Solving affine-clause satisfiability by maximum rank deficiency

OpenAI Codex and Yichuan Wang
(Dated: November 23, 2026)

An affine clause is a disjunction of parity equations over the binary field. We study the satisfiability of conjunctions of such clauses. For a subfamily of clauses, its rank deficiency is the number of clauses minus the rank of their coefficient vectors. Let k be the maximum deficiency over all subfamilies. We give a polynomial-space randomized algorithm that finds a satisfying assignment, when one exists, in $2^k \operatorname{poly}(L) \log(1/\epsilon)$ bit operations with probability at least $1-\epsilon$, where L is the input length. The bound allows arbitrary clause width and arbitrary parity equations. It gives polynomial time at fixed k even when the dimension of the space of literal values grows without bound. The proof reduces the formula to $r+k$ clauses of rank r and encodes their solutions in a weighted sum over the Boolean cube. A change of variables eliminates r pivot variables. The remaining coefficient can be computed by summing over the subsets of the k remaining variables, one subset at a time.

I. INTRODUCTION

A system of binary parity equations can be solved by Gaussian elimination. Suppose instead that each clause offers a choice of equations, at least one of which must hold. For instance, the clause

$$(x_1 + x_2 + x_3 = 0) \ \vee \ (x_2 + x_4 = 1)$$

allows either parity condition to satisfy it. Choosing one equation from each clause produces a linear system, but the choices in different clauses must be consistent. If the chosen coefficient vectors are linearly independent, the system is solvable for every choice of right-hand sides. This observation suggests measuring how many clauses must be omitted before such an independent choice becomes possible.

An affine literal on $x \in \mathbb{F}_2^n$ is an equation $a \cdot x = b$, with coefficient vector (or normal) $a \in \mathbb{F}_2^n$ and right-hand side $b \in \mathbb{F}_2$. An affine-clause formula is a conjunction of disjunctions of these literals.

Index the M clauses by $[M] = \{1, \dots, M\}$. For $J \subseteq [M]$, let V_J be the span over $\mathbb{F}_2$ of the coefficient vectors a of all literals in the clauses indexed by J. Define the *maximum rank deficiency* by

$$k = \max_{J \subseteq [M]} \big(|J| - \dim V_J \big). \tag{1}$$

The empty subfamily gives $k \geq 0$. A set of clauses can have a large deficiency even when other clauses raise the rank of the whole formula. Rado's theorem identifies k with the minimum number of clauses one must omit to choose independent representatives from all remaining clauses [1]. We write L for the bit length of the explicit formula, including the variable list and literal coefficients, and O^* for a bound up to factors polynomial in L.

Theorem 1. For $0 < \epsilon \leq 1/2$, there is a randomized algorithm that uses $2^k \operatorname{poly}(L) \log(1/\epsilon)$ time and space polynomial in $L + \log(1/\epsilon)$. On a satisfiable input it returns a satisfying assignment with probability at least $1-\epsilon$. Every assignment returned is correct. On all other runs the algorithm reports failure.

For fixed error probability, Theorem 1 gives polynomial time whenever $k = O(\log L)$. The proof first identifies $r+k$ clauses whose normals span a space of dimension r. Satisfying these clauses suffices because a solution extends to the remaining clauses by linear algebra. The main algorithmic step is to detect a solution on this subfamily in time exponential in k even when r is large.

A. Deficiency and subspace dimension

For ordinary Boolean literals, each normal is a coordinate vector, so $\dim V_J$ counts the distinct variables appearing in J. In this case Eq. (1) recovers the maximum deficiency of a formula in conjunctive normal form (CNF). Szeider's Theorem 4 gives a deterministic $O(2^k n^3)$ algorithm for ordinary CNF on n variables, with either a satisfying assignment or a regular resolution refutation as certificate [2]. Theorem 1 retains the 2^k dependence for arbitrary parity tests, with a one-sided randomized guarantee. At $k = 0$, satisfiability also follows from Brouwer's bound on irredundant covers by affine subspaces [3]. The falsifying sets of a minimally unsatisfiable family form such a cover, which requires at least one more clause than the rank of its normals.

