CONWAY'S 99-GRAPH PROBLEM: A BOOLEAN SATISFIABILITY APPROACH

NATHANIEL SELUB

Abstract. In this paper, we present a framework for solving Conway's 99-graph problem using a Boolean satisfiability problem (SAT) solver. We begin by encoding the problem into conjunctive normal form (CNF) using the Tseytin transformation. Then, we append symmetry-breaking clauses to the CNF encoding in order to improve SAT solver performance by preventing it from exploring isomorphic regions of the search space. Finally, we describe techniques to determine essential graph substructures, which aid in fixing variables in our SAT problem and simplifying the CNF encoding.

Contents

1. Introduction

Conway's 99-graph problem poses the following question: Does there exist an undirected graph with 99 vertices such that each edge belongs to a unique triangle and each non-adjacent vertex pair belongs to a unique quadrilateral?

A *Boolean formula* is an expression built from Boolean variables (variables that are either `TRUE` or `FALSE`), Boolean operators (`AND`, `OR`, `NOT`), and parentheses. An assignment of truth values to these variables is called an *interpretation*. The *Boolean Satisfiability Problem* (SAT) is to determine if a given Boolean formula can be made true. If there exists an interpretation that results in the formula being `TRUE`, the formula is called *satisfiable*, and if every possible interpretation results in the formula being `FALSE`, the formula is called *unsatisfiable*. A program designed to check the satisfiability of a SAT instance is called a SAT solver.

Conway's 99-graph problem can be expressed as a SAT problem. Consider $4851 = \binom{99}{2}$ Boolean variables $e_{1,2}, e_{1,3}, \ldots, e_{98,99}$. Here, $e_{i,j}$ indicates the presence (`TRUE`) or absence (`FALSE`) of an edge between vertices i and j. Let F be a Boolean formula whose arguments are the aforementioned Boolean variables such that F is `TRUE`if and only if each edge belongs to a unique triangle and each non-adjacent vertex pair belongs to a unique quadrilateral. Then, Conway's 99-graph problem is equivalent to determining the satisfiability of F.

A decision problem yields a "yes" or "no" answer for an input. In computational complexity theory, the nondeterministic polynomial time class (NP) is the set of all decision problems such that "yes" instances have solutions whose correctness can be verified in polynomial time. A problem is NP-complete if it is in NP and can emulate any other problem in NP with a solution that can be verified quickly. It is known that SAT is NP-complete.

In general, NP-complete problems are hard to solve. However, modern SAT solvers are very effective. As of 2023, they can process instances with nearly one billion variables and billions of clauses [**IntelPaper**]. Conway's 99-graph problem can be encoded as a SAT instance whose number of variables and clauses is well within these bounds, which makes it a candidate for resolution with a SAT solver.

This paper is structured as follows: In Section 2, we define conjunctive normal form (CNF) and show how to use Tseytin transformations to encode Conway's 99-graph problem as a Boolean formula in CNF.

2. Encoding the Problem in Conjunctive Normal Form

2.1. Conjunctive Normal Form. A *truth function* is a function function that takes truth values as input and yields a unique truth value as output. For example:

- The `AND` operator, represented as $\wedge$, is the truth-functional operator of conjunction.
- The `OR` operator, represented as $\vee$, is the truth-functional operator of disjunction.

A *clause* is a propositional formula constructed from a finite collection of literals (either atoms or their negations) combined using logical connectives.

We define a *clause* as a disjunction of literals. A formula is said to be in *conjunctive normal form* (CNF) if it consists of a conjunction of one or more clauses. Thus, one can view a CNF formula as an "`AND` of `OR`'s".

Given the variables A, B, C, D, E, and F, the following are examples of formulas in conjunctive normal form:

(1) $(A \vee \neg B \vee \neg C) \wedge (\neg D \vee E \vee F)$
(2) $(A \vee B) \wedge (C)$
(3) $(A \vee B)$
(4) (A)

By convention, a SAT instance needs to be in CNF in order to be processed by a SAT solver.

