (or the maximum number of colors unavailable to it[1]) to label it. In case of a tie between two or more neighboring nodes in this parameter, it chooses a node which has a degree higher than that of all other neighboring nodes. If there's another tie in terms of the degree also, it chooses a path randomly. The salient feature of moving through only (consecutively) connected nodes makes the algorithm follow a 'trailing path'. Again, the involvement of randomness in every non-trivial step imparts an evolutionary attribute to the heuristic where the corresponding number of colors determines the fitness of a 'trailing path'.

2.2 Comment on convergence

Let p be the probability of finding a trailing path which gives the minimum coloring, i.e., the probability to choose a spanning tree that gives the chromatic number. Then the probability to find the desired trailing path in two iterations will be $2p - p^2$. Similarly the probability to choose the desired trailing path in n iterations may be written as $n \cdot p - p^2 - p^3 \ldots - p^n$ which equals to $n \cdot p - p^2 \cdot \frac{1-p^{(n-1)}}{1-p}$ which happens to be the probability of convergence. Hence if $n > \frac{p}{1-p}$, it is expected to obtain the desired trailing path at least once. However, the number of spanning trees of a graph, in general, is of the order of n^{n-2}. Therefore, it is clear that ensuring convergence of chromatic number will lead the algorithm to a complexity of the non-polynomial time domain—which is obvious for any NP-hard problem.

2.3 Novelty and apt of the algorithm

Effectively, the 'trailing path' algorithm is a subtle combination of the search patterns of two existing heuristics, namely DSATUR and largest first (LF). LF is based on a search along a descending order of degrees of nodes while the search in DSATUR is based on the color availability of nodes to be colored. This very meticulous manner of combining the search patterns of DSATUR and LF imparts in the current algorithm its novelty and effectiveness, which is further enhanced by its unique feature of traversing through a trailing path of consecutively connected nodes while coloring. To the best of our knowledge, there is no algorithm currently which follows such a trailing path (or a continuous coloring scheme).

2.4 Algorithmic complexity

The algorithm in implementation repeats the inner loop until the value of the chromatic number converges to a single value (subsequent iterations are not able to find any

[1] For any finite color array.

smaller chromatic number). As explained in the previous section, to obtain the exact value of chromatic number, this loop will continue for exponential time in the worst case. Consider a graph with n edges. Each iteration of the loop first tries to find and extract the maximum degree node which can be done in log n time using a heap data structure. In the subsequent steps, the algorithm iterates over all the neighbors of a node to find one with least number of colors. This again is possible using log ($n-1$) time in the worst case (assuming $n-1$ neighbors). The rest of the operations in the loop (e.g., assigning color, deleting color and deleting edges/vertices) can be done in at most log (n) time. Thus, each iteration of the loop can be completed in at most O(log (n)) time. Thus, if the algorithm continues for k iterations before converging to a value, the complexity of the algorithm would be O(k log (n)).

3 The software: *Chromnum*

The software package with detailed documentation is available at: https://github.com/nemo8130/Chromnum containing two different versions, one (*chromnum.m*) which was originally developed in MATLAB (version: R2016a) using its advanced 'graph' module for visualizing the colored graph. This version returns the colored graph in both a circular and a forced layout. However, to make the software more student-friendly, another version (*chromnum_octave.m*) is provided which runs on both MATLAB and Octave replacing the more sophisticated graph-visualization part by a simple display of a colored graph in a random layout developed using the trivial 'plot' command alone (Supplementary Figure S1) which is inbuilt in both standard distributions (i.e., MATLAB and Octave).

Apart from returning (an approximate solution for) the chromatic number and the visual display of the colored graphs, the program also returns the corresponding colormap (i.e., which node could be labeled by what color) which is definitely more informative and useful in investigating real-world networks than just the chromatic number alone. However, it should be noted that this colormap may in principle be potentially degenerate and the program returns just one of the possible solutions obtained following a particular 'trailing path'. Repeating the program more than once on the same adjacency matrix may result in obtaining different colormaps on different runs. An illustrative example is shown in Fig. 1. Here, as illustrated in the figure legend, the 'trailing path algorithm' leads to four possible degenerate paths (in the second iterative step), which eventually results in the same number of minimum colors required to color the graph, though leading to the attainment of two alternative colormaps. Here, as illustrated in the figure, the 'trailing path algorithm' leads to four possible degenerate paths (in the second iterative step), subsequent to coloring the *highest degree node* (N6 → red) in the first step. These four paths are: N6 → N14 (path 1), N6 → N9 (path 2), N6 → N17 (path 3) and N6 → N18 (path 4). Note that all of the possible four nodes have identical degrees of five and identical lengths of their color array (19–1 = 18). In four independent runs covering all four degenerate paths, the algorithm returns the same chromatic number (Nc = 3) while exploring two possible alternative colormaps (say, cm1: path1, path4 and cm2: path2, path3).

To aid a variety of trade-offs between the run time and the accuracy, the program has also been built with the provision of accepting the desired number of iterations as input by the user. Thus, the program can be run in single as well as multiple iteration mode, as might be required for a given context.

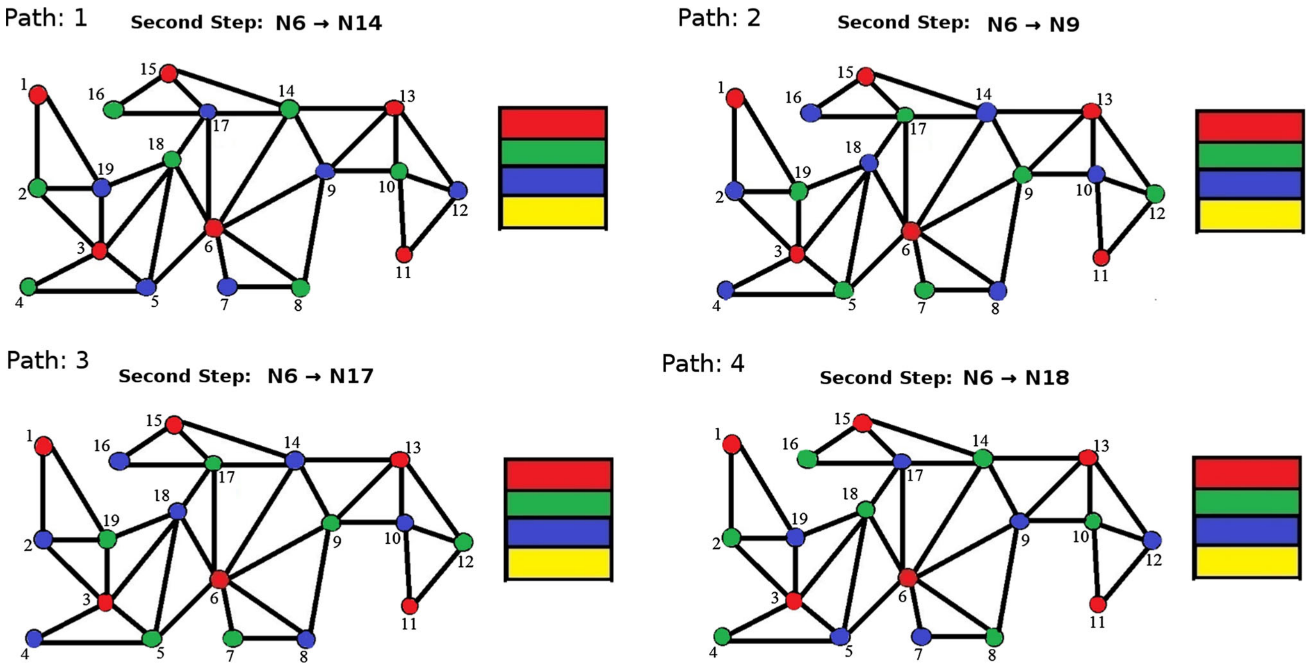
As discussed in the introduction, vertex coloring is the heart of the graph coloring problem and an edge coloring of a graph could be transformed into the vertex version of its line graph. With this understanding, we also provide with the distribution, a small script (linegraph.m) that can take an original graph and return its corresponding line graph, so that an edge coloring problem can also be addressed by the same package.

4 Results and discussion

The Results and Discussion section may broadly be classified into four major parts: (1) a thorough discussion on the different properties of chromatic number (particularly emphasizing on its stability and absorptive property) with systematic calculations carried out on different structured networks (4.2. random graphs, 4.3. small world, 4.4. scale free, 4.5. modular, 4.8. regular graphs) coupled with a comparative study (4.7); (2) a whole section on computational complexity (4.9); (3) comparison with other heuristics (4.10) and (4) with a few case studies demonstrating the application of graph coloring (4.11–4.13). In order to initiate a systematic discussion of the stability and absorptive property of chromatic number, we required a method (preferably a numeric scheme) to account for the topological variation in each category of structured networks—which has been introduced and elaborated in Sect. 4.1.

4.1 Accounting for topological variability in relation to graph partitioning

Graph coloribility is essentially a demonstration of graph partitioning, wherein the nodes (or edges) are partitioned on the basis of their relative adjacencies. On the other hand,


Path: 1 Second Step: N6 → N14

Path: 2 Second Step: N6 → N9

Path: 3 Second Step: N6 → N17

Path: 4 Second Step: N6 → N18

Fig. 1 The trailing path algorithm. The figures illustrate a demonstrative example of how the trailing path algorithm operates (see Main Text)

it is the different combinations of adjacencies that lead to the variations in network topologies. However, a combination of specific network parameters (e.g., network size, average degree, rewiring probability, etc.) can, in principle, be so sampled (systematically) that graphs with roughly equal (or at least similar) link densities (Ld) can be constructed with small-to-large variations in their network topologies. Therefore, it would be really interesting to explore if there exists any empirical correlation between 'partitioning' (in terms of the chromatic number) and 'topological variation' in graphs. As a complementary analysis, it is also important to find out the relation between 'partitioning' and global network descriptors like Ld.

Detection of topological variation between graphs essentially approaches to the famous 'graph isomorphism' problem (RJLipton+KWRegan 2015) which is a subject on its own and falls outside the scope of the current work. To simplify matters, we used a modified version of a previously proposed numeric scheme to identify unique graphs from a statistical ensemble of different structured (and unstructured) networks.

In order to explore the 'network view' of the internal architecture of globular proteins (Basu et al. 2011), a novel numeric scheme (namely, the 'motif identifier') was proposed in a previous study which found its efficacy in characterizing and classifying contact networks within proteins as a gradual and context dependent assembly from a finite yet non-rigid basis set of unique graphs, namely 'packing motifs'. In effect, it demonstrated a nucleation-condensation model in protein packing. Protein contact networks, however, were restricted in the extent of possible topological variation (like any other real-world networks) due to molecular steric constraints. Here, in this current study, we explored the potentiality of this 'numeric scheme' to identify unique graphs generally in systematically sampled statistical ensembles of random and structured networks. We also take the opportunity to discuss the limitation of the numeric scheme for the particular case of topological variations in a subset of k-regular graphs ($k > 2$) of identical network size.

