hexagon:2610.00307v2

Set Cover by Cyclic Rotations Is NP-Complete

Yuran Xiu

October 15, 2026

Abstract

Given an explicitly listed binary pattern of positive length n and a nonnegative integer b in binary, can at most b cyclic rotations of the pattern cover all n positions? We prove that this decision problem is NP-complete. The reduction constructs a pattern in a cyclic group of polynomial order and outputs its entire indicator word. Its principal ingredient is a replicated-marker construction: a cover within the budget must choose one translate in every layer, and each failed difference constraint leaves a block of more holes than the remaining translates can cover. Combining this construction with the known embedding of graph cliques into circulant graphs using distinct three-term sums gives a polynomial-time reduction from 3SAT. The theorem allows every subset, including the empty pattern, and every binary budget. A Lean formalization checks the mathematical reduction, the full-word polynomial-time programs, NP membership, and the unconditional NP-hardness foundation.

1 Introduction

Let S be a subset of the cyclic group $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$, where $n \geq 1$. Its translates are $S+t = \{s+t : s \in S\}$, for $t \in \mathbb{Z}_n$. The covering number is

$$\tau(S, \mathbb{Z}_n) = \min\{|T| : T \subseteq \mathbb{Z}_n,\ S + T = \mathbb{Z}_n\},$$

with value $+\infty$ if there is no cover. Thus the only pattern without a cover is $S = \varnothing$. All of the sets available to the covering algorithm are rotations of a single input pattern. This restriction is substantial: deleting a chosen set and the positions it covers generally does not leave an instance of the same form.

Wirth introduced this problem as *pattern rotation* in Section 2.7 of his dissertation [2]. He asked about its computational complexity and distinguished finding a minimum covering family from deciding a covering-number bound. The problem appeared again as Open Problem 4, *Set-Cover by Rotations*, attributed to Anthony Wirth in the WADS 2023 open-problems collection [1]. Covering numbers of this kind also form part of the broader study of covering groups by translates of a set; see Bollobás, Janson, and Riordan [3]. We settle the decision classification for an explicitly given pattern.

Theorem 1.1. *The following language is NP-complete under polynomial-time many-one reductions: an instance consists of an explicitly listed word $s \in \{0,1\}^n$, with $n \geq 1$, and a nonnegative integer b encoded in binary; it is accepted precisely when, for $S = \{i \in \mathbb{Z}_n : s_i = 1\}$, there exists $T \subseteq \mathbb{Z}_n$ such that*

$$|T| \leq b \qquad \text{and} \qquad S + T = \mathbb{Z}_n.$$

NP-hardness already holds for nonempty patterns and budgets $b \leq n$.

The pattern in this statement is the complete n-bit word. Its length is part of the input size; the theorem does not concern a succinct description of an exponentially long pattern. Neither nonemptiness of S nor an upper bound on b is a promise in the unrestricted language.

The central construction is a reduction from cyclic difference constraints to translate covering. In $\mathbb{Z}_m \times \mathbb{Z}_L$, the template has one dense slice and $m-1$ sparse slices. The dense slice has blocks of m holes. Each block represents a constraint between two chosen phases. A sufficiently large L forces a cover of size at most m to select exactly one translate in every layer. The markers are spaced so that a sparse slice covers either its entire intended block or at most one marker of a block. Consequently, $m-1$ unintended contributions cannot cover a failed block of m holes. We choose $L \equiv 1 \pmod{m}$, so the product group is an ordinary cyclic group of order mL.

To supply the difference constraints, we use the established three-sum-distinct clique embedding. Codenotti, Gerace, and Vigna used this method for circulant graphs [4]; Godsil and Rooney state and prove the corresponding finite abelian-group lemma [5, Lemma 6.1]. We give the required proof, including an explicit polynomial-size family of labels. The replicated-marker covering construction is the contribution of this paper; the clique embedding and the 3SAT occurrence-graph reduction are credited ingredients.

Sections 2–5 give a mathematical proof and an explicit polynomial-time algorithm. Section 6 describes the checked language, its correspondence with the set formulation, and the formal reproduction evidence.

2 The encoded problem and NP membership

We fix a binary encoding to include malformed inputs and arbitrarily large budgets in the statement. For words s and β, define $\mathrm{pair}(s,\beta)$ by doubling every bit of s ($0 \mapsto 00$ and $1 \mapsto 11$), appending the separator 01, and then appending β verbatim. Its length is $2|s|+2+|\beta|$. Reading consecutive pairs from the start identifies the first field and separator uniquely. A word without this form is rejected. The second field is a little-endian binary numeral:

$$b(\beta) = \sum_{j=0}^{|\beta|-1} \beta_j 2^j.$$

The empty numeral represents zero, and redundant high zero bits are allowed. Reversing the numeral or using a conventional nonempty encoding of zero gives polynomial-time equivalent ordinary binary encodings.