2.2. Encoding the problem as a Boolean formula. Consider $\binom{n}{2}$ Boolean variables $e_{1,2}, e_{1,3}, \ldots, e_{n-1,n}$. Here, $e_{i,j}$ indicates the presence (`TRUE`) or absence (`FALSE`) of an edge between vertices v_i and v_j. The ordering of the indices is unimportant, i.e., $e_{i,j} = e_{j,i}$. We denote the inverse of a Boolean variable $e_{i,j}$ as $\bar{e}_{i,j} = \neg e_{i,j}$, where $\neg$ is the logical `NOT` operator. For each variable $e_{i,j}$, there are two Boolean formulae associated with it

- If $e_{i,j}$, then there must exist a unique triangle containing $e_{i,j}$.
- If $\bar{e}_{i,j}$, then there must exist a unique quadrilateral containing $e_{i,j}$.

We refer to the first of the above Boolean formulae as the triangle condition for $e_{i,j}$, which we denote as $T_{i,j}$, and the second of the above formulae as the quadrilateral condition for $e_{i,j}$, which we denote as $Q_{i,j}$.

First, we encode the triangle condition for $e_{i,j}$ as a Boolean formula. Observe that if $e_{i,j}$ is `TRUE`, then there exists a triangle containing $e_{i,j}$ if and only if there exists a vertex v_k such that k is distinct from i, j and both $e_{i,k}$ and $e_{j,k}$ are `TRUE`. Therefore, if $e_{i,j}$ is `TRUE`, then a triangle containing $e_{i,j}$ exists if and only if

$$T_{i,j,E} \equiv \bigvee_{\substack{1 \le k \le n \\ k \ne i,j}} e_{i,k} \wedge e_{j,k}$$

is `TRUE`. We refer to $T_{i,j,E}$ as the triangle existence condition for $e_{i,j}$.

Observe that if $e_{i,j}$ is `TRUE`, then there does not exist more than one triangle containing $e_{i,j}$ if and only if there do not exist two vertices v_{k_1}, v_{k_2} such that k_1, k_2, i, j are distinct and e_{i,k_1}, e_{j,k_1}, e_{i,k_2}, e_{j,k_2} are `TRUE`. Therefore, if $e_{i,j}$ is `TRUE`, then there does not exist more than one triangle containing $e_{i,j}$ if and only if

$$\begin{aligned} T_{i,j,U} &\equiv \bigwedge_{\substack{1 \le k_1 < k_2 \le n \\ k_1, k_2 \ne i,j}} \neg \left(e_{i,k_1} \wedge e_{j,k_1} \wedge e_{i,k_2} \wedge e_{j,k_2}\right) \\ &= \bigwedge_{\substack{1 \le k_1 < k_2 \le n \\ k_1, k_2 \ne i,j}} \bar{e}_{i,k_1} \vee \bar{e}_{j,k_1} \vee \bar{e}_{i,k_2} \vee \bar{e}_{j,k_2}, \end{aligned}$$

where we have used De Morgan's Law. We refer to $T_{i,j,U}$ as the triangle uniqueness condition for $e_{i,j}$.

Using the definition of $a \implies b$, the triangle condition for $e_{i,j}$ can be written as

$$\begin{aligned} T_{i,j} &= \bar{e}_{i,j} \vee (T_{i,j,E} \wedge T_{i,j,U}) \\ &= (\bar{e}_{i,j} \vee T_{i,j,E}) \wedge (\bar{e}_{i,j} \vee T_{i,j,U}). \end{aligned}$$

Next, we encode the quadrilateral condition as a Boolean formula. Observe that there exists a quadrilateral containing v_i and v_j if and only if there exist two vertices k_1, k_2 such that k_1, k_2, i, j are distinct and e_{i,k_1}, e_{j,k_1}, e_{i,k_2}, e_{j,k_2} are TRUE. Therefore, a quadrilateral containing $e_{i,j}$ exists if and only if

$$Q_{i,j,E} \equiv \bigvee_{\substack{1 \le k_1 < k_2 \le n \\ k_1, k_2 \ne i,j}} e_{i,k_1} \wedge e_{j,k_1} \wedge e_{i,k_2} \wedge e_{j,k_2}$$

is TRUE. We refer to $Q_{i,j,E}$ as the quadrilateral existence condition for $e_{i,j}$.

