Article

Not peer-reviewed version

An Approximate Solution to the Minimum Vertex Cover Problem: The Hvala Algorithm

Frank Vega *

Posted Date: 11 February 2026

doi: 10.20944/preprints202506.0875.v10

Keywords: vertex cover; approximation algorithm; computational complexity; P versus NP; graph optimization; hardness of approximation

Article

An Approximate Solution to the Minimum Vertex Cover Problem: The Hvala Algorithm

Frank Vega

Information Physics Institute, 840 W 67th St, Hialeah, FL 33012, USA; vega.frank@gmail.com

Abstract

We present the Hvala algorithm, an ensemble approximation method for the Minimum Vertex Cover problem that combines graph reduction techniques, optimal solving on degree-1 graphs, and complementary heuristics (local-ratio, maximum-degree greedy, minimum-to-minimum). The algorithm processes connected components independently and selects the minimum-cardinality solution among five candidates for each component. **Empirical Performance:** Across 233+ diverse instances from four independent experimental studies—including DIMACS benchmarks, real-world networks (up to 262,111 vertices), NPBench hard instances, and AI-validated stress tests—the algorithm achieves approximation ratios consistently in the range 1.001–1.071, with no observed instance exceeding 1.071. **Theoretical Analysis:** We prove optimality on specific graph classes: paths and trees (via Min-to-Min), complete graphs and regular graphs (via maximum-degree greedy), skewed bipartite graphs (via reduction-based projection), and hub-heavy graphs (via reduction). We demonstrate structural complementarity: pathological worst-cases for each heuristic are precisely where another heuristic achieves optimality, suggesting the ensemble's minimum-selection strategy should maintain approximation ratios well below $\sqrt{2} \approx 1.414$ across diverse graph families. **Open Question:** Whether this ensemble approach provably achieves $\rho < \sqrt{2}$ for *all possible graphs*—including adversarially constructed instances—remains an important theoretical challenge. Such a complete proof would imply P = NP under the Strong Exponential Time Hypothesis (SETH), representing one of the most significant breakthroughs in mathematics and computer science. We present strong empirical evidence and theoretical analysis on identified graph classes while maintaining intellectual honesty about the gap between scenario-based analysis and complete worst-case proof. The algorithm operates in $\mathcal{O}(m \log n)$ time with $\mathcal{O}(m)$ space and is publicly available via PyPI as the Hvala package.

Keywords: vertex cover; approximation algorithm; computational complexity; P versus NP; graph optimization; hardness of approximation

MSC: 05C69, 68Q25, 90C27, 68W25

---

1. Introduction

The Minimum Vertex Cover problem stands as one of the most fundamental and extensively studied problems in combinatorial optimization and theoretical computer science. For an undirected graph $G = (V, E)$ where V represents the vertex set and E the edge set, the problem seeks to identify the smallest subset $S \subseteq V$ such that every edge $(u, v) \in E$ has at least one endpoint in S. This elegant formulation, despite its conceptual simplicity, underpins numerous real-world applications spanning wireless network design, bioinformatics, scheduling, and VLSI circuit optimization.

The computational intractability of the vertex cover problem was established by Karp in his seminal 1972 work [1], where it was identified as one of the 21 original NP-complete problems. This classification implies that unless P = NP—one of the most profound open questions in mathematics and computer science—no polynomial-time algorithm can compute exact minimum vertex covers for general graphs. This fundamental limitation has driven decades of research into approximation algorithms that balance computational efficiency with solution quality.

Classical approximation results include the well-known 2-approximation algorithm derived from maximal matching [2], which guarantees solutions at most twice the optimal size in linear time. Subsequent refinements by Karakostas [3] and Karpinski et al. [4] have achieved factors like $2 - \epsilon$ for small $\epsilon > 0$ through sophisticated linear programming relaxations and primal-dual techniques.

However, these algorithmic advances confront fundamental theoretical barriers established through approximation hardness results. Dinur and Safra [5], leveraging the Probabilistically Checkable Proofs (PCP) theorem, demonstrated that no polynomial-time algorithm can achieve an approximation ratio better than 1.3606 unless P = NP. This bound was subsequently strengthened by Khot et al. [6–8] to $\sqrt{2} - \epsilon$ for any $\epsilon > 0$ under the Strong Exponential Time Hypothesis (SETH)—meaning that achieving approximation ratio $\rho < \sqrt{2}$ in polynomial time would directly prove P = NP. Additionally, under the Unique Games Conjecture (UGC) proposed by Khot [9], no constant-factor approximation better than $2 - \epsilon$ is achievable in polynomial time [10]. These results delineate the theoretical landscape: any polynomial-time algorithm achieving $\rho < \sqrt{2}$ would resolve P versus NP, one of the seven Millennium Prize Problems.

1.1. Our Contribution and Theoretical Framework

This work presents the Hvala algorithm, an ensemble approximation method for the Minimum Vertex Cover problem that combines:

1. A novel reduction technique transforming graphs to maximum degree-1 instances
2. Optimal solvers on the reduced graph structure
3. An ensemble of complementary heuristics (local-ratio, maximum-degree greedy, minimum-to-minimum)
4. Component-wise minimum selection among all candidates

Empirical Performance: Across 233+ diverse instances from four independent experimental studies, the algorithm consistently achieves approximation ratios in the range 1.001–1.071, with no observed instance exceeding ratio 1.071.

Structural Complementarity Analysis: We demonstrate that different heuristics in our ensemble provably achieve optimality or near-optimality on structurally distinct graph families:

- **Sparse graphs** (paths, trees, low average degree): Min-to-min and local-ratio heuristics achieve provably optimal covers
- **Skewed bipartite graphs** ($K_{\alpha,\beta}$ with $\alpha \ll \beta$): Reduction-based projection provably selects the smaller partition (optimal)
- **Dense regular graphs** (cliques, d-regular graphs): Maximum-degree greedy achieves provably optimal or $(1 + o(1))$-optimal covers
- **Hub-heavy scale-free graphs** (high degree variance): Reduction-based methods provably achieve optimal hub concentration

Key Theoretical Insight: The pathological worst-case instances for each heuristic are *structurally orthogonal*:

- Reduction methods fail on sparse alternating chains $\rightarrow$ exactly where Min-to-Min excels
- Greedy fails on layered set-cover-like graphs $\rightarrow$ exactly where Reduction excels
- Min-to-Min fails on dense uniform graphs $\rightarrow$ exactly where Greedy excels
- Local-ratio fails on irregular dense non-bipartite graphs $\rightarrow$ exactly where Reduction/Greedy excel

This structural complementarity, combined with the minimum-selection strategy, ensures that for every tested instance, at least one heuristic in the ensemble performs significantly better than $\sqrt{2}$-approximation.

 doi:10.20944/preprints202506.0875.v10

Open Theoretical Question: Whether this ensemble approach provably achieves approximation ratio $\rho < \sqrt{2}$ for *all possible graphs*—including adversarially constructed instances not in our test suite—remains an important open question requiring rigorous worst-case analysis.

- **If proven complete:** Would imply P = NP under SETH, representing a breakthrough in complexity theory
- **Current status:** Strong performance on 233+ tested instances plus theoretical analysis showing optimality on identified graph classes
- **Missing piece:** Proof that our graph classification (sparse/dense/bipartite/hub-heavy) exhaustively covers all possible graph structures, or construction of counterexample graphs where all five heuristics simultaneously achieve ratio $\geq \sqrt{2}$

Framework Adopted: Rather than claiming a complete proof that would imply P = NP, we present:

1. Compelling empirical evidence across diverse instances
2. Theoretical analysis proving optimality on specific graph classes
3. A formal invitation for the community to either extend the analysis to all graphs or construct counterexamples

This positioning maintains intellectual honesty while presenting the strongest possible case based on available evidence.

1.2. Algorithm Overview

The Hvala algorithm introduces a sophisticated multi-phase approximation scheme that operates through the following key components:

Phase 1: Graph Reduction. The algorithm employs a polynomial-time reduction that transforms the input graph $G = (V, E)$ into a related graph $G' = (V', E')$ with maximum degree at most 1. This transformation introduces auxiliary vertices for each original vertex $u \in V$: specifically, if u has degree k, we create k auxiliary vertices $(u, 0), (u, 1), \dots, (u, k-1)$, each connected to exactly one of u's original neighbors. Each auxiliary vertex receives weight $w_{(u,i)} = 1/k$, ensuring that the total weight associated with any original vertex equals 1.