Definition 2.1. The language CyclicCover consists of the words $\mathrm{pair}(s,\beta)$ for which $n = |s| > 0$ and there is a word $t \in \{0,1\}^n$ satisfying

$$\mathrm{wt}(t) \le b(\beta), \tag{1}$$

$$\forall x \in \{0,\dots,n-1\}\ \exists i \in \{0,\dots,n-1\}: \quad t_i = s_{(x+n-i) \bmod n} = 1. \tag{2}$$

Here $\mathrm{wt}(t)$ denotes the number of ones in t.

The word t is the indicator of a set $T \subseteq \mathbb{Z}_n$. Its weight is $|T|$, and $(x+n-i) \bmod n$ represents $x-i$ in $\mathbb{Z}_n$. Therefore (2) is exactly $S+T=\mathbb{Z}_n$. A list of shifts with repetitions gives no stronger notion: deleting repetitions preserves its union and can only decrease its size.

Lemma 2.2. *For every $n \ge 1$, $S \subseteq \mathbb{Z}_n$, and $b \in \mathbb{N}$, a cover with at most b shifts exists if and only if one with at most $\min(b,n)$ shifts exists. The empty pattern is rejected at every budget, budget zero is rejected for every pattern, and every nonempty pattern is accepted when $b \ge n$.*

Proof. Every subset of $\mathbb{Z}_n$ has at most n elements, which proves the first claim. If $S = \varnothing$ or $T = \varnothing$, then $S + T = \varnothing$, whereas $\mathbb{Z}_n \neq \varnothing$. If $s_0 \in S$, every $x \in \mathbb{Z}_n$ belongs to $S + (x - s_0)$. Thus taking all n shifts covers $\mathbb{Z}_n$. □

Proposition 2.3. CyclicCover $\in$ NP.

Proof. Use the n-bit word t as the certificate. Parse the two fields, reject malformed inputs and $n = 0$, count $\mathrm{wt}(t)$, and test (2) for all n^2 pairs (x, i). These operations have polynomial running time in the input length on a multitape Turing machine: indices have $O(\log(n + 1))$ bits, and even sequential scans to access the specified bits incur only a polynomial overhead.

The binary budget causes no large-integer expansion. Read its bits from most significant to least significant, starting with $q = 0$, and update

$$q \longleftarrow \min(n, 2q + c)$$

when the next bit is c. Inductively q is the decoded prefix capped at n, since $\min(n, 2\min(n, u)+c) = \min(n, 2u + c)$ for every $u \in \mathbb{N}$ and $c \in \{0, 1\}$. All intermediate states have $O(\log(n + 1))$ bits. The final comparison with $\mathrm{wt}(t)$ is correct by Lemma 2.2. The certificate length is at most the input length, and all verification steps are polynomial in $n + |\beta|$. □

3 Replicated markers realize difference constraints

Fix integers $m \geq 2$ and $D \geq 0$. For each nonzero $d \in \mathbb{Z}_m$, let $A_d \subseteq [-D, D] \cap \mathbb{Z}$ be a given finite set. We will construct a single cyclic pattern whose covers of size at most m are equivalent to assignments of phases x_r satisfying

$$x_r - x_{r-d} \in \overline{A_d} \quad (r \in \mathbb{Z}_m,\ d \in \mathbb{Z}_m \setminus \{0\}). \tag{3}$$

Here and below a bar denotes reduction of an integer or integer set modulo the chosen positive integer L.

3.1 Construction and marker spacing

Set

$$h = m(m - 1), \qquad u_j = j + 2hj^2 \quad (0 \leq j < h), \qquad M = 2D + 1, \qquad H_{\max} = Mu_{h-1}. \tag{4}$$

The marker with index j is the integer Mu_j. Partition the markers into $m - 1$ blocks of size m: using representatives $1 \leq d < m$, put

$$H_d = \{\overline{Mu_j} : (d - 1)m \leq j < dm\}, \qquad H = \bigcup_{d=1}^{m-1} H_d. \tag{5}$$