Observe that there does not exist more than one quadrilateral containing $e_{i,j}$ if and only if there do not exist three vertices $v_{k_1}, v_{k_2}, v_{k_3}$ such that k_1, k_2, k_3, i, j are distinct and e_{i,k_1}, e_{j,k_1}, e_{i,k_2}, e_{j,k_2}, e_{i,k_3}, e_{j,k_3} are TRUE. Therefore, there do not exist more than one quadrilateral containing $e_{i,j}$ if and only if

$$\begin{aligned} Q_{i,j,U} &\equiv \bigwedge_{\substack{1 \le k_1 < k_2 < k_3 \le n \\ k_1, k_2, k_3 \ne i,j}} \neg \left(e_{i,k_1} \wedge e_{j,k_1} \wedge e_{i,k_2} \wedge e_{j,k_2} \wedge e_{i,k_3} \wedge e_{j,k_3} \right) \\ &= \bigwedge_{\substack{1 \le k_1 < k_2 < k_3 \le n \\ k_1, k_2, k_3 \ne i,j}} \bar{e}_{i,k_1} \vee \bar{e}_{j,k_1} \vee \bar{e}_{i,k_2} \vee \bar{e}_{j,k_2} \vee \bar{e}_{i,k_3} \vee \bar{e}_{j,k_3}, \end{aligned}$$

where we have used De Morgan's Law. We refer to $Q_{i,j,U}$ as the quadrilateral uniqueness condition for $e_{i,j}$. Using the definition of $a \implies b$, we can write the quadrilateral condition for $e_{i,j}$ as

$$\begin{aligned} Q_{i,j} &= e_{i,j} \vee (Q_{i,j,E} \wedge Q_{i,j,U}) \\ &= (e_{i,j} \vee Q_{i,j,E}) \wedge (e_{i,j} \vee Q_{i,j,U}) \end{aligned}$$

Define the Boolean formula

$$F_n \equiv \bigwedge_{1 \le i < j \le n} T_{i,j} \wedge Q_{i,j}.$$

Then, a graph satisfies the conditions of Conway's 99-graph problem if and only if the Boolean formula F_{99} evaluates to TRUE for the Boolean variables corresponding to the graph's edges. Furthermore, a graph satisfying the conditions of Conway's 99-problem exists if and only F_{99} is satisfiable.

2.3. **Converting the Boolean formula to Conjunctive Normal Form.** Observe that in order to transform F_n into CNF, it suffices to transform each occurrence of $\bar{e}_{i,j} \vee T_{i,j,E}$ and each $e_{i,j} \vee Q_{i,j,E}$ into CNF. Fix $i < j$ and consider

$$T_{i,j,E} = \bigvee_{\substack{1 \le k \le n \\ k \ne i,j}} e_{i,k} \wedge e_{j,k}.$$

If we attempt to convert this to CNF by expanding this formula, the resulting formula is

$$(e_{i,1} \vee e_{i,2} \vee \cdots \vee e_{i,n}) \wedge (e_{j,1} \vee e_{i,2} \vee \cdots \vee e_{i,n}) \wedge (e_{i,1} \vee e_{j,2} \vee \cdots \vee e_{i,n}) \wedge (e_{j,1} \vee e_{j,2} \vee \cdots \vee e_{i,n}) \wedge \cdots \wedge (e_{j,1} \vee e_{j,2} \vee \cdots \vee e_{j,n}),$$

where we note that all terms of the form $e_{i,j}$, $e_{i,i}$, and $e_{j,j}$ are absent in the above formula. The result is a formula with 2^{n-2} clauses, which, for $n = 99$, is far too large for any computer to handle.

Let F and F' be two Boolean formulae. Then, F and F' are *equisatisfiable* if F is satisfiable if and only F', and *equivalent* if $F = F'$ for all possible input variables. There exist transformations into CNF that avoid an exponential increase in size by transforming the original formula into a new formula such that the original formula and the new formula are equisatisfiable rather than equivalent. These transformations are guaranteed to only linearly increase the size of the formula, but introduce new variables.

We now define a new CNF formula $T'_{i,j,E}$ such that $T_{i,j,E}$ and $T'_{i,j,E}$ are equisatisfiable. Introduce auxiliary Boolean variables $t^k_{i,j}$ for $k = 1, \ldots, i-1, i+1, \ldots, j-1, j+1, \ldots n$, where the ordering of the subscript indices is unimportant, i.e., $t^k_{i,j} = t^k_{j,i}$. Let

$$T'_{i,j,E} \equiv \left[\bigvee_{\substack{1\le k\le n\\ k\ne i,j}} t^k_{i,j} \right] \wedge \left[\bigwedge_{\substack{1\le k\le n\\ k\ne i,j}} (\bar{t}^k_{i,j} \vee e_{i,k}) \wedge (\bar{t}^k_{i,j} \vee e_{j,k}) \right].$$

By inspection, every interpretation that satisfies $T_{i,j,E}$ satisfies $T'_{i,j,E}$ if at least one of the new variables is true, and $T'_{i,j,E}$ is unsatisfiable if $T_{i,j,E}$ is unsatisfiable.

