Local iterative algorithms for approximate symmetry guided by network centralities

David Hartman[1 [0000−0003−3566−8214]], Jaroslav Hlinka[1 [0000−0003−1402−1470]], Anna Pidnebesna[1 [0000−0002−9391−8886]], and František Szczepanik[1]

Institute of Computer Science of the Czech Academy of Sciences,
Pod Vodárenskou věží 271/2, 182 07 Prague, Czech Republic
{hartman,hlinka,pidnebesna,szczepanik}@cs.cas.cz

Abstract. Recently, the influence of potentially present symmetries has begun to be studied in complex networks. A typical way of studying symmetries is via the automorphism group of the corresponding graph. Since complex networks are often subject to uncertainty and automorphisms are very sensitive to small changes, this characterization needs to be modified to an approximate version for successful application. This paper considers a recently introduced approximate symmetry of complex networks computed as an automorphism with acceptance of small edge preservation error, see Liu 2020 [18]. This problem is generally very hard with respect to the large space of candidate permutations, and hence the corresponding computation methods typically lead to the utilization of local algorithms such as the simulated annealing used in the original work. This paper proposes a new heuristic algorithm extending such iterative search algorithm method by using network centralities as a heuristic. Centralities are shown to be a good tool to navigate the local search towards more appropriate permutations and lead to better search results.

Keywords: Approximate symmetry · automorphism · graph matching · simulated annealing · complex network · centrality

1 Introduction

Complex networks are graphs that represent various natural and man-made systems such as, for example, the Internet [1], public transportation [16], or the human brain [27]. A typical way of analyzing such systems is to use several characteristics, both local (bound to vertices or edges) and global (bound to the whole graph). One notable global property is network symmetry, which received attention since MacArthur et al. [19] showed that many real-world networks contain high symmetry. Network symmetry has proven useful in revealing patterns of synchronized node clusters, helping understand forms of collective behavior [6].

All types of network characteristics can be burdened with uncertainty problems. This uncertainty can be due to the construction of the corresponding network, an example is the uncertainty of local characteristics caused by nonlinearity in time series, see [11,14,10]. The second type of uncertainty is the sensitivity of the corresponding characteristics to the overall behavior of the system,

exemplified by the sensitivity of the global characteristic called a small-world coefficient, see [13,15,26]. Both types of uncertainty are combined in the graph symmetry represented by the automorphic group mainly because of its sensitivity to small changes.

For this reason, a more robust approximate symmetry based on the distance to the originally introduced automorphic symmetry was recently introduced, see Liu 2020 [18]. The proposed numerical solution was based on simulated annealing omitting the fixed points of the corresponding permutation. In a recent improvement of this method, see [21], this approach has been extended by introducing fixed points and also by using a relaxed problem with continuous optimization.

In practice, simulated annealing struggles to find an optimal permutation in the factorially large solution space. To address these limitations, we introduce a new heuristic method. This heuristic is based on the observation that permutations close to automorphisms tend to display similar vertices to themselves. We measure vertex similarity using graph centralities such as eigenvector centrality, PageRank, or betweenness. We compare the performance of the original and heuristic algorithm on several random network models, building on the experiments carried out by Pidnebesna et al. [21]. Presented results are based on the results of the bachelor thesis of F. Szczepanik, where one can find further details about methods used and some additional observations [25].

2 Preliminaries

2.1 Basic notation and definitions.

The object of interest is a graph (undirected, unweighted, loop-less), defined by a set of vertices (nodes) V and edges (links) E, denoted as $G = (V, E)$. We use the standard notion of an adjacency matrix denoted as A, where $a_{ij} = 1$ if $\{v_i, v_j\} \in E$ and $a_{ij} = 0$ otherwise. We denote the set of vertices adjacent to a given vertex as $N_i = \{v_j | \{v_i, v_j\} \in E\}$. We call its size as $k_i = |N_i|$ and call it the degree of a node i. For a set of nodes $U \subseteq V$ we denote $E(U)$ as the set of edges induced by U, i.e. set of edges where each edge has at least one node in U. The Cartesian product $G \times H$ of two graphs G and H is a graph such that 1) $V(G \times H) = V(G) \times V(H)$, and 2) $\{(u, u'), (v, v')\} \in E(G \times H)$ if $u = v$ and $\{u', v'\} \in E(H)$ or $u' = v'$ and $\{u, v\} \in E(G)$.

An automorphism of a graph $G = (V, E)$ is a permutation π of V preserving adjacencies, i.e.: $\{\pi(v_x), \pi(v_y)\} \in E \iff \{v_x, v_y\} \in E$. We represent such permutations via matrices: the permutation matrix P for a permutation π on n elements is an $n \times n$ matrix where:

$$P_{ij} = \begin{cases} 1 & \text{if } i = \pi(j), \\ 0 & \text{otherwise.} \end{cases}$$

PA permutes A's rows and AP^T permutes its columns according to π. Considering the orthogonality of permutation matrices, we obtain the following [3]: *the permutation π is an automorphism of G with an adjacency matrix A if*

and only if $A = PAP^{\mathrm{T}}$. Lastly, the L1 norm of an $n \times m$ real matrix A is: $\|A\|_1 = \sum_i^m \sum_j^n |a_{ij}|$.