Arvind and Guruswami studied affine-clause satisfiability in the equivalent form of ordinary CNF restricted to an affine subspace [4]. To see the equivalence, map an assignment x to the vector of truth values of its literal occurrences. The image is an affine subspace on which the original clauses are ordinary disjunctions. Conversely, parameterizing an affine subspace by free binary coordinates turns each ordinary literal into a parity test. For width two, Theorem 2 of Ref. [4] gives a randomized $O^*(1.5^d)$ bound in the dimension d of the prescribed subspace, as well as bounds involving ambient dimension and codimension. Their Theorem 25 applies to unbounded width when there are at most cd clauses for a fixed constant c. Indeed, a clause with literal truth bits u_j gives the equation $\prod_j (1 + u_j) = 0$ in d free coordinates. Every nonconstant irreducible factor of a nonzero product has degree one, giving a randomized bound $O^*(2^{(1-\alpha)d})$ for a constant $\alpha > 0$ depending on c.

B. Bounded deficiency at unbounded dimension

The dimension d can grow even when k is fixed. For each $i \in [n]$, take a clause with two literals, $x_i = b_i$ and an arbitrary parity test, and add three more clauses of width two. The literal values recover x from the n distinguished tests, so their affine span has dimension n. Any subfamily containing s of the first n clauses has rank at least s and contains at most $s + 3$ clauses. The full family attains deficiency three, hence $k = 3$ for every n.

The example can also force the unique subfamily of maximum deficiency to have rank n. Give one literal in the last three clauses the all-ones normal. Omitting any of the last three clauses then leaves deficiency at most two. If all three are present but an initial clause is absent, the all-ones vector is independent of the selected coordinate vectors, again giving deficiency at most two. On this family the dimension bound for width two has exponential factor 1.5^n, whereas Theorem 1 gives polynomial time with factor 2^3, even with dense parity tests.

II. REDUCTION TO A FULL-RANK CORE

A partial independent transversal selects at most one vector from each clause and requires the selected vectors to be linearly independent. By Rado's theorem its maximum size is $M - k$ [1]. It can be found in polynomial time by intersecting the partition matroid of clauses with the binary linear matroid of vectors [5]. Column-rank tests use Gaussian elimination.

Lemma 2. One can compute a subfamily $\mathcal{C}$ of $m = r+k$ clauses with $r = \dim V_{\mathcal{C}}$, and a set of r matched clauses in $\mathcal{C}$ whose representatives form a basis. Every solution expressed on this basis extends to a solution of the full formula in polynomial time.

Proof. Start the final transversal augmenting search at every unmatched clause. Within a reached clause inspect all candidate vectors. A candidate in the span of the current representatives has a unique expansion. Mark the matched clauses supplying the nonzero coefficients, and continue from them. A reached vector outside the span would terminate an augmenting path, contradicting maximality. This is the usual exchange search for a linearly independent transversal [5].

Let $\mathcal{C}$ consist of the reached clauses. Their matched representatives span every reached vector by the stopping rule, so they form a basis of $V_{\mathcal{C}}$. All k unmatched clauses were initial vertices. This gives $m = r + k$. The remaining matched representatives are independent modulo $V_{\mathcal{C}}$ because the entire matched family is independent. For any assignment to the r basis evaluations, one may therefore prescribe, in addition, that each outside representative satisfies its own equation. These prescriptions form a consistent linear system and yield the promised extension. In the reverse direction, restrict any satisfying assignment to the core. □

The evaluation of the core basis defines coordinates $y \in \mathbb{F}_2^r$, and every such y is attainable. An empty core immediately yields a solution by extension. When $r = 0$, test the constant core clauses directly. The remaining argument assumes $r > 0$.

III. A POLYNOMIAL WITNESS CRITERION

Write $u_{ij}(y) = 1 + b_{ij} + a_{ij} \cdot y$ for the truth bit of the jth literal in core clause i, using its core-coordinate normal a_{ij}. Give each occurrence its own formal variable λ_{ij}. In characteristic two define

$$Z(\lambda) = \sum_{y \in \mathbb{F}_2^r} \prod_{i=1}^{m} \left(\sum_{j} \lambda_{ij} u_{ij}(y) \right). \tag{2}$$

An empty clause makes the polynomial zero.

Lemma 3. The core is satisfiable if and only if Z is a nonzero formal polynomial.

Proof. If every y violates some clause, each summand vanishes. For the converse fix a solution y_*. In each of the r matched clauses fix one literal v true at y_*. Keep every literal true there. For each false literal u, add the form $u + v$, which is true there as well. The normals of these true forms have the same span as the original normals because the normal of u is recovered from $u + v$ and v, and every new form is a linear combination in the clause.