To that end, we adapted a modified version of the previously proposed 'numeric scheme' to represent unique graphs that can directly be calculated from their adjacency matrices. In line with the earlier formulation (Basu et al. 2011), each node of a graph was initially assigned a string of numbers of length ($d + 1$) (where d is the degree of the node) starting with its own degree, followed by the degrees of its connected nodes (direct neighbors) sorted in a descending order, and separated by two distinct delimiters (say $\sim$, –). These delimited and concatenated numeric strings (viz. nodal motifs) were then collected as elements of an array and converted into a hash table, tabulating the unique number strings and their respective counts. Generally, for any two given graphs (except for regular graphs), if their corresponding hash tables were found to be identical, that is to say that if both graphs (hash tables) contained identical set of nodal motifs (number strings) with identical counts, they could be treated as identical graphs. Thus, the motif identifier essentially discriminates between two graphs based on the combined distribution of degrees

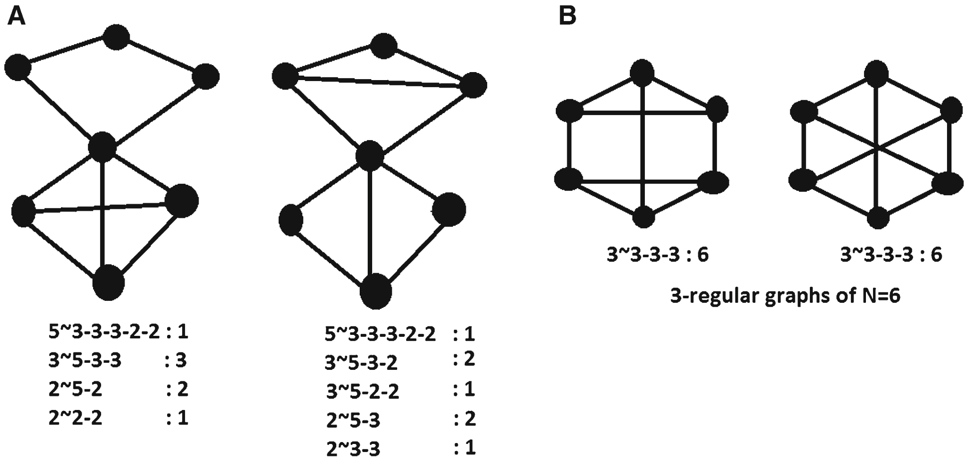


Fig. 2 The motif identifier: accounting for topological variations in graphs. The motif identifier (presented in the form of a hash table) is a collection of numeric strings representative of each unique nodal motif and their corresponding counts. The first number in each numeric string stands for the degree of a node, and the other numbers represent the degrees of its direct neighbors sorted in a descending order. The degrees of the neighbors are concatenated by a hyphen (-) and their concatenated number strings are further joined to the degree of their corresponding source nodes by a tilde ($\sim$)

of their constituent nodes, coupled with the degrees of their neighboring nodes, and will potentially signal for any variability in these network parameters between two given graphs (Fig. 2a). In fact, this is precisely the reason why the identifier fails to discriminate between two non-identical (non-trivial) k-regular graphs ($k > 2$) (Fig. 2b). In other words, since for regular graphs, all nodes have identical degrees, no variability can be accounted for in terms of their degree and/or the degrees of their neighboring nodes, even if the two k-regular graphs (of the same size) are topologically non-identical. The motif identifier will hence return identical hash tables for both the graphs. In Fig. 2, panel A shows a case where the motif identifier successfully discriminates between two non-identical graphs while panel (b) shows the limiting case of two non-identical 3-regular graphs—where the motif identifier fails to discriminate between the two.

It is important to note that although the motif identifier may incorrectly signal identity for the case of two or more non-identical graphs (in case of k-regular graphs; $k > 2$), it will never signal non-identity for any two identical graphs. Hence, if it ascertains n unique graphs to be found from a statistical ensemble of N graphs ($N > n$), then there is definitely at least n unique graphs (if not more) in the set of N.

4.2 Chromatic number as a function of link density: theoretical and statistical bounds in random graphs

As discussed in the Theory section, there are theoretical upper-bounds of chromatic number for typical 'structured' graphs. A complete graph of N nodes will trivially be N-colorable and an entirely disconnected (edge-less) graph will be 1-colorable by definition. Again, all trees[2] containing more than one node will always be 2-colorable irrespective of its length and extent of branching. Since, trees are acyclic connected graphs, by definition, they lack any embedded closed triplet clique which has been revealed as the unit of clustering per se (Basu et al. 2011). They therefore essentially have zero-clustering (i.e., clustering coefficient[3] = 0) and will require just two colors alternatively put to the nodes along a trailing path to color them minimally and exhaustively. To that end, it is obvious that any graph containing an embedded triplet clique (i.e., clustering coefficient > 0) will at least be 3-colorable. Similarly, any even cycle (or closed cyclic graphs constituted of even number of nodes) will be at least 2-colorable, and any odd cycle will be at least 3-colorable, irrespective of the actual graph size.

However, most real-world networks are non-trivial and offer far greater complexity and variability in their topologies. Hence, exhaustive and systematic analyses were felt necessary to perform, varying the link density of a graph within its entire theoretical range, [0,1] and then computing the chromatic number for all graphs and carry out a thorough statistical analysis.

First the graph size (i.e., the number of nodes, N) was fixed at a certain value (say $N = 10$), and the link density

[2] Trees are undirected graphs where any two nodes are connected by exactly one path.

[3] Clustering coefficient of a node in a graph is the ratio the total number of actually existing connections in its direct neighborhood and the number of maximum possible connections within the same set. For a graph or subgraph, the average clustering over all nodes is considered.

(Ld) was varied within its entire range, [0,1], giving rise to the construction of random graphs[4] with connection probabilities same as their Ld values. Thus, for a given Ld value, an ensemble of random graphs was sampled covering a wide variety of possible range of topologies. Chromatic numbers were calculated for each of these graphs, and the minimum and maximum along with the average (and standard deviations) plotted as a function of link density (Fig. 3). All three parameters converged to 1 (the theoretical lower bound) for Ld = 0 and to N for Ld = 1 (upper bound). As expected, the chromatic number followed an ascending trend as a function of link density. As could be seen from the error bars associated with the average plots, chromatic number generally varies within a narrow range ($\sim \pm 0.25$ to 1.00) for lower values of Ld and gradually increases at its higher end. The variation in chromatic number at the higher ends of Ld is certainly non-negligible. Increasing the size of the graph ($N = 10, 15, 20$: illustrated, respectively, in panels (A) (B) and (C) of Fig. 3) did not seem to alter the overall trend. In other words, the trajectory of chromatic number as a function of link density appears to be characteristic, irrespective of the graph size.

4.3 Chromatic number of small-world networks

Small-world networks are characterized by the combined features of local cohesiveness and global reach. That is to say that a small-world network is essentially locally cohesive, attaining a reasonably high clustering coefficient scaling to that of a regular network[5] of the same size. At the same time, it is also globally reachable represented by a trademark of low characteristic (or mean shortest) path length[6] equivalent to that of a random network of the same size. From definition of clustering, it is obvious that small-world networks essentially sustain one or more closed triplets (i.e., 3 cliques) (Basu et al. 2011).

There are different algorithms to generate small-world networks out of which the most famous is surely the 'Watt–Strogatz' algorithm (Watts and Strogatz 1998)—which was adapted in the current study. In this approach, a regular graph is first generated taking its size and the degree of each node as inputs. Then, the edges of the given random network are rewired with a given rewiring probability, i.e., an edge of the graph is deleted at the cost of one long range link to be created with this same rewiring probability. This operation on the template regular network will lead to the generation of a small-world network of the given size and an average degree identical to that of the template regular graph. Even a low rewiring probability will lead to the generation of a network showing small-world properties. Trivially, the resultant graph would remain unaltered (i.e., the initial regular network) if the rewiring probability is exactly 'zero' and would map to a purely random graph when the rewiring probability is exactly 1. So, as a result of varying the rewiring probability from 0 to 1, the adapted algorithm would generate small-world networks trending from regular to purely random.

To map the distribution of chromatic number for a wide variety of small-world networks, different graph sizes were considered ranging from 5 to 30 at an interval of 5 nodes. For each of these graph sizes (N), degree of each node (k) was made to vary from 1 to ($N-1$), while the rewiring probability (p_{rw}) was sampled in the range of 0.1–1 at an interval of 0.1, and the chromatic number of each graph was calculated. Thus, this calculation covers a wide spectrum of networks ranging from graphs close to being regular (say, at $p_{\mathrm{rw}} \sim 0.1$) to graphs that are purely random (at $p_{\mathrm{rw}} = 1.0$).

Based on these network parameters (N, k), link density ($\mathrm{Ld}_{\mathrm{smw}}$) can be analytically derived by the following expression (Eq. 1).

$$\mathrm{Ld}_{\mathrm{smw}} = \frac{\frac{N \cdot k}{2}}{{}^{N}C_{2}} \tag{1}$$

where ${}^{N}C_{2}$ is the maximum possible number of links that the graph of N nodes can accommodate. Noteworthy is that the link density is trivially independent of the rewiring probability (p_{rw}) for small-world networks. The reason is that, for a given rewiring probability, an edge is generated at the cost of an existing edge at that probability, and hence, the total number of connections should ideally remain the same in both the template (regular) and the resultant (small-world) networks.

To estimate the stability of the calculated chromatic number (crn) across a wide 'topological' variety of graphs, constructed from an identical set of network parameters (N, k, p_{rw}), all calculations were repeated 100 times for each set of sampled network parameters, and the average (μ_{smw}) and standard deviations (σ_{smw}) in their chromatic numbers were recorded. Hence, we were actually looking at a statistical ensemble of 'potentially different' networks having identical network parameters rather than a randomly selected single graph. The topological variation in each statistical ensemble was enumerated by the numeric scheme (motif identifier) elaborated earlier. Other than the really small networks ($N = 5$), most of the resulting graphs were found to be unique (Supplementary Table S1). Except

[4] A graph where all the connections (edges) are randomly assigned.

[5] A graph where all nodes have identical degrees.

[6] Shortest path of a pair of nodes in a graph is the minimum number of links connecting the two. The average over all pairs of nodes in a graph (or subgraph) defines the characteristic (or mean shortest) path-length of the graph (or subgraph).

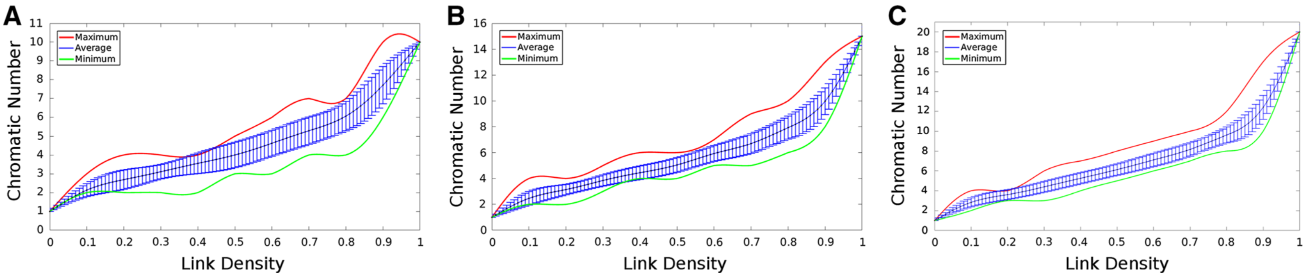


Fig. 3 Chromatic number as a function of link densities for random graphs. The red, green and blue curves represent maximum, minimum and the average chromatic numbers while the standard deviations are given by means of error bars centering the corresponding average values (color figure online)

for the trivial cases of complete graphs (resulting from $k = N{-}1$), the number of topological variants (unique graphs) was found to be increasing proportionally with the rewiring probability (p_{rw}) while eventually saturating at the highest possible value, 100, for a large majority of sampled network parameters ($N > 5$, $p_{\mathrm{rw}} \geq 0.4$ say). Note that a p_{rw} of precisely 1 would lead to the generation of purely random graphs—which by definition are expected to be unique. On the other hand, a very low rewiring probability would result in graphs close to the template regular graphs, with a restricted scope of topological variability, particularly relevant for small networks. Hence, it was no surprise to find only 5 unique graphs for $N = 5$, $k = 2$ (Table S1). Considering this, the parameters were so chosen that could judiciously eliminate the trivial and limiting cases of regular graphs (by setting $p_{\mathrm{rw}} > 0$). Hence, the topological variations obtained by the motif identifier should be treated unambiguous, reflecting the fact that most networks in a sampled statistical ensemble (as a function of identical network parameters) were indeed unique.

Given such large topological variability in the sampled graphs, the average chromatic numbers, however, were found remarkably stable, reflected in significantly low standard deviations relative to their corresponding means (Supplementary Figure S2) attained for virtually all statistical ensembles ($|\sigma|_{\mathrm{smw}} \sim 7.6 \pm 4.9\%$ of $|\mu|_{\mathrm{smw}}$). Therefore, small-world networks generated with identical network parameters can be represented by a characteristic chromatic number.