2.2 Approximate Symmetry

The approximate symmetry of a graph $G = (V, E)$ with an adjacency matrix A is defined as [18]:

$$E_D(A) = \frac{1}{4} \min_P (\left\|A - PAP^{\mathrm{T}}\right\|_1).$$

$E_D(A)$ minimizes the number of mismatches $A - PAP^{\mathrm{T}}$ over permutation matrices P. Due to the constant $\frac{1}{4}$, it directly expresses the number of mismatched edges that differ after the permutation: $A_{ij} \neq A_{\pi(i)\pi(j)}$. So, if $A = PAP^{\mathrm{T}}$, then P corresponds to an automorphism, and $E_D(A)$ is equal to zero.

A parametrized symmetry version $\epsilon(A, P)$ is then defined as follows:

$$\epsilon(A, P) = \frac{1}{4} \left\|A - PAP^{\mathrm{T}}\right\|_1,$$

yielding $E_D(A) = \min_P \epsilon(A, P)$. For comparability across graphs, Liu introduced *normalized symmetry* $S(A)$, which normalizes $E(A)$ by the maximal $\epsilon(A, P)$. The maximal value of $\epsilon(A, P)$ is reached when the graph has exactly half of all possible edges, that is $\frac{1}{2}\binom{n}{2}$, and all of them are mismatched by P. The normalized approximate symmetry of a graph $G = (V, E)$ with an adjacency matrix A is thus:

$$S'(A) = \frac{E_D(A)}{\frac{1}{2}\binom{n}{2}} = \frac{\min_P \left(\left\|A - PAP^{\mathrm{T}}\right\|_1\right)}{n(n-1)}.$$

Let π be a permutation on the set V. The element $v \in V$ is a fixed point of π if $v = \pi(v)$. Liu originally considered only permutations with no fixed points (FPs), which reduced the problem space at the risk of excluding some potentially promising solutions. This motivated Pidnebesna et al. [21] to experiment with an annealing variant that allows up to K FPs, where K is a parameter. They set $K = n/2$ in their experiments, i.e., half of all the nodes.

An alternative approach avoids specifying an upper limit on FPs and instead determines the number of FPs dynamically by penalizing them in the objective function, resulting in the following version of approximate symmetry:

$$S(A) = \frac{E_D(A)}{\frac{1}{2}\binom{n}{2}} = \frac{\min_P \left(\left\|A - PAP^{\mathrm{T}}\right\|_1\right)}{n(n-1) - F(P)(F(P)-1)},$$

where $F(P)$ denotes the number of fixed points in the permutation P. In the aforementioned bachelor's thesis by Szczepanik [25], it is shown that both approaches to FPs in simulated annealing - fixing $K = n/2$ or penalizing them in the objective function - yield values of S' (the normalized symmetry as originally defined by Liu) with no statistically significant differences. Based on this finding, all subsequent experiments use the dynamic FPs penalization method.

Identifying the optimal approximate automorphism P minimizing $\epsilon(A, P)$ is computationally hard as it involves searching through a factorially large solution space of vertex permutations. One of the suitable metaheuristics for this setting is Simulated Annealing, a probabilistic optimization technique [23]. Annealing explores the solution space by transitioning from one state to another while decreasing *energy*, which represents the value of the objective function. The process is initialized at a random state and initially explores worse states to avoid getting stuck in local optima according to the current *temperature*. The temperature is gradually reduced according to a chosen *cooling schedule* until some threshold temperature is reached or all move iterations are carried out.

2.3 Graph Centralities

Graph centralities are characteristics that evaluate the importance of a vertex from a structural perspective. More details about centralities can be found e.g. in [20]. One of the simplest measures is *degree centrality* defined as the number of adjacent edges. This centrality captures mainly local effects around the vertices. The generalization of the node degree is *eigenvector centrality*, which considers the importance of the vertex neighbors in addition to the number of its neighbors. Moreover, we apply this consideration to more distant vertices. This centrality can be defined recursively. If we have defined the initial centrality as $e_i(0)$ defined for a node i, then the total eigenvector centrality for this node can be defined as

$$e_i(t) = \sum_{j=1}^{n} A_{ij} e_j(t-1).$$

The exact value of the centrality is computed for $t \to \infty$. It is possible to show further that this equation can be rewritten into the following characteristic equation $e_i = \frac{1}{\lambda} \sum_j A_{ij} k_j$, where λ is the greatest eigenvalue of A. Thus eigenvector centrality can be expressed as the eigenvector corresponding to the largest eigenvalue of A [20]. The last generalization in this direction is *PageRank*, which originated as a system for ranking the importance of web pages [5]. The main generalization of this centrality over the eigenvector centrality is the normalization of steps with degrees

$$e_i(t) = \alpha \sum_{j=1}^{n} A_{ij} \frac{e_j(t-1)}{k_j},$$

where k_j denotes the degree of vertex j and α is a damping factor typically set to 0.85, see full conditions in [20].

Instead of studying the number of neighbors, one can also study the density of edges in the neighborhood. This is measured by the *clustering coefficient* c_i of a node i defined as

$$c_i = \frac{2|E(N_i)|}{|N_i||N_i - 1|}.$$

This centrality measures how locally clustered the neighborhood of a vertex is, i.e. how much the vertex is part of the denser graphs [2].