Choose

$$Q = \max\{2H_{\max} + 2D,\ m^2(2D + 1),\ 3D\}, \tag{6}$$

$$L = m(\lfloor Q/m \rfloor + 1) + 1, \qquad N = mL. \tag{7}$$

Then

$$L > 2H_{\max} + 2D, \qquad L > m^2(2D + 1), \qquad L > 3D, \qquad L \equiv 1 \pmod{m}. \tag{8}$$

Indeed, writing $Q = ma + r$ with $0 \le r < m$ gives $L - Q = m + 1 - r > 0$. The markers are distinct in $\mathbb{Z}_L$: the u_j increase strictly and all Mu_j lie in $[0, H_{\max}] \subset [0, L)$. Thus the blocks in (5) are disjoint and each has exactly m elements.

Define the template S^* in $G = \mathbb{Z}_m \times \mathbb{Z}_L$ by its slices:

$$S_0 = \mathbb{Z}_L \setminus H, \qquad S_d = H_d + \overline{A_d} \quad (d \neq 0), \qquad S^* = \{(d, y) : y \in S_d\}. \tag{9}$$

The slice S_0 is dense and the other slices are sparse. In particular,

$$|S_d| \le m|A_d| \le m(2D + 1) \qquad (d \neq 0). \tag{10}$$

The template is nonempty, since $|H| = h < m^2 < L$.

Lemma 3.1 (Unique marker differences)**.** *The nonzero ordered differences $u_j - u_i$, for $0 \le i, j < h$ and $i \neq j$, determine (i, j) uniquely.*

Proof. First suppose $i < j$, $k < \ell$, and $u_j - u_i = u_\ell - u_k$. Reducing modulo $2h$ gives $j - i \equiv \ell - k \pmod{2h}$. Both differences are between 1 and $h - 1$, so they are equal to a common positive integer t. The original equality now reads

$$t + 2ht(i + j) = t + 2ht(k + \ell).$$

Canceling $2ht \neq 0$ gives $i + j = k + \ell$. Equal sums and differences give $i = k$ and $j = \ell$. Since u_j is strictly increasing, the sign of a nonzero difference identifies its orientation. Reversing both pairs therefore proves the negative-difference case as well. □

Lemma 3.2 (Overlap for every phase)**.** *Let $d, e \in \mathbb{Z}_m \setminus \{0\}$ and $\delta \in \mathbb{Z}_L$. Then*

$$\left|\{z \in H_e : z + \delta \in H_d + \overline{A_d}\}\right| = m \quad \text{if } e = d \text{ and } \delta \in \overline{A_d},$$

and this cardinality is at most one in every other case.

Proof. A covered marker has an index i in block e, and there are an index j in block d and $a \in A_d$ such that

$$\delta = \overline{M(u_j - u_i) + a}. \tag{11}$$

Every integer on the right before reduction lies in $[-H_{\max} - D, H_{\max} + D]$. This interval has diameter strictly less than L by (8). Therefore two representations of δ, with indices (i, j) and (k, ℓ) and offsets a, a', are equal as integers. Subtracting them gives

$$M\big((u_j - u_i) - (u_\ell - u_k)\big) = a' - a.$$

The right side has absolute value at most $2D < M$. Since the left side is a multiple of M, both sides vanish. Hence $a = a'$ and $u_j - u_i = u_\ell - u_k$.

If the common marker difference is nonzero, Lemma 3.1 identifies the ordered index pairs; in particular $i = k$, so the two covered markers are the same. If it is zero, strict increase gives $i = j$ and $k = \ell$. A marker belongs to only one block, so $e = d$, and (11) gives $\delta = \overline{a} \in \overline{A_d}$. Outside this exceptional case, any two covered markers must therefore coincide. In the exceptional case, $z + \delta \in H_d + \overline{A_d}$ for every $z \in H_d$, giving exactly m covered markers. This reasoning makes no restriction on the representative or size of the phase δ. □

3.2 Forcing the translates and the constraints

Lemma 3.3 (One translate in each layer)**.** *If $T \subseteq G$ satisfies $|T| \le m$ and $S^* + T = G$, then for every $r \in \mathbb{Z}_m$ there is exactly one $x_r \in \mathbb{Z}_L$ with $(r, x_r) \in T$.*

Proof. Suppose no selected shift has first coordinate r. A shift $(q, x) \in T$ then has $q \ne r$ and covers, in target row r, precisely the positions $x + S_{r-q}$. By (10), this set has at most $m(2D+1)$ elements. The union of at most m such sets has size at most $m^2(2D+1) < L$, and cannot cover the row. Every one of the m layers must consequently contain a selected shift. The total budget m implies there is exactly one in each layer. □