Each subfamily of the r matched clauses originally had rank at least its size. This remains true after the replacement, so Rado's theorem gives one true form from each matched clause with r independent normals. Their common truth set is the singleton $\{y_*\}$. Select any literal true at y_* in each remaining clause. These selections can all be realized by choosing weights in $\mathbb{F}_2$ in Eq. (2), including the two weights needed to realize a sum $u + v$. The product is now the indicator of y_*, so the sum equals one. □

Pick a binary extension field K of size q with $4(m+1) \leq q < 8(m+1)$. Choose all weights independently and uniformly in K, retaining $\mathbb{F}_2^r$ as the domain of the sum. Since Z has degree at most m, a trial misses a nonzero Z with probability at most $m/q < 1/4$ [6]. At a nonzero value, replace the weight vector of one clause by a unit vector preserving a nonzero value. Such a unit vector exists by linearity in those weights. Repeating over the clauses selects one original literal per clause. The final sum in K is the parity of the number of common Boolean solutions of the selected equations. A nonempty affine solution set has power-of-two size, so nonzero parity means exactly one solution. Solve by elimination, extend by Lemma 2, and verify all clauses. It remains to evaluate Eq. (2) within the stated resource bounds.

IV. COMPUTING THE WEIGHTED BOOLEAN SUM

For a numerical weight choice, write the ith factor as $w_{i0} + \sum_{j=1}^{r} w_{ij} y_j$. The array $W = (w_{ij})_{i\in[m],j\in[r]}$ excludes the constant column. We first derive a coefficient formula in the regular case $\operatorname{rank} W = r$, then give a polynomial-space evaluation and remove the rank hypothesis.

A. Row allocation and pivot elimination

A monomial in the y_j sums to one on $\mathbb{F}_2^r$ precisely when every y_j occurs. Each missing coordinate contributes a factor of two. Introduce one variable z_i per row and impose $z_i^2 = 0$. In this quotient let

$$P_0(z) = \prod_{i=1}^{m} (1 + w_{i0} z_i),$$
$$P_j(z) = \prod_{i=1}^{m} (1 + w_{ij} z_i) - 1 \qquad (j \in [r]). \tag{3}$$

In the product $P_0 \prod_{j=1}^{r} P_j$, each positive column claims a nonempty set of rows, while the constant column can claim any set. Assigning a row twice gives zero. Requiring every row to appear once therefore gives the exact sum

$$Z = [z_1 \cdots z_m]\, P_0(z) \prod_{j=1}^{r} P_j(z). \tag{4}$$

Brackets extract the indicated squarefree coefficient.

Reorder rows so that the first r give an invertible minor of W. Retain the names $z_1, \ldots, z_r$ for their variables and call the rest $t_1, \ldots, t_k$. Let $B = K[t_1, \ldots, t_k]/(t_1^2, \ldots, t_k^2)$. Write $P = (P_1, \ldots, P_r)^{\mathsf{T}}$ and $D = (w_{ij})_{j,i\in[r]}$, with row index j. A vector $s \in B^r$ with zero constant term and $P(s,t) = 0$ can be constructed one homogeneous degree at a time. If degrees below h have been cancelled, a correction v of degree h changes the degree-h residual by Dv. The nonlinear terms all gain at least one further degree. Invertibility of D gives a unique correction. Starting with $s = 0$ and continuing to degree k constructs the unique such s.

Lemma 4. Let J_{ji} be the partial derivative of the displayed product defining P_j with respect to pivot variable z_i, evaluated at (s,t). Then

$$Z = [t_1 \cdots t_k] P_0(s,t) \det J, \qquad J_{ji} = w_{ij} + w_{ij}^2 s_i. \tag{5}$$

Proof. Every element of B with zero constant term squares to zero, since the characteristic is two and every positive-degree monomial contains a square-zero variable. Thus $z_i \mapsto z_i + s_i$ respects the defining relations. Translation leaves the coefficient of $z_1 \cdots z_r$ invariant because no monomial can acquire a missing pivot variable and the top monomial retains coefficient one. After translation each P_j with $j > 0$ has zero pivot-constant term. To reach pivot degree r, each of these r factors must contribute one linear term, and P_0 must contribute its pivot-constant term. The coefficient is the permutation sum for J, equal to its determinant in characteristic two. Finally, $P_j(s,t) + 1 = 1$. Removing the factor $1 + w_{ij} s_i$ from this product multiplies it by the same factor, because $(1 + w_{ij} s_i)^2 = 1$. This gives the stated entries. □