The last centrality is representative of a class of characteristics measuring the role of a vertex in communication through the network. In more detail, *betweenness centrality* assesses a node's importance by counting the number of shortest paths between all pairs of nodes passing through it [22]. The betweenness centrality of a node v_i is defined as:

$$b_i = \sum_{\substack{x,y \in V(G) \\ x \neq y \neq v_i}} \frac{\sigma_{xy}(v_i)}{\sigma_{xy}},$$

where σ_{xy} denotes the number of shortest paths between nodes x and y and $\sigma_{xy}(v_i)$ is the number of shortest paths between x and y passing through v_i [22].

2.4 Random network models

Random network models provide abstractions of real-world systems and aim to capture their structural properties. In this section, we introduce several random graphs and random network models that serve as the dataset for comparing and evaluating the original and heuristic versions of simulated annealing. For such testing, we follow the methodologies of Straka [24] and Pidnebesna et al. [21].

The first class of graphs is not random but suitable for symmetry testing due to its high regularity. A *grid graph* $R_{n_1,n_2,\ldots,n_d}$ in d dimensions and lengths n_i in dimension i is defined as a graph product of path graphs $P_{n_1} \times P_{n_2} \times \ldots \times P_{n_d}$ [21].

The basic random graph is the standard *Erdős–Rényi* (ER) model $G(n,p)$ [7]. $G(n,p)$ is generated on n vertices by adding an edge between every pair with probability p. The ER graph is simple to analyze but it represents the real networks poorly, e.g. lacks the small-world as well as scale-free character [26,20].

A more realistic model is the *Barabási-Albert* model (BA) that is s network growing model [2]. The generation procedure for the selected parameters n and k is as follows. We start with a small ($m \ll n$) graph and then add a vertex at each step, which we connect to k existing vertices, preferring higher degree vertices, namely the probability of connecting to an existing node i is $p_i = k_i / \sum_j k_j$. This model captures some characteristics such as power-law degree distribution (scale-free) and the emergence of important high-degree nodes (hubs) [2].

The *duplication-divergence* (DD) model simulates the evolution of protein-protein interaction (PPI) networks. PPI networks evolve by the mechanism of gene duplication, which generates new proteins that are initially identical to pre-existing ones but gradually diverge, with most of the new proteins not surviving. The duplication–divergence is described by the following growth mechanism [17]:

- Duplication: A node is randomly selected and duplicated. The newly created node inherits all links of the duplicated node.
- Divergence: Each link of the new node is retained with *divergence probability* σ. If no links survive, the duplication fails, and the node is discarded.

This model represents a real network but has some form of symmetry directly embedded in the generative process.

3 Heuristic method

Various combinatorial-heuristic methods including simulated annealing, genetic algorithms, artificial neural networks, and others can be used to search the space of permutations to minimize $E_D(A)$ or $S(A)$. In this study, however, we concentrate on improving previously proposed and tested methods (see [18,21]). The reason for this is a basic investigation of the effect of centralities on search efficiency for well-constructed approaches. It can be assumed that the results will be translatable to more robust methods.

In particular, our primary objective is to investigate how guiding annealing with graph centralities improves the detection of approximate network symmetry. Thus, in this study, we introduce a modified annealing algorithm, *guided annealing*, which uses centrality metrics as a heuristic for selecting transpositions when moving across states.

Roughly speaking, a basic approach to exploring the solution space of approximate automorphisms involves applying random transpositions to the current permutation. Enhancing the transition strategy by incorporating some form of guidance could improve the efficiency of the annealing process. An optimal approximate automorphism should align vertices with similar structural roles in the graph. For instance, mapping a hub node onto a peripheral node will likely result in a large number of mismatched edges and therefore suboptimal approximate symmetry. Centralities allow us to quantify vertex "similarity" and thus identify vertices that should be aligned. An important property of graph centralities is that *automorphisms preserve centralities* [22,9]. Therefore, we expect nodes with similar centrality values to be aligned in an optimal permutation.

3.1 Move function in guided annealing

We reimplement the *move* function, which generates a new permutation π' from the current permutation π and thus determines the next state. In unmodified simulated annealing, the transposition is constructed by randomly swapping the images of two vertices under the current permutation (moving from $a, \pi(a)$ and $b, \pi(b)$ to $a, \pi(b)$ and $b, \pi(a)$, where a and b are vertices and π the current permutation). In our guided version, the transpositions aligning similar vertices are chosen with higher probability.

Similarity matrix and implementation description We express the similarity of vertex pairs for a given centrality Γ in an $n \times n$ matrix M, indexed by vertices. To construct M, we start by computing a *difference matrix* D, where each element is defined as $d_{ij} = |\Gamma(i) - \Gamma(j)|$, i.e. the absolute difference of the centrality of v_i and v_j. To transform this into a similarity measure where higher values imply higher similarity, we compute inverses over the elements of D. Specifically, the entries of M are defined as $m_{ij} = (d_{ij} + \beta)^{-1}$, where $\beta > 0$ is a division constant preventing the denominator from being zero and moderating the variance of the matrix values; as β increases, the entries of M become more uniform, guiding the annealing less aggressively.