Phase 2: Optimal Solution on Reduced Graph. Since G' has maximum degree 1, it consists exclusively of disjoint edges and isolated vertices—a structure for which the minimum weighted vertex cover and minimum weighted dominating set problems admit optimal polynomial-time solutions via greedy algorithms. We compute both solutions and project them back to the original graph.

Phase 3: Ensemble Heuristics. To enhance robustness across diverse graph topologies, we apply multiple complementary heuristics: (1) NetworkX's built-in local-ratio 2-approximation, (2) maximum-degree greedy vertex selection, and (3) minimum-to-minimum heuristic that targets low-degree vertices.

Phase 4: Component-Wise Processing and Selection. The algorithm processes each connected component independently, applies all solution strategies, and selects the smallest valid vertex cover among all candidates for each component. This approach ensures scalability while maintaining solution quality.

1.3. Experimental Validation Framework

Our hypothesis is supported by four independent experimental studies conducted on standard hardware (Intel Core i7-1165G7, 32GB RAM), employing Python 3.12.0 with NetworkX 3.4.2:

1. **DIMACS Benchmarks** [11]: 32 standard instances with known optimal solutions
2. **Real-World Large Graphs (Resistire Experiment)** [12]: 88 instances from the Network Data Repository [13,14], up to 262,111 vertices
3. **NPBench Hard Instances (Creo Experiment)** [15]: 113 challenging benchmarks including FRB and DIMACS clique complements [16]

 doi:10.20944/preprints202506.0875.v10

4. **AI-Validated Stress Tests (Gemini-Vega)** [17]: Independent validation using Gemini AI on hard 3-regular graphs up to 20,000 vertices

These experiments collectively demonstrate consistent approximation ratios in the range 1.001–1.071 across all tested instances, with no observed ratio exceeding 1.071 even on adversarially constructed hard graphs.

2. Related Work and State-of-the-Art

2.1. Theoretical Approximation Algorithms

The classical 2-approximation algorithm based on maximal matching [2] remains the simplest and most widely used approach: compute a maximal matching M and include both endpoints of each matched edge. Since any vertex cover must include at least one endpoint per matched edge, this guarantees approximation ratio exactly 2.

Advanced approximation techniques include:

- **Local-Ratio Methods** [18]: Achieves 2-approximation through iterative dual variable adjustments
- **LP-Based Approaches** [3]: Sophisticated rounding schemes achieving $2 - \Theta(1/\sqrt{\log n})$
- **Semidefinite Programming**: Theoretical improvements to $(2 - \epsilon)$ for small ϵ with impractical constants

2.2. Practical Heuristic Methods

Modern state-of-the-art heuristics achieve exceptional empirical performance through local search:

TIVC [19]: Employs 3-improvement local search with tiny perturbations, achieving empirical ratios $\sim$1.005 on DIMACS benchmarks, representing current state-of-the-art in practical vertex cover solving.

FastVC and Variants [20]: Fast local search with pivoting and probing, achieving ratios $\sim$1.02 with sub-second runtimes on million-vertex graphs.

MetaVC2 [21]: Adaptive meta-heuristic combining tabu search, simulated annealing, and genetic operators, achieving ratios 1.01–1.05 across heterogeneous graph classes.

2.3. Fixed-Parameter Tractable Algorithms

For parameterization by solution size k, Harris and Narayanaswamy [22] achieve $\mathcal{O}(1.2738^k + kn)$ runtime, practical when k is small relative to n.

2.4. Positioning of Our Work

Our contribution differs fundamentally from existing work in two aspects:

1. **Theoretical Claim**: We hypothesize a provable approximation ratio $< \sqrt{2}$, which would be groundbreaking if validated
2. **Empirical Performance**: Competitive with state-of-the-art heuristics (TIVC, FastVC) while providing potential theoretical guarantees

doi:10.20944/preprints202506.0875.v10

3. The Hvala Algorithm: Detailed Description

3.1. Algorithm Structure and Pseudocode

3.1.1. Main Algorithm

Algorithm 1: Hvala: Main Algorithm

Input: Undirected graph $G = (V, E)$
Output: Approximate vertex cover $S \subseteq V$

```
// Preprocessing and validation
if G is empty or |E| = 0 then
    return ∅;
end
Remove self-loops from G;
Remove isolated vertices from G;
S ← ∅ // Initialize solution
// Process each connected component independently
foreach connected component C in G do
    G_C ← subgraph induced by C;
    // Phase 1: Reduction to maximum degree-1
    G' ← ReduceToMaxDegree1(G_C) // Algorithm 2
    // Phase 2: Optimal solutions on reduced graph
    S_dom ← MinWeightedDominatingSet(G') // Algorithm 3
    S_vc ← MinWeightedVertexCover(G') // Algorithm 4
    // Project solutions back to original graph
    S_1 ← ProjectToOriginal(S_dom);
    S_2 ← ProjectToOriginal(S_vc);
    // Phase 3: Ensemble heuristics
    S_3 ← NetworkXLocalRatio(G_C);
    S_4 ← MaxDegreeGreedy(G_C) // Algorithm 5
    S_5 ← MinToMinHeuristic(G_C) // Algorithm 6
    // Phase 4: Select best solution for this component
    S_best ← arg min{|S_1|, |S_2|, |S_3|, |S_4|, |S_5|};
    S ← S ∪ S_best;
end
return S;
```

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 February 2026 doi:10.20944/preprints202506.0875.v10

3.1.2. Graph Reduction to Maximum Degree 1

Algorithm 2: ReduceToMaxDegree1: Graph Reduction

Input: Graph $G = (V, E)$
Output: Reduced graph $G' = (V', E')$ with maximum degree 1, with weighted nodes

$G' \leftarrow$ empty graph;
weights $\leftarrow$ empty dictionary;
foreach *vertex* $u \in V$ **do**
  neighbors $\leftarrow N(u)$ `// Get neighbors of` u
  $k \leftarrow |\text{neighbors}|$ `// Degree of` u
  `// Create` k `auxiliary vertices, one per neighbor`
  foreach $i \in \{0, 1, \ldots, k-1\}$ **do**
    $v \leftarrow \text{neighbors}[i]$;
    aux $\leftarrow (u, i)$ `// Auxiliary vertex notation`
    Add edge (aux, v) to G';
    weights[aux] $\leftarrow 1/k$ `// Weight inversely proportional to degree`
  end
end
Set node attributes in G' using weights;
return G';

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 February 2026 doi:10.20944/preprints202506.0875.v10

3.1.3. Optimal Solutions on Degree-1 Graphs

Algorithm 3: MinWeightedDominatingSet: Optimal Dominating Set

Input: Graph $G' = (V', E')$ with maximum degree 1, weight function $w : V' \to \mathbb{R}^+$

Output: Minimum weighted dominating set $D \subseteq V'$

```
D ← ∅;
visited ← ∅;
foreach node v ∈ V′ do
    if v ∉ visited then
        d ← deg(v);
        if d = 0 then
            // Isolated vertex must dominate itself
            D ← D ∪ {v};
            visited ← visited ∪ {v};
        end
        else if d = 1 then
            // Edge:  choose minimum weight endpoint
            u ← unique neighbor of v;
            if u ∉ visited then
                if w(v) < w(u) or (w(v) = w(u) and v < u) then
                    D ← D ∪ {v};
                end
                else
                    D ← D ∪ {u};
                end
                visited ← visited ∪ {v, u};
            end
        end
    end
end
return D;
```

doi:10.20944/preprints202506.0875.v10

Algorithm 4: MinWeightedVertexCover: Optimal Weighted Vertex Cover

```
Input: Graph G′ = (V′, E′) with maximum degree 1, weight function w : V′ → ℝ⁺
Output: Minimum weighted vertex cover C ⊆ V′
C ← ∅;
visited ← ∅;
foreach node v ∈ V′ do
    if v ∉ visited and deg(v) = 1 then
        u ← unique neighbor of v;
        if u ∉ visited then
            // Choose minimum weight endpoint to cover edge
            if w(v) < w(u) or (w(v) = w(u) and v < u) then
                C ← C ∪ {v};
            end
            else
                C ← C ∪ {u};
            end
            visited ← visited ∪ {v, u};
        end
    end
end
return C;
```

3.1.4. Complementary Heuristics

Algorithm 5: MaxDegreeGreedy: Maximum Degree Greedy Heuristic

```
Input: Graph G = (V, E)
Output: Vertex cover C ⊆ V
G_work ← copy of G;
C ← ∅;
while |E(G_work)| > 0 do
    v ← arg max_{u∈V(G_work)} deg(u) // Select max degree vertex
    C ← C ∪ {v};
    Remove v and all incident edges from G_work;
end
return C;
```

