ISSN 2590-9770
The Art of Discrete and Applied Mathematics 8 (2025) #P1.08
https://doi.org/10.26493/2590-9770.1789.4e7
(Also available at http://adam-journal.eu)

Advanced clique algorithms for protein product graphs*

Janez Konc† , Dušanka Janežič‡

University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies, Glagoljaška ulica 8, SI-6000 Koper, Slovenia

Received 1 April 2024, accepted 13 January 2025, published online 25 February 2025

Abstract

In this paper, we give a comprehensive overview of the development of clique algorithms and their use for drug design based on the search for cliques in protein product graphs. The maximum clique problem is a computational problem of finding largest subsets of vertices in a graph that are all pairwise adjacent. A related problem is the maximum weight clique problem and the highest weight k-clique problem, which both extend the algorithm to weighted graphs. The review covers our developed algorithms, starting with our improved branch-and-bound algorithm for finding maximum cliques in undirected graphs from 2007 up to the recent developments of algorithms for weighted graphs in 2024. We show the application of these algorithms to early stages of drug discovery, in particular to protein binding site detection based on protein similarity search in large protein databases and to protein-ligand molecular docking.

Keywords: Cliques, protein product graphs, applications.

Math. Subj. Class.: 05C69

1 Introduction

The maximum clique problem (MCP) is a computational challenge in which the goal is to find the largest subset of vertices within a graph in which every vertex is directly connected to every other vertex in the subset. MCP is classified as NP-hard due to its inherent difficulty in finding an optimal solution [3].

Another version of this problem, the maximum weight clique problem (MWCP), operates with weighted graphs. In such graphs, each vertex is assigned a numerical weight,

*Dedicated to Professor Dragan Marušič on the occasion of the 70th birthday.

†Equally Contributed.
‡Corresponding author. Equally Contributed.
E-mail addresses: konc@cmm.ki.si (Janez Konc), dusanka.janezic@upr.si (Dušanka Janežič)

usually indicating its importance, utility or cost. A maximum weight clique is a clique in the weighted graph with the highest sum of the weights of its vertices.

The clique problem is useful for a wide range of applications. Its versatility goes beyond graph theory, as it serves as a model for a variety of challenges in different disciplines. The clique algorithms have been used in many research and industry settings [12].

In this review, we first present the clique algorithms for the MCP and MWCP we have developed between years 2007 and 2024 (Figure 1). We then present their use in drug design through the use of specially constructed protein product graphs. We focus on two areas, namely protein binding sites detection and protein-ligand molecular docking. The developed tools, collectively known as Protein Binding Site (ProBiS) tools, enable highly efficient drug design [7].

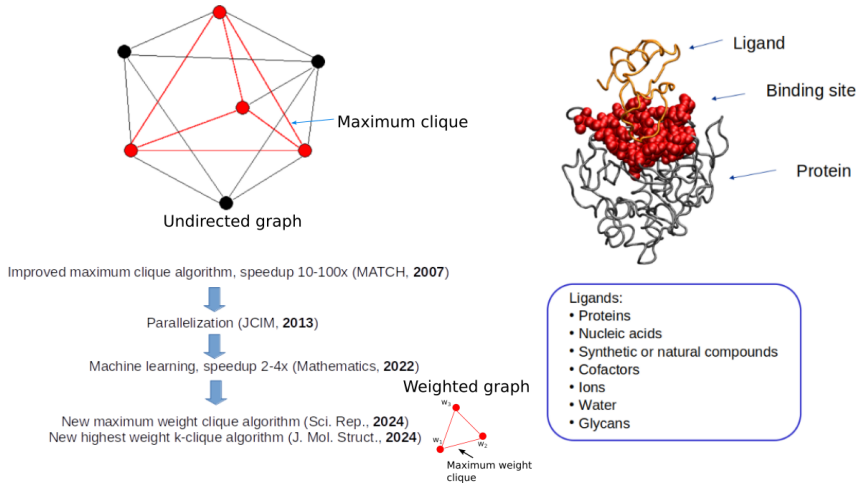

Figure 1: Development of clique algorithms for protein binding sites and ligands detection.

