Breaking the Trade-Off

Adaptive Optimization for Scalable, Minimal RBAC

Chrysanthos Kindynis

Delft University of Technology

TU Delft

Breaking the Trade-Off

Adaptive Optimization for Scalable, Minimal RBAC

by

Chrysanthos Kindynis

to obtain the degree of Master of Science in Computer Science
at the Delft University of Technology,
to be defended publicly on Tuesday June 24, 2025 at 11:00 AM.

Student number:	5289394	
Thesis committee:	Professor Dr. George Smaragdakis	TU Delft, Chair
	Assistant Professor Dr. Yury Zhauniarovich	TU Delft, Supervisor
	Assistant Professor Dr. Megha Khosla	TU Delft
	Dr. Eduardo Barbaro	External Supervisor
Project duration:	October 2024 – June 2025	

This thesis is confidential and cannot be made public until October 31, 2025.

Cover:	The background of the cover image was generated using ChatGPT Pro.
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

TU Delft

Acknowledgments

I am deeply grateful to Prof. Dr. George Smaragdakis for his unconditional support and to Asst. Prof. Dr. Yury Zhauniarovich for his constructive feedback throughout the months. I needed both, and I deeply appreciate their guidance. I would also like to thank Dr. Eduardo Barbaro for the opportunity he provided me to work closely with his team and apply my research directly within an organization. I am sincerely grateful for the freedom I had, and for all the experience, guidance, and knowledge I gained through this collaboration; thank you.

I also wish to thank Asst. Prof. Dr. Megha Khosla for her willingness to participate and assist with the completion of my master's thesis.

Throughout this thesis, especially in its final stages, I have received tremendous support from lots of people. I am sincerely grateful for all the support, and I appreciate every one of you.

This thesis I dedicated to my father in gratitude for all he has invested in me throughout the years.

Contents

Abstract

Role-Based Access Control (RBAC) is foundational to enterprise security, yet manual role engineering remains error-prone and unscalable. Although automated role mining addresses this, existing methods face a critical trade-off: exact approaches guarantee minimal roles but fail on real-world scales, while heuristics scale but lack formal guarantees. This inconsistency forces enterprises into suboptimal, insecure configurations—increasing vulnerability risks and compliance costs. We resolve this instability through a four-level resource-aware framework that dynamically adapts: (1) a memory-light heuristic, (2) optimality-preserving reductions, (3) a greedy heuristic with logarithmic approximation bounds, and (4) an ILP-based exact solver. Notably, our approach eliminates more than 99% of edges in 26 out of 31 real-world systems, enabling globally optimal role configurations and achieving an average 53% simplification of existing RBAC systems. Our heuristics achieve near-optimal solutions, while providing significant speedups over prior heuristics. Beyond individual components, the unified, adaptive framework minimizes suboptimal decisions at any scale. We open-source this framework to enable minimal RBAC deployment at any scale.

1

Introduction

In modern enterprises, cybersecurity plays a pivotal role in protecting sensitive data, ensuring regulatory compliance, and maintaining operational integrity. Among its foundational components is Identity and Access Management (IAM) — the discipline responsible for ensuring that only the right individuals have access to the right resources at the right times. A cornerstone of IAM is access control, the mechanism by which systems determine and enforce who is authorized to perform certain actions.

One of the most widely adopted models for enterprise access control is Role-Based Access Control (RBAC). Originally proposed by Ferraiolo and Kuhn in the early 1990s, RBAC introduces roles as an abstraction layer between users and permissions, simplifying permission management by grouping access rights according to job functions or responsibilities. Its popularity stems from its simplicity, alignment with organizational structures, and ability to reduce administrative errors and security risks [11]. Decades later, RBAC remains a dominant paradigm, with commercial IAM platforms and regulatory standards continuing to rely heavily on it [30].

However, as organizations grow and evolve, managing roles becomes a significant challenge. Enterprise environments often consist of thousands of users, complex departmental hierarchies, and evolving access requirements. This results in RBAC inefficiencies—such as redundant, overlapping, or obsolete roles—that bloat access control policy and impair auditing, compliance, and system performance [32]. Manual role engineering becomes infeasible at scale, motivating the need for automated techniques.

This is where role mining becomes essential. Role mining aims to automatically infer a compact and meaningful set of roles from existing user-permission assignments, reducing administrative burden and improving policy quality. However, as an NP-hard problem [26], the Role Mining Problem (RMP) presents significant computational challenges. Extensive research has been conducted in recent years on the RMP, utilizing various methods that focus on diverse optimization objectives, constraints, and algorithmic approaches.

These methods broadly fall into two categories. First, *Optimal Approaches* (ILP-based) that guarantee solution optimality but face exponential complexity. Reduction rules, such as Ene et al.'s [9], improve scalability through deterministic simplifications; however, they still require days to converge on modern benchmarks [38], making their direct application to enterprise environments impractical. Second, *Heuristic Approaches* that scale effectively but sacrifice optimality guarantees. Most lack theoretical quality bounds, though rare exceptions like Huang et al. [17] offer limited approximation guarantees for specific problem variants. This absence of robust quality assurances limits their reliability for medium-sized enterprise graphs where near-optimality is critical.