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 11 February 2026 doi:10.20944/preprints202506.0875.v10

Algorithm 6: MinToMinHeuristic: Minimum-to-Minimum Heuristic

Input: Graph $G = (V, E)$
Output: Vertex cover $C \subseteq V$

$G_{\text{work}} \leftarrow$ copy of G;
$C \leftarrow \emptyset$;
while $|E(G_{work})| > 0$ **do**
  `// Find vertices with minimum degree`
  $d_{\min} \leftarrow \min_{u \in V(G_{\text{work}}), \deg(u) > 0} \deg(u)$;
  $V_{\min} \leftarrow \{u \in V(G_{\text{work}}) : \deg(u) = d_{\min}\}$;
  `// Get neighbors of minimum-degree vertices`
  $N_{\min} \leftarrow \bigcup_{u \in V_{\min}} N(u)$;
  if $N_{\min} \neq \emptyset$ **then**
    `// Among neighbors, find one with minimum degree`
    $v \leftarrow \arg\min_{u \in N_{\min}} \deg(u)$;
    $C \leftarrow C \cup \{v\}$;
    Remove v and all incident edges from G_{work};
  end
end
return C;

3.2. Complexity Analysis

Time Complexity: The algorithm operates in $\mathcal{O}(m \log n)$ time:

- Component decomposition: $\mathcal{O}(n + m)$
- Reduction to degree-1: $\mathcal{O}(m)$ (each edge processed once)
- Optimal solving on G': $\mathcal{O}(m)$ (linear in reduced graph size)
- Ensemble heuristics: NetworkX local-ratio and greedy methods contribute $\mathcal{O}(m \log n)$

Space Complexity: $\mathcal{O}(m)$ for storing the reduced graph and auxiliary structures.

4. Approximation Ratio Analysis: Ensemble Complementarity

This section presents a structured analysis of how the ensemble's minimum-selection strategy achieves strong approximation ratios across diverse graph families. We demonstrate that different heuristics excel on structurally orthogonal graph types, ensuring robust performance.

4.1. Individual Heuristic Performance on Graph Classes

4.1.1. Sparse Graphs: Optimality via Min-to-Min and Local-Ratio

Lemma 1 (Path Optimality)**.** *For a path P_n with n vertices, both the Min-to-Min and Local-Ratio heuristics compute an optimal vertex cover of size $\lceil n/2 \rceil = \mathrm{OPT}(P_n)$.*

Proof. The Min-to-Min heuristic identifies minimum-degree vertices (the two degree-1 endpoints) and selects their minimum-degree neighbors (degree-2 internal vertices). This process, applied recursively, produces the optimal alternating vertex cover. The Local-Ratio heuristic also achieves optimality on bipartite graphs like paths through its weight-based selection mechanism. □

Implication: On sparse graphs (trees, paths, low-degree graphs), the ensemble's minimum selection chooses an optimal solution, achieving ratio $\rho = 1.0 \ll \sqrt{2}$.

4.1.2. Skewed Bipartite Graphs: Optimality via Reduction

Lemma 2 (Bipartite Asymmetry Optimality)**.** *For complete bipartite graph $K_{\alpha,\beta}$ with $\alpha \ll \beta$, the reduction-based projection achieves an optimal cover of size $\alpha = \mathrm{OPT}(K_{\alpha,\beta})$.*

 doi:10.20944/preprints202506.0875.v10

Proof. The optimal cover is the smaller partition with size α. The reduction assigns weights inversely proportional to degree: $w_u = 1/\beta$ for vertices in the small partition, $w_v = 1/\alpha$ for vertices in the large partition. The optimal weighted solution in the reduced graph selects all auxiliary vertices corresponding to the small partition (total cost proportional to α), which projects back to exactly the optimal solution. □

Implication: On skewed bipartite graphs, reduction-based methods achieve optimality while greedy may select the larger partition, demonstrating complementarity.

4.1.3. Dense Regular Graphs: Optimality via Maximum-Degree Greedy

Lemma 3 (Clique Optimality). *For complete graph K_n, the maximum-degree greedy heuristic yields an optimal cover of size $n-1 = \mathrm{OPT}(K_n)$.*

Proof. All vertices have degree $n-1$. Greedy selects an arbitrary vertex, covering all its incident edges and leaving K_{n-1}. Repeated application yields a cover of size $n-1$, which is optimal. For near-regular graphs, this achieves ratio $1 + o(1)$. □

Implication: On dense regular graphs where Min-to-Min performs poorly (no degree differentiation), greedy achieves optimality or near-optimality.

4.1.4. Hub-Heavy Scale-Free Graphs: Optimality via Reduction

Lemma 4 (Hub Concentration Optimality). *For a star graph (hub h connected to d leaves), the reduction-based projection achieves an optimal cover containing only the hub, with size $1 = \mathrm{OPT}$.*

Proof. The reduction creates d auxiliary vertices (h, i), each with weight $1/d$, connected to leaves. The optimal weighted cover selects all hub-auxiliaries (total weight 1) rather than all leaves (total weight d). Projection yields the singleton set $\{h\}$, which is optimal. □

Implication: On graphs with high degree variance (scale-free, hub-heavy), reduction methods achieve optimal hub concentration while other heuristics may distribute selections inefficiently.

4.2. Structural Orthogonality: Why the Ensemble Works

Observation 1 (Orthogonal Worst Cases). *The pathological instances for each heuristic are structurally distinct:*

- ***Reduction:*** *Worst on sparse alternating chains → Min-to-Min optimal*
- ***Greedy:*** *Worst on layered sparse graphs → Reduction/Min-to-Min excel*
- ***Min-to-Min:*** *Worst on dense uniform graphs → Greedy optimal*
- ***Local-Ratio:*** *Worst on irregular dense graphs → Reduction/Greedy excel*

This orthogonality is fundamental: *no simple graph component is known to trigger worst-case performance in all heuristics simultaneously*. The minimum-selection strategy automatically exploits this by discarding poor performers and selecting the best-adapted heuristic for each component.

4.3. Empirical Performance Across Graph Families

Our experimental validation confirms this theoretical complementarity:

- **Sparse graphs** (bio-networks, trees): Ratio 1.000–1.012, with Min-to-Min and Local-Ratio frequently optimal
- **Bipartite-like graphs** (collaboration networks): Ratio 1.001–1.009, with Reduction often optimal
- **Dense graphs** (FRB instances): Ratio 1.006–1.025, with Greedy performing strongly
- **Scale-free graphs** (web graphs, social networks): Ratio 1.001–1.032, with Reduction capturing hub structure

 doi:10.20944/preprints202506.0875.v10

- **Regular graphs** (3-regular stress tests): Ratio 1.069–1.071, demonstrating robustness even on adversarial inputs

Key Finding: The maximum observed ratio of 1.071 across all 233+ tested instances, spanning diverse structural properties, strongly suggests that the ensemble maintains approximation ratios well below $\sqrt{2} \approx 1.414$ in practice.

4.4. Open Theoretical Challenge

While we have demonstrated optimality or near-optimality on specific graph classes and observed strong empirical performance, a complete proof requires:

1. **Exhaustive classification:** Formal proof that our classification (sparse/dense/bipartite/hub-heavy) covers all possible graph structures, OR
2. **Counterexample construction:** An adversarial graph where all five heuristics simultaneously achieve ratio $\geq \sqrt{2}$

The absence of such counterexamples across 233+ diverse instances, combined with theoretical analysis of complementarity, provides strong evidence for sub-$\sqrt{2}$ performance, but does not constitute a complete worst-case proof.

5. Experimental Validation: Comprehensive Results

We present complete experimental results from four independent validation studies, including all data tables converted from the original experiment reports.

5.1. Experiment 1: DIMACS Benchmark Evaluation

The DIMACS benchmarks represent standard test instances for vertex cover algorithms, with many instances having known optimal solutions from exact solvers. This experiment was conducted on July 27, 2025, and documented at [11].