2 Protein product graphs for protein structural comparisons

For many years now, we have used the developed clique algorithms in a real drug design application. First, we represented proteins as graphs, which is a natural way to represent molecules, since they are composed of atoms and bonds, which are natural graphs. In these, so called "protein graphs", vertices are positioned at the geometric centers of functional groups of protein surface amino acids, with each vertex representing a physicochemical property of the underlying group: acceptor, donor, π-π stacking, aliphatic, or acceptor-donor. Two vertices, u_i and u_j, within a protein graph G are considered adjacent if the distance between them (u_i, u_j) is less than 15 Å [5].

The "protein product graph" G_p of two protein graphs, G_1 and G_2, is defined on the vertex set $V(G_p) = V(G_1) \times V(G_2)$. To compare a pair of protein graphs, we identify a maximum clique within their product graph, where the maximum clique corresponds to the superimposition aligning the most vertices of the compared protein graphs. Each vertex in the protein product graph (u_i, v_i) consists of two component vertices: one from the first protein graph $(u_i \in V(G_1))$ and one from the second protein graph $(v_i \in V(G_2))$. Generally, a protein product graph has $n_1 \times n_2$ vertices if the respective protein graphs have

n_1 and n_2 vertices. However, we reduce its size by considering only vertices with identical component vertex colors (physicochemical properties). Additionally, component vertices must exhibit similar neighborhoods in their corresponding protein graphs. This similarity is determined based on the comparison of distance matrices representing discretized distances between all pairs of vertices in the neighborhood. Two protein product graph vertices (u_i, v_i) and (u_j, v_j) are considered adjacent (we connect them by an edge) if the distances between their respective component vertices in both protein graphs are nearly identical, i.e., they differ by less than 2 Å [5].

We constructed ten benchmark protein product graphs, each derived from a pair of protein structures obtained from the Protein Data Bank (PDB). To assess the impact of protein size (i.e., the number of amino acids) on the performance of the maximum clique search, we generated product graphs from proteins ranging from approximately 50 to 2000 amino acids. Additionally, we considered protein pairs sharing sequence identities ranging from approximately 10% to 95%. These resulting protein product graphs serve as representative standard tests for evaluating newly developed maximum clique algorithms. The benchmark set is available at http://insilab.org (see MaxCliquePara software) [1].

3 Maximum clique algorithm

In a work from 2007 [4], we developed a new algorithm named MaxClique for finding a maximum clique in an undirected graph, in which we improved an approximate coloring algorithm originally used in an algorithm of Tomita et al. [17]. We used our new coloring algorithm to provide bounds to the size of the maximum clique in a basic algorithm, also developed in [6]. Additionally, we extended this basic algorithm to include dynamically varying bounds, which resulted in our MaxCliqueDyn algorithm that proved to be significantly faster than the basic algorithm that we started with. The MaxCliqueDyn algorithm, as described in [4], offers significant advances over previous such algorithms, in particular over the previous algorithm [17] through the integration of two innovative features.

First, we developed an improved approximate coloring algorithm called ColorSort. This algorithm preserves the nodes within the candidate set of graph vertices in a strategically arranged descending order based on their degrees. This ordering is based on the insight that the assignment of nodes to color classes, which normally disturbs the ordering, is only necessary above a certain threshold. This threshold is calculated as $kmin = |Qmax| - |Q| + 1$, where $|Qmax|$ is the size of the current maximum clique and $|Q|$ is the size of the clique discovered on the current branch of the search tree. Vertices with colors below $kmin$ cannot be used to construct subsequent cliques on the current branch, and therefore they can retain their original order. This approach consistently reduces both the number of steps required to identify a maximal clique and the time spent doing so.

