# Solving hard optimization problems of packing, covering, and tiling via clique search

Sándor Szabó
sszabo7@hotmail.com
University of Pécs
Pecs, Hungary

Bogdán Zaválnij
bogdan@renyi.hu
HUN-REN Alfred Renyi Institute of Mathematics
Budapest, Hungary

## Abstract

In the paper we propose to convert NP-hard combinatorial optimization problems of packing, covering, and tiling types into maximum or $k$-clique problems. The key step is to come up with a tactically constructed auxiliary graph whose maximum or $k$-cliques correspond to the sought combinatorial structure. As an example, we will consider the problem of packing a given cube by copies of a brick. The aim of the paper is two fold to illustrate (i) the modeling power and (ii) the feasibility of the clique approach. Since theoretical tools are not readily available to study the effectiveness of the solution of the resulting clique problems we will carry out carefully conducted numerical experiments.

## Keywords

mathematical programming, $k$-clique problems, combinatorial optimization

## 1 Introduction

One can see graphs as a mathematical models that can describe various fields of interest. Like numbers, functions, or Linear Programming graph based approach can model interesting problems and aid us in solving them. Some of these approaches are quite straightforward like cliques of people in a social interaction graphs or shortest path problem in a road map. Other approaches are less obvious but still easily constructed, like conflict graphs in a set of codewords where a maximum independent set represents a maximum set of suitable error correcting codes [9].

But the approach of modeling and solving various problems by graphs are more versatile. Namely, we can see graphs as a language for mathematical programming – if certain combinatorial problems can be solved by constructing a suitable auxiliary graph and finding a maximum or $k$-clique of this graph gives the solution. The authors have already used this approach in connection with mathematical conjectures [1], hyper graph coloring [11], subgraph isomorphism [2], scheduling problems [12], graph coloring problems [13] and protein docking problems in chemistry [8].

Here we would like to give an example, where a hard combinatorial optimization problem can be solved by this approach. For this we chose a simple to understand but numerically hard to solve problem of brick packing popularized by M. Gardner. We will focus on different approaches of how to construct an auxiliary graph in order that to translate this problems into a clique search problem. We will try to investigate how these different approaches – based on packing, covering and tiling– affect the solving time and if they have other consequences as well. First, we describe the basic problem, then we present theoretical discussion of different reformulations, and finally we describe the results of numerical experiments. The emphasis is on the modeling aspect of the computation and not on reaching new records, as the proposed problem was solved in theoretical manner within months of its formulation. Here we use it as a prototype of similar problems, and our aim to show the versatility of our approach, that is model a problem by a graph.

Graphs in this paper will be finite simple graphs. Further all graphs we use will not have loops or double edges. A finite simple graph $G$ can be described with its set of nodes $V$ and a subset $E$ of the Cartesian product $V \times V$. The subset $E$ can be identified by the set of edges of $G$.

Let $G = (V, E)$ be a finite simple graph. A non-empty subset $C$ of $V$ is called a $k$-clique if each two distinct nodes of $C$ are adjacent in $G$ and in addition $C$ has exactly $k$ elements. If $C$ has only one element, then we consider it a 1-clique. The 2-cliques of $G$ are the edges of $G$. A $k$-clique $C$ of $G$ is called a maximum clique if $G$ does not have any $(k+1)$-clique. For each finite simple graph $G$ there is an integer $k$ such that $G$ contains a $k$-clique but $G$ does not contain any $(k+1)$-clique. This well defined integer $k$ is called the clique number of $G$. We state two clique problems formally.

Problem 1. *Given a finite simple graph $G$ and an integer $k$. Decide if $G$ has a $k$-clique.*

Problem 2. *Compute the clique number of a given finite simple graph.*

Problem 1 is a decision problem, it is referred as the $k$-clique problem, and it is an NP-complete problem included in the original list of 21 NP-complete problems by Karp [7]. Problem 2 is an optimization problem and referred as the maximum clique problem, and as the decision problem belongs to the NP-complete class it follows that it belongs to the NP-hard class.

We color the nodes of a finite simple graph $G$ with the colors $1, 2, \ldots, k$ such that each node receives exactly one color and adjacent nodes never receive the same color. Such a coloring of the nodes of $G$ is called a well coloring, a proper coloring, or a legal coloring (the terminology is not unified). The set of nodes of $G$ receiving the color $i$ is called the $i$-th color class. Clearly, a color class is an independent


*Information Society 2024, 7–11 October 2024, Ljubljana, Slovenia*

https://doi.org/https://doi.org/10.70314/is.2024.sikdd.9

set of $G$, that is, two nodes from a fixed color class are never adjacent.