Despite the impressive advances in Role Mining, a fundamental gap persists: *No methodology adjusts the algorithm used, based on the dynamic size of the underlying problem*. So far, practitioners must choose upfront between optimality and scalability, unable to switch approaches mid-execution when the dynamic problem size allows. Alternatively, the unified framework we propose can leverage fast

heuristic approaches to reduce large input graphs effectively, and as the underlying problem size decreases, it can transition to more optimal approaches, thereby minimizing the approximation error on large graphs while still allowing exact solutions on small instances.

This thesis addresses three central limitations in existing role-mining methods by answering the following research questions:

- **RQ1:** *Can classical role mining reduction rules be systematically reinterpreted through set cover theory, and does this reformulation enable more efficient or theoretically grounded reductions compared to traditional neighbor-based approaches?*
- **RQ2:** *Can a heuristic be designed that offers logarithmic approximation guarantees for the Role Mining Problem, thereby bridging the scalability-optimality gap?*
- **RQ3:** *Can we develop a role mining methodology that dynamically decides between heuristic and exact methods to balance scalability and solution quality adaptively?*

To answer these questions, we introduce a four-level, resource-aware framework for role mining. Each level of the framework corresponds to a progressively more accurate and computationally expensive technique. The pipeline is designed to adapt based on the input graph size and available system resources, ensuring the most effective feasible method is used at any time. This structured approach enables rapid, approximate decisions when necessary and exact, optimal solutions when feasible. At the first level, the framework uses a memory-light heuristic algorithm that iteratively selects large bicliques without computing the entire set of maximal bicliques. This approach ensures scalability even on massive graphs. At the second level, it applies biclique-based reductions, offering a reinterpretation of classical reduction rules such as domination, isolation, and subset through the lens of set cover theory. These reductions simplify the input while preserving solution quality. At the third level, the framework adopts a greedy heuristic applied to the set cover formulation. This stage provides a logarithmic approximation bound and offers the first theoretical guarantee in role mining heuristics to our knowledge. Finally, at the fourth level, when the graph is sufficiently small, the framework invokes an ILP-based exact solver to compute the optimal solution. This layered architecture forms a multi-step resource-aware decision framework, where each level defers to a more powerful layer only when required. In doing so, it bridges the gap between runtime performance and role-set quality, adapting to the practical constraints of real-world deployment.

Our contributions are as follows:

- A principled reformulation of classical role mining reductions by drawing on ideas from the set cover problem, providing theoretical foundations, improving modularity compared to previous neighbor-based approaches, and opening the doors for further exploration of the efficiency of these reduction rules.
- A greedy role selection heuristic with theoretical guarantees that functions as an effective middle layer between heuristic and exact approaches.
- A pure memory-light heuristic that can efficiently handle input graphs of the size while also delivering quality role selections through well-informed choices.
- A four-level adaptive framework that integrates heuristic, reduction, approximate, and exact methods in a resource-aware fashion, bridging the gap between rigid heuristic or exact role mining methodologies.
- An open-source implementation of the whole framework, along with the predecessor's competitive approaches, to support adoption and further research.
- A comprehensive experimental evaluation across real-world, synthetic, and benchmark datasets, validating theoretical assumptions and demonstrating the competence of each component as well as the overall framework.

The remainder of the thesis is organized as follows. Chapter 2 introduces the necessary background and preliminary information in graph theory, bicliques, and the set cover problem. Chapter 3 reviews related work in exact, heuristic, and hybrid role-mining techniques. Chapter 4 outlines the theoretical foundations of our reductions and establishes the methodology. Chapter 5 presents the complete

four-level framework. Chapter 6 evaluates its performance empirically. Chapter 7 discusses practical observations, limitations, and future directions. Chapter 8 concludes.

2 Background

The RMP, particularly in its bottom-up formulation, can be rigorously studied through the lens of graph theory. A central concept in this representation is that of *bicliques* [13], which provide a natural structural interpretation of roles in an RBAC system. This section introduces the necessary background on bicliques and bipartite graphs, followed by a brief overview of constraint solvers in the role mining context and an introduction to the Set Cover Problem.

2.1. Bipartite Graphs, Bicliques and Roles

A *bipartite graph* $G = (V, E)$ partitions its vertex set V into two disjoint subsets A and B, where every edge $e \in E$ connects vertices exclusively between A and B. This structure naturally models access control systems: A represents *users*, B represents *permissions*, and edges denote direct user-permission assignments, as illustrated in Figure 2.1.

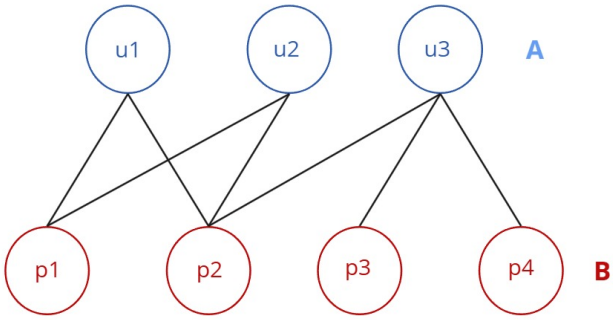

Figure 2.1: User-permission assignments represented as a bipartite graph. Users (u_1-u_3) and permissions (p_1-p_4) form disjoint partitions A and B. Edges indicate access rights.