Theorem 3.4 (Replicated-marker equivalence)**.** *The template S^* has a cover in G of size at most m if and only if there exist $x_r \in \mathbb{Z}_L$, for $r \in \mathbb{Z}_m$, satisfying (3).*

Proof. Suppose first that the phases satisfy (3). Select $T = \{(r, x_r) : r \in \mathbb{Z}_m\}$. In row r, the shift (r, x_r) covers $\mathbb{Z}_L \setminus (H + x_r)$. For $d \ne 0$, the constraint gives $x_r - x_{r-d} \in \overline{A_d}$ and hence

$$H_d + x_r \subseteq H_d + \overline{A_d} + x_{r-d} = S_d + x_{r-d}.$$

The shift $(r - d, x_{r-d})$ therefore covers the entire block $H_d + x_r$. These blocks partition the holes of the dense slice, so every position in every row is covered.

Conversely, a cover of size at most m has the form above by Lemma 3.3. If a constraint fails for some (r, e), consider the m points of $H_e + x_r$ in row r. The own-row shift (r, x_r) misses all of them. The shift $(r - d, x_{r-d})$, for $d \ne 0$, covers one of these points $z + x_r$ precisely when

$$z + (x_r - x_{r-d}) \in H_d + \overline{A_d}.$$

If $d \ne e$, Lemma 3.2 bounds its contribution by one. If $d = e$, the same bound holds because the constraint was assumed to fail. The $m - 1$ other shifts thus cover at most $m - 1$ of the m points, a contradiction. Every constraint holds. □

3.3 Returning to a single cyclic group

Since $L \equiv 1 \pmod{m}$, the integers m and L are coprime. The map

$$\phi : \mathbb{Z}_{mL} \longrightarrow \mathbb{Z}_m \times \mathbb{Z}_L, \qquad \phi(z) = (z \bmod m, z \bmod L)$$

is an additive bijection. For completeness, if $0 \le r < m$ and $0 \le y < L$, its inverse is represented by

$$\psi(r, y) = y + L\big((r - y) \bmod m\big). \tag{12}$$

The last remainder lies in $\{0, \dots, m-1\}$, so $0 \le \psi(r, y) < mL$. The expression has residue y modulo L and, using $L \equiv 1 \pmod{m}$, residue r modulo m. Uniqueness follows because an integer divisible by both coprime factors is divisible by their product. Define

$$P = \phi^{-1}(S^*) \subseteq \mathbb{Z}_N, \qquad N = mL. \tag{13}$$

An additive bijection preserves translates, unions, and cardinalities. Thus Theorem 3.4 holds with P in $\mathbb{Z}_N$ in place of S^* in G. Only the two polynomial-size factors m and L will be used; there is no product of an unbounded family of moduli.

4 The credited circulant clique embedding

Let X be a finite simple undirected graph with vertices $\{1, \dots, v\}$, where $v \geq 2$. Put

$$B = 3v^3 + 1, \qquad g_i = i + Bi^2 + B^2 i^3, \qquad D = g_v, \tag{14}$$

and let

$$A = \{g_i - g_j : \{i, j\} \in E(X)\}, \tag{15}$$

including both orientations of every edge. All elements of A lie in $[-D, D]$, and A is symmetric and does not contain zero. Let $L > 3D$, and write the labels also for their residues when the group is clear.

Lemma 4.1 (Distinct three-term sums)**.** *If $a, b, c, d, e, f \in \{1, \dots, v\}$ and*

$$g_a + g_b + g_c = g_d + g_e + g_f \quad \text{in } \mathbb{Z}_L,$$

then the multisets $\{a, b, c\}$ and $\{d, e, f\}$ are equal. Repeated indices are allowed.

Proof. All integer sums in question lie in $[0, 3D]$, whose diameter is less than L. Their congruence is therefore an equality of integers. For each $r \in \{1, 2, 3\}$, a sum of three rth powers of indices is at most $3v^r \leq 3v^3 < B$. There are no carries between the three base-B coordinates in (14). Uniqueness of the base-B expansion implies that the two triples have the same first three power sums p_1, p_2, p_3. If e_1, e_2, e_3 denote their elementary symmetric polynomials, the identities

$$e_1 = p_1, \qquad 2e_2 = p_1^2 - p_2, \qquad 6e_3 = p_1^3 - 3p_1p_2 + 2p_3$$

show that these also agree. The two monic cubics $z^3 - e_1 z^2 + e_2 z - e_3$ are consequently identical. Their roots, counted with multiplicity, are the respective index triples, proving the assertion. □