If the nodes of a finite simple graph can be legally colored using $k$ colors, then we say that $G$ is a $k$-partite graph. The reason is that in this situation the nodes of $G$ form a union of $k$ independent sets and these sets are pair-wise disjoint.

In this paper we will focus on the following clique problem.

Problem 3. *Given a finite simple graph $G$ whose nodes are legally colored using $k$ colors. Decide if $G$ has a $k$-clique.*

Problem 3 is a $k$-clique problem particularized to case of $k$-partite graphs. This problem is still an NP-complete problem, as the graph coloring problem can be reduced to such question as shown in [13], and should not be confused with the problem of complete graphs.

The problem class we will be focusing on in the present paper consists of packing, covering, or tiling problems. Obviously many real world and mathematical problems fall into this class, and here we would show some ideas how such problems can be modeled by a suitably constructed auxiliary graph where a $k$-clique search would solve the original problem.

## 2 Packing, covering, and tiling

First, we describe the problem class in question. Second, we draw up some basic concepts how these problems can be modeled by graphs.

Let $U$ be a finite ground set and let

$$A_1, \ldots, A_m \quad (1)$$

be subsets of $U$. A family of subsets

$$B_1, \ldots, B_n \quad (2)$$

with $\{B_1, \ldots, B_n\} \subseteq \{A_1, \ldots, A_m\}$ is called a packing of $U$ if the members of the family (2) are pair-wise disjoint. A family of subsets (2) is called a covering of $U$ if the union of (2) is equal to $U$. Phrasing it differently, a family of subsets (2) is a covering of $U$ if each element of $U$ belongs to at least one member of the family (2). If a family of subsets (2) is a packing and a covering of $U$ in the same time, then it is called a tiling of $U$. A tiling of $U$ some times referred as exact covering of $U$.

A packing of $U$ is called a $k$-packing if it consists of $k$ subsets of $U$. Similarly, a covering of $U$ is called a $k$-covering if it consists of $k$ subsets of $U$. Finally, a tiling of $U$ is called a $k$-tiling if it consist of $k$ subsets of $U$. For a given ground set $U$ and for its given subsets (1) there is an integer $k$ such that $U$ has a $k$-packing using subsets of the family (1) but there is no any $(k+1)$-packing of $U$ using members of the family (1). This well defined integer $k$ is the packing number of $U$ with respect to the family (1). If the packing number of $U$ is equal to $k$, then each $k$-packing of $U$ is called maximum packing of $U$.

For a given ground set $U$ and for its given subsets (1) there is an integer $k$ such that $U$ has a $k$-covering using subsets of the family (1) but there is no any $(k-1)$-covering of $U$ using members of the family (1). This well defined integer $k$ is the covering number of $U$ with respect to the family (1). If the covering number of $U$ is equal to $k$, then each $k$-covering of $U$ is called minimum covering of $U$.

We state five problems related to packings, coverings, and tilings in a formal manner. Given a finite set $U$ and its subsets (1).

Problem 4. *Decide if $U$ has a $k$-packing using the members of the family (1).*

Problem 5. *Decide if $U$ has a $k$-covering using the members of the family (1).*

Problem 6. *Decide if $U$ has a $k$-tiling using the members of the family (1).*

Problem 7. *Compute the packing number of $U$ with respect to the family (1).*

Problem 8. *Compute the covering number of $U$ with respect to the family (1).*

Problem 4 can be reduced to Problem 1. We construct a finite simple graph $G$. The nodes of $G$ are the members of the family (1). Two distinct nodes $A_i$ and $A_j$ are adjacent in $G$ whenever $A_i$ and $A_j$ are disjoint. A $k$-clique in $G$ corresponds to a $k$-packing of $U$.

Problem 5 can be reduced to Problem 3. We sketch the main points of this reduction. We construct a finite simple graph $G$. The first type of nodes of $G$ are ordered pairs $(B, x)$, where $B \in \{A_1, \ldots, A_m\}$, $1 \leq x \leq k$. The intuitive meaning of the pair $(B, x)$ that the subset $B$ is the $x$-th member of a $k$ element family of (1). To the node $(B, x)$ we assign the color $x$. Two nodes receiving the same color will be non-adjacent in $G$. Therefore the first type nodes of $G$ are legally colored with $k$ colors.