Re-implementing the move function We now describe the re-implemented move function. We start by randomly choosing the first vertex a [1], where $\pi(a)$ is its image under the current permutation and $m_{a\pi(a)}$ is their similarity. We evaluate a potential similarity increase by considering swaps with each vertex b and its image $\pi(b)$. The difference in the similarity of the original and proposed setting can be calculated as $\Delta M_b = m_{a\pi(b)} + m_{b\pi(a)} - m_{a\pi(a)} - m_{b\pi(b)}$.

Before normalizing ΔM into a probability distribution over vertices, we again introduce a smoothing parameter $\phi > 0$, which we call the probability constant, and set $\Delta E_b = \max(\Delta M_b, \phi)$ for each b. Similarly to the division constant β, higher values of ϕ imply all vertices (even dissimilar pairs) have higher chances of being chosen. We then randomly draw b according to the created probability distribution. We present the pseudocode of the re-implemented move function, where we assume π is the current permutation represented by the matrix P:

Algorithm 1 Move function in guided annealing.

1: $a \leftarrow$ random vertex
2: **for** every vertex b **do**
3: $\quad \Delta E_b = max(m_{a,\pi(b)} + m_{b,\pi(a)} - m_{a,\pi(a)} - m_{b,\pi(b)}, \phi)$
4: **end for**
5: $\Pi \leftarrow$ probability distribution obtained by normalizing ΔE
6: $b \leftarrow$ vertex randomly drawn from Π
7: $P' \leftarrow$ permutation generated from P by transposing images of a, b
8: return P'

Time complexity of the new version The similarity matrix is computed only once at the beginning of a run for each graph, and most of the complexity in this process lies in computing the centralities themselves.

Optimization of parameters by grid search The values of the two introduced parameters, β and ϕ, i.e., the division and probability constants, were determined via grid search on about instances 400 of various random network models with varying sizes (50 to 150 vertices) and parameters. ER, BA, and Stochastic Block Models were included in the dataset used in grid search (we later excluded Stochastic Block Model graphs from further analysis due to their structural similarity to ER graphs). The examined values of β and ϕ ranged from 0.001 to 10. For every combination of the mentioned parameters, 30 simulations were conducted. We then selected the parameters with the best average results. When working with PageRank, we also conducted a grid search to optimize the

[1] We also experimented with a "one-step" approach, which does not select the first vertex randomly but instead considers all possible transpositions. However, this requires $O(n^2)$ evaluations in each step or a one-time pre-computation for all pairs of pairs, which has a complexity of $O(n^4)$, both practically unusable for large graphs.

damping factor α, but without significant differences in results; therefore, we keep using the default value of $\alpha = 0.85$. Generally, the few top combinations of parameters performed similarly, whereas the other combinations performed significantly worse. We do not claim the global optimality of the parameters we selected and recognize that different datasets could produce different parameters. However, the parameters we selected were sufficient to answer our hypothesis about guided annealing's improvement in symmetry calculation.

3.2 Evaluation methodology

We now describe statistical methods used to evaluate the performance of different annealing versions. Violin plots are utilized to visually represent the distribution of measured symmetry (i.e. values of S, where lower values imply higher symmetry) across different instances of the models. Paired t-tests are used to assess whether the mean values of symmetry values obtained from two different annealing versions differ. The null hypothesis assumes that there is no significant difference, and the standard significance level of 0.05 is used to reject the null hypothesis. Finally, Cohen's d quantifies the effect size.

4 Results

Grid We evaluate improved annealing on grid graphs with $50, 100, 150$ vertices in 2 and 3 dimensions. In 2D, the lengths of sides are always set to $5 \times x$ and in 3D to $2 \times 5 \times x$. For all grid parameters, guided versions significantly improved computed symmetry, with betweenness yielding the highest improvements, followed by eigenvector centrality (see Fig. 1, the first subplot).

Erdős–Rényi model For ER graphs with $n = 20, 50, 100$ and edge densities $p = 0.1, 0.3, 0.5$, we observe that there were no statistically significant differences between results produced by the different annealing versions, as presented in the second subplot of Figure 1. A possible explanation would be that the ER model is entirely random; it lacks any internal structure stemming from a generation process and has no predictable substructures that can be reflected in each other.

Barabási–Albert model We conduct evaluations on instances with 50, 100, and 150 nodes and k values of 3, 5, and 7. The results for BA graphs on 150 nodes and $k = 5$ are presented in the third subplot of Figure 1. While not significant for 50 nodes, as graph size increases to $n = 100$ and 150, the improvements grow to become more significant. Versions of annealing guided by eigenvector centrality, betweenness, and PageRank yield some level of improvement over original annealing.