A *biclique* (complete bipartite subgraph) consists of subsets $A' \subseteq A$ and $B' \subseteq B$ where every $a \in A'$ connects to every $b \in B'$. In Figure 2.1, vertices $\{u_1, u_2\}$ and $\{p_1, p_2\}$ demonstrate such a biclique structure.

In role mining, it is essential to adhere to fundamental principles, such as the *Least Privilege principle*, which ensures the RBAC system does not grant users permissions they did not initially possess [31]. Based on this constraint, we can establish a **fundamental mathematical equivalence**: the users and permissions of any role in an RBAC system correspond precisely to a biclique in the original user–permission graph, where all assigned users (A') are connected to all role permissions (B'). Figure 2.2 demonstrates this critical relationship.

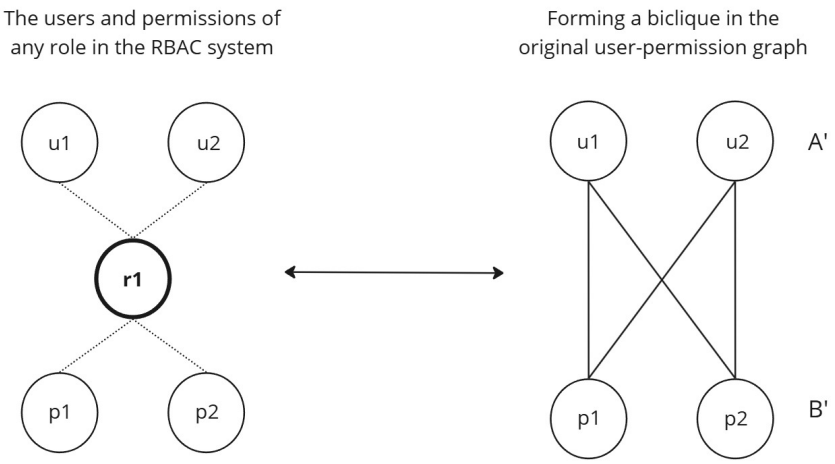

Figure 2.2: Equivalence between a role in RBAC (left: users assigned to a role, which is connected to permissions) and the corresponding biclique in the user-permission graph (right: users and permissions forming fully connected subsets, A' and B').

This mapping between roles and bicliques enables a powerful formulation: **constructing a set of roles that collectively explain the observed access assignments is equivalent to covering the user–permission graph with bicliques**. Figure 2.3 demonstrates this equivalence through a concrete example.

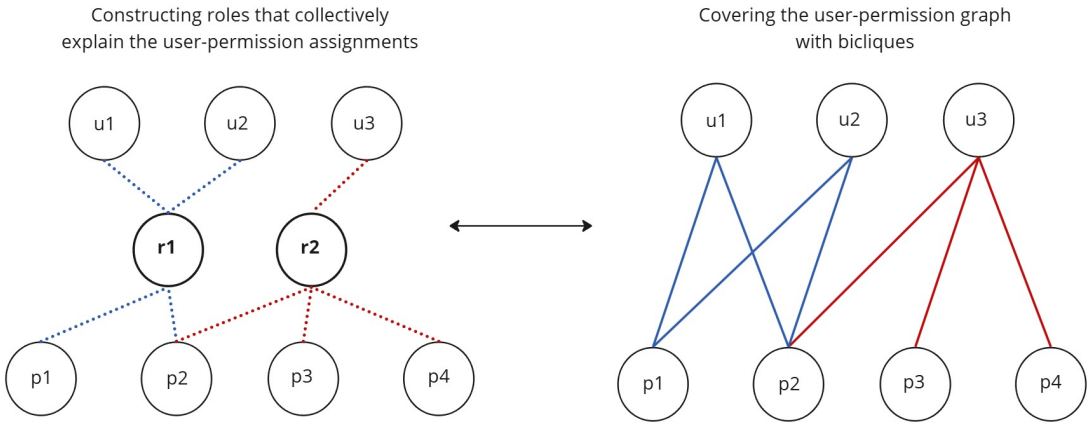

Figure 2.3: Equivalence between constructing a set of roles that collectively explain the user-permissions assignments (left) and covering the user-permission graph using bicliques (right)

2.2. Constraint Solvers

Constraint solvers are general-purpose tools designed to solve optimization problems involving mathematical constraints. These solvers operate on formal models such as Integer Linear Programs (ILPs), Boolean satisfiability (SAT), or Constraint Satisfaction Problems (CSPs), finding variable assignments that optimize an objective function while satisfying specified constraints.

Prominent ILP solvers such as *Gurobi* [15], *CPLEX* [19], and *SCIP* [35] employ sophisticated tech-

niques including branch-and-bound, cutting planes, and presolving heuristics to navigate solution spaces efficiently, even for large instances with thousands of variables and constraints.