Appending the same label to two equal two-term sums and canceling it from the resulting equal multisets shows that two-term sums are also distinct up to order. In particular,

$$g_a - g_b = g_c - g_d \text{ in } \mathbb{Z}_L, \quad a \neq b \quad \Longrightarrow \quad a = c \text{ and } b = d. \tag{16}$$

Indeed, equality gives $\{a, d\} = \{c, b\}$; the matching $a = b$, $d = c$ is excluded. The labels themselves are distinct modulo L, since they increase strictly in $[1, D] \subset [0, L)$. Thus $0 \notin \overline{A}$.

Let Y be the undirected Cayley graph on $\mathbb{Z}_L$ with connection set $\overline{A}$: distinct x, y are adjacent when $x - y \in \overline{A}$.

Lemma 4.2 (Codenotti–Gerace–Vigna; Godsil–Rooney)**.** *The graphs X and Y have the same clique number. More precisely, every clique of Y has a clique of the same size in X, and conversely.*

Proof. The map $i \mapsto g_i$ is injective and sends every edge of X to an edge of Y. In fact (16) shows that it is an induced embedding. This proves one direction.

For the other direction, let C be a nonempty clique of Y. Translate it so that $0 \in C$. Each $x \in C \setminus \{0\}$ has a unique representation $x = g_a - g_b$ with (a, b) an oriented edge of X, by (16). Consider two distinct such points $x = g_a - g_b$ and $y = g_c - g_d$. Since they are adjacent, write $y - x = g_e - g_f$ for another oriented edge. Rearrangement gives

$$g_a + g_e + g_d = g_c + g_b + g_f.$$

Lemma 4.1 implies equality of the multisets $\{a, d, e\} = \{b, c, f\}$. Also $a \neq b$, $c \neq d$, and $e \neq f$. To spell out the matching argument, a must match either c or f. If it matches c, the matching

$d = b, e = f$ is forbidden, leaving $d = f, e = b$. If it matches f, the matching $d = c, e = b$ is forbidden, leaving $d = b, e = c$. Hence

$$a = c \qquad \text{or} \qquad b = d. \tag{17}$$

This argument applies even when some indices on the same side repeat: multiset equality always admits a matching of the three occurrences.

A family of distinct ordered pairs satisfying (17) pairwise must have a coordinate common to the whole family. If some two pairs have the same first coordinate and different second coordinates, any third pair with a different first coordinate would have to agree with both distinct second coordinates, which is impossible. Otherwise, either all first coordinates agree already or a pair with distinct first coordinates forces all second coordinates to agree by the symmetric argument. For a family of one pair the assertion is immediate.

Suppose the common first coordinate is a. Then the pairs have the form (a, b_x) with distinct b_x, each adjacent to a in X. For two of them, adjacency of the corresponding points implies $g_{b_y} - g_{b_x} \in \overline{A}$. By (16), $\{b_x, b_y\}$ is an edge of X. Thus $\{a\} \cup \{b_x : x \in C \setminus \{0\}\}$ is a clique of size $|C|$. If the second coordinate is common, the same reasoning applies to the distinct first coordinates. A singleton C is realized by any vertex of X, and the empty clique is realized by the empty set. □

The preceding lemma is the known three-sum-distinct embedding from [4, 5]. Its proof is included to make the extraction of a source clique from arbitrary target phases explicit.

5 The reduction and its running time

5.1 From a graph to a cyclic pattern

For a graph X on v vertices and an integer k with $2 \le k \le v$, take the labels and D from (14), set $m = k$, and use $A_d = A$ for every nonzero $d \in \mathbb{Z}_k$ in the construction of Section 3. Let $P_X \subseteq \mathbb{Z}_N$ be the pattern in (13). Its budget is k.

Theorem 5.1. *For every finite simple undirected graph X and $2 \le k \le v$,*

$$X \text{ has a clique of size } k \iff \tau(P_X, \mathbb{Z}_N) \le k.$$

The constructed pattern is nonempty.

Proof. If $i_0, \ldots, i_{k-1}$ are the vertices of a clique, assign $x_r = g_{i_r} \in \mathbb{Z}_L$ to the k rows. The difference between any two distinct assigned phases lies in $\overline{A}$, so all constraints (3) hold. Theorem 3.4 and the additive bijection ϕ give a cover with k shifts.