We select eigenvector centrality, which by eye seems to perform similarly or better than other centralities, and evaluate its improvements in more detail. We

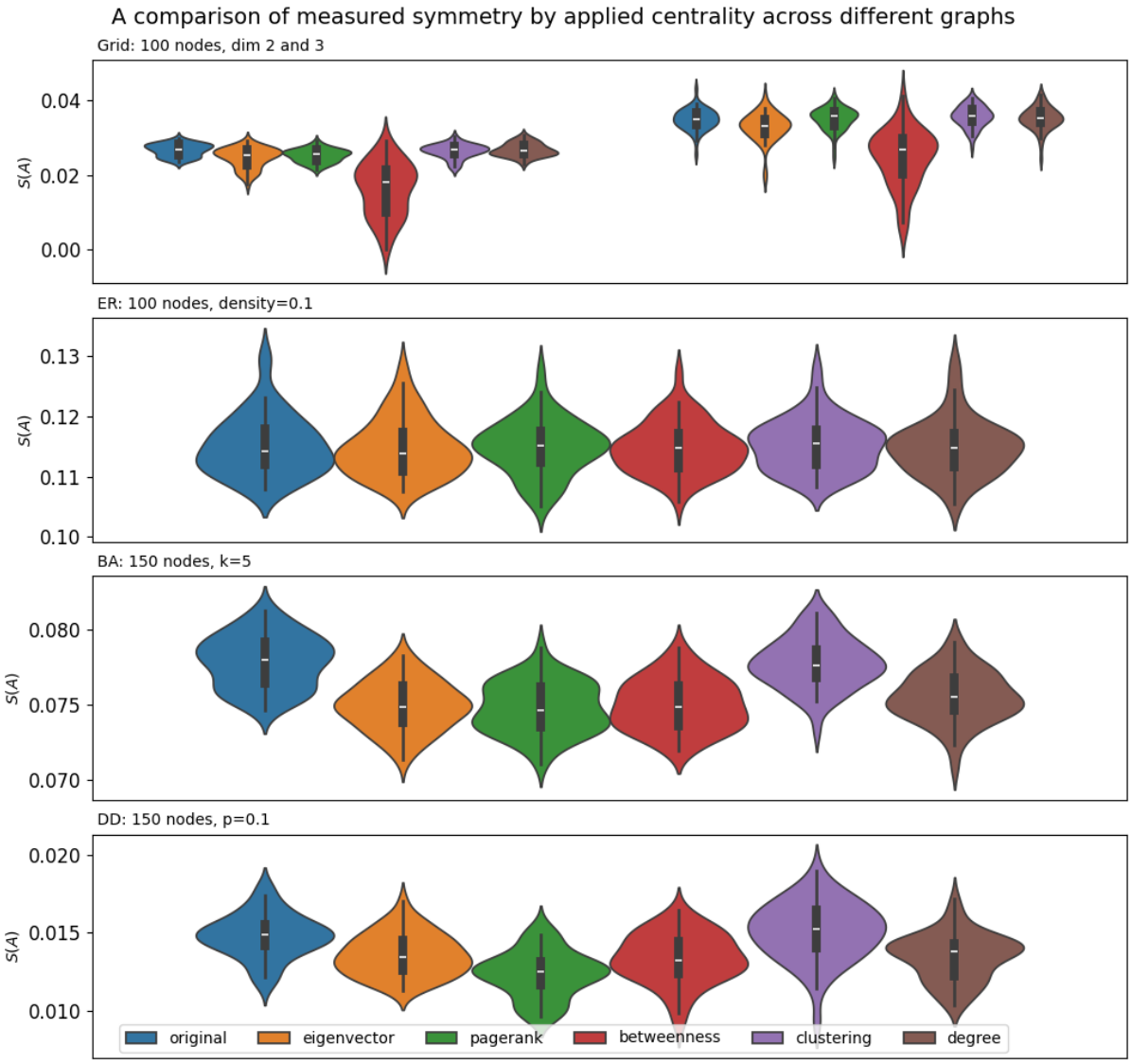


Fig. 1: Comparison of the performance of simulated annealing guided by different centralities across various random networks. The results are visualized in violin plots, where each color represents a version of annealing guided by a different centrality. For each configuration of parameters, we conduct 50 simulations. The first subplot presents results for grid graphs with 100 nodes in 2D and 3D. The second subplot presents results for Erdős–Rényi graphs with 100 nodes and edge density of 0.1. The third subplot presents results for Barabási–Albert graphs with 150 nodes and $k = 5$. The final subplot presents results for the Duplication–Divergence model with 150 nodes and a divergence probability 0.1.

compute paired t-tests to determine whether the two algorithms (eigenvector-centrality-guided and original annealing) compute symmetry values with the same mean, along with Cohen's d to measure the effect size (see Figure 2). After applying the Bonferroni correction for multiple hypothesis testing, annealing guided by eigenvector centrality produces significantly better outcomes compared to original annealing, apart from two combinations of parameters. We also see that the larger and sparser the graph is, the greater the improvement, as suggested by the increasing absolute value of Cohen's d.

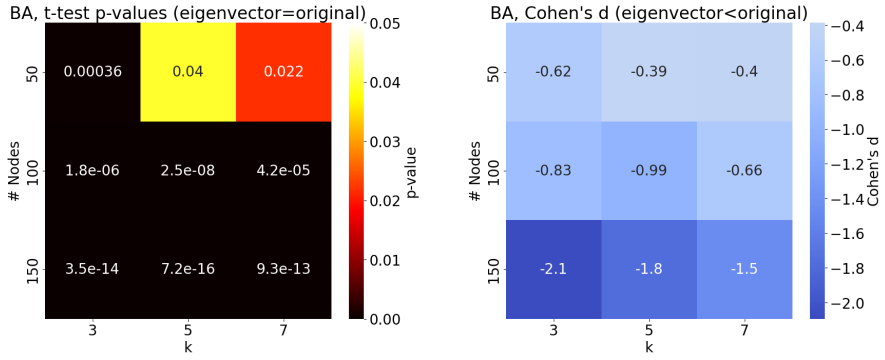


Fig. 2: Comparison of the performance of original simulated annealing and simulated annealing guided by eigenvector centrality, BA model, where k is the number of links of a new node.