Second, we designed the algorithm to adopts tighter, albeit more computationally intensive, upper bounds for a portion of the search space to speed up the process of identifying the maximum clique. In this context, the nodes within the set of graph vertices in the lower branches of the search tree are re-sorted based on their degrees. This sorting mechanism is supported by a counter of steps up to a certain level $Slevel$ of the search tree and a counter of total steps taken so far $Stotal$ and a parameter $Tlimit$ that dynamically activates or deactivates the sorting during the execution of the algorithm if $Slevel/Stotal < Tlimit$. This limits the use of computationally expensive resorting to lower levels (close to the root node) of the search tree, where such resorting has the greatest effect on the search effi-

ciency. This heuristic improves the overall performance of the algorithm on a variety of DIMACS and random graphs and provides the flexibility to fine-tune the performance for specific graph types by adjusting the $Tlimit$ parameter.

4 Parallel maximum clique algorithm

Recognizing the importance of modern multi-core computers and the possibility they give to algorithmic efficiency, in a 2013 work [1], we introduced a new exact parallel maximum clique algorithm called MaxCliquePara. This novel branch-and-bound algorithm for finding a maximum clique in undirected general and protein graphs is based on concepts from two well-established sequential algorithms. We first developed a version for a single computer core, a sequential MaxCliqueSeq algorithm. This algorithm proved to be particularly fast compared to the reference algorithms for both DIMACS benchmark graphs and protein-derived product graphs used for protein structure comparisons. We then parallelized the MaxCliqueSeq algorithm by splitting the branch-and-bound search tree across multiple cores, resulting in the MaxCliquePara algorithm. By efficiently utilizing all available cores, the new parallel MaxCliquePara algorithm significantly outperformed the other algorithms tested. On a 12-core computer system, parallelization resulted in up to two orders of magnitude faster execution on large DIMACS benchmark graphs and up to one order of magnitude faster execution on protein product graphs.

The MaxCliquePara algorithm uses the multithreading techniques that are supported in most programming languages without additional libraries or other software support. This makes the algorithm transferable to other languages and operating systems. It can be executed on most modern multicore computer architectures. The parallel algorithm behaves identically to the MaxCliqueSeq algorithm when run on a single core, but takes slightly longer to run due to the additional thread management code. However, when running on multiple cores, this extra effort is easily compensated if the maximum clique problem to be solved is not trivial, e.g. for graphs with high density. The parallel algorithm explores several branches simultaneously. This reduces the number of steps compared to the sequential algorithm and leads to a faster execution time (speedup) of the MaxCliquePara algorithm.

The experiments performed show significant speedups of the MaxCliquePara algorithm for a smaller number of parallel cores on most of the graphs tested. With the maximum 12 physical cores available, and even with the additional 12 hyperthreaded cores (24 cores in total), the speedup scales strongly for larger and more computationally intensive graphs. An important advantage of MaxCliquePara is the use of shared memory parallelism, which enables low overhead and thus fast execution on a wide range of graph sizes.

Superlinear speedups are observed that are consistent with expectations for an algorithm that traverses a tree of possible solutions to search. For graphs that are “simple" for the sequential maximum clique algorithm, i.e., with an execution time on the order of milliseconds, parallelism does not provide significant benefits. The MaxCliquePara algorithm is currently considered to be one of the fastest general maximum clique algorithms for almost all but the most trivial graphs.

5 Maximum clique algorithm using machine learning

In another work from 2022 [14], we introduced machine learning into the MaxCliqueDyn algorithm. We extended the MaxCliqueDyn algorithm with a first phase of machine learning to predict the dynamic parameter ($Tlimit$) that is best suited for each input graph,

resulting in the new MaxCliqueDyn-ML algorithm. $Tlimit$ is an empirical parameter that determines the course of the algorithm as described in a previous section. Such graph type adaptability based on modern machine learning is a novel approach that has not yet been used in most graph-theoretic algorithms. We showed empirically that the resulting new algorithm MaxCliqueDyn-ML improves the search speed for certain types of graphs, in particular for molecular docking graphs used in drug discovery that determine energetically favorable conformations of small molecules at a protein binding site. In such cases, the increase in speed is twofold.

6 Maximum weight clique algorithm

Weighted graphs are important for drug design because they can represent binding strengths, with the maximum weighted clique in these graphs indicating the strongest bonding site.