Conversely, a cover within the budget yields phases satisfying (3). Every pair of distinct rows is of the form $(r, r - d)$ for a nonzero d. Thus all pairwise phase differences lie in $\overline{A}$. They are nonzero, so the phases are distinct and form a k-clique in Y. Lemma 4.2 produces a k-clique in X. Nonemptiness was proved just after (10). □

To define the reduction on all graph instances, let the fixed YES instance be the length-one pattern $S = \{0\}$ with budget 1, and the fixed NO instance the same pattern with budget 0. For $k = 0$ output YES. For $k = 1$ output YES if $v \ge 1$ and NO otherwise. For $k > v$ output NO. Use Theorem 5.1 in all remaining cases. These branches cover the empty graph and every natural k. An edgeless graph with $2 \le k \le v$ has $A = \varnothing$ and is handled by the same construction: its difference constraints are impossible. Malformed graph encodings can be sent to the fixed NO instance after an explicit validity check. All outputs in this graph reduction have a positive-length, nonempty pattern and a budget no larger than their length.

5.2 Polynomial output and an explicit emitter

Lemma 5.2. *In the nontrivial branch $2 \le k \le v$, the output length satisfies $N \le 422v^{16}$.*

Proof. Since $v \ge 1$,

$$B \le 4v^3, \qquad D = v + Bv^2 + B^2v^3 \le v + 4v^5 + 16v^9 \le 21v^9.$$

Also $h = k(k-1) \le v^2$ and $u_{h-1} < 3h^3$. Since $D \ge 1$, $M = 2D + 1 \le 3D$ and

$$H_{\max} \le 9Dh^3 \le 189v^{15}.$$

The three terms in (6) are bounded respectively by $420v^{15}$, $63v^{11}$, and $63v^9$. Consequently

$$Q \le 420v^{15}, \qquad L \le Q + k + 1 \le 422v^{15}, \qquad N = kL \le 422v^{16}.$$

□

This length bound is accompanied by an actual algorithm for writing the full indicator. Use an explicit graph representation, for example a $v \times v$ adjacency matrix with its vertex count, so that v is polynomially bounded by the source length. First check symmetry and the zero diagonal and handle the branches above. In the nontrivial branch, compute B, D, h, M, Q, L, N. For each integer $z = 0, \dots, N-1$, compute $r = z \bmod k$ and $y = z \bmod L$ and write the following bit:

$$\begin{cases} 1 \iff y \not\equiv Mu_j \pmod{L} \text{ for all } 0 \le j < h, & r = 0, \\ 1 \iff y \equiv Mu_j + g_a - g_b \pmod{L} \text{ for some } (r-1)k \le j < rk \text{ and } \{a,b\} \in E(X), & r \ne 0. \end{cases} \tag{18}$$

Equation (9) proves that this is exactly the indicator of P_X under ϕ. Finally pair the entire indicator word with the binary encoding of k. The pattern field has exactly N bits; the encoded output has length $2N + 2 + O(\log(k+1))$.

For clarity, the emitter is polynomial-time without unit-cost arithmetic or random access. There are N output iterations. A dense-row test examines at most h markers, and a sparse-row test examines at most kv^2 marker/ordered-vertex pairs. These are polynomially many tests by Lemma 5.2. Every numerical quantity involved, including a signed edge difference, has $O(\log(v+2))$ bits. Schoolbook addition, multiplication, comparison, and division with remainder have polynomial bit cost on these operands. One may implement every adjacency lookup by scanning the explicit v^2-bit matrix, and compute labels and marker values directly from their indices. Binary loop counters and sequential output therefore give a deterministic multitape algorithm with a polynomial running time. If k is supplied in binary, comparison with v and the exceptional branches take polynomial time in the length of that field; only $k \le v$ reaches the large emitter. This proves a runtime bound separately from the numerical output bound. No particular optimal exponent for the Turing-machine running time is asserted.

5.3 A complete source reduction

We recall the standard occurrence-graph reduction to specify every source witness. A 3CNF formula has clauses $C_1, \dots, C_q$, each with exactly three literal occurrences; repeated literals are permitted. Make one vertex (i, j) for each of the three occurrences j in clause i. Join two vertices exactly when they belong to different clauses and their literals are not complementary. The graph X_F has $v = 3q$ vertices.