Role Mining Applications
Different modeling approaches exist for leveraging constraint solvers for role mining. Some formulations leave role construction entirely to the solver, generating all possible user-permission combinations—an approach that scales poorly beyond toy examples due to combinatorial explosion. More practical modeling approaches provide *candidate roles* to the solver (e.g., identified through biclique analysis) and optimize for the minimal subset that ensures complete coverage of user-permission assignments.

This candidate-based approach leverages domain knowledge to constrain the solution space while maintaining optimality guarantees, making constraint solvers viable for role mining on datasets of small to moderate size.

2.3. Set Cover Problem

The *Set Cover Problem (SCP)* is a fundamental NP-hard problem in combinatorial optimization [6]. Given a universe $U = \{e_1, e_2, \ldots, e_n\}$ and a collection of subsets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ where $S_i \subseteq U$, the objective is to find the smallest subcollection $\mathcal{C} \subseteq \mathcal{S}$ satisfying:

$$\bigcup_{S \in \mathcal{C}} S = U \quad \text{with} \quad |\mathcal{C}| \text{ minimized}$$

This problem has numerous applications in fields like vehicle routing [12], airline crew scheduling [4], information retrieval [8] and more [7].

Due to its NP-hardness [21], finding an exact solution is computationally infeasible for large instances. Practical heuristic approaches include: greedy approximation algorithm, genetic algorithms [**chen2024genetic**] and local search heuristics [**balaji2024improved**].

Integer Linear Programming Formulation

The SCP can be formulated as an ILP with binary decision variables:

$$x_i = \begin{cases} 1 & \text{if } S_i \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

The ILP minimizes set count while ensuring element coverage:

$$\text{Minimize} \quad \sum_{i=1}^{m} x_i$$

subject to:

$$\sum_{i : e_j \in S_i} x_i \geq 1 \quad \forall e_j \in U, \quad x_i \in \{0, 1\} \quad \forall i$$

The constraints ensure that every element $e_j \in U$ is covered by at least one selected set, while the objective function minimizes the total number of sets chosen. This ILP formulation provides an exact solution to the SCP but can be computationally expensive for large instances due to its NP-hardness. Nonetheless, it serves as a foundation for many exact algorithms and optimization techniques [6].

Greedy approximation algorithm

The greedy algorithm is among the most effective polynomial-time approximation techniques [36] due to its balance of efficiency and solution quality. It iteratively selects the subset covering the maximum number of uncovered elements until U is covered.

The algorithm achieves an approximation ratio of H_n (the n-th harmonic number), bounded by $H_n \leq \ln n + 1$ where $n = |U|$. This means: The solution size is at most H_n times the optimal cover size. For

example, if we have a number of elements, and the minimal number of subsets we need to use is m, then the greedy heuristic is guaranteed to find a solution that uses at most $m * (ln(n + 1))$ subsets. Under standard complexity assumptions, this logarithmic factor is the best achievable in polynomial time [10].

Reduction Rules

Preprocessing reduction rules simplify SCP instances without affecting optimality [6]:

1. **Singleton Rule**: If $\exists e_j$ contained only in S_k, select S_k and remove all covered elements.
2. **Subset Rule**: If $S_i \subseteq S_j$, remove S_j (since S_i covers its elements more efficiently).
3. **Element Domination**: If $N(e_a) \supseteq N(e_b)$ (where $N(e)$ is sets containing e), remove e_b (covered when e_a is covered).

These rules significantly reduce instance size and accelerate both exact and heuristic solutions.

3

Related work

The RMP has been the focus of extensive research over the past two decades, particularly within the broader field of RBAC systems [26] [20]. As organizations increasingly rely on RBAC to manage access control at scale, the need for efficient, automated techniques to infer meaningful role structures from large and complex user-permission datasets has grown accordingly. This chapter surveys the evolution of methods for solving the RMP, with an emphasis on bottom-up, data-driven approaches. We begin by contrasting the foundational paradigms of top-down and bottom-up role engineering, then proceed to discuss key algorithmic strategies in role mining—particularly those that reduce the problem to well-known combinatorial formulations such as the Minimum Biclique Cover problem. Special attention is given to two influential papers that represent milestones in this research trajectory: Ene et al. [9], who introduced a graph-reduction-based exact and heuristic role mining framework, and Tripunitara et al. [38], who extended this line of work through the use of Maximal Biclique Enumeration (MBE). Their respective contributions, strengths, and limitations are discussed in detail, forming the foundation upon which this thesis builds.

3.1. Role Mining and recent methods

Two general approaches to role engineering in RBAC exist: *top-down* and *bottom-up*. A top-down approach involves a thorough analysis of business processes, which are systematically decomposed into smaller functional units [27, 33]. The permissions necessary to perform each task are identified, and business roles are constructed accordingly. This process typically requires extensive manual effort and collaboration among domain experts from various departments, who collectively possess a deep understanding of organizational workflows. Top-down approaches are particularly effective for new systems lacking existing permission data or for organizations with stable, compliance-driven environments. However, this method is time-consuming, demands substantial human involvement [18], and may become obsolete as business needs evolve. Consequently, top-down approaches often struggle to scale in modern, dynamic enterprise settings.