In a work from 2024 [15], we introduce a new algorithm MaxCliqueWeight for identifying a maximum weight clique in a weighted graph, and its variant MaxCliqueDynWeight with dynamically varying bounds. This algorithm uses an efficient branch-and-bound approach with a new weighted graph coloring algorithm that efficiently determines upper weight bounds for a maximum weighted clique in a graph. The algorithm is based on our MaxClique fast branch and bound algorithm for finding a maximum clique in an undirected graph. We test the newly developed algorithm and its dynamic variant on random weighted graphs of up to 10,000 vertices and on standard benchmark DIMACS graph, used in different research fields and industries, available at http://insilab.org/maxcliqueweight.

We show that our newly developed algorithms are up to three orders of magnitude faster than a comparable Cliquer algorithm on random graphs, with the largest speedup achieved by the MaxCliqueDynWeight algorithm on a graph with 100 vertices and an edge density of 0.95; the MaxCliqueWeight algorithm is up to six orders of magnitude faster on DIMACS graphs, with the highest speedup on the san200_0.7_2 graph with 200 nodes and an edge density of 0.7. The main difference of the MaxCliqueWeight algorithm over the MaxClique algorithm is that in the coloring algorithm, we use the assignment of vertices to color classes as a means to calculate weight upper bounds for the maximum weight of a clique that is reachable from each vertex by following down the branch from that vertex. The idea is to calculate the weight of each color class as the maximum weight of its containing vertices. The sum of color class weights is then used as the upper bound to the weighted clique in a graph.

7 Highest weight k-clique algorithm

In drug design, searching for k-cliques is important because both subgraph isomorphism and docking site search involve identifying specific substructures of a given size. The k parameter represents this size, allowing researchers to match smaller graphs or docking molecules to potential binding sites accurately.

In a work from 2024 [16], we introduced a new algorithm for identifying up to N highest-weight k-cliques in a vertex-weighted graph, where k and N are given as parameters. This newly developed algorithm is a versatile graph-theoretical algorithm suitable for various types of vertex-weighted graphs and universal problem solving. To the best of our knowledge, this is the first algorithm of its kind. It is an extension of the MaxClique algorithm. The new algorithm finds up to N highest weighted k-cliques in an undirected vertex-weighted graph. A k-clique is defined as a clique with exactly k vertices, where

k can take any positive integer value. If k is less than or equal to the size of the maximum clique in a given graph, then the decision problem will have at least one k-clique as a solution.

This algorithm is used in molecular docking, so we have created a benchmark set of weighted docking graphs that is available at http://insilab.org/kcliqueweight for other researchers to test their algorithms. The algorithm allows to determine the conformation with the lowest energy of a docked ligand within a protein binding site. In this context, the different ligand conformations are represented as a docking graph. This graph consists of vertices, each denoting a ligand fragment with weights corresponding to their docking energy, and edges indicating the connectivity between the fragments. A k-clique with the highest weight in this graph, where the parameter k is the number of fragments of a ligand, corresponds to the conformation of the ligand with the lowest energy.

8 Applications of clique algorithms in drug design

Based on the developed clique algorithms, we have developed new ProBiS tools - web servers, databases and algorithms to recognize structurally similar protein binding sites based on the fact that protein surface structures are conserved in binding site regions [6]. These tools search for local similarities in the physico-chemical properties of different protein surface structures, independent of sequence or folding. The proteins are modeled as protein graphs, i.e. as rigid 3D objects consisting of vertices and edges. The developed algorithms are used for the prediction of protein binding sites and ligand transposition in experimental and artificial intelligence modeled protein structures.

Based on the MaxCliqueDyn algorithm, we have developed the ProBiS algorithm [5], which aligns and overlays complete protein surfaces, surface motifs or protein binding sites. It enables pairwise alignments of entire protein structures or selected binding sites as well as a fast search of hundreds of thousands of proteins for similar protein binding sites. The algorithm can find similar binding sites even in proteins with different folds and without prior knowledge of their location. The ProBiS algorithm can be used in parallel on one or more CPU platforms.