In parallel, link densities (Ld) were also computed, both analytically ($\mathrm{Ld}_{\mathrm{exp}}$) as described above, as well as from the actual networks ($\mathrm{Ld}_{\mathrm{obs}}$), as a complementary measure of the topological variability. Mean ($|\mathrm{Ld}_{\mathrm{obs}}|$) and standard deviations ($\sigma_{\mathrm{Ld\_obs}}$) of the observed link densities ($\mathrm{Ld}_{\mathrm{obs}}$) were recorded for each statistical ensemble (i.e., generated from an identical set of network parameters) and compared with the corresponding expected value ($\mathrm{Ld}_{\mathrm{exp}}$). $|\mathrm{Ld}_{\mathrm{obs}}|$ and $\mathrm{Ld}_{\mathrm{exp}}$ were found to be practically identical with negligibly small standard deviations (Supplementary Figure S3) for small-world networks of all sizes.

For each network size (N), the average chromatic numbers obtained from the aforementioned statistical ensembles were then plotted as surfaces, as a bivariate function of the average degree (k) and the rewiring probability (p_{rw}). As could be seen from the surface plots (Fig. 4), the surfaces were reasonably smooth, resembling a 'floating carpet' from the Arabian Nights, elevated in the breeze toward the diagonal corner with high k and p. Similar patterns were obtained for all tested network sizes ($N = 5, 10, 15, 20, 25, 30$), although, for the smallest network size ($N = 5$), the surface was relatively more corrugated due to relatively smaller number of sampled data points. It is also noteworthy that for $N = 5$, $k = 2$, the standard deviations were consistently on the higher side (average chromatic numbers: 2.75 ± 0.45) in the whole range of p going from 0.3 to 1. The reason is that the template regular graph obtained for $N = 5$, $k = 2$ is the 'pentagon' (or 5-cycle) which is trivially (at least) 3-colorable (see Theory and Algorithm) and thus, any rewiring generally leads to a random hopping of the average chromatic number between 2 and 3. The nature of the surface plots physically means that a small-world network generally attains a higher chromatic number for higher average degree of the network, which, in turn, is proportional to the link density of the network. A higher rewiring probability also leads to a higher chromatic number, but the growth of chromatic number is generally more as a function of the average degree than the rewiring probability.

4.4 Chromatic number of scale-free networks

A scale-free network is formally defined as one for which the degree distribution follows a power law (Clauset et al. 2009), at least asymptotically. That is to say that the fraction of nodes $P(k)$ in the network having k connections goes for large values of k as $P(k) \sim k^{-\gamma}$; where γ is a parameter falling typically in the open interval of (2, 3), while occasionally lying outside these bounds. In the current work, the Barabasi–Albert algorithm (Albert and Barabási 2002) was adapted to generate a range of scale-

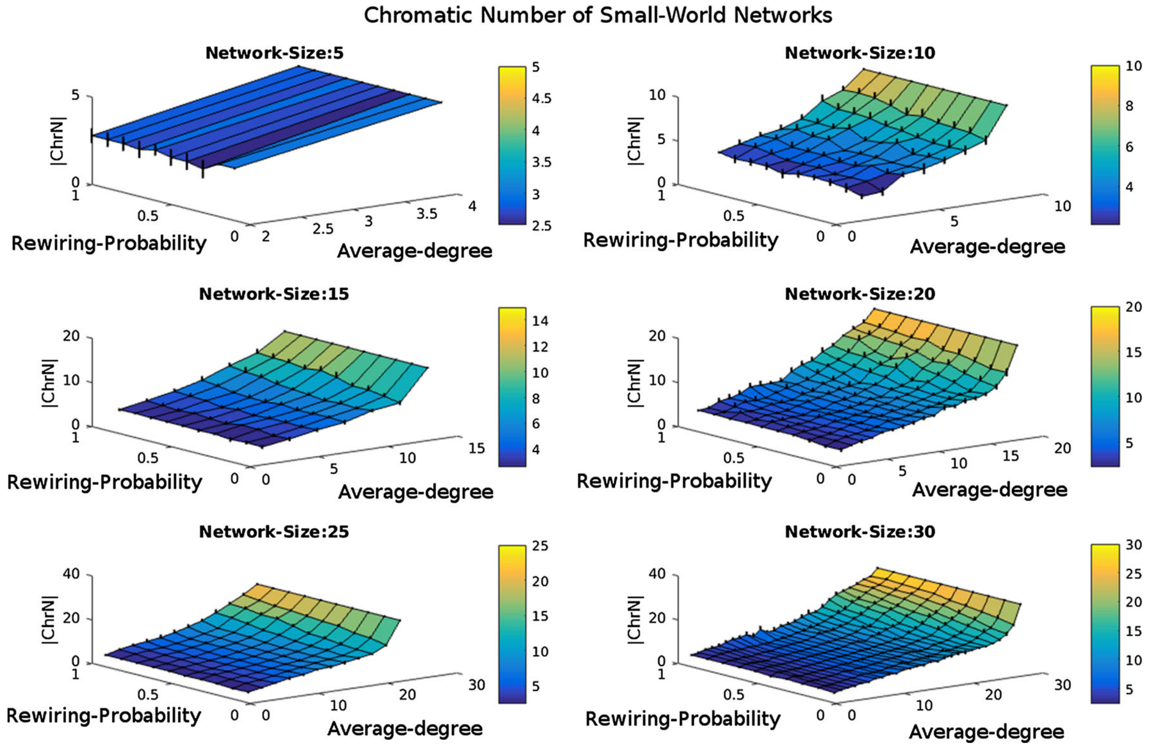


Fig. 4 Chromatic number of small-world networks. Average chromatic numbers (|ChrN|) are plotted (as surfaces) as bivariate functions of the small-world network parameters (average degree: k, rewiring probability: P_{rw}) pertaining to the statistical ensembles for each sampled network size

free networks. This algorithm follows the principle of 'preferential attachment (Choromański et al. 2013) of new nodes to previously existing high degree nodes' and is based on two parameters, namely the network size (N) and the average degree of nodes (k). Starting from two connected nodes, each time a new node is attached in the network with the probability of creating an edge between this new node and an existing node being proportional to the degree of the existing node. In other words, new edges incident to the new node are preferentially attached to existing nodes based on a cumulative degree distribution of existing nodes computed at each step of the growing network in (N−2) steps for a network of size N. Hence, here, chromatic number may be treated and tested as a bivariate function of two scale-free parameters (N, k).

Similar to the case of small-world networks, link density ($\mathrm{Ld}_{\mathrm{smw}}$) of scale-free networks can also be analytically derived based on these network parameters (N, k), giving an identical expression to that of the small-world networks (Eq. 2).

$$\mathrm{Ld}_{\mathrm{scf}} = \frac{\frac{N \cdot k}{2}}{{}^{N}C_2} \tag{2}$$

where ${}^{N}C_2$ is the maximum possible number of links if the N-graph was complete.

Following the above algorithm, the network size (N) was sampled from 5 to 50 at an interval of 5 nodes; the average degree (k) was made to vary from 1 to (N−1) and the chromatic number calculated for each graph. Similar to the small-world networks, a statistical ensemble of 100 potentially different graphs was considered for each given set of network parameters (N, k), and the average (μ_{scf}) and standard deviations (σ_{scf}) in their chromatic numbers were recorded.

Topological variability (as enumerated in terms of the motif identifier) was found to be even greater than the small-world networks. Only for the exception of N = 5 (the smallest network size considered) which trivially presents a restricted combinatorial space of topological variability, the number of unique graphs was otherwise invariably found to reach the maximum possible value of 100 (Supplementary Table S2). Hence, by and large, we were indeed looking at non-identical networks throughout, in the overwhelming majority of statistical ensembles, sampled at fixed sets of network parameters (N, k).

Similar to that of the small-world networks, here also the average chromatic numbers were found to be remarkably stable reflected in their low standard deviations (Supplementary Figure S4) with respect to their corresponding means ($|\sigma|_{\mathrm{scf}} \sim 8.9 \pm 5.6\%$ of $|\mu|_{\mathrm{scf}}$), and hence, the statistical ensembles can be represented by the corresponding 'characteristic' average chromatic numbers.

Likewise to the small-world networks, link densities (Ld) were also computed, both analytically ($\mathrm{Ld}_{\mathrm{exp}}$) and from the actual networks ($\mathrm{Ld}_{\mathrm{obs}}$) and their mean ($|\mathrm{Ld}_{\mathrm{obs}}|$) and standard deviations ($\sigma_{\mathrm{Ld\_obs}}$) recorded for each statistical ensemble. Average observed and expected link densities ($|\mathrm{Ld}_{\mathrm{obs}}|$ and $\mathrm{Ld}_{\mathrm{exp}}$) were found to be largely matching (Supplementary Figure S5) with minor variations in the two measures being observed mostly for high link densities; say at $k \sim (N{-}1)$. Standard deviations in the observed link densities were found to be low all throughout the whole range of sampled network sizes (N going from 5 to 50).

The distribution of average chromatic number as a function of the scale-free parameters (N, k) was then plotted as a surface plot (Fig. 5) which looks like a partially unfolded Chinese hand fan, with chromatic number increasing along both N and k in a discrete step-wise hierarchical manner. The variation in the average chromatic number is realized more in a wavy manner along the average degree than along the network size.

In a closer look, when the average chromatic numbers were plotted individually as a function of the average degree (k) for each network (Supplementary Figure S6), its ascending trend was further confirmed with increasing k which eventually saturates at the higher end of k. To exemplify this saturation event, an ensemble of scale-free networks of size 50 is presented (Supplementary Figure S7), generated by varying the k in the range of 41–49. Noteworthy is the fact that all these graphs are 17- to 19-colorable, while the variation in the corresponding total number of edges in these graphs is in the range of 885–967 (average: 924.4 ± 24.1). Thus, a tiny bin (Δ) of 3 in the value of chromatic number could actually accommodate for 82 new edges in this particular example—indicative of the remarkable absorptive property of graph coloring for closely related yet distinct graph topologies.

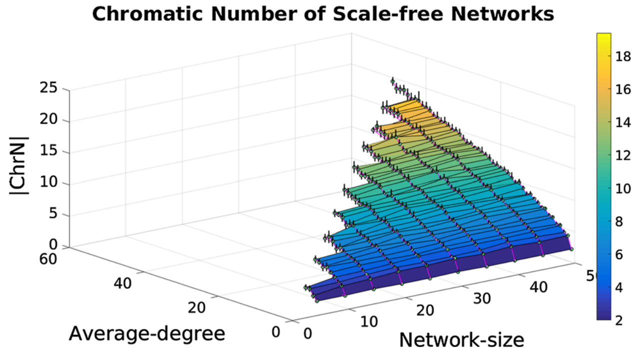


Fig. 5 Chromatic number of scale-free networks. Average chromatic numbers (|Chrn|) are plotted by means of a surface as bivariate functions of the scale-free network parameters (network size: N, average degree: k) pertaining to the sampled statistical ensemble

When the trends among different network sizes were compared, mild and gradual increases in the highest value of the average chromatic number were observed for larger networks (Fig. 5). Further, all of these ‘average degree versus average chromatic number’ plots for different network sizes (from 5 to 50) could be best fitted to straight lines having almost identical slopes (Fig. 6) characterized by a negligibly small standard deviation among them (0.083). This implies that the distribution of chromatic number for these graphs is independent of the network size (N)—which is the hallmark feature of scale freeness. Also, from the average slope ($\sim$ 0.39) of these best fitted straight lines, one can, in principle, approximately calculate the chromatic number of a scale-free network from the average degree (k) alone (given the negligibly small standard deviation obtained in their slopes).

4.5 Chromatic number of modular networks

Modularity is one key measure to demonstrate the topological structure of graphs. It could be viewed as a measure of strength of division of a network into modules (i.e., groups, clusters or communities). In other words, modular networks are those in which the connection density is significantly higher within modules compared to that of the nodes between different modules (Newman 2006). Modularity is frequently used to detect the community structure in real-world networks as part of optimization methods. However, small communities are often left undetected by ‘modularity’ due to its inherent limit in resolution. Biological networks of diverse origins including animal brains, contact networks in multi-domain proteins, molecular interaction networks in complex signal transduction and cross-talking metabolic pathways exhibit high degree of modularity.