Duplication–divergence model Finally, we conduct evaluations on the DD model. The dataset consists of DD graphs with node counts $n = 50, 100, 150$, and divergence probabilities 0.1 and 0.3. The results for DD graphs on 150 nodes and divergence probability 0.1 are presented in the fourth subplot of Figure 1.

Our findings indicate that guided annealing significantly outperforms original annealing. It also holds that instances with divergence probability 0.3 are less symmetric than those with divergence probability 0.1, which we attribute to the fact that as σ decreases, the graph becomes sparser and starts to resemble trees, which are intuitively somewhat regular (at least in the sense of having many leaf nodes likely sharing similar properties).

For a more rigorous evaluation, we select PageRank, which seems to perform at least as well as other centralities. Calculating paired t-tests and Cohen's d reveals that for smaller graph sizes, we cannot reject the null hypothesis. If we move to larger graph sizes, however, we see PageRank-guided annealing outperforms the original, unguided version, as evident in Figure 3, which also holds after applying the Bonferroni correction on the multiple (8 in this case) hypoth-

esis testing. The improvements are more significant in sparser DD instances with lower σ and also become more pronounced with growing graph size.

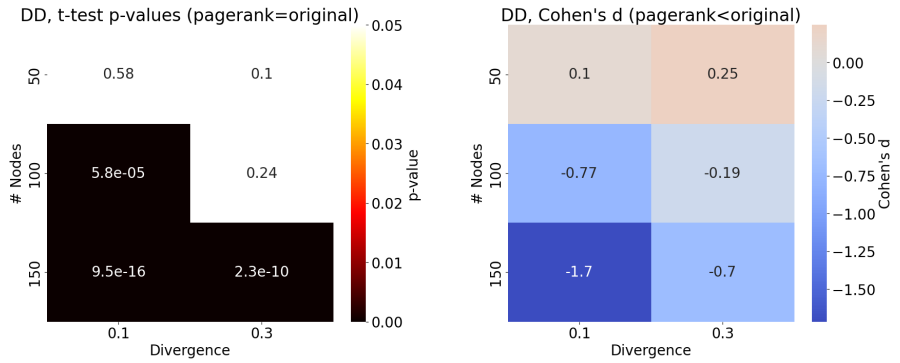


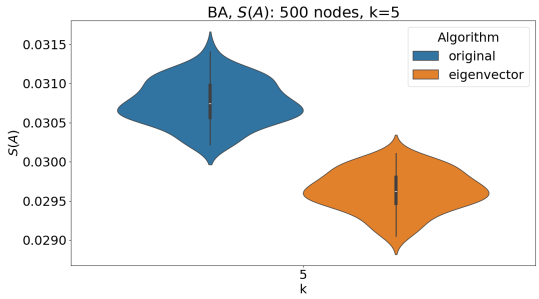
Fig. 3: Comparison of the performance of original simulated annealing and simulated annealing guided by eigenvector centrality, DD model.

Larger graphs In the previous section, we observed a trend in both the BA and DD models; improvements achieved by guided annealing became more pronounced with growing graph size. To further explore this trend, we extend our analysis to graphs of sizes 300 and 500. Specifically, we focus on the performance of PageRank-guided annealing on larger DD graphs, and eigenvector-centrality-guided annealing on larger BA graphs. Figure 4 shows the distribution of measured symmetry for $n = 500$ and Cohen's d effect size for other combinations of the parameters. In both cases, the guided versions yield even more significant improvements over original annealing than in smaller graph instances.

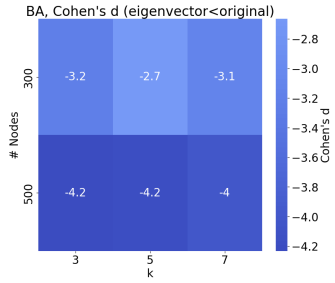
5 Discussions

This study investigated the impact of guiding simulated annealing with graph centralities as heuristics to improve the computation of approximate network symmetry. Given that graph centralities are preserved by automorphisms, we hypothesized that aligning vertices with similar centrality values would lead to improved values of approximate symmetry.

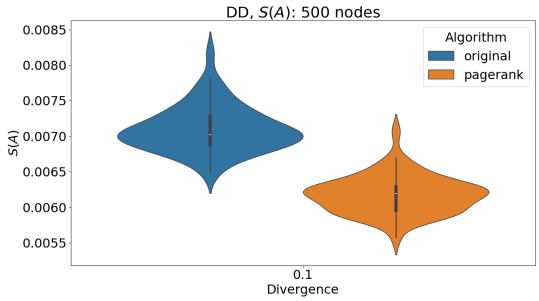
We introduced a new heuristic algorithm for finding approximate symmetry extending the existing simulated annealing approach. As a heuristic, we used network centralities, namely degree centrality, eigenvector centrality, betweenness centrality, and clustering coefficient. We also defined an efficient way of selecting similar vertices to be transposed. The effectiveness of this new method was evaluated across several graph classes or random models, including the grid,



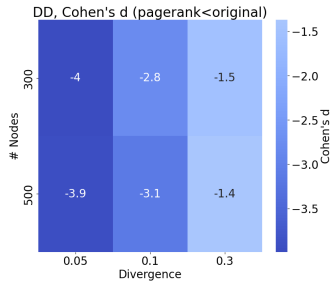
(a) BA graphs: Measured symmetry.