Let

$$\begin{aligned}\tilde{T}_{i,j,E} &\equiv \bar{e}_{i,j} \vee T'_{i,j,E}\\ &= \left[\bar{e}_{i,j} \vee \left(\bigvee_{\substack{1\le k\le n\\ k\ne i,j}} t^k_{i,j} \right) \right] \wedge \left[\bigwedge_{\substack{1\le k\le n\\ k\ne i,j}} (\bar{e}_{i,j} \vee \bar{t}^k_{i,j} \vee e_{i,k}) \wedge (\bar{e}_{i,j} \vee \bar{t}^k_{i,j} \vee e_{j,k}) \right]\end{aligned}$$

and

$$\tilde{T}_{i,j,U} \equiv \bar{e}_{i,j} \vee T_{i,j,U}.$$

By inspection, $\tilde{T}_{i,j,E}$ is equisatisfiable with $\bar{e}_{i,j} \vee T_{i,j,E}$, and by definition, $\tilde{T}_{i,j,U}$ is equivalent to $\bar{e}_{i,j} \vee T_{i,j,U}$ so it follows that

$$\tilde{T}_{i,j} \equiv \tilde{T}_{i,j,E} \wedge \tilde{T}_{i,j,U}$$

is equisatisfiable with $T_{i,j}$.

For the analogous formula $e_{i,j} \vee Q_{i,j,E}$ resulting from the quadrilateral existence condition, we can avoid the corresponding exponential growth in clauses that results from a naive conversion to CNF by proceeding in exactly the same manner. Introduce auxiliary Boolean variables $q^{k_1,k_2}_{i,j}$ for $1 \le k_1 < k_2 \le n$ such that $k_1, k_2 \ne i, j$, where the ordering of the subscript indices and the ordering of the superscript indices are irrelevant, i.e., $q^{k_1,k_2}_{i,j} = q^{k_1,k_2}_{j,i}$ and $q^{k_1,k_2}_{i,j} = q^{k_2,k_1}_{i,j}$. Let

$$\begin{aligned}Q'_{i,j,E} \equiv &\left[\bigvee_{\substack{1\le k_1<k_2\le n\\ k_1,k_2\ne i,j}} q^{k_1,k_2}_{i,j} \right]\\ &\wedge \left[\bigwedge_{\substack{1\le k_1<k_2\le n\\ k_1,k_2\ne i,j}} (\bar{q}^{k_1,k_2}_{i,j} \vee e_{i,k_1}) \wedge (\bar{q}^{k_1,k_2}_{i,j} \vee e_{j,k_1}) \wedge (\bar{q}^{k_1,k_2}_{i,j} \vee e_{i,k_2}) \wedge (\bar{q}^{k_1,k_2}_{i,j} \vee e_{j,k_2}) \right].\end{aligned}$$

By inspection, every interpretation that satisfies $Q_{i,j,E}$ satisfies $Q'_{i,j,E}$ if at least one of the new variables is true, and $Q'_{i,j,E}$ is unsatisfiable if $Q_{i,j,E}$ is unsatisfiable.

Let

$$\begin{aligned}\tilde{Q}_{i,j,E} &\equiv e_{i,j} \vee Q'_{i,j,E}\\ &= \left[e_{i,j} \vee \left(\bigvee_{\substack{1\le k_1<k_2\le n\\ k_1,k_2\ne i,j}} q^{k_1,k_2}_{i,j} \right) \right]\\ &\wedge \left[\bigwedge_{\substack{1\le k_1<k_2\le n\\ k_1,k_2\ne i,j}} (e_{i,j} \vee \bar{q}^{k_1,k_2}_{i,j} \vee e_{i,k_1}) \wedge (e_{i,j} \vee \bar{q}^{k_1,k_2}_{i,j} \vee e_{j,k_1}) \wedge (e_{i,j} \vee \bar{q}^{k_1,k_2}_{i,j} \vee e_{i,k_2}) \wedge (e_{i,j} \vee \bar{q}^{k_1,k_2}_{i,j} \vee e_{j,k_2}) \right],\end{aligned}$$

and

$$\tilde{Q}_{i,j,U} = e_{i,j} \vee Q_{i,j,U}.$$

By inspection, $\tilde{Q}_{i,j,E}$ is equisatisfiable with $e_{i,j} \vee Q_{i,j,E}$, so it follows that

$$\tilde{Q}_{i,j} \equiv \tilde{Q}_{i,j,E} \wedge \tilde{T}_{i,j,U}$$

is equisatisfiable with $Q_{i,j}$.