In contrast, bottom-up methods, also known as role-mining, aim to automatically discover a set of roles by analyzing existing user-permission assignments. This approach is data-driven and scales better to large, complex systems. Recent advances in bottom-up techniques have also enabled the integration of business constraints into the mining process [42], enhancing compliance with evolving business requirements. In this thesis, we are primarily interested in constructing roles for large real-world enterprise systems, so we focus on bottom-up, role-mining approaches.

In the field of role mining, Vaidya et al. [39] is the first to define the RMP formally. Given a bipartite graph that represents users and permissions, the RMP's objective is to identify the *smallest* set of roles that accurately reproduces the existing user-permission connections. This definition sets a clear objective for what the role-mining algorithm should achieve - a *compact* RBAC system. The authors map the RMP to other well-known NP-complete problems, demonstrating the RMP's NP-completeness and enabling the application of algorithmic approaches from these other problems to solve the RMP.

A central contribution to the literature is the comprehensive survey by Mitra et al. [26], which classifies role-mining problems and associated solution methods. At the core of problems is the previously mentioned *Basic Role Mining Problem (Basic-RMP)*, which is defined as finding the smallest set of roles that accurately reconstructs the observed user-permission matrix. This optimization formulation based on a single metric forms the problem basis for many subsequent studies. Other variants include *Edge-RMP*, which minimizes the total number of assignments, and *MinNoise-RMP*, which allows limited deviation from the input matrix in exchange for a reduced number of roles. The most general variant is the *Weighted Structural Complexity Optimization (WSCO)* model, which permits weighted optimization over all RBAC components, including roles, user-role and role-permission assignments, role hierarchies, and direct exceptions. This thesis adopts the Basic-RMP formulation due to its conceptual clarity, widespread adoption in the literature, and its ability to be effectively reduced to established combinatorial problems.

The survey by Mitra et al. also categorizes the main algorithmic paradigms for solving RMP. These include permission grouping-based heuristics, which cluster similar permission sets to define roles (e.g., CompleteMiner and FastMiner [40]); problem-mapping methods that translate RMP into known optimization problems like tiling or biclique cover; matrix decomposition-based strategies using Boolean Matrix Factorization (BMD [23], EBMD [24]); and graph-based approaches that exploit structural patterns in the user-permission matrix (e.g., RH-Builder and RH-Miner [14]). Each family of methods balances trade-offs between computational efficiency, solution interpretability, and role model compactness.

We now explore recent works from different paradigms and then focus on those that our research more closely relates to.

3.1.1. Recent Role Mining methods

Over the last decade, research in role mining has expanded into multiple paradigms—ranging from matrix decomposition and set cover approximations to evolutionary search and dynamic policy adaptation. In this section, we review representative recent contributions from each of these directions, highlighting their core methodologies, strengths, and limitations before transitioning to the two works that most directly inform our framework.

MFC-RMA: Matrix Factorization and Constraint-Role Mining Algorithm. Zhu et al. [42] propose MFC-RMA, which formulates the role mining problem as a constrained Boolean matrix factorization task. By clustering users and permissions using k-means and then enforcing cardinality and exclusion constraints in a post-processing step, MFC-RMA efficiently reduces role-permission edge counts while accommodating real-world business rules. Although this yields more manageable role sets, the approach remains entirely heuristic. On the one hand, optimal solutions are not guaranteed, even for small instances. On the other hand, the approach lacks formal approximation guarantees, which does not provide confidence that even in worst-case application scenarios, the heuristic's performance will still be within.

addRole-EA: A New Evolutionary Algorithm for the Role Mining Problem. Anderer et al. [2] introduce a genetic algorithm–based framework, addRole-EA, in which chromosomes encode variable-length role sets, and fitness is balanced between reconstruction error and role count. This flexibility enables adaptive exploration of the solution space; however, the method provides no theoretical performance guarantees and cannot be handed off to an exact solver once the candidate space is reduced—leading to unpredictable performance on large or structured datasets.

Dynamic Optimization of Role Concepts Using Evolutionary Algorithms. Anderer et al. [1] address the dynamic evolution of access control policies using custom crossover and mutation operators to adapt role hierarchies over time. Their approach supports policy stability in the face of organizational changes and enforces business constraints such as separation of duties. Nonetheless, their framework prioritizes incremental updates over global optimality. It lacks theoretical guarantees, and by preserving evolving structures rather than re-optimizing globally, it may accumulate inefficiencies over time.

IAM Role Diet: A Scalable Approach to Detecting RBAC Data Inefficiencies. Moratore et al. [32] present a fast and scalable method for detecting and removing redundancies in RBAC systems, focusing on practical inconsistencies and inefficiencies rather than complete role mining. By leveraging efficient graph-based analysis, the approach identifies and eliminates obvious redundancies, such as duplicate or subsumed assignments, at a large scale, making it particularly suitable as a pre-processing step for role-mining pipelines. Although not a complete role mining solution, its efficiency and ability to clean RBAC data suggest promising avenues for integration before more computationally intensive role mining algorithms are applied. The method thus provides a practical and lightweight means to enhance data quality and facilitate subsequent analysis.