Modular networks were so constructed that the chromatic number could be viewed as a multivariate function of five network parameters, namely the network size (N), the number of modules (m), deviation in the number of nodes in the modules (e), and two probability measures defined as the (i) probability of having an edge within each module (the intra-modular connection probability: p_1) and (2) that of having an edge between different modules (the inter-modular connection probability: p_2). To simplify matters, only bi-modular graphs were considered (i.e., $m = 2$) where the modules were made to vary in their size by a single node ($e = 1$). Therefore, we were essentially looking at large communities of roughly equal size of each module (Supplementary Figure S8). Graphs were constructed from 10 nodes to 40 with an increment of 10 nodes, p_1 varying from 0.2 to 1.0 and p_2 from 0.1 to p_1, both at an interval of 0.1. By this setup, it was ensured that in the resulting networks, the probability of an intra-modular edge

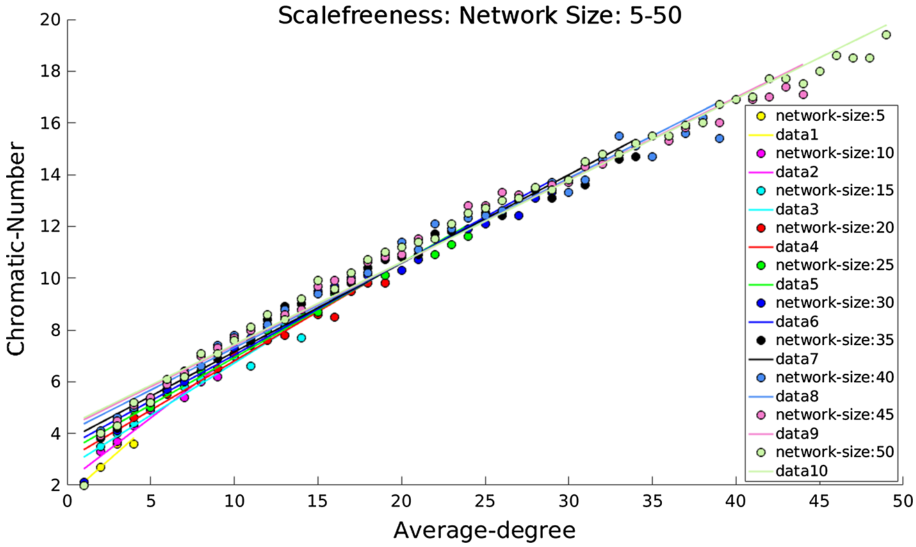


Fig. 6 Average chromatic number versus average degree plots exhibiting scale freeness. For each network size, the points could be best fitted to straight lines having almost identical slopes (~ 0.39). The standard deviation in the slopes was negligibly small (0.083)—implying that the distributions are independent of the network size (N) and hence scale free

formation is always higher than that of an inter-modular edge ($p_1 > p_2$) implementing modularity, with the sole exception of the extreme case of purely random networks resulting at their equality ($p_1 = p_2$) (other than the trivial case of the complete networks resulting from $p_1 = p_2 = 1$). In other words, there is no point in covering the whole range of probability values for both p_1 and p_2 since $p_2 > p_1$ will represent anti-modularity (rather than modularity), which is ambiguous in the given context. On the contrary, it is rather interesting to note that an alternative approach to generate modular networks as prescribed above would in fact result in k-partite graphs made of k modules by inverting the trends of the two probability parameters (i.e., $p_1 = 0,\ \ p_2 > p_1$). These graphs, by definition will be k-colorable and a prominent example should be the case of three regular bi-partite graphs (Supplementary Figure S9).

It is to be noted that the choice of $m = 2$ and $e = 1$ in effect will result in the formation of bi-modular networks of size n and n−2 where $n + (n-2) = N$.

Likewise the structured networks described before, link density for modular networks ($\mathrm{Ld}_{\mathrm{mod}}$) can also be derived analytically based on these network parameters (N, p_1, p_2) by the following expression (Eq. 3).

$$\mathrm{Ld}_{\mathrm{mod}} = \frac{p_1 \cdot \left({}^{n}C_2 + {}^{(n-2)}\,C_2\right) + p_2 \cdot n \cdot (n-2)}{{}^{(2n-2)}C_2} \qquad (3)$$

where $N = 2n-2$ and hence the denominator stands for the maximum possible links in the N-graph.

Consistent with the earlier calculations, again a statistical ensemble of 100 graphs was sampled for each set of fixed network parameters (N, m, e, p_1, p_2), the topological variability in the ensemble was accounted for by the motif identifier, and the average (μ_{mod}) and standard deviations (σ_{mod}) in the chromatic numbers were recorded.

Except for the trivial case of the complete graphs resulting from $p_1 = p_2 = 1$, most other combinations of network parameters exclusively gave rise to unique graphs (i.e., 100 out of 100 cases) (Supplementary Table S3). In contrast, chromatic numbers of the graphs constituting each of the statistical ensembles were again found to be remarkably stable, reflected in their low standard deviations (Supplementary Figure S10) compared to their corresponding means ($|\sigma|_{\mathrm{mod}} \sim 15.0 \pm 10.9\%$ of $|\mu|_{\mathrm{mod}}$). The standard deviations, however, were somewhat higher than those obtained for small-world and scale-free networks, although the average chromatic numbers could still be considered characteristic of the statistical ensembles. It was noteworthy that the standard deviations reduced gradually with increasing network size, attaining significantly low values for larger networks. This is expected since the modular networks are inherently partitioned into modules which itself is an influential, causal factor for the stability of chromatic number. Furthermore, the average chromatic numbers for larger modular networks should be able to buffer the local perturbations caused by the fluctuations in the probability parameters (p_1, p_2) in a better way compared to smaller networks.

Likewise to the small-world and scale-free networks, link densities (Ld) were also computed, both analytically ($\mathrm{Ld}_{\mathrm{exp}}$) and from the actual networks ($\mathrm{Ld}_{\mathrm{obs}}$) and their mean ($|\mathrm{Ld}_{\mathrm{obs}}|$), standard deviations ($\sigma_{\mathrm{Ld\_obs}}$) recorded for each statistical ensemble. Average observed and expected link densities ($|\mathrm{Ld}_{\mathrm{obs}}|$ and $\mathrm{Ld}_{\mathrm{exp}}$) were found to be almost identical (Supplementary Figure S11), with the extent of their agreement increasing with increasing network size (N), while the standard deviations in the observed link densities ($\sigma_{\mathrm{Ld\_obs}}$) followed an inverse trend.

Similar to the earlier analyses, the distribution of average chromatic numbers was plotted as a function of (p_1, p_2)

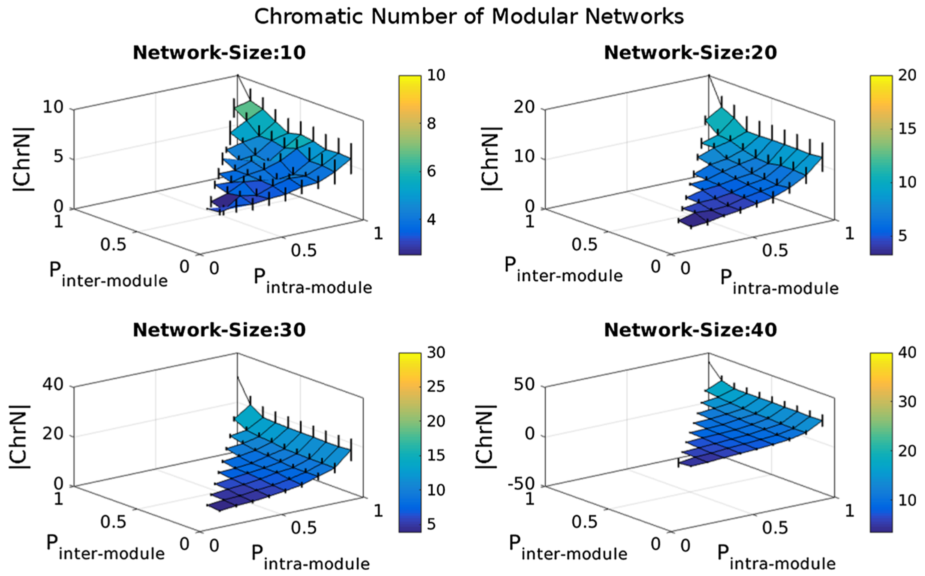


Fig. 7 Chromatic number of modular networks. Average chromatic numbers (|Chrn|) are plotted (as surfaces) as bivariate functions of the modular network parameters, the intra-modular connection probability: $p_{\text{intra-module}}$ (p_1 in the Main Text) and the inter-modular connection probability: $p_{\text{inter-module}}$ (p_2 in the Main Text)

as a surface (Fig. 7) for each distinct network size ($N = 10$, 20, 30, 40). The 'average chromatic number' surfaces were nearly similar in shape for all network sizes, growing steeply along increasing p_1 (i.e., along higher intra-modular connection densities) right up to its maximum possible value equaling the network size (N) upon reaching a complete graph for $p_1 = 1$, $p_2 = 1$.

Although the growth of average chromatic number along p_1 is fairly steep throughout its entire range, there is a clear point of saturation for the parameter before reaching completeness (i.e., $p_1 < 1.0$). As could be seen from the surface plots (Fig. 7), this point of saturation for chromatic number is attained at ~ 8 for $N = 10$; ~ 14 for $N = 20$; ~ 22 for $N = 30$; ~ 24 for $N = 40$; after which, there is an abrupt jump directly reaching the theoretical maxima, equaling the corresponding network size (N) for complete graphs. This physically means that the corresponding number of colors can actually take care of a whole range of connections (and graphs) until the critical threshold of completeness is attained at $p_1 = 1$, $p_2 = 1$.

It should be noted that the chromatic number of modular networks are largely dominated by the chromatic number of its largest module. With this understanding, bi-modular networks were judiciously designed where the size of the modules were varied by 2 nodes (by setting $m = 2$, $e = 1$). Now, since p_1 (the probability of having an intra-modular edge) was kept identical throughout both modules of the graph, the larger module will have a few more connections with approximately having the same connection density to that of the smaller module. This implies that the chromatic number of these bi-modular networks is unambiguously dominated by the chromatic number of the larger module. That is to say that the larger module is generally expected to cover the set of colors required to color the smaller module as well.

4.6 Stability of vertex coloring

It is highly unlikely that an identical network will reappear in an unbiased statistical ensemble for all the structured networks described till this point, since all of them have one or more random components in their construction. A careful re-investigation of the actual topological identity of the resultant networks (by implementing the 'motif identifier') could actually validate this hypothesis. Given such large topological variation, the statistics of the chromatic number shows remarkable stability of graph partitioning. This indicates that in certain structural problems, chromatic number has the potential to remain invariant against a flow of considerable topological variation in graphs.

The stability of chromatic number is in fact similar in character to chemical buffers (McIlvaine 1921), wherein change in pH due to the addition of acid or alkali is resisted by an adequate storage of an alkali or proton reserve, respectively. Here, rewiring of links maps parallel to the addition of acid or alkali, whereas the presence of a buffer acts as a restoring force similar to the absorptive property inherent in graph partitioning.

4.7 A comparative outlook of vertex coloring across different structured networks

Apparently, one technical difficulty to compare the trends of chromatic number across different network types is that they were systematically constructed by distinctly different network parameters. However, the network size (N) and the link density (Ld) as an ordered pair can potentially be viewed as a reduced representation of a graph that can bridge the gap across different network types and provide a common conceptual platform to discuss their relative trends. From that platform, here we attempt to point out some key observations by means of a comparative study.

The network size (N) of 20 was chosen as a test case which is common to all three statistical samplings (i.e., scale free, small world and modular). The average chromatic numbers (crn) were then plotted as a function of link density (Ld) for each network type (Supplementary Figure S12). All three 'crn versus Ld' plots could be best fitted to quadratic polynomials with R^2 values of 1.00, 0.92 and 0.99, respectively, for scale-free, small-world and modular networks.