(b) BA graphs: Cohen's d.



(c) DD graphs: Measured symmetry.



(d) DD graphs: Cohen's d.

Fig. 4: Comparison of measured symmetry and Cohen's d values for larger BA and DD graphs.

the Erdős–Rényi random graph, the Barabási–Albert model, and the duplication–divergence model.

Our results demonstrated that centrality-guided annealing improved approximate symmetry calculation concerning the model used. In grid graphs, betweenness centrality significantly improved computed approximate symmetry. This can be explained by the fact that this centrality considers communication using the shortest paths and in a grid graph this centrality characterizes vertices relatively uniquely. This graph, however, is not a proper test instance of the principle.

In contrast, no significant improvements were observed for Erdős–Rényi graphs, which we attribute to some of their known properties, e.g. small clustering of ER model implies no improvement using clustering coefficient, whereas degree or even PageRank are potentially confused by many similar nodes. Similarly, given the short distances and generally good connectivity of the ER model, neither betweenness centrality is a suitable discriminating characteristic. This may seem like a disappointing result, however, given the small reproducibility of real networks using the ER model, this result is not crucial.

In the Barabási–Albert and duplication-divergence networks, eigenvector centrality, and PageRank led to measurable improvements, particularly in instances

that were inherently more symmetric. With graph size increasing from 100-150 vertices to 300-500 vertices, the improvements in Barabási–Albert and duplication–divergence networks became even more pronounced. These results correspond to intuition in the BA model due to the specific connection of vertices to each other and the creation of a certain hierarchical structure characteristic also typical for real networks [2]. For the duplication-divergence model, the results obtained are more surprising but can be attributed to some structure in the repeated generation of the duplicated vertex. Considering the model parameters, the goal is to test possible further variations in the future. It should also be mentioned that for both networks betweenness centrality works almost similarly to the above-mentioned spectral characteristics. This is probably due to the better resolution of this centrality in terms of its more comprehensive use of more distant areas from the corresponding node. On the other hand, the clustering coefficient performed poorly, likely due to the high locality of this characteristic and the fact that the networks were highly clustered.

There are several ways to extend the results mentioned here. One weakness seems to be the use of simulated annealing itself. Therefore, it would be beneficial to explore other metaheuristic approaches in the context of approximate symmetry computation, such as evolutionary algorithms, and examine whether centrality-based guidance can improve their performance.

In the BA model, for example, simple degree centrality performs comparably to the more complex eigenvector centrality. Since this centrality is a generalization of the vertex degree, it would be interesting to distinguish for each model the degree of locality needed for the centrality used. Similarly, betweenness centrality could be considered in its more local versions, such as k-betweenness [4].

Instances of graphs that have uniform centralities across vertices, such as regular graphs having identical vertex degrees, can cause a certain problem for the whole algorithm. These graphs can often be quite specific. However, for possible use in algorithms, it would be beneficial to identify a well-described class of such uniform graphs, especially for the characteristics yielding good results as a heuristic for approximate symmetry computation. In the case of graphs with uniform betweenness centralities, this is ongoing research [12,8]. The contribution of such a potential classification can be seen in the fact that such betweenness uniform graphs need not to be regular.

6 Conclusion

We introduced centrality-based heuristics into the computation of approximate graph symmetries, enhancing the guidance of local search algorithms such as simulated annealing. By leveraging structural cues from centralities, the method more effectively navigates the vast space of vertex permutations, especially in complex or larger networks. This approach lays the groundwork for making approximate symmetry computation more practical, and future exploration with more advanced algorithms—such as evolutionary methods—could further expand its applicability.