Handling Least Privilege and Role Mining via Set Cover Approximation. Huang et al. [17] explicitly map the Basic-RMP and Least Privilege User Assignment Problem (LPUAP) to Set Cover and propose greedy algorithms with provable $\rho H(\gamma)$-approximation bounds. Their methods support SSD constraints and dynamic role modifications (addition and deletion) and include a UPA matrix reduction step to compress the problem prior to applying Set Cover. Despite the strength of the formulation, several limitations persist: the reductions are applied only once, cannot be modularly reused, and the framework offers no principled escalation from heuristics to exact optimization.

Our work builds on this theoretical foundation by recasting role mining reductions as classical kernelization rules, allowing for iterative applications, and introducing a four-level pipeline that dynamically selects the best strategy based on problem complexity. This approach provides both flexibility and formal guarantees while scaling to real-world systems using commodity hardware.

Overall, recent work has aimed to extend the basic role-mining problem with various interesting, helpful, and impressive ideas/directions. However, the central problem of basic role mining problem is still missing a central framework. It employs a variety of heuristic approaches and some optimal approaches but lacks a unified framework that can integrate these and leverage the strengths of each.

We now turn to the two prior works that most directly influence our design: Ene et al. [9] and Tripunitara et al. [38], whose contributions form the theoretical and practical basis for our improvements.

3.2. Fast exact reduction rules and fast heuristic method for RMP

Among all approaches surveyed by Mitra et al.[26], the problem-mapping method introduced by Ene et al. [9] stands out for its superior performance across all datasets. Ene et al. [9] view the RMP as the MCP problem, which involves covering all edges of a bipartite graph with the fewest possible bicliques. Their main contributions consist of an exact algorithm that includes deterministic reductions and a fast approximate algorithm that exhibits polynomial-time complexity.

Exact Approach & Reduction Rules

The exact algorithm aims to derive a minimum biclique cover for the input graph by identifying a minimum clique partition for the dual graph, capitalizing on the inherent duality between these two problems. The authors implement two iteratively applied reduction rules that simplify the dual graph while ensuring the integrity needed to recover an optimal solution.

The first reduction rule functions by identifying and temporarily eliminating user-permission edges that are logically implied by others. The authors denote such edges as dominator edges. In the following chapter 4.1, we provide a detailed explanation of when these edges appear. The dominator edges are removed from the graph and are later retrieved from roles that contain their dominated edges. The computational complexity associated with this reduction is $O(|E|^3|V|\log|V|)$.

At the same time, any edge that becomes isolated and cannot be merged with other edges to form a larger biclique prompts the immediate creation of a new role for that edge. In this scenario, the only biclique that includes the isolated edge is the singleton biclique, which establishes a direct connection between the specific user and the corresponding permission. Since we need to create roles that describe all user-permission connections, the singleton biclique must be part of the solution, thereby allowing us to definitively establish it as a role while removing the isolated edge from the graph.

In summary, the first graph reduction rule removes redundant edges, while the second one addresses

isolated edges by creating a corresponding role for them. Interestingly, these reductions were impressively effective on the datasets present at that time. Specifically, the reduction left no edge on six out of eight datasets. On the remaining two, only about 2.15 percent of the edges remained.

Once no dominator or isolated edges remain, the reduction ends, and the exact algorithm solves the remaining MCP problem instance using an ILP solver. This approach guarantees optimality but is only feasible for small graphs since it is exponential in the worst case. Notably, since the reductions were highly effective on the existing datasets, the dual graphs presented to the solver were relatively small. As a result, the algorithm successfully found an optimal solution for all the instances.

Heuristic Approach

The second approach is a fast heuristic that constructs a cover by iteratively selecting and adding one biclique at a time until all user-permission edges are covered. A node is greedily selected at each iteration, and its neighborhood biclique gets activated as a role. This results in a fast and greedy heuristic. A lattice-based post-processing algorithm follows that merges or removes roles to reduce redundancy. Experimental results show that the heuristic performs remarkably well, often producing results within a few percent of the optimal, with much lower computational cost.

3.3. Role Mining through MBE

Despite being effective on earlier datasets, the method proposed by Ene et al. [9] encounters scalability challenges when applied to newer role-mining benchmarks [3]. In particular, the initial reduction rules, which previously eliminated most edges, are far less effective on complex, modern datasets, often leaving a substantial portion of the graph intact. Moreover, the exact approach relies on reducing the Minimum Biclique Cover problem to Minimum Clique Partition and subsequently to graph coloring—transformations that drastically inflate the problem size. The resulting graph sometimes becomes up to 10,000 times larger than the original bipartite input, making the method infeasible for large-scale instances.

In response to these limitations, Tripunitara et al. [38] propose an alternative framework that remains within bipartite graphs and bicliques to avoid the size blowup.

The primary theoretical contribution of this study is establishing an optimal Role-Based Access Control (RBAC) system for any given graph where each role corresponds to a maximal biclique. A maximal biclique is defined as one that cannot be included as a proper subset within any other biclique and cannot be further expanded without losing its biclique properties.