Scale-free networks attained the lowest maximum value for the average chromatic number ($\sim$ 9.8), while the other two types (small-world and modular) could climb to their theoretical maxima of 20 upon completeness. It should be noted here that scale-free networks would never attain completeness since the degree distribution in complete graphs does not follow a power law. This is true even when average degree, k, is sampled to be $N-1$, the maximum possible value the parameter can attain. Dynamic real-world networks growing in size with time generally corresponds to scale-free networks. This, in effect, should be the best possible manifestation of the absorptive property of chromatic number, wherein newly added nodes will get absorbed within the partition of already existing color classes. The other network types can, however, attain completeness, by definition. In more precise terms, a complete network is in fact the one with the best display of small-world properties, since it is locally the most cohesive (clustering coefficient = 1) and globally the most reachable (characteristic path length = 1). On the other end, the extreme case of modularity can be extrapolated to completeness where both intra- and inter-modular connection probabilities converge to their maximum value of 1.

For the corresponding small-world networks, a thin bin of link densities (or practically the same value) could actually give rise to a much thicker array of average chromatic numbers—thereby attaining the least of the R^2 values in its quadratic fit compared to the other network types. This physically means that the topological variation is quite wide given the same density estimates mostly influenced by different rewiring probabilities giving rise to different chromatic numbers. However, it should not be mistaken with the fact that chromatic number is generally remarkably stable against topological variations for the same global network parameters (N, Ld).

The average chromatic numbers of the corresponding modular networks are less stable than the other network types which are reflected in their scatter (Supplementary Figure S12). This is perhaps anticipated because of at least two plausible causal factors: (1) in modular networks, each module could potentially have a diverse plethora of topological variability; however, the chromatic number is dominated by the colorability of the largest (or the more connected) module, as discussed earlier; (2) modular networks approach completeness as a function of two parameters simultaneously, both approaching their maximum values (Lim, $p_1 \rightarrow 1$, $p_2 \rightarrow 1$), whereas small-world networks proceed toward completeness as a function of just one parameter (viz. average degree, k) approaching its maxima (Lim, $k \rightarrow (N-1)$), while scale-free networks rule out completeness. This also potentially adds extra variation in the coloribility of modular networks compared to either small world or scale free.

4.8 Chromatic number of regular networks

As the name suggests regular networks are the most ordered among the different types of structured networks. They are formally defined as networks where all the nodes have identical degrees; i.e., a k-regular graph of N nodes has k edges incident to each of the N nodes. Regular networks are prevalent in different three-dimensional physicochemical extended structures, e.g., crystal lattice, consecutively concatenated carbon chains in graphenes, polybenzene hydrocarbons, hydrogen bonded networks in water structures, etc. All these structural ensembles are extended in all three dimensions and potentially infinite in their growth. From a network point of view, however, all the above examples have essentially low regular degrees assigned to each node, e.g., 3 for carbon in graphenes, 4 for the oxygen in water and so on.

An independent small calculation was carried out to investigate the chromatic numbers of regular networks. Unlike the previous analyses (for small-world, scale-free, and modular networks) here, no statistical ensembles were considered for two reasons: (1) since regular networks are generally restricted (compared to other structured or random networks) in their topological space for a given set of fixed network parameters and (2) since the motif identifier is incompetent to discriminate between non-isomorphic k-regular networks of N nodes (as elaborated in Sect. 4.1). Thus, the regular networks were built based on two network parameters namely the network size N and the degree of each node (k). The sampling range for this network parameters was set in accordance with the calculation for small-world networks, varying N from 4 to 30 at an interval of 2, while varying k from 2 to ($N-1$). Setting $k = 1$ would lead to a collection of single, mutually disjoint edges which is ambiguous in the given context. On the other end of the spectra, $k = (N-1)$ would lead to complete graphs which are also regular. Also, as a rule, N and k cannot have odd values simultaneously, which was trivially avoided by the sampling setup. Likewise, to the previous analyses the chromatic numbers were considered as a bivariate function of the two network parameters (N, k) and plotted as a

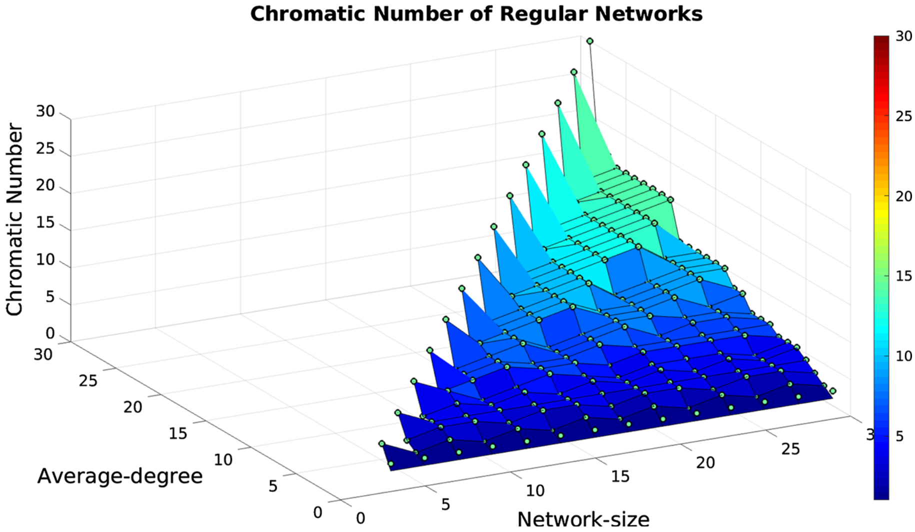


Fig. 8 Chromatic number of regular networks. Chromatic numbers are plotted by (as surfaces) as a bivariate function of the network size (N) and the average degree (k) for regular networks

surface plot (Fig. 8). In noticeable contrast to all earlier surface plots (for small-world, scale-free, and modular networks) here, the surface was found to be considerably rough. In greater detail, for any given network size (N) the increase in chromatic numbers was step-wise as an ascending function of k, resembling staircase like patterns. Similar trajectories are reminiscent of transitions involving multiple intermediate steady states. Effectively, the whole surface looks like a rocky mountain with approximately planar valleys, in between hierarchically increasing steepness to climb. In other words, the chromatic number for a given network size (N) follows a characteristic hopping of sequential linear increase and saturation. This physically is again reflective of the remarkable absorptive property of chromatic number, this time in regular graphs. That is to say that for a particular size of regular graph, the same chromatic number can absorb a range of average degrees, that is an increasing number of connections, till a critical point of disjuncture is reached, triggering a further increase in chromatic number.

4.9 Computational complexity

As is well known, the graph coloring problem belongs to the NP-hard category (which is most relevant for characteristic k-regular graphs with a non-trivial average degree, $k > 2$) (Dailey 1980; McDiarmid and Sánchez-Arroyo 1994). An NP-complete problem will have both NP (non-deterministic in polynomial time) and NP-hard components, wherein an NP-hard algorithm contains at least one component (say a subroutine) which is NP and is hence said to be at least as hard as an NP (Arora and Barak 2009). For these problems (e.g., problems in combinatorics), the complexity of the algorithm, $f(N)$ increases in a non-polynomial manner (i.e., at least exponentially) with the order of the problem, N,[7] implying $f(N) = O(2^N)$. Again, there could be many other problems (Karp 1972) with even greater complexity than that of exponential, also falling into the same 'NP' category. For example, it is trivial to think of a 'blind search' for the very problem of finding the chromatic number of a graph. In such an algorithm, N steps will be required to color N nodes *in turn*, and hence, there are N! non-degenerate sequences of nodes to be colored sequentially in N steps, accounting for N! ways to color the graph exhaustively. This will lead to a complexity of $O(N!)$—which is even a higher-order function than that of exponential ($O(2^N)$).

The other category is said to be P-problems (deterministic in polynomial time) where the complexity follows an order of a polynomial function (i.e., $f(N) = O(N^\alpha)$; $\alpha > 1$). To test whether the current algorithm is able to find the approximate solution for the chromatic number with a descent accuracy within the polynomial time domain, regular graphs were constructed of size (N) varying from 4 to 512 following a geometric progression with a common ratio of 2 (i.e., 4, 8, 16, 32 … 512). To simplify matters, the other parameter, namely the average degree (k), was set to $N/2$; chromatic numbers were calculated and the run time was recorded for each run. By this setup, the complexity (run time) could effectively be envisaged as a mono-variate function of N, since k is automatically fixed upon fixing N. Consistently throughout this entire calculation, the program was run in its multiple iteration mode (in contrast to the faster single iteration mode) to render the best possible accuracy of the predicted chromatic numbers. All the calculations were consistently run under a 8 core dual processor Intel(R) Xeon(R) CPU E5-2609 v4 @1.70 GHz in CentOS, Linux. A plot (Fig. 9) of complexity versus

[7] The order of the problem refers to the graph-size (N) in the current context of the 'graph coloring' problem.

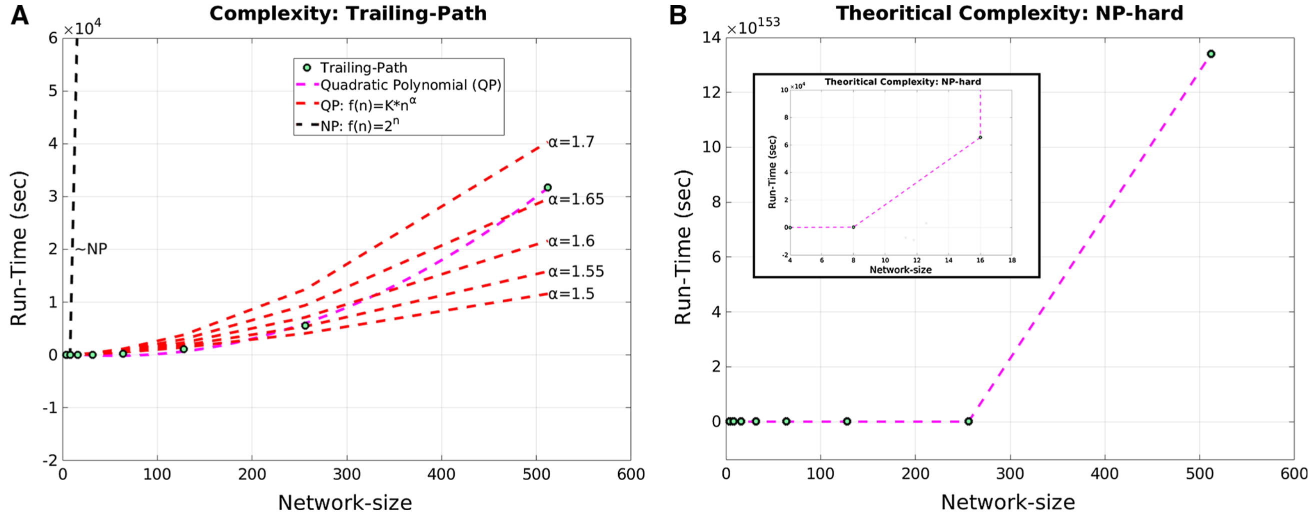


Fig. 9 Complexity of the algorithm. Panel A of the composite figure, plots the calculated complexity (*f*(*N*)) versus order (*N*) which fits exceptionally ($R^2 = 0.96$) to a quadratic polynomial (blue dashed line). The predicted chromatic numbers (filled circles in 9A) are 2, 4, 6, 11, 22, 45, 89 and 178, respectively, in the ascending order of graph size from 4 to 516 (see Main Text). The red dashed lines represent fractional exponents (α—alpha) fed to the polynomial function ($f(N) \sim N^\alpha$) in an attempt to envelope the complexity profile, wherein an exponent of 1.7 was found to be optimal (see Main Text) (color figure online)

order (*N*) fits exceptionally ($R^2 = 0.96$) to a quadratic polynomial (i.e., $f(N) = O(N^2)$) which reflects the fact that doubling the order of the problem would increase the run time merely by a factor of 4. Fractional exponents were sampled in an attempt to narrow-down the margin of the polynomial functions ($f(N) = O(N^\alpha)$) that would be able to envelope the complexity profile. An exponent of 1.7 was found to be good enough for the purpose (Fig. 9a). In dramatic contrast, the (theoretical) complexity derived from a corresponding NP profile shoots exponentially as a function of the same order (Fig. 9b). For example, ~ 1.56 s is all that the algorithm takes (even in a higher-level interpreted language like that of MATLAB) to complete the run for a regular network of size 16 (with an average degree of nodes set to 8), whereas the corresponding NP run time is $O(2^{16})$ amounting to ~ 18.20 years. Thus, the run time maps to the polynomial time domain even with the 'multiple iteration mode' of the program. Due to the extreme difference in the two scales in Fig. 9a, b (polynomial and non-polynomial), the NP-complexity profile is independently drawn as a separate panel (Panel B) and a section corresponding to small network size (up to $N \sim 20$) is magnified in the inner window of the same panel by means of comparison.