We have developed the ProBiS-Dock docking algorithm [8], which enables fast docking of small molecules to proteins using a newly developed fast highest weight k-clique algorithm. Small molecules and proteins are treated as fully flexible entities that allow conformational changes in both upon ligand binding.

An overview of specific case studies in which these algorithms have been used shows their practical application in drug discovery. For example, ProBiS-based methods were used in the inverse docking of various plant substances. Notable studies include the docking of these compounds reported in Refs. [9, 10, 11]. In addition, ProBiS-based methods have contributed to the identification of water-binding sites, with important results published in [2, 13].

9 Conclusions

In summary, the research presented in this paper demonstrates significant advances in clique algorithms and their applications, particularly in the field of drug design. The review begins with an investigation of the maximum clique problem (MCP) and its weighted variant (MWCP), both of which are fundamental challenges in complexity theory. Based on insights from previous work, we developed novel algorithms to efficiently solve these prob-

lems, including MaxCliqueDyn, MaxCliquePara and MaxCliqueDyn-ML, each of which is tailored to specific computational challenges and improves solution accuracy.

Our investigations went beyond traditional clique problems to include the identification of k-cliques with the highest weight in vertex-weighted graphs, a groundbreaking endeavor that is of great use in various graph-theoretic applications. These innovations culminated in the development of the ProBiS suite of tools that utilize clique algorithms to facilitate key tasks in drug discovery, including protein binding site recognition and ligand docking.

By constructing protein product graphs and conducting extensive empirical evaluations, we were able to demonstrate the efficacy and scalability of our algorithms across a spectrum of graph sizes and complexities. In particular, our parallelization efforts resulted in significant speedups, especially on multicore architectures, underscoring the practical relevance of using modern computational paradigms to improve algorithmic efficiency.

Furthermore, our integration of machine learning techniques into clique algorithms represents a breakthrough approach to algorithmic optimization, enabling adaptive parameter selection and improving solution quality for specific graph types. This fusion of traditional algorithmic principles with contemporary machine learning methods is an example of a synergistic paradigm shift in computational problem solving.

The culmination of these efforts in the ProBiS suite provides researchers in the fields of molecular biology, pharmaceuticals and structural bioinformatics with unprecedented capabilities for protein structure analysis, binding site prediction and molecular docking studies. By providing accessible, powerful tools for complex biological analysis, our work paves the way for accelerated drug discovery processes and deeper insights into protein-ligand interactions.

In essence, our contributions highlight the transformative potential of clique algorithms in addressing real-world challenges at the intersection of graph theory, computational biology and pharmaceutical science. As we continue to refine and expand these methods, we anticipate further advances that will catalyze innovation and support researchers in their quest to develop new therapeutics and elucidate biological mechanisms.

ORCID iDs

Janez Konc https://orcid.org/0000-0003-0160-3375
Dušanka Janežič https://orcid.org/0000-0003-4067-0116

References

[1] M. Depolli, J. Konc, K. Rozman, R. Trobec and D. Janežič, Exact parallel maximum clique algorithm for general and protein graphs, *J. Chem. Inf. Model.* **53** (2013), 2217–2228, doi: 10.1021/ci4002525, https://doi.org/10.1021/ci4002525.

[2] M. Jukič, J. Konc, D. Janežič and U. Bren, ProBiS H2O MD approach for identification of conserved water sites in protein structures for drug design, *ACS Med. Chem. Lett.* **11** (2020), 877–882, doi:10.1021/acsmedchemlett.9b00651, https://doi.org/10.1021/acsmedchemlett.9b00651.

[3] R. M. Karp, Reducibility among combinatorial problems, in: R. E. Miller, J. W. Thatcher and J. D. Bohlinger (eds.), *Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations*, Springer US, Boston, MA, pp. 85–103, 1972, doi:10.1007/978-1-4684-2001-2_9, https://doi.org/10.1007/978-1-4684-2001-2_9.

[4] J. Konc and D. Janežič, An improved branch and bound algorithm for the maximum clique problem, *MATCH Commun. Math. Comput. Chem.* **58** (2007), 569–590.