The authors substantiate their claims through constructive proof. According to the definition of a non-maximal biclique within a specific RBAC framework, it is possible to identify a maximal biclique that encompasses both the users and permissions of the non-maximal biclique. Consequently, this allows for expanding the role into the identified maximal biclique without granting any additional permissions to the users since all users already possess access to these permissions.

As such, by demonstrating that they can transform any existing RBAC system into one that exclusively employs maximal biclique roles without increasing the overall number of roles, the authors prove that any optimal solution may be reformulated into another optimal configuration comprised solely of maximal bicliques. This characterization implies that limiting roles to maximal bicliques rather than allowing for any biclique significantly reduces the search space (logarithmically) without compromising the solution's optimality.

Following the insight that any optimal role configuration can be composed solely of maximal bicliques, Tripunitara et al. [38] propose exact and heuristic approaches based on MBE. A key contribution of their work lies in proposing a principled strategy for assessing the difficulty of a given input graph. They identify the number of maximal bicliques in the graph as a natural measure of the instance's hardness.

The authors use a practical threshold based on empirical analysis to determine which method to apply. Specifically, if the number of maximal bicliques in the input graph is fewer than three million, the exact approach is used; otherwise, they use their heuristic. To check the number of maximal bicliques in a graph without exhausting memory, they run their MBE algorithm in a memory-light mode that counts

maximal bicliques without storing them. This enables the system to quickly estimate problem difficulty without incurring the full cost of biclique generation.

Exact Approach

The process begins by applying the reduction rules introduced by Ene et al. [2], which remove dominating and isolated edges to simplify the graph. Next, an algorithm enumerates all maximal bicliques within the reduced graph. These bicliques are then submitted to an integer linear programming (ILP) solver to select the smallest subset that covers all user-permission edges. The authors find that this process reliably terminates within a reasonable time and produces optimal results for instances containing up to three million maximal bicliques.

The optimal solution found here forms the optimal solution to the initial role-mining problem. Bicliques represent roles, and there is an optimal solution that utilizes only maximal bicliques. As a result, by providing the solver with all maximal bicliques, we equip it with all the necessary roles to construct a minimal, complete set of roles. Consequently, the solver can cover all user-permission edges with the least number of roles, thus solving the initial role-mining problem.

Heuristic Approach

When the number of maximal bicliques exceeds the three-million threshold, the authors resort to a heuristic algorithm. This approach iteratively enumerates maximal bicliques and, upon finding one that is sufficiently large (defined in their experiments as covering at least 200 edges), activates it as a role and removes the covered edges from the graph. This cycle continues, progressively shrinking the graph until it becomes small enough for the exact method to be applied. Interestingly, after every role selection, they remove the covered edges from the input graph and run the memory-light MBE algorithm to see if the remaining graph is small enough. Overall, this approach deliberately chooses maximal and relatively large bicliques for their roles and leverages the exact approach as soon as possible. The loss of optimality comes from the fact that not every large biclique is necessarily part of an optimal solution. Importantly, this research takes the first step towards combining approximate approaches with exact ones, a valuable property for safeguarding the quality of the results.

3.4. Comparison and Limitations of Related Works

In this section, we compare the methodology of the two foundational works: the reduction-based method of Ene et al. [9] and the MBE-based method proposed by Tripunitara et al. [38]. Afterward, we present the overall limitations of both the foundational and recent methods and present the methodological contributions of our approach.

3.4.1. Comparison of foundational approaches

In this section, we compare the two foundational approaches to bottom-up role mining. Both methods share common goals, reducing problem complexity and improving role quality, but differ in their algorithmic design and computational trade-offs. We analyze their exact and heuristic components separately, focusing on scalability, solution quality, and the ability to integrate with broader frameworks.

Comparison of Exact Approaches

The exact method used by Tripunitara et al. [38] and that of Ene et al. [9] begin with the same graph reduction techniques. However, introducing the maximality constraint in the former significantly reduces the number of candidate roles that must be considered, thereby shrinking the solution space. Moreover, by operating directly on maximal bicliques, Tripunitara et al. avoid the costly graph transformations used in Ene et al.'s approach (i.e., the reduction to clique partition and further to graph coloring), which, as discussed earlier, can introduce orders-of-magnitude blowup in the graph size. This results in a more scalable and memory-efficient exact approach for modern datasets.

Comparison of Heuristics

The heuristic proposed by Ene et al. constructs roles incrementally by greedily selecting a root node and forming a biclique around it. This method is fast but shortsighted, as the largeness of the chosen biclique is solely based on the greedy choice of the root node.

In contrast, the heuristic by Tripunitara et al. works only with maximal bicliques and activates only those that exceed a predefined size threshold. This yields a more informed role selection process, leading to better coverage with fewer roles. The experimental results of [38] confirm this advantage: Tripunitara et al.'s heuristic consistently produces higher-quality solutions than its predecessor, though at the cost of significantly higher computation time, sometimes requiring days instead of seconds.

3.4.2. Limitations of previous work

While effective, the initial reduction phase becomes prohibitively expensive for large graphs and can even require multiple days of computation (see Table 2 of [38]). Therefore, their direct application on large graphs is prohibited.