4.10 Comparison with other heuristics

There have been many attempts to come up with greedy heuristics for finding out the chromatic number of a graph—as surveyed in several reviews and comparative studies, time and again (Kosowski and Manuszewski 2004; Lai et al. 2006). There are the following existing heuristics that are mostly used in applications. The 'largest first' heuristic finds and considers the nodes with largest degree at first for coloring. The 'independent set' heuristic finds an independent set of the graph and colors them with same color. The 'Connected sequential bfs' and 'Connected sequential dfs' heuristics color the nodes in the order in which they are encountered in BFS (Breadth First Search) and DFS (Depth First Search) traversal, respectively. The DSATUR heuristic considers nodes in the order of their saturation which has been explained later in details. Finally, the 'Random sequential' heuristic colors the nodes in random order.

Generalizations to the problem have also been proposed, by means of partition (Fidanova and Pop 2016) and selective (Demange et al. 2014) graph coloring problems, implementing evolutionary algorithms like hybrid ant colony optimization, etc., leading to improved performances. Traditionally, two of the most popular heuristics with contrasting strategies have been the largest first (LF) (Welsh and Powell 1967) and the DSATUR (Brélaz 1979), both of which find an approximate solution for the chromatic number with reasonably good accuracy. As is suggested in the name, LF and its derivative strategies [e.g., Distributed LF (Hansen et al. 2004)] follow a descending order of degrees to color nodes and use random seeds to break ties in degrees wherever applicable. On the other hand, DSATUR is unique in its strategy to change the order of the nodes (http://cs.indstate.edu/tdu/mine1.pdf) based on maximizing the saturation (or color unavailability) to select new candidate node(s) to color (San Segundo 2012). In most cases, DSATUR is known to yield an optimal coloring for different types of graphs (Janczewski 2001; Lai et al. 2006).

As it turns out, the present algorithm may be envisaged as a meticulous combination of the search patterns of LF and DSATUR, as it is compiled of two similar heuristics, one that follows a descending order of degrees (likewise to that of LF), and another that finds out the next neighboring node to be colored, with a higher saturation or a less number of available colors (likewise to DSATUR). However, the most salient feature of the present algorithm remains to be its strategic choice to traverse along a 'trailing path' of consecutively connected nodes for the entire course of coloring, which distinguishes it from both LF and DSATUR. The adaption of the trailing path enables the algorithm to always follow a path of connected nodes (which is effectively a spanning tree) through the graph, which, in turn, results in a continuous coloring scheme. This continuous coloring scheme is critical to avoid assigning redundant colors in potentially conflicting cases that may arise due to a discontinuous local coloring otherwise. Note that for such discontinuous coloring schemes (e.g., LF and DSATUR), there is a greater chance to assign redundant colors for conflict due to local coloring, under the constraint of not assigning the same color to any two adjacent nodes. However, keeping track of the color saturation (or availability) for all the vertices at each trivial step requires a bit more space (i.e., computer memory) in the continuous coloring scheme (i.e., in the 'trailing path'), compared to the discontinuous ones.

One other salient feature of the present algorithm is that it deletes a node along with all its incident edges subsequent to coloring it, which enables it to reevaluate and update the true degrees of the remaining nodes in context of the new (resultant) graph. Thus, if and when there is a case of more than one disjoint components remaining to be colored, the algorithm has the advantage (over the earlier approaches) to color them simultaneously as if coloring two independent (sub-)graphs, which would further reduce the computational complexity.

While an efficient implementation of single iteration of coloring of the graph in the present algorithm will run in $O((m + n) \log n)$ (where *m is the average degree of nodes and n is the graph size*), same to that of the running time of DSATUR, in addition, 'trailing path' explores not only one sequence of possible coloring but various different coloring sequences as may be potentially needed in a given context. This is done by exploring different possible paths, when the algorithm has to break a tie in either LF or DSATUR, thereby attaining more precision and remaining flexible at the same time. This is equivalent to running the algorithm repeatedly and exploring a solution which has not been explored before. Afterward, the minimum value (for the approximate solution of the chromatic number) coming from all (independent) runs is considered, where the number of runs may be specified by the end user. Due to the nature of the problem (NP-Complete), it is evident that to get the optimal chromatic number of a graph, the algorithm is required to be run exhaustively for an exponential time. The advantage of the current approach is that the choice of trade-off is left to the user so that the user can actually decide as to how many different ways the iteration(s) need to run. The more number of iteration is used, the more is the probability of getting the optimal chromatic number for a graph. For the cases where DSATUR and LF find the chromatic number very easily, the algorithm can detect and stop further iterations by itself. These algorithms are heuristics rather than (merely) approximation algorithms (Lewis 2016) as both of them use random seeds to break ties in their corresponding parameters (i.e., color availability and degree) wherever applicable. Likewise, the involvement of random selection of neighbors and different order of nodes to be colored makes the 'trailing path' algorithm a heuristic as well. In practice, it is seen that the present algorithm performs almost always better than the existing heuristics due to the meticulous amalgamation of the two approaches.

4.10.1 A case study of performance comparison on regular graphs

A detailed comparison of accuracy given by different heuristics (Kosowski and Manuszewski 2004) can be found in Table 1, wherein seven existing heuristics (H1 to H7) represent strategies, named largest_first (H1), independent_set (H2), Connected_sequential_bfs (H3), Connected_sequential_dfs (H4), Smallest_last (H5), Saturation_largest_first (H6) and Random_sequential (H7) and TP represents the Trailing_path algorithm. For the comparison, regular graphs of size (n) 4–512 were constructed, sampled at a geometric progression with a common ratio of 2 (i.e., 4, 8, 16 … 512), while the degree of each node was set to the half of the graph size (i.e., $k = n/2$). The choice of regular graphs for the comparison was based on the well-known difficulty in finding the exact solution for their chromatic number, due to the indiscernible nature of their nodes. Each heuristic was made to run on each size of graph for 100 times, and for algorithms returning non-identical values for the computed chromatic number, the range of values obtained was recorded. Note that there are methods (H1 to H6 in Table 1) which return an identical value in each iteration and their performances are equally suboptimal. Noticeably, its only heuristics involving randomness which return different solutions on different runs on the same graph(s), and, more importantly, they are the ones that produce the best (lowest) solutions,

Table 1 Comparison of accuracy among different heuristics tested on regular graphs

Graph size	Degree	Solution (minimum number of colors)								Elapsed time (s) for TP
		H1	H2	H3	H4	H5	H6	H7	TP	
4	2	2	2	2	2	2	2	2	2	0.00814 (± 0.00041)
8	4	5	5	5	5	5	5	4	4	0.01965 (± 0.00055)
16	8	6	6	6	6	6	6	7	6	0.06010 (± 0.00145)
32	16	14	14	14	14	14	14	11–15	11–12	0.20721 (± 0.00520)
64	32	30	30	30	30	30	30	22–25	22–23	0.78163 (± 0.01583)
128	64	62	62	62	62	62	62	44–47	43–47	3.19818 (± 0.05391)
256	128	126	125	126	126	126	126	87–92	88–94	14.20557 (± 0.26600)
512	256	254	254	254	254	254	254	172–180	171–188	63.42610 (± 0.92533)

See Main Text for description of the methods. For H7 and TP, there are variations in the computed minimum number of colors (beyond graph size 16) and in these cases the range of values returned has been tabulated (i.e., the minimum and the maximum)

or solutions closest to the chromatic number (Table 1). On that note, the current algorithm unequivocally performs far better than all other heuristics except for the 'random sequential' (H7 in Table 1) approach to which it performs comparably (if not better). And, it is highly dependent on the context of a given graph as to which of the two heuristics (both involving randomness) perform better. It is important to take a note of that H1 and H6 in Table 1 represent LF and DSATUR, respectively, and, as is evident from the table, the 'trailing path' (TP) algorithm unambiguously performs better than both LF and DSATUR.

The complexity of the current method, however, can not be directly compared with that of the existing heuristics in terms of computational run time (or elapsed time) due to non-uniformity of the languages used to code them. More precisely, the competing methods have been compiled in a highly optimized python environment, implementing C, C++ libraries (obtained from https://networkx.github.io/documentation/networkx-1.1/install.html), whereas the current (trailing path) algorithm has been coded in MATLAB which is a higher-level interpreted language. One of the prime reasons behind the choice of MATLAB is because of its advanced and interactive graphics so that the program returns the desired visual displays of the colored graph, as a default, without having the necessity for additional coding or installing complex libraries. However, the display can also be skipped (if not desired) from the user interface by resetting appropriate variables. It takes even less time for the program to run in a no-display mode. It is noteworthy that even in such a high-level interpreted language environment, the current MATLAB code takes about only a minute to run on a massively large and complicated regular graph of size 512 (with the degree of each node set to 256) to output a solution appreciably close to the actual chromatic number.

4.11 A special example of map coloring

As described vividly in the introduction, the very idea of graph coloring was brought about from the context of coloring of physical geographic maps which corresponds to graphs embedded in a plane. In fact, a coloring problem of such planar maps can actually be transformed into the vertex coloring problem of their dual graphs. In these maps, any geographic location is a node and a link exists between those nodes which share a common 'border' (or 'boundary'). In such a context, there are examples of geographic maps consisting of several consecutively connected territories (forming a closed loop/cycle) further connected to a central territory (hub node). The corresponding graph representation in such cases consists of one or more overlapping closed triplet cliques (for any number of neighboring nodes greater than one), and hence the lower bound of chromatic number is trivially 3. However, the upper bound of the chromatic number varies as the number of members (neighbors) in the closed loop being even or odd. For an even cycle with a central hub node, the chromatic number remains 3; however, for an odd cycle with a central hub node, the graph has to be at least 4-colorable, due to topological constraints. Interestingly and importantly, increase in the number of neighbors in the cycle, in steps of two, does not further increase the chromatic number beyond these theoretical bounds (i.e., even: 3, odd: 4). They are therefore characteristically called 3- and 4-chromatic planar graphs (Harary 1969), respectively. Such graphs are prevalent geographically, and in the current study, four examples of the latter case (viz., 4-chromatic planar graphs) have been presented (Supplementary Figure S13) where the central hub nodes are represented by Indian states: Madhya pradesh and Chhattisgarh; and US states: Nevada, Georgia.

4.12 Perspective

Graph coloring is a manifestation of 'partitioning' of graph (Andreev and Räcke 2004), wherein nodes (and edges) are partitioned on the basis of their relative adjacencies. In fact, the very concept of 'partitioning' is far more general than to be restricted to graphs alone and rather serves as a compartmentalization of any structural problem. A prominent example may be the very well posed and yet unsolved 'protein folding' problem (Dill and MacCallum 2012) in structural biology—where one of the major concerns is the dynamic stability of the folded proteins (Roy et al. 2015). This is common in any dynamic natural system, and the current study reveals that 'coloribility' of graphs can actually be envisaged as a parallel to stability estimates in such dynamic natural systems. Different packing modes (Basu et al. 2011), alternative to that of the native protein fold, have been designed (Street and Mayo 1999) and experimentally verified (Berhanu and Masunov 2012) to be stable (Jiang et al. 2000) and active[8] satisfying certain overall 'packing' and 'electro-chemical' constraints (2013). Such constraints include atomic packing density (Gerstein et al. 1995; Tsai et al. 1999), global electrostatic balance (Basu et al. 2012), shape complementarity of the core residues with their local neighborhood (Banerjee et al. 2003; Basu et al. 2011), distribution of hydrophobicity/polarity in terms of burial of solvent exposure of amino-acid residues (Lee and Richards 1971) and may be a few more (Basu et al. 2014). Likewise, different graph topologies with appreciable variation in their corresponding adjacencies can lead to the same partitioning, and hence, the same coloribility results in identical chromatic numbers. Furthermore, a small interval (Δ) of chromatic number centering about a mean-value can actually accommodate for a combinatorial expansion of morphologically distinct non-identical networks—which could be potentially generated either from a fixed set of network parameters or from their slight variations. In noticeable similarity, the aforementioned physico-chemical constraints in protein folding can accommodate for several alternatively designed (packed) hydrophobic protein cores (Munson et al. 1996), and hence different amino acid sequences. On that note, there are well-characterized 'twilight' (Rost 1999) and 'midnight' (Pirun et al. 2005) zones of pairs of proteins belonging to the 'low sequence identity—same fold' category. The fact that graph partitioning based on 'coloribility' is a stable and absorptive property for structured networks (namely small-world, scale-free, modular, etc.) allegorically matches with the concept of stability in protein folding in terms of the physico-chemical constraints, allowing for guided variation in the designed amino acid sequences.