We are adding second type nodes to $G$. Namely, we are adding the ordered pairs $(A, u)$, where $A \in \{A_1, \ldots, A_m\}$, $u \in U$ and in addition $u \in A$ holds. The intuitive meaning of the pair $(A, u)$ is that the element $u$ is covered by set $A$. To the node $(A, u)$ we assign $u$ as a color. Two nodes receiving the same color will not be adjacent in $G$. Thus the second type nodes of $G$ are legally colored using $t = |U|$ colors. Now if we are locating a $(k+t)$-clique in $G$, then we select exactly $k$ subsets from (1) and each element of $U$ will belong to at least one of these subsets. The missing part of the construction, what we left for the reader, is how the first and second types of nodes are connected by edges.

Problem 6 can be reduced to Problem 3. As a tiling is a packing and covering at the same time, we can add the packing restrictions, namely not connecting two sets if they intersect, to the second type of nodes. On the other hand – in case of equal size sets –, we do not need to count the used sets, so we won't need the first type of nodes, they can be omitted.

The computational difficulties of the $k$-packing, $k$-covering, and $k$-tiling problems are different. It seems that the covering problems are the computationally most demanding and the tiling problems are the most manageable.

## 3 Gardner's bricks problem

We picked Gardner's problem because it is intuitive and easy to comprehend among such problems that can be reduced to Problem 3 and so it serves as a good illustration of the kind of clique modeling we are dealing with. We do not claim any originality in connection with the problem. We do not prove any new results. Each of the facts we use are known from the folklore and we present them only for the reader convenience. The problem was raised by Foregger in March 1975 [10], popularized by Gardner in February 1976 [5], and solved by Foregger and Mather in November 1976 [3].

Let us consider a brick $B$ of dimensions $1 \times 2 \times 4$. The brick $B$ is a union 8 unit cubes whose edges are parallel to the coordinate axis. From some reason unknown for us the brick $B$ is referred as canonical brick. Suppose we have a large supply of congruent copies of $B$ and we want to pack as many as possible into a $7 \times 7 \times 7$ cube $C$. The cube $C$ is a union of 343 unit cubes. Let us divide 343 by 8 with remainder. As $343 = (42)(8) + (7)$, 43 copies of $B$ cannot be packed into $C$. M. Gardener advanced the question if 42 copies of $B$ can be placed into $C$. One can place a copy of $B$ into $C$ in any possible rotated position as long the edges of $B$ are parallel to the coordinate axis. (The answer to this question is actually: No, one cannot place 42 bricks into a cube of size $7 \times 7 \times 7$.)

Gardner's problem can be expressed in terms of computing the clique number of a suitable constructed graph $G$. In other words, Gardner's problem can be reduced to an instance of the maximum clique problem. Let us denote the set of the 343 unit cubes forming $C$ by $U$. An 8 elements subset $v$ of $U$ is a vertex of $G$ if the union of the elements of $v$ is a congruent copy of $B$. As it turns out $G$ has 1008 nodes. Two distinct nodes $v$ and $v'$ of $G$ are adjacent in $G$ if $v$ and $v'$ are disjoint. If $G$ contains a (42)-clique, then 42 congruent copies of $B$ can be packed into $C$. During our numerical experiments a greedy coloring procedure provided a legal coloring of the nodes of $G$ using 42 colors. Note that this is just a coincidence, it could've happened otherwise. Thus we are facing with a particular case of the $k$-clique problem stated in Problem 3. The nodes of $G$ are legally colored with 42 colors and we are looking for a (42)-clique in $G$. Phrasing it differently, we are looking for a $k$-clique in a $k$-partite graph, where $k = 42$.

We introduce a coordinate system whose origin coincides with a corner of the cube $C$.

Observation 1. *If 42 congruent copies of the brick $B$ can be packed into $C$, then there is such a packing which contains the congruent copy of $B$ whose one corner is the origin. Further the edges of lengths 1, 2, 4 are parallel to the first, second and third coordinate axis, respectively.*

Proof. As $343 = (42)(8) + (7)$ holds, 7 unit cubes of $C$ are not contained by any bricks of the packing. The cube $C$ has 8 corners and so at least one of the corners must be contained by a brick. At this point we introduce a coordinate system whose origin is this corner of $C$. Then we introduce the first, second, and third coordinate axis to satisfy our requirement. □

The cube $C$ can be sliced into 7 slabs using planes perpendicular to the first coordinate axes. Each slab is a $1x7x7$ slice of the big cube, that is a union of 49 unit cubes. The centers of these cubes are in a plane perpendicular to the first coordinate axis. The 7 unit cubes of $C$, that are not contained by any brick of the packing, are referred as unpacked unit cubes.