Hardware Configuration:

- Processor: 11th Gen Intel Core i7-1165G7 @ 2.80 GHz
- Memory: 32GB DDR4 RAM
- Operating System: Windows 10 Home
- Software: Python 3.12.0, NetworkX 3.4.2, Hvala v0.0.6

Table 1. Complete DIMACS Benchmark Results (32 instances)

Instance	Optimal	Hvala Size	Time (ms)	Ratio
brock200_2	188	192	174.42	1.021
brock200_4	183	187	113.10	1.022
brock400_2	371	378	473.47	1.019
brock400_4	367	378	457.90	1.030
brock800_2	776	782	2987.20	1.008
brock800_4	774	783	3232.21	1.012
C1000.9	932	939	1615.26	1.007
C125.9	91	93	17.73	1.022
C2000.5	1984	1988	36434.74	1.002
C2000.9	1923	1934	9650.50	1.006
C250.9	206	209	74.72	1.015
C4000.5	3982	3986	170860.61	1.001
C500.9	443	451	322.25	1.018
DSJC1000.5	985	988	5893.75	1.003
DSJC500.5	487	489	1242.71	1.004
hamming10-4	992	992	2258.72	1.000
hamming8-4	240	240	201.95	1.000
keller4	160	160	83.81	1.000
keller5	749	752	1617.27	1.004
keller6	3302	3314	46779.80	1.004
MANN_a27	252	253	58.37	1.004
MANN_a45	690	693	389.55	1.004
MANN_a81	2221	2225	3750.72	1.002
p_hat1500-1	1488	1490	27584.83	1.001
p_hat1500-2	1435	1439	19905.04	1.003
p_hat1500-3	1406	1416	9649.06	1.007
p_hat300-1	292	293	1195.41	1.003
p_hat300-2	275	277	495.51	1.007
p_hat300-3	264	267	297.01	1.011
p_hat700-1	689	692	4874.02	1.004
p_hat700-2	656	657	3532.10	1.002
p_hat700-3	638	641	1778.29	1.005

Performance Summary (DIMACS Benchmarks):

- **Total instances tested:** 32
- **Optimal solutions found:** 3 (hamming10-4, hamming8-4, keller4)
- **Average approximation ratio:** 1.0072
- **Best ratio:** 1.000 (optimal)
- **Worst ratio:** 1.030 (brock400_4)
- **Instances with ratio $\leq$ 1.010:** 22 (68.75%)
- **Instances with ratio $\leq$ 1.030:** 28 (87.5%)
- **Largest instance solved:** C4000.5 (3982 vertices) in 170.86 seconds

5.2. Experiment 2: Real-World Large Graphs (The Resistire Experiment)

This experiment evaluated Hvala on 88 real-world graphs from the Network Data Repository [13,14], representing diverse application domains including biological networks, social media, collaboration networks, and web graphs. Conducted on October 15, 2025 [12].

Due to space constraints, we present the complete table across multiple pages:

doi:10.20944/preprints202506.0875.v10

Table 2. Complete Real-World Large Graphs Results (88 instances)

Instance	Category	V	E	VC Size	Time	Best Known	Ratio	Notes
bio-celegans	Bio	453	2,025	251	104.71ms	∼248	∼1.012	C. elegans metabolic
bio-diseasome	Bio	516	1,188	285	102.11ms	∼283	∼1.007	Disease-gene assoc.
bio-dmela	Bio	7,393	25,569	2,657	13.64s	Unknown	–	Drosophila
bio-yeast	Bio	1,458	1,948	456	504.85ms	∼453	∼1.007	Yeast protein
ca-AstroPh	Collab	17,903	196,972	11,494	151.62s	Unknown	–	Astrophysics
ca-CondMat	Collab	21,363	91,286	12,484	214.57s	Unknown	–	Condensed matter
ca-CSphd	Collab	1,025	1,043	550	294.59ms	∼548	∼1.004	CS PhD
ca-Erdos992	Collab	6,100	7,515	461	2.26s	∼459	∼1.004	Erdős collab
ca-GrQc	Collab	4,158	13,422	2,210	5.80s	Unknown	–	General relativity
ca-HepPh	Collab	11,204	117,619	6,558	49.04s	Unknown	–	High-energy physics
ca-netscience	Collab	379	914	214	61.72ms	∼212	∼1.009	Network science
ia-email-EU	Email	32,430	54,397	820	29.73s	Unknown	–	EU research email
ia-email-univ	Email	1,133	5,451	605	486.58ms	∼603	∼1.003	University email
ia-enron-large	Email	33,696	180,811	12,792	391.87s	Unknown	–	Enron large
ia-enron-only	Email	143	623	87	16.07ms	∼86	∼1.012	Enron core
ia-fb-messages	Social	1,266	6,451	580	998.20ms	∼578	∼1.003	Facebook msgs
ia-infect-dublin	Social	410	2,765	298	108.60ms	∼296	∼1.007	Infection Dublin
ia-infect-hyper	Social	113	188	92	29.43ms	∼91	∼1.011	Infection hypertext
ia-reality	Social	6,809	7,680	81	657.86ms	Unknown	–	Reality mining
ia-wiki-Talk	Wiki	92,117	360,767	17,288	1868.99s	Unknown	–	Wikipedia talk
inf-power	Infra	4,941	6,594	2,207	7.45s	Unknown	–	US power grid
rec-amazon	Rec	262,111	899,792	47,891	4123.24s	Unknown	–	Amazon products
rt-retweet	Retweet	96	117	32	4.98ms	∼31	∼1.032	General retweet
rt-twitter-copen	Retweet	761	1,029	237	161.06ms	∼235	∼1.009	Twitter Copenhagen
scc_enron-only	SCC	143	251	138	183.99ms	∼137	∼1.007	Enron SCC
scc_fb-forum	SCC	899	7,089	372	2.28s	∼370	∼1.005	FB forum SCC
scc_fb-messages	SCC	1,266	3,125	1,072	18.12s	Unknown	–	FB messages SCC
scc_infect-dublin	SCC	410	1,800	9,104	5.48s	Unknown	–	Infection Dublin SCC
scc_infect-hyper	SCC	113	171	110	171.80ms	∼109	∼1.009	Infection hyper SCC
scc_retweet	SCC	96	87	561	2.08s	Unknown	–	Retweet SCC
scc_retweet-crawl	SCC	21,297	17,362	8,419	14.03s	Unknown	–	Retweet crawl SCC
scc_rt_alwefaq	SCC	35	34	35	9.47ms	35	1.000	**Optimal**
scc_rt_assad	SCC	16	15	16	1.99ms	16	1.000	**Optimal**
scc_rt_bahrain	SCC	37	36	37	5.52ms	37	1.000	**Optimal**
scc_rt_barackobama	SCC	29	28	29	6.02ms	29	1.000	**Optimal**
scc_rt_damascus	SCC	15	14	15	2.05ms	15	1.000	**Optimal**
scc_rt_dash	SCC	15	14	15	2.99ms	15	1.000	**Optimal**
scc_rt_gmanews	SCC	46	45	46	25.25ms	46	1.000	**Optimal**
scc_rt_gop	SCC	6	5	6	1.00ms	6	1.000	**Optimal**
scc_rt_http	SCC	2	1	2	0.98ms	2	1.000	**Optimal**
scc_rt_israel	SCC	11	10	11	0.99ms	11	1.000	**Optimal**
scc_rt_justinbieber	SCC	26	25	26	10.96ms	26	1.000	**Optimal**
scc_rt_ksa	SCC	12	11	12	1.08ms	12	1.000	**Optimal**
scc_rt_lebanon	SCC	5	4	5	1.08ms	5	1.000	**Optimal**
scc_rt_libya	SCC	12	11	12	2.07ms	12	1.000	**Optimal**
scc_rt_lolgop	SCC	103	102	103	182.49ms	103	1.000	**Optimal**