4.13 The plausible use of graph partitioning in protein design—a case study

To demonstrate the above proposition by means of an actual calculation, here we present a case study of designed and corresponding native proteins. The designed proteins were borrowed from Zhu et al. (2016) which aims to backtrack the origin of a folded repeat protein toward a possible intrinsically disordered evolutionary ancestor. The desired protein fold was expected to contain a tetratricopeptide repeat (TPR) where the repeat units are helical hairpins which interact via specific geometry involving knobs-into-holes packing (Crick and IUCr 1953).

A ribosomal protein, namely RPS20, was chosen as a representative of the TPR fold—which lacks an ordered three-dimensional structure outside the ribosomal context (Peng et al. 2014), thereby belonging to the class of intrinsically disordered proteins. Two native experimental high-resolution X-ray crystal structures from the protein data bank (Berman et al. 2000) (PDB ID[9]: 1na0_A; and 2vqe_T) were fused to serve for the structural templates for the in silico design of a series of multi-mutants (namely M2, M4E, M4N, M4RD, M5 and M4NΔC) spanning a wide range of 8 potential candidate mutation sites. In addition, other native proteins homologous to the repeating units were also identified, serving to generate a sequence consensus (PDB ID: 3ax3_A; 2v6y_A; 2v1s_A; 3ax3_A; 1wfd_A, 2rpa_A; 2v6y_A; 2v6y_B; 4a5x_A) and create a profile based on that. A fraction of the designed mutants was found to retain their stable folds in solutions out of

[8] Designed protein cores have been found to follow a 'activity-stability trade-off', wherein, the higher stability is gained at the cost of loosing activity and vice versa, analogous to the event of enthalpy-entropy compensation.

[9] The four letter accession code followed by the chain identifier.

which the structure of three were solved experimentally by X-ray crystallography (M4N: 5fzq, M4NΔC: 5fzr, 5fzs).

To investigate the similarity/difference in contact networks found at the interior of these helical (all-α) proteins, a previously standardized protocol (Basu et al. 2011; Roy et al. 2015) was implemented based on pairwise surface complementarity and overlap (Banerjee et al. 2003) cutoffs set to completely or partially buried interacting amino acid side-chains to filter out the real contacts. Thus, each protein interior was mapped to one or more surface contact networks (Basu et al. 2011) sustaining specific geometry consistent with the native fold. The minimum network size considered was 3 with no limit set to the maximum. The collection of these networks was then subjected to the identification of topological variability (Basu et al. 2011) by implementation of the motif identifier (as defined in Sect. 4.1) along with the calculation of their chromatic numbers. As expected, the explored topological space from these networks showed considerable variability giving rise to a large set of unique and distinct packing motifs, mostly falling into the motif-families (Basu et al. 2011) previously explored within globular proteins (e.g., trees, 3 cliques, etc.). Out of a collection of 28 contact networks obtained from a list of 12 (native + designed) proteins, 22 unique graphs (motifs) could be identified, 19 of which were found just once in the list, 2 (namely, trivial linear trees consisting of 5 and 7 nodes) were found twice, and the most trivial possible motif, namely the linear tree of 3 nodes (1 ∼ 2: 2, 2 ∼ 1–1: 1), found five times (Supplementary Table S4). Regular graphs were absent throughout, which confirms the correctness of the topological variability accounted for by the motif identifier (as elaborated in earlier sections). In fact it is worth making a note that other than the simple 4 cycles, all other k-regular graphs ($k > 2$) are structurally forbidden due to atomic steric constraints attributed to protein contact networks (Basu et al. 2011). Most intriguingly, all the motifs found in the current calculation were either at least 2- or 3-colorable irrespective of their size and topological variability (Supplementary Table S4). Visual inspection of the motifs (Fig. 10, Supplementary Figure S13) confirmed that all the graphs so obtained were planar, without a single exception—which is reflected in their characteristic low values of the chromatic numbers (2 or 3). This result is also consistent with the theory of atomic steric clashes among tightly packed amino acid side-chains leaving almost no realistic scope for mutually intersecting edges on a plane[10] to occur within folded proteins. This effectively means that there is a theoretical upper bound to be considered for chromatic numbers for contact networks within folded proteins, which is four. Recall that four colors are sufficient to color any map (Appel and Haken 1977) which essentially corresponds to the coloring of its dual planar graph. This should serve as an important rule of thumb in the general guidelines for protein design. Furthermore, the study also highlights the plausible use of graph partitioning (chromatic number) as a critical bottleneck filter in a computational pipeline aiming for directed design of proteins. The results strongly support the theory of alternative packing modes leading to the same stable fold within native and/or designed protein interiors.

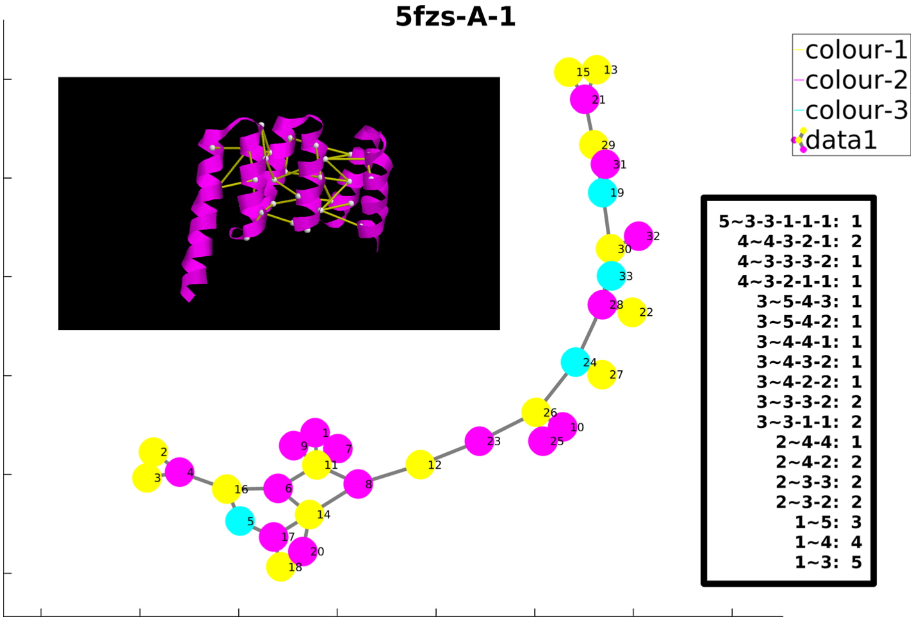
4.14 Future application

In future, the current algorithm may also be designed to handle additional constraints in the ‘graph coloring problem’ applicable to highly specific graphs. For example, say, all nodes distant from each other having a certain ‘high’ degree be assigned the same color. This will benefit to visually analyze the graph in greater detail, and identify, at a glance, those nodes having the same ‘high’ degree, analogous to identifying nucleation sites. The physical interpretation will of course be contextual. It is likely that these nodes share common implicit properties, thereby falling not only to the same color class but also attaining the same degree. Also, in a follow-up task for the future, a separate (third) heuristic may be designed to break the current ties between nodes having both an identical degree and an equivalent color saturation—which is done randomly in the present algorithm.

5 Conclusion

Graph coloring is a challenging problem in mathematics and computer science. The current study presents an elegant approximate solution to the problem by the strategic implementation of a ‘trailing path’ (effectively, a continuous coloring scheme), alongside meticulously combining two previous approaches to lead to a novel compound heuristic and a corresponding software code (in MATLAB and Octave). A single iteration of the program returns optimally accurate solutions, favorably comparing to the *state of the art* (actually performing better than most heuristics in the business). The program can also be run in a multiple iteration mode to make the algorithm trail through different random paths of consecutively connected nodes (effectively mapping to spanning trees of the graph) while minimizing the approximate solution for the chromatic number, aimed potentially for a better accuracy. Setting this (desired) trade-off between accuracy and run time remains a choice of the user. The program consistently hits running times of the polynomial order with respect to the input graph size. The study of graph

[10] By definition, edges do not intersect in planar graphs.



5~3-3-1-1-1:	1
4~4-3-2-1:	2
4~3-3-3-2:	1
4~3-2-1-1:	1
3~5-4-3:	1
3~5-4-2:	1
3~4-4-1:	1
3~4-3-2:	1
3~4-2-2:	1
3~3-3-2:	2
3~3-1-1:	2
2~4-4:	1
2~4-2:	2
2~3-3:	2
2~3-2:	2
1~5:	3
1~4:	4
1~3:	5

Fig. 10 Example of a 3-chromatic planar graph within a designed protein interior. The figure displays a contact network within a designed protein interior (PDB ID: 5FZS) mapping to a 3-chromatic planner graph. The inner (black) window represents the actual contact network (represented by yellow sticks) displayed on the actual protein structure (helices displayed as magenta ribbons), while the corresponding colored graph returned by chromnum is a reduced representation of the same, displayed as a ball-and-stick view. The network topology enumerated by the motif identifier (see Main Text) is presented as the hash table within the box at the right bottom of the picture (color figure online)

partitioning in random, structured and real-world networks shows its remarkable stability and absorptive property across a wide variety of graph topologies. This is an important finding in the context of compartmentalization in any structural problem. The software can directly be implemented in the targeted design of real-world networks. For example, interior designers of different disciplines should be potentially benefited by the software. Furthermore, there are several day-to-day real-life problems like guarding an art gallery, round robin sports and aircraft scheduling, etc., which should also be addressed adequately by the general algorithmic solution. We also discuss the interesting special cases of four chromatic planar graphs in the context of map coloring. Finally, the application of graph partitioning in compartmentalization has been demonstrated by means of an actual calculation in structural biology, wherein the chromatic number was envisaged as a measure of stability and uniqueness in protein contact networks, effectively serving as a plausible bottleneck filter in a protein design pipeline.

Acknowledgement The work was supported by the Department of Science and Technology—Science and Engineering Research Board (DST-SERB research Grant PDF/2015/001079). We take the opportunity to thank Mr. Arnab Kar (Department of IT, IIIT Alahabad) for his brief participation during the revision.

Authors' contributions SB conceived the problem. AB and SB designed the algorithm. AB wrote the initial MATLAB code which was improved at different stages by both AB and SB. For the analysis, AB provided small scripts which were executed by SB to carry out the computational experiments. AB and SB analyzed the results. SB wrote the paper with help from AB. AKD participated in the comparison with other heuristics and provided crucial notes at different portions of the manuscript. All authors read and approved the final manuscript.

Compliance with ethical standards

Conflict of interest None of the authors have any competing interests in the manuscript.