Lemma 5.3. *The formula F is satisfiable if and only if X_F has a clique of size q.*

Proof. A satisfying assignment permits the choice of one true occurrence in each clause. Two such occurrences are never complementary, so the chosen q vertices form a clique. Conversely, a clique contains at most one vertex from each clause, because there are no within-clause edges. A q-clique consequently contains exactly one from each of the q clauses. Its selected literals are pairwise compatible. Set each variable that appears among them to the truth value making its selected sign true; pairwise compatibility guarantees that this is well defined. Give all other variables any value. Every clause contains its selected true literal, so the assignment satisfies F. When $q = 0$, the empty conjunction is true and the empty graph has the required zero-clique. □

Proof of Theorem 1.1. The NP-completeness of 3SAT is the standard source theorem, also proved in the formal foundation used here [6]. Given a valid 3CNF formula, construct X_F and use the total graph reduction with budget $k = q$. Lemma 5.3 and Theorem 5.1, together with its exceptional branches, give the required equivalence for every formula. In particular, an empty formula produces the fixed YES instance and a one-clause formula does also. Constructing the graph requires only polynomially many literal comparisons; its adjacency matrix has $(3q)^2$ bits. The full cyclic indicator is emitted in polynomial time as proved above, with $N \leq 422(3q)^{16}$ in the nontrivial branch. Source words that fail the chosen 3CNF syntax are detected in polynomial time and mapped to NO. Thus this is a total polynomial-time many-one reduction. It outputs only nonempty patterns and budgets at most their lengths, proving the stronger NP-hardness statement. Proposition 2.3 supplies NP membership both for the unrestricted language and for that restriction. □

6 Lean formalization and reproduction

The associated Lean development checks the complete decision theorem, including its computational interpretation. Its final theorem is

`CyclicCover.BinaryCover.NPComplete_language`.

The language is defined on finite Boolean lists with the pairing and binary-budget conventions in Section 2. The exact semantic correspondence is

`CyclicCover.BinaryCover.mem_encode_indicator_iff`.

It states, for every natural n with `NeZero n`, every finite set $S \subseteq \mathrm{ZMod}\, n$, and every natural b, that encoding the n-bit indicator of S and the ordinary binary numeral of b belongs to the language if and only if

$$\exists T \subseteq \mathrm{ZMod}\, n, \qquad |T| \leq b \ \wedge (\forall x,\ \exists t \in T,\ x - t \in S).$$

There is no nonempty-pattern or $b \leq n$ hypothesis. The theorem `CyclicCover.BinaryCover.mem_pair_iff` extends this interpretation to arbitrary binary budget words, including redundant high zero bits. Zero-length patterns and malformed pairs are rejected.

6.1 Correspondence with the mathematical proof

The arithmetic and set-theoretic core is organized as follows. `CyclicCover.marker_representation_unique` supplies the short-interval and scaled-difference argument of Lemma 3.2. `CyclicCover.Concrete.overlap_bound` instantiates it with the explicit marker blocks. The layer argument and counting contradiction are assembled in `CyclicCover.Concrete.concrete_cover_iff`;

the transport to $\mathbb{Z}_{mL}$ is `CyclicCover.Concrete.cyclic_cover_iff`. The repeated-summand label property and the clique extraction are checked in the label and graph modules, culminating in `CyclicCover.GraphEncoding.clique_iff_assignment`. The total graph reduction is verified by `CyclicCover.GraphEncoding.reduceGraph_correct`, and its full indicator length by `CyclicCover.GraphEncoding.reduceGraph_indicator_length`.

The word-level reduction is checked separately. In particular, `CyclicCover.GraphEncoding.GraphString.emit_mem_FP` proves that the graph-word emitter computes its entire output in polynomial time, and `CyclicCover.GraphEncoding.GraphString.emit_correct` proves its semantic equivalence on every input word. The source parser preserves individual literal occurrences and checks the exact 3CNF syntax. The theorem `CyclicCover.SourceReduction.threeSAT_reduction` composes its explicit adjacency emitter with the graph-to-cover program. The `GeneralCover` and `BinaryCover` modules then remove intermediate nonemptiness, capped-budget, and unary-budget restrictions. In particular, `CyclicCover.BinaryCover.budget_fp` certifies capped parsing of an arbitrary binary budget, and `CyclicCover.BinaryCover.fromUnary_fp` together with `CyclicCover.BinaryCover.fromUnary_correct` supplies the total binary-output reduction. The final semantics therefore cover every input pattern and every natural budget, not just outputs of the hardness gadget.