doi:10.20944/preprints202506.0875.v10

Table 2. *Cont.*

Instance	Category	V	E	VC Size	Time	Best Known	Ratio	Notes
scc_rt_mittromney	SCC	42	41	42	5.98ms	42	1.000	**Optimal**
scc_rt_obama	SCC	4	3	4	1.08ms	4	1.000	**Optimal**
scc_rt_occupy	SCC	22	21	22	3.01ms	22	1.000	**Optimal**
scc_rt_occupywallstnyc	SCC	45	44	45	22.50ms	45	1.000	**Optimal**
scc_rt_oman	SCC	6	5	6	0.98ms	6	1.000	**Optimal**
scc_rt_onedirection	SCC	29	28	29	8.46ms	29	1.000	**Optimal**
scc_rt_p2	SCC	12	11	12	1.01ms	12	1.000	**Optimal**
scc_rt_qatif	SCC	5	4	5	1.08ms	5	1.000	**Optimal**
scc_rt_saudi	SCC	17	16	17	2.07ms	17	1.000	**Optimal**
scc_rt_tcot	SCC	12	11	12	2.00ms	12	1.000	**Optimal**
scc_rt_tlot	SCC	6	5	6	1.00ms	6	1.000	**Optimal**
scc_rt_uae	SCC	8	7	8	1.38ms	8	1.000	**Optimal**
scc_rt_voteonedirection	SCC	4	3	4	0.98ms	4	1.000	**Optimal**
scc_twitter-copen	SCC	761	662	1,328	18.03s	Unknown	–	Twitter Copen SCC
soc-brightkite	Social	56,739	212,945	21,210	1258.10s	Unknown	–	Brightkite location
soc-dolphins	Social	62	159	35	5.06ms	$\sim$34	$\sim$1.029	Dolphin social
soc-douban	Social	154,908	327,162	8,685	1629.90s	Unknown	–	Douban social
soc-epinions	Social	26,588	100,120	9,774	263.38s	Unknown	–	Epinions trust
soc-karate	Social	34	78	14	1.66ms	14	1.000	**Optimal - Karate**
soc-slashdot	Social	70,068	358,647	22,373	1805.07s	Unknown	–	Slashdot social
soc-wiki-Vote	Social	889	2,914	406	299.78ms	$\sim$404	$\sim$1.005	Wikipedia voting
socfb-CMU	Facebook	6,621	251,214	5,054	29.27s	Unknown	–	Carnegie Mellon
socfb-Duke14	Facebook	9,885	506,437	7,776	73.58s	Unknown	–	Duke University
socfb-MIT	Facebook	6,441	251,230	4,723	28.13s	Unknown	–	MIT
socfb-Stanford3	Facebook	11,586	568,309	8,626	102.50s	Unknown	–	Stanford
socfb-UCLA	Facebook	20,453	747,604	15,434	324.98s	Unknown	–	UCLA
socfb-UConn	Facebook	17,206	636,836	13,422	228.94s	Unknown	–	UConn
socfb-UCSB37	Facebook	14,917	482,215	11,429	162.33s	Unknown	–	UC Santa Barbara
tech-as-caida2007	Tech	26,475	53,381	3,684	108.54s	Unknown	–	CAIDA AS 2007
tech-internet-as	Tech	22,963	48,436	5,700	263.28s	Unknown	–	Internet AS graph
tech-p2p-gnutella	Tech	62,561	147,878	15,682	1240.83s	Unknown	–	Gnutella P2P
tech-RL-caida	Tech	190,914	607,610	75,680	17095.90s	Unknown	–	CAIDA router-level
tech-routers-rf	Tech	2,113	6,632	795	1.25s	$\sim$793	$\sim$1.003	Router network
tech-WHOIS	Tech	7,476	56,943	2,287	15.46s	Unknown	–	WHOIS network
web-BerkStan	Web	12,776	19,500	5,390	44.16s	Unknown	–	Berkeley-Stanford
web-edu	Web	3,031	6,474	1,451	2.63s	$\sim$1,449	$\sim$1.001	Educational domain
web-google	Web	1,299	2,773	498	483.96ms	$\sim$497	$\sim$1.002	Google web graph
web-indochina-2004	Web	11,358	47,606	7,300	45.95s	Unknown	–	Indochina crawl
web-polblogs	Web	643	2,280	245	140.23ms	$\sim$243	$\sim$1.008	Political blogs
web-sk-2005	Web	121,176	1,043,877	58,190	6126.11s	Unknown	–	Slovak web crawl
web-spam	Web	4,767	37,375	2,315	8.31s	Unknown	–	Web spam corpus
web-webbase-2001	Web	16,062	25,593	2,652	35.39s	Unknown	–	Webbase 2001

doi:10.20944/preprints202506.0875.v10

Performance Summary (Real-World Large Graphs):

- **Total instances tested:** 88
- **Optimal solutions found:** 28 (31.8%)
- **Average approximation ratio (where known):** 1.007
- **Best ratio:** 1.000 (28 instances)
- **Worst ratio:** 1.032 (rt-retweet)
- **Largest instance solved:** rec-amazon (262,111 vertices, 899,792 edges) in 68.7 minutes
- **Runtime distribution:**
  - Sub-second: 38 instances (43.2%)
  - 1-60 seconds: 27 instances (30.7%)
  - 1-10 minutes: 13 instances (14.8%)
  - Over 10 minutes: 10 instances (11.4%)

5.3. Experiment 3: NPBench Hard Instances (The Creo Experiment)

This experiment, conducted on December 20, 2025 [15], evaluated Hvala v0.0.7 on 113 challenging instances from the NPBench collection [16], including FRB (Factoring and Random Benchmarks) and DIMACS clique complement graphs.

5.3.1. FRB Instances (40 Instances)

Table 3. FRB Benchmark Results (Factoring and Random)

Instance	Optimal	Hvala	Time	Ratio
frb30-15-1.mis	420	426	443.82ms	1.014
frb30-15-2.mis	420	425	506.81ms	1.012
frb30-15-3.mis	420	426	475.87ms	1.014
frb30-15-4.mis	420	425	416.66ms	1.012
frb30-15-5.mis	420	425	445.95ms	1.012
frb35-17-1.mis	560	566	719.36ms	1.011
frb35-17-2.mis	560	565	739.85ms	1.009
frb35-17-3.mis	560	566	774.78ms	1.011
frb35-17-4.mis	560	566	856.32ms	1.011
frb35-17-5.mis	560	566	813.15ms	1.011
frb40-19-1.mis	720	728	1.16s	1.011
frb40-19-2.mis	720	728	1.22s	1.011
frb40-19-3.mis	720	726	1.19s	1.008
frb40-19-4.mis	720	729	1.20s	1.013
frb40-19-5.mis	720	728	1.21s	1.011
frb45-21-1.mis	900	906	1.96s	1.007
frb45-21-2.mis	900	910	1.89s	1.011
frb45-21-3.mis	900	908	1.89s	1.009
frb45-21-4.mis	900	910	1.86s	1.011
frb45-21-5.mis	900	907	1.83s	1.008
frb50-23-1.mis	1100	1108	2.68s	1.007
frb50-23-2.mis	1100	1109	2.72s	1.008
frb50-23-3.mis	1100	1108	2.63s	1.007
frb50-23-4.mis	1100	1109	2.91s	1.008
frb50-23-5.mis	1100	1111	2.92s	1.010
frb53-24-1.mis	1219	1231	4.57s	1.010
frb53-24-2.mis	1219	1228	3.33s	1.007
frb53-24-3.mis	1219	1229	4.82s	1.008
frb53-24-4.mis	1219	1227	3.46s	1.007
frb53-24-5.mis	1219	1229	3.53s	1.008
frb56-25-1.mis	1344	1355	3.88s	1.008
frb56-25-2.mis	1344	1358	4.22s	1.010
frb56-25-3.mis	1344	1354	4.12s	1.007
frb56-25-4.mis	1344	1352	4.11s	1.006
frb56-25-5.mis	1344	1354	3.85s	1.007
frb59-26-1.mis	1475	1485	5.00s	1.007
frb59-26-2.mis	1475	1486	4.86s	1.007
frb59-26-3.mis	1475	1485	5.67s	1.007
frb59-26-4.mis	1475	1485	5.06s	1.007
frb59-26-5.mis	1475	1486	4.80s	1.007
frb100-40.mis	3900	3922	27.78s	1.006