References

Albert R, Barabási A-L (2002) Statistical mechanics of complex networks. Rev Mod Phys 74:47–97. https://doi.org/10.1103/RevModPhys.74.47

Albertson MO, Cranston DW, Fox J (2010) Crossings, colorings, and cliques. ArXiv10063783 Math

Andreev K, Räcke H (2004) Balanced graph partitioning. In: Proceedings of the sixteenth annual ACM symposium on parallelism in algorithms and architectures. ACM, New York, pp 120–124

Appel K, Haken W (1977) Every planar map is four colorable. Part I: discharging. Ill J Math 21:429–490

Arora S, Barak B (2009) Computational complexity: a modern approach, 1st edn. Cambridge University Press, New York

Banerjee R, Sen M, Bhattacharya D, Saha P (2003) The jigsaw puzzle model: search for conformational specificity in protein interiors. J Mol Biol 333:211–226

Basu S, Bhattacharyya D, Banerjee R (2011) Mapping the distribution of packing topologies within protein interiors shows predominant preference for specific packing motifs. BMC Bioinform 12:195. https://doi.org/10.1186/1471-2105-12-195

Basu S, Bhattacharyya D, Banerjee R (2012) Self-complementarity within proteins: bridging the gap between binding and folding. Biophys J 102:2605–2614. https://doi.org/10.1016/j.bpj.2012.04.029

Basu S, Bhattacharyya D, Banerjee R (2014) Applications of complementarity plot in error detection and structure validation of proteins. Indian J Biochem Biophys 51:188–200

Berhanu WM, Masunov AE (2012) Alternative packing modes leading to amyloid polymorphism in five fragments studied with molecular dynamics. Biopolymers 98:131–144. https://doi.org/10.1002/bip.21731

Berman HM, Westbrook J, Feng Z et al (2000) The protein data bank. Nucleic Acids Res 28:235–242. https://doi.org/10.1093/nar/28.1.235

Blum M, Metcalf P, Harrison SC, Wiley DC (1987) A system for collection and on-line integration of X-ray diffraction data from a multiwire area detector. J Appl Crystallogr 20:235–242. https://doi.org/10.1107/S0021889887086783

Bollobás B, Catlin PA, Erdös P (1980) Hadwiger's conjecture is true for almost every graph. Eur J Comb 1:195–199. https://doi.org/10.1016/S0195-6698(80)80001-1

Brélaz D (1979) New methods to color the vertices of a graph. Commun ACM 22:251–256. https://doi.org/10.1145/359094.359101

Choromański K, Matuszak M, Miękisz J (2013) Scale-free graph with preferential attachment and evolving internal vertex structure. J Stat Phys 151:1175–1183. https://doi.org/10.1007/s10955-013-0749-1

Clauset A, Shalizi CR, Newman MEJ (2009) Power-law distributions in empirical data. SIAM Rev 51:661–703. https://doi.org/10.1137/070710111

Crick FHC, IUCr (1953) The packing of -helices: simple coiled-coils. In: Acta crystallogr. http://scripts.iucr.org/cgi-bin/paper?S0365110X53001964. Accessed 30 Nov 2016

Dailey DP (1980) Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. Discrete Math 30:289–293. https://doi.org/10.1016/0012-365X(80)90236-8

Demange M, Monnot J, Pop P, Ries B (2014) On the complexity of the selective graph coloring problem in some special classes of graphs. Theor Comput Sci 540–541:89–102. https://doi.org/10.1016/j.tcs.2013.04.018

Deng W, Chen R, Gao J et al (2012a) A novel parallel hybrid intelligence optimization algorithm for a function approximation problem. Comput Math Appl 63:325–336. https://doi.org/10.1016/j.camwa.2011.11.028

Deng W, Chen R, He B et al (2012b) A novel two-stage hybrid swarm intelligence optimization algorithm and application. Soft Comput 16:1707–1722. https://doi.org/10.1007/s00500-012-0855-z

Deng W, Yang X, Zou L et al (2013) An improved self-adaptive differential evolution algorithm and its application. Chemom Intell Lab Syst 128:66–76. https://doi.org/10.1016/j.chemolab.2013.07.004

Deng W, Zhao H, Liu J et al (2015) An improved CACO algorithm based on adaptive method and multi-variant strategies. Soft Comput 19:701–713. https://doi.org/10.1007/s00500-014-1294-9

Deng W, Yao R, Zhao H et al (2017a) A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm. Soft Comput. https://doi.org/10.1007/s00500-017-2940-9

Deng W, Zhao H, Yang X et al (2017b) Study on an Improved adaptive PSO algorithm for solving multi-objective gate assignment. Appl Soft Comput 59:288–302. https://doi.org/10.1016/j.asoc.2017.06.004

Deng W, Zhao H, Zou L et al (2017c) A novel collaborative optimization algorithm in solving complex optimization problems. Soft Comput 21:4387–4398. https://doi.org/10.1007/s00500-016-2071-8

Deng W, Zhang S, Zhao H, Yang X (2018) A novel fault diagnosis method based on integrating empirical wavelet transform and fuzzy entropy for motor bearing. IEEE Access 6:35042–35056. https://doi.org/10.1109/ACCESS.2018.2834540

Deng W, Xu J, Zhao H (2019) An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem. IEEE Access 7:20281–20292. https://doi.org/10.1109/ACCESS.2019.2897580

Díaz J, Petit J, Serna M (2002) A survey of graph layout problems. ACM Comput Surv 34:313–356. https://doi.org/10.1145/568522.568523

Dill KA, MacCallum JL (2012) The protein-folding problem, 50 years on. Science 338:1042–1046. https://doi.org/10.1126/science.1219021

Dong FM, Koh KM, Teo KL (2005) Chromatic polynomials and chromaticity of graphs. World Scientific, Singapore

Fidanova S, Pop P (2016) An improved hybrid ant-local search algorithm for the partition graph coloring problem. J Comput Appl Math 293:55–61. https://doi.org/10.1016/j.cam.2015.04.030

Gallian JA (2015) Graph labeling. Electron J Comb 1000:DS6

Garey MR, Johnson DS, Stockmeyer L (1974) Some simplified NP-complete problems. In: Proceedings of the sixth annual ACM symposium on theory of computing. ACM, New York, pp 47–63

Gerstein M, Tsai J, Levitt M (1995) The volume of atoms on the protein surface: calculated from simulation, using Voronoi polyhedra. J Mol Biol 249:955–966. https://doi.org/10.1006/jmbi.1995.0351

Hallórsson MM (1993) A still better performance guarantee for approximate graph coloring. Inf Process Lett 45:19–23. https://doi.org/10.1016/0020-0190(93)90246-6

Hansen J, Kubale M, Kuszner Ł, Nadolski A (2004) Distributed largest-first algorithm for graph coloring. In: Euro-Par 2004 parallel processing. Springer, Berlin, Heidelberg, pp 804–811

Harary F (1969) Graph theory. Addison-Wesley Publishing Company, Boston

Janczewski R (2001) T-coloring of graphs and its applications. Gdansk University of Technology, ETI Faculty, Gdansk

Jensen TR, Toft B (2011) Graph coloring problems. Wiley, New York

Jiang X, Farid H, Pistor E, Farid RS (2000) A new approach to the design of uniquely folded thermally stable proteins. Protein Sci Publ Protein Soc 9:403–416

Karp RM (1972) Reducibility among combinatorial problems. In: Miller RE, Thatcher JW, Bohlinger JD (eds) Complexity of computer computations. Springer, New York, pp 85–103

Kempe AB (1879) On the geographical problem of the four colours. Am J Math 2:193–200. https://doi.org/10.2307/2369235

Kosowski A, Manuszewski K (2004) Classical coloring of graphs. In: Graph colorings, pp 2–19

Lai H-J, Lin J, Montgomery B et al (2006) Conditional colorings of graphs. Discrete Math 306:1997–2004. https://doi.org/10.1016/j.disc.2006.03.052

Lee B, Richards FM (1971) The interpretation of protein structures: estimation of static accessibility. J Mol Biol 55:379–400

Lewis RMR (2016) A guide to graph colouring: algorithms and applications. Springer, Berlin

Lovasz L (2006) On the shannon capacity of a graph. IEEE Trans Inf Theor 25:1–7. https://doi.org/10.1109/TIT.1979.1055985

MacDougall JA, Miller M, Wallis WD (2002) Vertex-magic total labelings of graphs. Util Math 61:3–21

Marx D (2003) Graph colouring problems and their applications in scheduling

McDiarmid CJH, Sánchez-Arroyo A (1994) Total colouring regular bipartite graphs is NP-hard. Discrete Math 124:155–162. https://doi.org/10.1016/0012-365X(92)00058-Y

McIlvaine TC (1921) A buffer solution for colorimetric comparison. J Biol Chem 49:183–186

Munson M, Balasubramanian S, Fleming KG et al (1996) What makes a protein a protein? Hydrophobic core designs that specify stability and structural properties. Protein Sci Publ Protein Soc 5:1584–1593

Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci USA 103:8577–8582. https://doi.org/10.1073/pnas.0601602103

Peng Z, Oldfield CJ, Xue B et al (2014) A creature with a hundred waggly tails: intrinsically disordered proteins in the ribosome. Cell Mol Life Sci 71:1477–1504. https://doi.org/10.1007/s00018-013-1446-6

Pirun M, Babnigg G, Stevens FJ (2005) Template-based recognition of protein fold within the midnight and twilight zones of protein sequence similarity. J Mol Recognit JMR 18:203–212. https://doi.org/10.1002/jmr.728

RJLipton + KWRegan (2015) A big result on graph isomorphism. In: Gödels Lost Lett. PNP. https://rjlipton.wordpress.com/2015/11/04/a-big-result-on-graph-isomorphism/. Accessed 30 Nov 2016

Rost B (1999) Twilight zone of protein sequence alignments. Protein Eng 12:85–94

Roy S, Basu S, Dasgupta D et al (2015) The unfolding MD simulations of cyclophilin: analyzed by surface contact networks and their associated metrics. PLOS ONE 10:e0142173. https://doi.org/10.1371/journal.pone.0142173

San Segundo P (2012) A new DSATUR-based algorithm for exact vertex coloring. Comput Oper Res 39:1724–1733. https://doi.org/10.1016/j.cor.2011.10.008

Sanders DP, Zhao Y (2001) On improving the edge-face coloring theorem. Graphs Comb 17:329–341. https://doi.org/10.1007/pl00007248

Stiebitz M, Škrekovski R (2006) A map colour theorem for the union of graphs. J Comb Theory Ser B 96:20–37. https://doi.org/10.1016/j.jctb.2005.06.003

Street AG, Mayo SL (1999) Computational protein design. Structure 7:R105–R109. https://doi.org/10.1016/S0969-2126(99)80062-8

Tsai J, Taylor R, Chothia C, Gerstein M (1999) The packing density in proteins: standard radii and volumes1. J Mol Biol 290:253–266. https://doi.org/10.1006/jmbi.1999.2829

Wallis WD, Baskoro ET, Miller M, Slamin (2000) Edge-magic total labelings. Aust J Comb 22:177–190

Watts DJ, Strogatz SH (1998) Collective dynamics of "small-world" networks. Nature 393:440–442. https://doi.org/10.1038/30918

Welsh DJA, Powell MB (1967) An upper bound for the chromatic number of a graph and its application to timetabling problems. Comput J 10:85–86. https://doi.org/10.1093/comjnl/10.1.85

Zarrazola E, Gomez D, Montero J et al (2011) Network clustering by graph coloring: an application to astronomical images. In: 2011 11th international conference on intelligent systems design and applications (ISDA), pp 796–801

Zhang P (2015) Color-Induced graph colorings. Springer, Berlin

Zhao H, Li D, Deng W, Yang X (2017a) Research on vibration suppression method of alternating current motor based on fractional order control strategy. Proc Inst Mech Eng Part E J Process Mech Eng 231:786–799. https://doi.org/10.1177/0954408916637380

Zhao H, Sun M, Deng W, Yang X (2017b) A new feature extraction method based on EEMD and multi-scale fuzzy entropy for motor bearing. Entropy 19:14. https://doi.org/10.3390/e19010014

Zhao H, Yao R, Xu L et al (2018) Study on a novel fault damage degree identification method using high-order differential mathematical morphology gradient spectrum entropy. Entropy 20:682. https://doi.org/10.3390/e20090682

Zhu H, Sepulveda E, Hartmann MD et al (2016) Origin of a folded repeat protein from an intrinsically disordered ancestor. eLife. https://doi.org/10.7554/elife.16761

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.