Observation 2. *Two distinct uncovered unit cubes of $C$ cannot be in the same slab.*

Proof. Note that a fixed slab can contain only 0, 2 or 4 unit cubes from any brick of the packing. The point is that the numbers 0, 2, 4 are all even. Each slab consists of an odd number of unit cubes. Therefore, each slab must contain an odd number of unpacked unit cubes. The number of slabs is 7 and so each slabs must contain exactly one unpacked unit cube. □

We can also form slabs by slicing $C$ with planes perpendicular to the second coordinate axes. Each of these 7 slabs contains exactly one unpacked unit cube. Finally, slicing $C$ by planes perpendicular to the third axes we get that each of these slabs contains exactly one unpacked unit cubes. These constraints on the uncovered unit cubes are independent, but can also be checked independently during an extended search, and as such can reduce the search space well.

## 4 Numerical experiments

Gardner's brick packing problem can be turned into various clique search problems and we carried out numerical experiments with them. We will observe that the same geometric problem will lead to very different clique search problems. When we try to pack 42 congruent copies of the canonical brick $B$ into the the big cube $C$, we get a $k$-clique problem. When we notice that the nodes of the auxiliary graph can be legally colored using 42 colors we get a $k$-clique problem in a $k$-partite graph which is a more tractable search problem. When we try to pack 42 congruent copies of the brick into the cube $C$ together with 7 unit cubes we get tiling problem. When we try to pack 42 congruent copies of the brick into the cube $C$ together with 7 unit cubes and in addition we distinguish the unit cubes among each other we get yet another version of the tiling problem.

In the first approach the auxiliary graph $G_1$ had 1008 vertices. The nodes of $G_1$ were legally colored using 42 colors and we tried to locate a (42)-clique in $G$. Note, that although this graph can be colored with 42 colors it was just a coincidence. There is no theoretical background to this fact. Of course the expectation was that $G_1$ do not have any (42)-clique.

Let us assume that it is possible to pack 42 congruent copies of the $1 \times 2 \times 4$ canonical brick $B$ into the $7 \times 7 \times 7$ cube $C$. By Observation 1, we may assume that a brick appear in the packing such that one of the corners of the brick coincides with the origin of the coordinate system and the edges of lengths 1, 2, 4 are along the 1-st, 2-nd, 3-rd coordinate axis. This information can be interpreted such that there a (42)-clique $C_2$ in $G_1$ which has a specific node. Namely, the vertex $v_1$ of $G_1$ that corresponds to the special corner brick is a node of the of $C_2$. This suggests to restrict the graph $G_1$ to the neighbors of the vertex $v_1$ to get a new graph $G_2$. Then we are looking for a (41)-clique in $G_2$. Plainly, the nodes of $G_2$ are legally colored using 41 colors. This coloring is inherited from the coloring of the nodes of $G_1$. Since the graph $G_2$ has fewer vertices than $G_1$ (actually 960) and we are looking for a smaller clique in $G_2$ than in $G_1$. The new clique problem probably requires less computational effort because the graph is smaller, and because we introduced a symmetry breaking to it.

The problem of packing 42 bricks into a bigger cube can be viewed as a tiling problem. Namely, we try to tile the

$7 \times 7 \times 7$ cube $C$ by 42 copies of the canonical brick and 7 additional copies of a unit cube. Thus we are facing to a tiling problem using two different types of tiles and the number of the tiles is given. To ensure that we use 42 bricks we numerate the small cubes as $\{1, \ldots, 7\}$ and ensure in the graph that each small unit cube is used once, that is we do not connect nodes where the unit square is covered by the same small cube. This tiling problem can also be reduced to a clique search problem. We denote the corresponding graph $G_3$. Tiling problems are more manageable compared with packing problems as during the search back-tracking can be anticipated earlier. However, the graph associated with the tiling in our case has more vertices than the graph associated with the packing, namely it has 10 465 nodes. Therefore only computations can reveal which approach is preferable.

Obviously, in this case we can also fix a brick in the corner. This version will be the $G_4$ graph.

In the last clique search equivalent of Gardner's problem we construct a graph $G_5$. In this construction we handle a mixed tiling problem but we utilize the extra information that no two distinct unit cube can appear in the same slab. By Observation 2, this may be assumed. This is done by not connecting two nodes associated with unit cubes if those unit cubes lay in the same slab. This graph is the same size as $G_3$, as we only delete some edged from it. Also, we can fix a brick in the corner in this case as well, that shall be the $G_6$ graph.

Once again only numerical experiments can guide us in judging the merits of the possible clique search equivalents of the problems. Further, the preconditioning methods perform differently on the graphs $G_1$, $G_2$, $G_3$, $G_4$, $G_5$, $G_6$ and this adds an extra layer of difficulty to the numerical work. We used a computer with AMD EPYC 7643 processors, `C++`, and gcc v12.1 with settings `-O3 -arch=znver3`.