Acknowledgments. This work was supported by the Czech Science Foundation Grant No. 23-07074S.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abani, N., Braun, T., Gerla, M.: Betweenness centrality and cache privacy in information-centric networks. ACM SIGCOMM, 106–116 (2018). https://doi.org/10.1145/3267955.3267964
2. Barabási, A.-L.: Network Science. The Royal Society Publishing. http://networksciencebook.com/
3. Biggs, N.: Algebraic Graph Theory. Cambridge Mathematical Library, Cambridge University Press (1974).
4. Borgatti S.P., Everett M.G.. A graph-theoretic perspective on centrality. Soc Netw **28**(4):466–484 (2006), https://doi.org/10.1016/j.socnet.2005.11.005
5. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems **30**(1-7), 107–117 (1998). https://doi.org/https://doi.org/10.1016/S0169-7552(98)00110-X
6. Cho, Y., Nishikawa, T., Motter, A.: Stable Chimeras and Independently Synchronizable Clusters. Phys. Rev. Lett. **119**, 084101 (2017). https://doi.org/10.1103/PhysRevLett.119.084101
7. Erdös, P., Rényi, A.: On Random Graphs I. Publicationes Mathematicae Debrecen **6**, 290–297 (1959). https://static.renyi.hu/~p_erdos/1959-11.pdf
8. Ghanbari, B., Hartman, D., Jelínek, V., Pokorná, A., Šámal, R., Valtr, P. Structure of betweenness uniform graphs with low values of betweenness centrality. arXiv preprint arXiv:2401.00347, (2023).
9. Ghorbani, M., Dehmer, M., Lotfi, A., Amraei, N., Mowshowitz, A., Emmert-Streib, F.: On the relationship between PageRank and automorphisms of a graph. Information Sciences **582**, 874–885 (2021). https://doi.org/10.1016/j.ins.2021.08.013
10. Hartman, D., Hlinka, J., Paluš, M., Mantini, D., and Corbetta, M. The role of nonlinearity in computing graph-theoretical properties of resting-state functional magnetic resonance imaging brain networks. Chaos: An Interdisciplinary Journal of Nonlinear Science, **21**(1), 013119 (2011) https://doi.org/10.1103/PhysRevLett.119.084101
11. Hartman, D. and Hlinka, J. Nonlinearity in stock networks. Chaos: An Interdisciplinary Journal of Nonlinear Science, **28**(8), 083127 (2018), https://doi.org/10.1063/1.5023309
12. Hartman, D., Pokorná, A., Valtr, P. On the connectivity and the diameter of betweenness-uniform graphs. Discrete Applied Mathematics, 342, 27-37 (2024), https://doi.org/10.1016/j.dam.2023.08.017.
13. Hlinka, J., Hartman, D., and Paluš, M. Small-world topology of functional connectivity in randomly connected dynamical systems. Chaos: An Interdisciplinary Journal of Nonlinear Science, **22**(3), 033107 (2012), https://doi.org/10.1063/1.4732541

14. Hlinka, J., Hartman, D., Vejmelka, M., Dagmar Novotná and Milan Paluš Non-linear dependence and teleconnections in climate data: sources, relevance, nonstationarity. Clim. Dyn. **42**, 1873–1886 (2014). https://doi.org/10.1007/s00382-013-1780-2
15. Hlinka, J., Hartman, D., Jajcay, N., Tomeček, D., Tintěra, J., and Paluš, M. Small-world bias of correlation networks: From brain to climate. Chaos: An Interdisciplinary Journal of Nonlinear Science, **27**(3), 035812 (2017), https://doi.org/10.1063/1.4977951
16. Hong, J., Tamakloe, R., Lee, S., Park, D.: Exploring the Topological Characteristics of Complex Public Transportation Networks: Focus on Variations in Both Single and Integrated Systems in the Seoul Metropolitan Area. Sustainability **11**(19), 5404 (2019). https://doi.org/10.3390/su11195404
17. Ispolatov, Y., Krapivsky, P., Yuryev, A.: Duplication-divergence model of protein interaction network. Physical Review E **71**(6), 061911 (2005). https://doi.org/10.1103/physreve.71.061911
18. Liu, Y.: Approximate Network Symmetry. arXiv preprint arXiv:2012.05129 (2020), https://arxiv.org/abs/2012.05129
19. MacArthur, B., Sanchez-Garcia, R., Anderson, J.: Symmetry in complex networks. Discrete Applied Mathematics, **156**(18), 3525-3531 (2008), https://doi.org/10.1016/j.dam.2008.04.008
20. Newman, M.: Networks: An Introduction. Oxford University Press (2010). https://doi.org/10.1093/acprof:oso/9780199206650.001.0001
21. Pidnebesna, A., Hartman, D., Pokorná, A., Straka, M., Hlinka, J.: Computing Approximate Global Symmetry of Complex Networks with Application to Brain Lateral Symmetry. Inf Syst Front (2025) https://doi.org/10.1007/s10796-025-10585-3.
22. Pokorná, A.: Characteristics of network centralities. Prague: Charles University. Faculty of Mathematics and Physics, Computer Science Institute (2020). https://dspace.cuni.cz/handle/20.500.11956/118612
23. Kirkpatrick, S., Gelatt, D.C., Vecchi, M.P.: Optimization by Simulated Annealing. Science **220**(4598), 671–680 (1983). https://doi.org/10.1126/science.220.4598.671
24. Straka, M.: Approximative symmetries of complex networks. Bachelor thesis, Prague: Charles University. Faculty of Mathematics and Physics, Computer Science Institute (2022). https://dspace.cuni.cz/handle/20.500.11956/174639
25. Szczepanik, F.: Centralities in computation of approximate symmetries. Prague: Charles University. Faculty of Mathematics and Physics, Computer Science Institute (2024). http://hdl.handle.net/20.500.11956/192809
26. Watts, D., Strogatz, S.: Collective dynamics of 'small-world' networks. Nature **393**, 440–442 (1998). https://doi.org/10.1038/30918
27. Zhao, X., Liu, Y., Wang, X., Liu, B., Xi, Q., Guo, Q., Jiang, H., Jiang, T., Wang, P.: Disrupted small-world brain networks in moderate Alzheimer's disease: a resting-state FMRI study. PLOS One **7**(3), e33540 (2012). https://doi.org/10.1371/journal.pone.0033540