The complexity definitions come from the pinned `complexitylib` library [6]. Its FP functions are computed by finite-control deterministic multitape Turing machines with polynomial running-time bounds and exact output-tape semantics. Its NP class is nondeterministic polynomial time, requiring all computation paths to halt within the time bound. Polynomial many-one reductions quantify over all input words. The source NP-hardness is derived from the library's proved Cook–Levin and 3SAT theorems; it is not an additional axiom. NP membership uses the library's verifier-to-machine theorem with the linear-size translate indicator. The bound $422v^{16}$ is an output-size bound; the formal machine-time certificates are separate theorems.

6.2 Versions, trust, and commands

The checked environment uses Lean `v4.35.0-rc3`. The principal dependency commits are:

Dependency	Pinned commit
`complexitylib`	`add0aca8ee0d71e68969f2cb45cf91554a50f787`
`mathlib`	`728a93eeff833da3173895bb0575752fdc24edb0`
`cslib`	`311d27ad8458b61e9b7461fc83a480e4be97aef2`

The successful reproduction snapshot is identified by `20261014T124120Z`. It preserves the sources, dependency manifests, compiler and source fingerprints, compile commands, exit statuses, and axiom-audit logs. From that snapshot directory, the command

```
python3 reproduce_fresh.py
```

creates a new output directory and rebuilds the saved source sequence; an optional argument supplies a fresh destination directory. The `REPRODUCE.txt` file records the setup requirements. The expected base status is `all_compiled_audited`, followed by `compiled_and_audited=true` in the binary-extension status. The saved run compiled 307 modules, including all 24 authored modules and 283 selected foundation modules, and recorded 110 axiom queries. The final source and statement audit is saved as `PARENT_FINAL_AUDIT.json`; the manuscript-to-theorem mapping and source hashes are in `PAPER_AUDIT.json`.

The transitive axiom reports for the main theorem and semantic endpoint contain only `propext`, `Classical.choice`, and `Quot.sound`. No unfinished proof, new mathematical axiom, native evaluation certificate, or externally asserted hardness result is used. All authored, `complexitylib`, and selected `cslib` modules in the build closure were recompiled from copied source. The pinned official Lean and mathlib package caches were reused and fingerprinted; their complete sources were not rebuilt. The reproduction script checks the recorded compiler and cache locations, so a new machine must restore the pinned environment and adjust those locations as documented. The saved successful run does not assert portability to arbitrary library versions.

7 Consequences and scope

Theorem 1.1 answers the decision version of Set-Cover by Rotations. Computing the exact covering number, or producing a minimum cover, is consequently NP-hard, since either task decides the budget question. For a nonempty pattern the optimum lies between 1 and n; a decision oracle therefore determines its value by binary search. The proof does not provide a self-reduction that recovers an optimal set of shifts while keeping all oracle queries within the single-pattern cyclic problem, the distinction already raised by Wirth [2].

The reduction is intended to establish exact complexity. Its polynomial order bound is deliberately generous, and no approximation-hardness factor follows from the argument. Succinctly encoded patterns, restricted families of patterns, and improved quantitative bounds on the reduction remain separate questions.

References

[1] *Open Problems from WADS 2023*. Problem 4: Set-Cover by Rotations, contributed by Anthony Wirth, 2023. https://wadscccg2023.encs.concordia.ca/assets/pdf/wads_open_problems.pdf.

[2] Anthony Ian Wirth. *Approximation Algorithms for Clustering*. Ph.D. dissertation, Princeton University, January 2005. Section 2.7, "Pattern rotation," pp. 34–37. https://www.cs.princeton.edu/techreports/2004/716.pdf.

[3] Béla Bollobás, Svante Janson, and Oliver Riordan. *On covering by translates of a set*. arXiv:0910.3815v2, 2010. https://arxiv.org/abs/0910.3815v2.

[4] Bruno Codenotti, Ivan Gerace, and Sebastiano Vigna. Hardness results and spectral techniques for combinatorial problems on circulant graphs. *Linear Algebra and its Applications*, 285(1–3):123–142, 1998.

[5] Chris Godsil and Brendan Rooney. *Hardness of Computing Clique Number and Chromatic Number For Cayley Graphs*. arXiv:1502.00965v5, 2016. Section 6, especially Lemma 6.1. https://arxiv.org/abs/1502.00965v5.

[6] The complexitylib contributors. *Complexitylib: a Lean 4 formalization of computational complexity theory*. Source revision `add0aca8ee0d71e68969f2cb45cf91554a50f787`. https://github.com/SamuelSchlesinger/complexitylib/tree/add0aca8ee0d71e68969f2cb45cf91554a50f787.