Let

$$\tilde{F}_n = \bigwedge_{1\le i<j\le n} \tilde{T}_{i,j} \wedge \tilde{Q}_{i,j}.$$

By construction, $\tilde{F}_n$ is a Boolean formula in CNF that is equisatisfiable with F_n.

3. Breaking Symmetries in the Conjunctive Normal Form

The space of interpretations of $\tilde{F}_n$ is highly symmetric. In particular, let A be the following assignment of truth values:

$$\{e_{i,j} = t_{i,j}, t_{i,j}^k = b_{i,j}^k, q_{i,j}^{k_1,k_2} = c_{i,j}^{k_1,k_2}\}_{\substack{1\le i<j\le n \\ 1\le k\le n; k\ne i,j \\ 1\le k_1<k_2\le n; k_1,k_2\ne i,j}},$$

where each $a_{i,j}$, $b_{i,j}^k$, $c_{i,j}^{k_1,k_2}$ is either `TRUE` or `FALSE`, and for $\sigma \in S_n$, let $\sigma(A)$ be the following assignment of truth values:

$$\{e_{\sigma(i),\sigma(j)} = a_{\sigma(i),\sigma(j)}, t_{\sigma(i),\sigma(j)}^{\sigma(k)} = b_{\sigma(i),\sigma(j)}^{\sigma(k)}, q_{\sigma(i),\sigma(j)}^{\sigma(k_1),\sigma(k_2)} = c_{\sigma(i),\sigma(j)}^{\sigma(k_1),\sigma(k_2)}\}_{\substack{1\le i<j\le n \\ 1\le k\le n; k\ne i,j \\ 1\le k_1<k_2\le n; k_1,k_2\ne i,j}}.$$

Then, $\tilde{F}_n(A) = \tilde{F}_n(\sigma)$ for all $\sigma \in S_n$. Furthermore, the permutations $\sigma \in S_n$ partition the space of interpretations of $\tilde{F}_n$ into distinct orbits.

Without further modifications to the CNF encoding of our problem, the presence of this symmetry greatly reduce the performance of a SAT solver because without additional constraints, the SAT solver will spend a substantial amount of time exploring isomorphic parts of the search space. To determine the satisfiability of $\tilde{F}_n$, it suffices to consider a single representative interpretation from each orbit. To prevent the SAT solver from considering multiple interpretations from the same orbit, we can add additional clauses to the CNF encoding. Our method follows (Aloul).

An irredundant set of generators for S_n is

$$\{(1,2),(2,3),\ldots,(n-1,n)\}.$$

The transposition of vertices a and b interchanges the following variables in the CNF encoding for all distinct i, j, k distinct from each other and a, b:

$$\begin{aligned}
e_{a,i} &\leftrightarrow e_{b,i} \text{ for } i \ne a,b \\
t_{a,i}^j &\leftrightarrow t_{b,i}^j \text{ for } i \ne a,b; k \ne a,b,i \\
t_{a,i}^b &\leftrightarrow t_{b,i}^a \text{ for } i \ne a,b \\
t_{i,j}^a &\leftrightarrow t_{i,j}^b \text{ for } i,j \ne a,b \\
q_{a,i}^{k_1,k_2} &\leftrightarrow q_{b,i}^{k_1,k_2} \text{ for } i,k_1,k_2 \ne a,b \\
q_{a,i}^{b,k} &\leftrightarrow q_{b,i}^{a,k} \text{ for } i,k \ne a,b \\
q_{i,j}^{a,k} &\leftrightarrow q_{i,j}^{b,k} \text{ for } i,j,k \ne a.
\end{aligned}$$

4. A Review of Known Results About the Graph

5. Some Results from Graph Theory

6. Partially Solving the Graph

6.1. Solving Based on Trivial Constraints.

6.2. Solving Based on Eigenvalue Constraints.

7. CNF Files

First and foremost, I would like to sincerely thank my mentor Michael Klug for his insight, help, and guidance throughout the summer. I am also grateful to Peter May for organizing the 2023 University of Chicago REU during which this paper was written.