[5] J. Konc and D. Janežič, Probis algorithm for detection of structurally similar protein binding sites by local structural alignment, *Bioinformatics* **26** (2010), 1160–1168, doi:10.1093/bioinformatics/btq100, https://doi.org/10.1093/bioinformatics/btq100.

[6] J. Konc and D. Janežič, Probis tools (algorithm, database, and web servers) for predicting and modeling of biologically interesting proteins, *Prog. Biophys. Mol. Biol.* **128** (2017), 24–32, doi:10.1016/j.pbiomolbio.2017.02.005, https://doi.org/10.1016/j.pbiomolbio.2017.02.005.

[7] J. Konc and D. Janežič, Protein binding sites for drug design, *Biophys. Rev.* **14** (2022), 1413–1421, doi:10.1007/s12551-022-01028-3, https://doi.org/10.1007/s12551-022-01028-3.

[8] J. Konc, S. Lesnik, B. Skrlj, M. Sova, M. Proj, D. Knez, S. Gobec and D. Janežič, ProBiS-Dock: a hybrid multitemplate homology flexible docking algorithm enabled by protein binding site comparison, *J. Chem. Inf. Model.* **62** (2022), 1573–1584, doi:10.1021/acs.jcim.1c01176, https://doi.org/10.1021/acs.jcim.1c01176.

[9] K. Kores, J. Konc and U. Bren, Mechanistic insights into side effects of troglitazone and rosiglitazone using a novel inverse molecular docking protocol, *Pharmaceutics* **13** (2021), 315, doi:10.3390/pharmaceutics13030315, https://doi.org/10.3390/pharmaceutics13030315.

[10] K. Kores, S. Lesnik, U. Bren, D. Janežič and J. Konc, Discovery of novel potential human targets of resveratrol by inverse molecular docking, *J. Chem. Inf. Model.* **59** (2019), 2467–2478, doi:10.1021/acs.jcim.8b00981, https://doi.org/10.1021/acs.jcim.8b00981.

[11] S. Lešnik and U. Bren, Mechanistic insights into biological activities of polyphenolic compounds from rosemary obtained by inverse molecular docking, *Foods* **11** (2021), 67, doi:10.3390/foods11010067, https://doi.org/10.3390/foods11010067.

[12] R. Marino, L. Buffoni and B. Zavalnij, A short review on novel approaches for maximum clique problem: from classical algorithms to graph neural networks and quantum algorithms, 2024, arXiv:2403.09742 [math.CO].

[13] V. Ravnik, M. Jukič and U. Bren, Identifying metal binding sites in proteins using homologous structures, the MADE approach, *J. Chem. Inf. Model.* **63** (2023), 5204–5219, doi:10.1021/acs.jcim.3c00558, https://doi.org/10.1021/acs.jcim.3c00558.

[14] K. Reba, M. Guid, K. Rozman, D. Janežič and J. Konc, Exact maximum clique algorithm for different graph types using machine learning, *Mathematics* **10** (2021), 97, doi:10.3390/math10010097, https://doi.org/10.3390/math10010097.

[15] K. Rozman, A. Ghysels, D. Janežič and J. Konc, An exact algorithm to find a maximum weight clique in a weighted undirected graph, *Sci. Rep.* **14** (2024), Id/No 9118, doi:10.1038/s41598-024-59689-x, https://doi.org/10.1038/s41598-024-59689-x.

[16] K. Rozman, A. Ghysels, B. Zavalnij, T. Kunej, U. Bren, D. Janežič and J. Konc, Enhanced molecular docking: Novel algorithm for identifying highest weight k-cliques in weighted general and protein-ligand graphs, *J. Mol. Struct.* **1304** (2024), 137639, doi:10.1016/j.molstruc.2024.137639, https://doi.org/10.1016/j.molstruc.2024.137639.

[17] E. Tomita and T. Seki, An efficient branch-and-bound algorithm for finding a maximum clique, in: *International Conference on Discrete Mathematics and Theoretical Computer Science*, Springer, 2003 pp. 278–289.