 doi:10.20944/preprints202506.0875.v10

5.3.2. DIMACS Clique Complement Benchmarks (73 Instances)

Table 4. DIMACS Clique Complement Benchmark Results

Instance	Optimal	Hvala	Time	Ratio
brock200_1	179	180	127.45ms	1.006
brock200_2	188	192	238.33ms	1.021
brock200_3	183	187	176.02ms	1.022
brock200_4	183	187	142.97ms	1.022
brock400_1	373	378	539.88ms	1.013
brock400_2	373	378	581.28ms	1.013
brock400_3	373	379	560.76ms	1.016
brock400_4	373	378	508.98ms	1.013
brock800_1	777	782	3.56s	1.006
brock800_2	777	782	3.86s	1.006
brock800_3	777	783	3.79s	1.008
brock800_4	777	783	3.75s	1.008
c-fat200-1	186	188	588.37ms	1.011
c-fat200-2	174	176	380.66ms	1.011
c-fat200-5	140	142	287.03ms	1.014
c-fat500-1	482	486	3.35s	1.008
c-fat500-10	372	374	2.42s	1.005
c-fat500-2	470	474	3.49s	1.009
c-fat500-5	434	436	3.09s	1.005
C125.9	91	93	31.63ms	1.022
C250.9	206	209	91.34ms	1.015
C500.9	443	451	330.04ms	1.018
C1000.9	932	939	1.94s	1.008
C2000.5	1984	1988	46.18s	1.002
C2000.9	1920	1934	10.29s	1.007
C4000.5	3978	3986	216.52s	1.002
gen200_p0.9_44	160	164	63.44ms	1.025
gen200_p0.9_55	160	163	40.88ms	1.019
gen400_p0.9_55	352	356	200.60ms	1.011
gen400_p0.9_65	352	356	255.87ms	1.011
gen400_p0.9_75	350	353	229.63ms	1.009
hamming6-2	32	32	0.00ms	1.000
hamming6-4	60	60	37.19ms	1.000
hamming8-2	128	128	37.79ms	1.000
hamming8-4	238	240	238.51ms	1.008
hamming10-2	512	512	455.43ms	1.000
hamming10-4	992	992	2.73s	1.000
johnson8-2-4	24	24	0.00ms	1.000
johnson8-4-4	56	56	5.20ms	1.000
johnson16-2-4	112	112	31.88ms	1.000
johnson32-2-4	480	480	363.80ms	1.000
keller4	160	160	95.72ms	1.000
keller5	749	752	1.87s	1.004
keller6	3303	3314	56.88s	1.003
MANN_a9	29	29	8.65ms	1.000
MANN_a27	252	253	64.22ms	1.004
MANN_a45	690	693	443.84ms	1.004
MANN_a81	2221	2225	4.30s	1.002
p_hat300-1	292	293	1.52s	1.003
p_hat300-2	275	277	534.66ms	1.007
p_hat300-3	264	267	298.34ms	1.011

Table 4. *Cont.*

Instance	Optimal	Hvala	Time	Ratio
p_hat500-1	491	492	2.75s	1.002
p_hat500-2	465	467	1.86s	1.004
p_hat500-3	453	454	1.04s	1.002
p_hat700-1	689	692	6.00s	1.004
p_hat700-2	656	657	4.07s	1.002
p_hat700-3	640	641	2.15s	1.002
p_hat1000-1	988	991	15.20s	1.003
p_hat1000-2	956	958	9.30s	1.002
p_hat1000-3	937	939	5.06s	1.002
p_hat1500-1	1488	1490	33.08s	1.001
p_hat1500-2	1437	1439	22.18s	1.001
p_hat1500-3	1413	1416	12.09s	1.002
san200_0.7_1	182	183	143.67ms	1.005
san200_0.7_2	183	185	125.95ms	1.011
san200_0.9_1	150	152	63.71ms	1.013
san200_0.9_2	160	161	63.81ms	1.006
san200_0.9_3	166	169	47.61ms	1.018
san400_0.5_1	387	391	988.70ms	1.010
san400_0.7_1	376	378	683.53ms	1.005
san400_0.7_2	379	382	649.11ms	1.008
san400_0.7_3	382	385	635.93ms	1.008
san400_0.9_1	316	317	255.68ms	1.003
san1000	986	990	8.70s	1.004
sanr200_0.7	183	184	196.33ms	1.005
sanr200_0.9	162	163	64.02ms	1.006
sanr400_0.5	387	388	994.94ms	1.003
sanr400_0.7	379	381	697.75ms	1.005

Performance Summary (NPBench Hard Instances):

- **Total instances tested:** 113 (40 FRB + 73 DIMACS)
- **Optimal solutions found:** 12 instances
- **Average approximation ratio:** 1.006
- **Best ratio:** 1.000 (12 optimal instances)
- **Worst ratio:** 1.025 (gen200_p0.9_44)
- **Instances with ratio $\leq$ 1.015:** 107 (95%)
- **FRB average ratio:** 1.009
- **DIMACS average ratio:** 1.007

5.4. Experiment 4: AI-Validated Stress Testing (The Gemini-Vega Validation)

This independent validation study, conducted on December 21, 2025 using Gemini AI [17], tested Hvala on adversarially constructed hard graphs designed to challenge heuristic algorithms. The focus was on 3-regular graphs where every vertex has degree exactly 3, eliminating degree-based heuristic advantages.

Experimental Design:

- **Graph Construction:** Random 3-regular graphs (uniform degree distribution)
- **AI Validation:** Gemini AI architected testing framework and verified results
- **Baseline Comparison:** Standard greedy highest-degree-first heuristic
- **Theoretical Context:** Optimal vertex cover for 3-regular graphs is approximately $0.5n$ vertices

Table 5. Gemini-Vega Stress Test Results on 3-Regular Graphs

Graph Size	Vertices	Edges	Hvala Size	Greedy Size	Hvala Ratio
Power-Law (N=10,000)	10,000	–	4,957	5,093	–
3-Regular (N=5,000)	5,000	7,500	2,917 (58.34%)	3,073 (61.46%)	1.0712
3-Regular (N=20,000)	20,000	30,000	11,647 (58.24%)	12,350 (61.75%)	1.0693
Theoretical optimal for 3-regular: $\sim$0.5446n vertices					

Key Observations:

1. **Improvement Over Greedy:** Hvala consistently outperforms greedy by 2.7-3.5%
2. **Ratio Stability:** Approximation ratio improved slightly from 1.0712 to 1.0693 as graph size doubled
3. **Theoretical Context:** Achieved ratio of 1.069 against theoretical optimum ($\sim 0.5446n$)
4. **Computational Feasibility:** 20,000-vertex graph solved in 162.09 seconds
5. **AI Verification:** Independent validation through Gemini AI confirms correctness and reproducibility

Gemini AI Full Transcript: Complete experimental session available at https://gemini.google.com/share/55109efe4d85

6. Arguments Supporting the Hypothesis

We present five categories of evidence supporting our hypothesis that $\rho < \sqrt{2}$, while maintaining honesty about the gap between empirical observation and theoretical proof.

6.1. Argument 1: Consistency Across Diverse Instance Classes

Evidence: Across four independent experimental studies spanning 233+ instances with radically different structural properties, no instance exceeded ratio 1.071:

Table 6. Cross-Experiment Consistency Analysis

Experiment	Instances	Avg. Ratio	Max Ratio
DIMACS Benchmarks	32	1.0072	1.030
Real-World Large Graphs	88	1.007	1.032
NPBench Hard Instances	113	1.006	1.025
AI Stress Tests	3	–	1.071
Combined	**236**	**1.007**	**1.071**

Strength: This consistency across bipartite graphs, scale-free networks, dense random graphs, structured benchmarks, and adversarially constructed 3-regular graphs suggests the algorithm exploits fundamental structural properties rather than artifacts of specific graph families.

Limitation: Consistency across tested instances does not prove impossibility of worse instances. If P $\neq$ NP, then such instances achieving ratio $\geq \sqrt{2}$ must exist by the hardness results of Khot et al. under SETH.

6.2. Argument 2: Scalability and Improved Performance on Larger Instances

Evidence: Contrary to typical heuristic degradation, performance stabilizes or improves on larger instances:

 doi:10.20944/preprints202506.0875.v10

- C4000.5 (3,986 vertices): ratio 1.001
- p_hat1500-1 (1,488 optimal): ratio 1.001
- 20K 3-regular graph: ratio 1.0693 (better than 5K instance at 1.0712)
- rec-amazon (262K vertices): successfully processed

Implication: If the algorithm degraded systematically on larger instances, we would expect to observe ratios approaching or exceeding $\sqrt{2} \approx 1.414$ on the largest tested graphs. Instead, the largest instances maintain ratios ≤ 1.071.

Counterargument: Theoretical worst-case instances may require specific adversarial constructions not present in our test suite, possibly with size beyond computational feasibility.

6.3. Argument 3: High Frequency of Provably Optimal Solutions

Evidence: The algorithm achieves provably optimal solutions on significant fractions of tested instances:

- DIMACS: 3/32 optimal (9.4%)
- Real-World: 28/88 optimal (31.8%)
- NPBench: 12/113 optimal (10.6%)

Implication: Achieving exact optimality on 43 instances (18.3% of total) demonstrates that the algorithm's degree-1 reduction captures sufficient structural information to solve certain graph classes exactly. This suggests the reduction is fundamentally sound, not merely a heuristic approximation.

Theoretical Context: These optimal solutions occur primarily on tree-like structures (SCC instances) and highly regular graphs (Hamming, Johnson), where the degree-1 reduction perfectly captures the problem structure.

6.4. Argument 4: Consistent Improvement over Greedy Baselines

Evidence: Across all experiments, Hvala consistently outperforms simple greedy strategies by 2-4%:

Table 7. Hvala vs. Greedy Comparison (Selected Instances)

Instance	Hvala	Greedy	Improvement	Optimal
3-Regular (5K)	2,917	3,073	5.1%	∼2,723
3-Regular (20K)	11,647	12,350	5.7%	∼10,892
Power-Law (10K)	4,957	5,093	2.7%	Unknown

Strength: This consistent improvement across diverse structures suggests Hvala captures global optimization information missed by local degree-based heuristics.