We made all six graphs and performed $k$-clique search on them after preconditioning as described in [12, 13]. The preconditioning run for 1-2 hours for the bigger graph, and reduced it by half, namely to around 6 000 nodes for $G_3, G_4$; and to around 4 000 for $G_5, G_6$, that is the graphs where we allow only one small cube in a slab. For the smaller graphs $(G_1, G_2)$ the preconditioner runs for a couple seconds but cannot significally reduce the graph. Three of the six graph could be solved after preconditioning: $G_2$, $G_5$, and $G_6$.

The solution time of $G_2$ (the original graph with fixed brick in the corner) was 50 days. The solution of $G_5$ was a bit faster, 29 days. Finally, the graph $G_6$ could be solved more effectively. The running time was 123 484 seconds, that is 34 hours. This clearly show us the importance of the extra information of slabs.

## 5 Conclusions

We detailed several $k$-clique search reformulations of a certain combinatorial problem in terms of constructing suitable auxiliary graphs. We do not claim, that these methods result more efficient practical computations than other approaches. The point we are trying to make is that the clique reformulations open up a possibility to use well tuned clique solvers, including preconditioning, to handle different combinatorial problems in a unified manner as a general solver.

The results presented here have interesting consequences and suggest further research problems. First, and as anticipated, different auxiliary graphs lead to very different search space sizes. And although the usual concept in our research is that bigger graphs usually tend to be harder, that is not always the case. Remarkably, numerical results indicate that the size of the auxiliary graph alone is not as important as the type of the reformulation. Namely, the tiling type auxiliary graphs required less computational effort for clique search even if they were not the smallest graphs. Second, there are additional constraints that can be added to some reformulations while they seemingly cannot be incorporated into others. An example to such a constraint is the fact described after the proof of Observation 1. Namely, that no two distinct uncovered unit cubes can appear in the same slab in Gardner's brick packing problem. That kind of restriction could be incorporated into the tiling version of reformulation, and possibly not applicable to the packing reformulation. Taking advantage of the extra constraint made possible to solve the brick packing problem in reasonable time.

There are other problems that can be solved using similar approaches as detailed in the paper. Authors could solve smaller instances of the Golomb ruler problem or the Salem-Spencer set problem. The results, that lay outside the scope of the present paper, obtained with those instances open up even more interesting considerations.

## Acknowledgements

The present research was funded by National Research, Development and Innovation Office – NKFIH Fund No. SNN-135643.

## References

[1] K. Corrádi and S. Szabó, A combinatorial approach for Keller's conjecture. *Period. Math. Hungar.* Vol. 21, 91–100, 1990.
[2] M. Depolli, S. Szabó and B. Zaválnij, An Improved Maximum Common Induced Subgraph Solver. *MATCH Commun. Math. Comput. Chem.* 84 pp. 7–28. 2020.
[3] T. H. Foregger, and M. Mather, M. E2524. *The American Mathematical Monthly.* Vol. 83, No. 9 (Nov., 1976), pp. 741–742
[4] D. Hespe, Ch. Schulz, D. Strash. Scalable Kernelization for Maximum Independent Sets. *ACM Journal of Experimental Algorithmics.* Volume 24, Article No.: 1.16, pp 1–22. 2019.
[5] M. Gardner, MATHEMATICAL GAMES – Some elegant brick-packing problems, and a new order-7 perfect magic cube. *Scientific American.* Vol. 234, No. 2 (February 1976), pp. 122–127.
[6] M. R. Garey, and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, New York, 2003.
[7] R.M. Karp. "Reducibility Among Combinatorial Problems." In: Complexity of Computer Computations. New York: Plenum. pp. 85–103. 1972.
[8] 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. *JOURNAL OF MOLECULAR STRUCTURE.* 1304 p. 137639 Paper: 137639. 2024.
[9] N. J. A. Sloane. *Challenge Problems: Independent Sets in Graphs.* https://oeis.org/A265032/a265032.html
[10] T. H. Foregger. Elementary Problem E2524. *The American Mathematical Monthly.* Vol. 82, No. 3 (Mar., 1975), p. 300.
[11] S. Szabó and B. Zaválnij, Reducing hyper graph coloring to clique search. *Discrete Applied Mathematics.* 264. pp. 196–207. 2019.
[12] S. Szabó, and B. Zaválnij, Clique search in graphs of special class and job shop scheduling. *Mathematics.* 10(5), 697. 2022.
[13] S. Szabó, and B. Zaválnij, Graph Coloring via Clique Search with Symmetry Breaking. *SYMMETRY (BASEL).* 14 : 8 Paper: 1574, 16 p. 2022.