B. Polynomial-space evaluation

A direct computation in B could store 2^k coefficients. Instead, solve $P(s,t) = 0$ first in the ordinary formal power series ring $K[[t_1, \ldots, t_k]]$. Denote the solution from the degree recursion by $\widehat{s}$. Its reduction to B is s by uniqueness. Set

$$Q(t) = P_0(\widehat{s}, t) \det\big(w_{ij} + w_{ij}^2 \widehat{s}_i\big)_{j,i\in[r]}.$$

This choice lifts the expression in Eq. (5). Reduction modulo the squares preserves its $t_1 \cdots t_k$ coefficient.

For a subset $S \subseteq [k]$, put $t_i = u$ when $i \in S$ and $t_i = 0$ otherwise. Write the corresponding specialization as $Q_S(u)$. Then

$$[t_1 \cdots t_k] Q = [u^k] \sum_{S\subseteq[k]} Q_S(u). \tag{6}$$

A monomial with support E is counted $2^{k-|E|}$ times, which vanishes in K unless $E = [k]$. A surviving term of degree k must have exponent one in each coordinate. This proves Eq. (6), including $k = 0$.

For each S compute $\widehat{s}$ under this specialization directly modulo u^{k+1}. At step h, evaluate the degree-h residual of P at the current section and solve with D to determine its correction. Each row product and each determinant then uses polynomially many operations on length-$(k+1)$ coefficient lists. For the determinant, the constant matrix is D and is invertible, so pivoting on entries with nonzero constant coefficient permits ordinary elimination in the truncated ring. Accumulate $[u^k]Q_S$ and reuse the same storage for the next S. Specialization takes place in ordinary series, before reduction by t_i^2, since u^2 need not vanish in $K[u]/(u^{k+1})$.

C. Singular weights and the bit bound

Numerical weights can produce a singular W, including during witness recovery. Add an indeterminate T to each diagonal entry of its first r rows, keeping the constant column fixed. Their determinant is monic of degree r in T. At most r values of T are bad, while the Boolean sum computed with the perturbed matrix is a polynomial in T of degree at most m. Evaluate at $m + 1$ distinct good values in K and interpolate to $T = 0$. The field-size choice guarantees enough points. Interpolation concerns the numerical sum only and need not preserve a particular clause interpretation at the other points.

The cardinality q is $O(m+1)$. A representation of K can be constructed in polynomial bit time by enumerating monic binary polynomials of degree $\log_2 q$ and testing irreducibility. Field elements take $O(\log(m+1))$ bits. Each evaluation costs $2^k \operatorname{poly}(L)$ bit operations and polynomial space including interpolation. Recovery tries at most the number of original core literal occurrences in additional evaluations. After $\lceil \log_4(1/\epsilon) \rceil$ independent trials, the probability of missing a satisfying assignment is at most ϵ. Each trial reuses the same polynomial workspace. This proves Theorem 1. □

Formal verification. The Lean formalization covers the witness criterion for the matched core (Lemma 3, Lean 4.19.0), the regular-pivot identity in Lemma 4, and the squarefree coefficient identity underlying Eq. (6) (Lean 4.30.0 with mathlib). The complexity bound follows from the analysis above.

Use of AI. OpenAI Codex was used to develop the algorithm and proofs and to write the reference implementation, Lean formalizations, and manuscript.

[1] R. Rado, A theorem on independence relations, Q. J. Math. **13**, 83–89 (1942).

[2] S. Szeider, Minimal unsatisfiable formulas with bounded clause-variable difference are fixed-parameter tractable, J. Comput. Syst. Sci. **69**, 656–674 (2004).

[3] A. E. Brouwer, An inequality in binary vector spaces, Discrete Math. **59**, 315–317 (1986).

[4] V. Arvind and V. Guruswami, CNF satisfiability in a subspace and related problems, in *16th International Symposium on Parameterized and Exact Computation*, LIPIcs **214**, 5:1–5:15 (2021).

[5] J. Edmonds, Matroid intersection, Ann. Discrete Math. **4**, 39–49 (1979).

[6] J. T. Schwartz, Fast probabilistic algorithms for verification of polynomial identities, J. ACM **27**, 701–717 (1980).