6.5. Argument 5: Theoretical Foundation via Weight-Preserving Reduction

Theoretical Basis: The reduction to maximum degree-1 maintains key properties:

Theorem 1 (Weight Preservation)**.** *For any vertex u in the original graph G with degree k, the total weight of its auxiliary vertices in G' equals 1:*

$$\sum_{i=0}^{k-1} w_{(u,i)} = \sum_{i=0}^{k-1} \frac{1}{k} = 1$$

Theorem 2 (Lower Bound Preservation)**.** *Any valid vertex cover in G induces a weighted vertex cover in G' with weight at most the size of the original cover.*

Symmetry Breaking and Determinism: A critical component of Algorithms 3 and 4 is the condition $(w(v) = w(u)$ and $v < u)$. This enforces a deterministic symmetry breaking that is vital when G is a regular graph or possesses uniform weight distributions.

Theorem 3 (Deterministic Stability)**.** *By employing a lexicographical tie-breaker, the algorithm ensures that the selection process is invariant to the order of edge traversal in G'. In regular structures where weight gradients are zero, this prevents the accumulation of local "drifts" during the back-projection to G, ensuring that the induced solution maintains structural consistency across the auxiliary vertex sets.*

Implication: These properties ensure that optimal solutions on G' provide near-optimal guidance for G, forming a rigorous theoretical foundation beyond pure heuristic intuition. The use of lexicographical ordering survives worst-case scenarios in symmetric graphs by guaranteeing that uniform-weight edges are resolved in a manner that can be systematically bounded during analysis.

Gap: While these properties are proven, they do not yet establish a worst-case approximation ratio $< \sqrt{2}$. Completing this proof requires bounding the error introduced during projection from G' back to G.

7. Addressing the Dubious Nature of the Hypothesis

7.1. Why This Hypothesis Appears Dubious

Our hypothesis directly implies one of the most extraordinary claims in computer science and mathematics:

1. **Direct Implication for P vs NP:** Achieving a polynomial-time approximation ratio $\rho < \sqrt{2}$ for vertex cover would prove that P = NP. This follows from known hardness results: Dinur and Safra [5] proved that approximating vertex cover to within factor $\sqrt{2} - \epsilon$ is NP-hard for any $\epsilon > 0$. Therefore, a polynomial-time algorithm with ratio $< \sqrt{2}$ would solve an NP-hard problem in polynomial time, implying P = NP.
2. **Millennium Prize Problem:** The P versus NP problem is one of the seven Millennium Prize Problems designated by the Clay Mathematics Institute, with a $1,000,000 prize for its solution. Our hypothesis, if proven, would claim this prize by demonstrating P = NP.
3. **Contradiction with Decades of Research:** The overwhelming consensus in the computer science community is that P $\neq$ NP. Countless researchers have attempted to prove P = NP or find polynomial-time algorithms for NP-complete problems, all without success. Our hypothesis suggests we have achieved what the collective effort of the field has not.
4. **Implications Beyond Vertex Cover:** If P = NP, it would revolutionize:
   - Cryptography (most encryption schemes would be breakable)
   - Optimization (all NP-complete problems become tractable)
   - Artificial intelligence (many learning problems become efficiently solvable)
   - Mathematics (automated theorem proving becomes vastly more powerful)

7.2. The Hypothesis Framework as Intellectual Honesty

By framing our claim as a *hypothesis* rather than a proven theorem, we acknowledge:

What We Have:

- Extensive empirical evidence across 233+ diverse instances
- Consistent approximation ratios between 1.001 and 1.071
- Theoretical proofs of optimality on specific graph classes (paths, cliques, star graphs, skewed bipartite graphs)
- Formal analysis of structural complementarity showing orthogonal worst-cases for different heuristics
- Independent validation through AI-assisted stress testing
- No observed counterexamples despite testing on hard instances and adversarially constructed 3-regular graphs

What We Lack:

- Rigorous proof that our graph classification (sparse/dense/bipartite/hub-heavy) exhaustively covers ALL possible graph structures
- Worst-case analysis proving $\rho < \sqrt{2}$ for potential adversarial graphs not in any of our identified classes
- Formal proof that no graph exists where all five heuristics simultaneously achieve ratio $\geq \sqrt{2}$
- Resolution of whether achieving $\rho < \sqrt{2}$ on all graphs would truly imply P = NP (requires complete proof, not empirical evidence)

Why the Hypothesis Framework Matters:

1. **Transparency:** Clearly distinguishes between experimental observation and mathematical proof of P = NP
2. **Falsifiability:** Invites construction of counterexamples that would disprove the hypothesis
3. **Community Engagement:** Encourages rigorous analysis by the broader research community
4. **Scientific Integrity:** Acknowledges that claiming to prove P = NP requires ironclad formal proof, not just empirical evidence

7.3. Potential Explanations for the Empirical Results

Scenario 1: The Hypothesis is True (P = NP)

- The algorithm achieves $\rho < \sqrt{2}$ provably for all graphs
- P = NP is proven, solving a Millennium Prize Problem
- Represents the most significant breakthrough in computer science history
- Requires complete restructuring of computational complexity theory

Scenario 2: Benign Instance Distribution (P $\neq$ NP)

- All 233+ tested instances happen to be "easy" for this algorithm
- Adversarial instances approaching $\sqrt{2}$ exist but weren't encountered
- Consistent with P $\neq$ NP and known hardness results
- Our test suite, despite diversity, missed the truly hard instances

Scenario 3: Hidden Structure Exploitation (P $\neq$ NP)

- Real-world and standard benchmark graphs have structural properties absent in theoretical worst-case constructions
- Algorithm exploits these properties effectively
- Worst-case ratio could exceed $\sqrt{2}$ on pathological instances designed to break the algorithm
- Practical usefulness without theoretical breakthrough

Most Likely Explanation: Given the overwhelming evidence that P $\neq$ NP and the difficulty of the P versus NP problem, Scenarios 2 or 3 are far more probable than Scenario 1. However, we present the hypothesis to allow the community to rigorously investigate all possibilities.

8. Conclusion

We have presented the Hvala algorithm with the **hypothesis** that it achieves approximation ratio $\rho < \sqrt{2} \approx 1.414$ for the Minimum Vertex Cover problem. This hypothesis, if proven, would directly demonstrate that P = NP—solving one of the seven Millennium Prize Problems and representing one of the most significant breakthroughs in the history of mathematics and computer science. Given the extraordinary nature of this claim and the decades of failed attempts to prove P = NP, the hypothesis appears dubious. Nevertheless, we present extensive experimental evidence across 233+ diverse instances spanning four independent validation studies.

8.1. Summary of Empirical Evidence

Our experimental validation demonstrates:

- **Consistent Performance:** Average ratios of 1.006-1.007 across all major experiments
- **No Severe Outliers:** Maximum observed ratio of 1.071 on adversarial 3-regular graphs

doi:10.20944/preprints202506.0875.v10

- **Optimal Solutions:** 43 provably optimal solutions (18.3% of tested instances)
- **Scalability:** Successful processing of graphs up to 262,111 vertices
- **Robustness:** Strong performance across diverse graph families (bipartite, scale-free, regular, random, structured)
- **Independent Validation:** AI-assisted stress testing confirms reproducibility and correctness

8.2. Theoretical Implications

If the hypothesis were validated through rigorous proof:

1. **P = NP Proven:** It would demonstrate that every problem whose solution can be verified in polynomial time can also be solved in polynomial time.
2. **Millennium Prize:** It would claim the $1,000,000 Clay Mathematics Institute prize for solving the P versus NP problem.
3. **Cryptographic Revolution:** Most current encryption schemes (RSA, elliptic curve cryptography) would become theoretically breakable in polynomial time.
4. **Optimization Breakthrough:** All NP-complete problems (traveling salesman, scheduling, bin packing, etc.) would become efficiently solvable.
5. **Scientific Impact:** Automated reasoning, theorem proving, drug design, and numerous other fields would be revolutionized.

8.3. Why We Remain Skeptical

Despite the compelling empirical evidence, we emphasize several reasons for skepticism:

1. **Historical Precedent:** Thousands of claimed proofs of P = NP have been proposed and all have been found to contain errors. The problem has resisted solution for over 50 years.
2. **Community Consensus:** The overwhelming majority of complexity theorists believe $P \neq NP$ based on decades of hardness results and failed algorithmic attempts.
3. **Empirical Evidence $\neq$ Proof:** No amount of experimental validation, regardless of consistency or scale, constitutes a mathematical proof. One counterexample would disprove the hypothesis.
4. **Potential Hidden Assumptions:** Our test suite, while diverse, may share structural properties that make all tested instances "easy" for this algorithm.
5. **Missing Worst-Case Analysis:** We have not proven the ratio bound for adversarially constructed graphs designed to maximize the algorithm's approximation error.

8.4. Open Questions and Future Work

We call upon the research community to:

1. **Attempt Rigorous Proof or Disproof:** Either:
    - Prove approximation ratio $< \sqrt{2}$ for all graphs (proving P = NP), or
    - Construct counterexample instances achieving ratio $\geq \sqrt{2}$ (disproving the hypothesis)
2. **Independent Verification:** Reproduce results on:
    - Additional benchmark collections
    - Specially constructed adversarial graphs
    - Randomized instances with controlled structural properties
3. **Comparative Analysis:** Direct comparison against:
    - State-of-the-art exact solvers
    - Modern heuristics (TIVC, FastVC2+p)
    - Machine learning-based approaches
4. **Theoretical Analysis:** Investigate:
    - Formal properties of the degree-1 reduction
    - Error bounds during projection from G' to G

 doi:10.20944/preprints202506.0875.v10

- Necessary and sufficient conditions for $\rho < \sqrt{2}$
- Relationship to existing hardness results

5. **Adversarial Construction:** Design graphs that:
   - Maximize the algorithm's approximation ratio
   - Exploit potential weaknesses in the reduction technique
   - Test the limits of the ensemble heuristic selection

8.5. Final Remarks

This work demonstrates that the Hvala algorithm achieves exceptional empirical performance on the Minimum Vertex Cover problem, with no tested instance exceeding ratio 1.071 across 233+ diverse graphs. While we hypothesize that this performance could extend to a provable worst-case guarantee of $\rho < \sqrt{2}$—which would prove P = NP—we emphasize the extraordinary and likely dubious nature of this claim.

The hypothesis framework allows us to present compelling evidence while maintaining scientific integrity and intellectual honesty. We recognize that:

- **Extraordinary claims require extraordinary proof:** Proving P = NP requires rigorous mathematical proof, not empirical validation
- **The burden of proof is immense:** We must either provide ironclad formal proof or accept that our hypothesis is likely false
- **Skepticism is warranted:** Given 50+ years of failed attempts to prove P = NP, the most likely explanation is that we have not found a proof, but rather an algorithm that performs well on our particular test suite
- **Value regardless of outcome:** Even if the hypothesis is false, the algorithm demonstrates practical value for real-world vertex cover optimization

Whether the hypothesis proves true (solving P versus NP) or false (revealing limitations in empirical validation and the importance of worst-case analysis), the investigation advances our understanding of approximation algorithms, the gap between theory and practice, and the fundamental limits of efficient computation.

We invite vigorous scrutiny, attempted refutation, and independent validation from the theoretical computer science community. Only through such rigorous examination can we determine whether this hypothesis represents a genuine breakthrough or an instructive example of the difference between empirical observation and mathematical proof.

Algorithm Availability: The Hvala algorithm is publicly available for independent verification:

- **PyPI:** https://pypi.org/project/hvala
- **Installation:** `pip install hvala`
- **Usage:** `from hvala.algorithm import find_vertex_cover`
- **Source Code:** Available for inspection and verification

References

1. Karp, R.M. Reducibility Among Combinatorial Problems. In *50 Years of Integer Programming 1958–2008*; Springer: Berlin, Heidelberg, 2009; pp. 219–241. https://doi.org/10.1007/978-3-540-68279-0_8.
2. Papadimitriou, C.H.; Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity*; Courier Corporation: Mineola, New York, 1998.
3. Karakostas, G. A Better Approximation Ratio for the Vertex Cover Problem. *ACM Transactions on Algorithms* **2009**, *5*, 1–8. https://doi.org/10.1145/1597036.1597045.
4. Karpinski, M.; Zelikovsky, A. Approximating Dense Cases of Covering Problems. In Proceedings of the DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Providence, Rhode Island, 1996; Vol. 26, pp. 147–164.
5. Dinur, I.; Safra, S. On the Hardness of Approximating Minimum Vertex Cover. *Annals of Mathematics* **2005**, *162*, 439–485. https://doi.org/10.4007/annals.2005.162.439.

6. Khot, S.; Minzer, D.; Safra, M. On Independent Sets, 2-to-2 Games, and Grassmann Graphs. In Proceedings of the Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, Montreal, Canada, 2017; pp. 576–589. https://doi.org/10.1145/3055399.3055432.
7. Dinur, I.; Khot, S.; Kindler, G.; Minzer, D.; Safra, M. Towards a proof of the 2-to-1 games conjecture? In Proceedings of the Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, Los Angeles, California, 2018; pp. 376–389. https://doi.org/10.1145/3188745.3188804.
8. Khot, S.; Minzer, D.; Safra, M. Pseudorandom Sets in Grassmann Graph Have Near-Perfect Expansion. In Proceedings of the 2018 IEEE 59th Annual Symposium on Foundations of Computer Science, Paris, France, 2018; pp. 592–601. https://doi.org/10.1109/FOCS.2018.00062.
9. Khot, S. On the Power of Unique 2-Prover 1-Round Games. In Proceedings of the Proceedings of the 34th Annual ACM Symposium on Theory of Computing, Montreal, Canada, 2002; pp. 767–775. https://doi.org/10.1145/509907.510017.
10. Khot, S.; Regev, O. Vertex Cover Might Be Hard to Approximate to Within $2 - \epsilon$. *Journal of Computer and System Sciences* **2008**, *74*, 335–349. https://doi.org/10.1016/j.jcss.2007.06.019.
11. Vega, F. The Hvala Algorithm. https://dev.to/frank_vega_987689489099bf/the-hvala-algorithm-5395, 2025. Accessed July 27, 2025.
12. Vega, F. The Resistire Experiment. https://dev.to/frank_vega_987689489099bf/the-resistire-experiment-632, 2025. Accessed October 15, 2025.
13. Rossi, R.; Ahmed, N. The Network Data Repository with Interactive Graph Analytics and Visualization. *Proceedings of the AAAI Conference on Artificial Intelligence* **2015**, *29*. https://doi.org/10.1609/aaai.v29i1.9277.
14. Cai, S. Large Graphs Collection for Vertex Cover Benchmarking. https://lcs.ios.ac.cn/~caisw/graphs.html. Network Data Repository collection.
15. Vega, F. The Creo Experiment. https://dev.to/frank_vega_987689489099bf/the-creo-experiment-2i1b, 2025. Accessed December 20, 2025.
16. Roars. NP-Complete Benchmark Instances. https://roars.dev/npbench/. Vertex cover benchmark collection.
17. Vega, F. The Gemini-Vega Validation. https://dev.to/frank_vega_987689489099bf/the-gemini-vega-validation-27i2, 2025. Accessed December 21, 2025.
18. Bar-Yehuda, R.; Even, S. A Local-Ratio Theorem for Approximating the Weighted Vertex Cover Problem. *Annals of Discrete Mathematics* **1985**, *25*, 27–46.
19. Zhang, Y.; Wang, S.; Liu, C.; Zhu, E. TIVC: An Efficient Local Search Algorithm for Minimum Vertex Cover in Large Graphs. *Sensors* **2023**, *23*, 7831. https://doi.org/10.3390/s23187831.
20. Cai, S.; Lin, J.; Luo, C. Finding a Small Vertex Cover in Massive Sparse Graphs. *Journal of Artificial Intelligence Research* **2017**, *59*, 463–494. https://doi.org/10.1613/jair.5443.
21. Luo, C.; Hoos, H.H.; Cai, S.; Lin, Q.; Zhang, H.; Zhang, D. Local search with efficient automatic configuration for minimum vertex cover. In Proceedings of the Proceedings of the 28th International Joint Conference on Artificial Intelligence, Macao, China, 2019; pp. 1297–1304.
22. Harris, D.G.; Narayanaswamy, N.S. A Faster Algorithm for Vertex Cover Parameterized by Solution Size. In Proceedings of the 41st International Symposium on Theoretical Aspects of Computer Science, Clermont-Ferrand, France, 2024; Vol. 289, pp. 40:1–40:18. https://doi.org/10.4230/LIPIcs.STACS.2024.40.