The MBE-based method exhibits two core limitations despite its promising performance:

1. Computational Inefficiency
First, the chosen enumeration process is inefficient and computationally expensive, which cascades numerous problems down the line. Unavoidably, the slow MBE algorithm directly limits the efficiency of the proposed MBE-based heuristic algorithm. The heuristic is responsible for handling the largest graphs; therefore, when it is inefficient, it limits the scalability and efficiency of the entire methodology. We verify this claim by examining the running times presented in their experiments, which span several days of computation. Notably, these long runtimes were obtained not on standard moderate computational resources but on a powerful 64-core server equipped with 256 GB of RAM.

2. Threshold Sensitivity
Solution quality is susceptible to the "large biclique" threshold parameter. Excessively high thresholds fail to identify bicliques in locally connected subgraphs, while overly low thresholds increase role count and compromise optimality. This necessitates dataset-specific calibration, complicating deployment across diverse environments.

Collectively, these approaches suffer from four fundamental gaps:

- **Methodological Rigidity**: Inability to dynamically transition between heuristic and exact methods based on problem size or resource availability
- **Theoretical-Practical Disconnect**: No middle ground exists between exact methods (with guarantees) and heuristics (with scalability)
- **Reduction Limitations**: Reduction rules are applied only once, missing opportunities for iterative simplification
- **Implementation Constraints**: The MBE heuristic's dataset-specific thresholds and computationally intensive, memory-bound sequential algorithm prevent out-of-the-box applicability

3.4.3. Summary and Contribution Layout

Table 3.1 synthesizes the comparative strengths and limitations of recent approaches. Crucially, existing methods universally lack three critical capabilities: theoretical guarantees for heuristics, optimality-preserving reductions, and dynamic adaptation to varying problem sizes. These gaps directly motivate our framework's architecture.

Method	Heur. Guarantees	Opt. Reductions	Adaptive
MFC-RMA [42]	No	No	No
addRole-EA [2]	No	No	No
Dynamic EA [1]	No	No	No
Set-Cover Approx. [17]	**Yes**	No	No
BMD [23]	No	No	No
Fast Reductions [9]	No	**Yes**	No
MBE [38]	No	**Yes**	**Partially**
Unified Framework (Ours)	**Yes**	**Yes**	**Yes**

Table 3.1: Comparative analysis of bottom-up role mining approaches. **Heur. Guarantees**: Solution quality bounds for heuristic methods; **Opt. Reductions**: Optimality-preserving input reductions; **Adaptive**: Runtime method switching. Our resource-aware framework uniquely achieves all three.

As Table 3.1 demonstrates, existing methods exhibit fundamental limitations: heuristic approaches lack guarantees, application of reduction based methods is limited, and the same goes for techniques that can adjust the method they use based on the size of the underlying graph. This landscape reveals a critical research gap: **no methodology dynamically integrates theoretical guarantees, optimal reductions, and resource-aware adaptation**.

Our framework bridges this gap through:

- **Set cover reformulation** enabling logarithmic-approximation heuristics
- **Optimal biclique reductions** applied directly up to medium sized graphs, and post-heuristic reduction to the larger graphs
- **Four-level resource monitoring** triggering real-time method transitions

This integrated approach delivers consistent solution quality while adapting computational effort to dynamic problem characteristics—a capability absent in prior work.

4

Reframing Reductions in Role Mining: Theoretical Foundations and Design Trade-Offs

This chapter revisits the reduction algorithm introduced by Ene et al. [9] and reinterprets it through the lens of classical kernelization rules from the set cover literature. By explicitly aligning the reduction logic with these well-established rules, we decouple them from specific implementation details and propose a more modular and reusable reduction strategy.

A central insight of our analysis is that achieving optimality with these reductions requires access to the complete set of maximal bicliques before any transformation is applied. This requirement, though implicit in prior work, has not been formally articulated in the literature.

Our motivation for exploring a biclique-based alternative is twofold. First, neighbor-based reductions, as initially proposed, can take several days to complete on large-scale datasets [38]. Moreover, their contribution is fundamental to role mining, but to the best of our knowledge, no research has attempted to suggest alternative implementations. Through such exploration, we may find directions that further enhance the efficiency of these methods. Second, their design typically assumes single-pass application, limiting adaptability in iterative frameworks. Building on these observations, we investigate two key questions: (a) can biclique-based reductions offer improved efficiency under certain conditions, and (b) does this formulation enable more modular and iterative reduction strategies

4.1. Reframing Prior Reductions and Theoretical Foundations

We reinterpret the core reductions of prior role mining methods [9, 38] using classical kernelization rules from the set cover problem. This mapping offers both theoretical clarity and implementation flexibility.

Element domination rule Two user-permission edges, e and f, whose users and permissions can form a biclique, are called neighbors. This occurs when the user associated with edge e has access to the permission of edge f, and the user of edge f has access to the permission of edge e. Notably, these edges may or may not share an endpoint.

A collection of neighbors for an edge includes all edges with which it can form a biclique. If all the neighbors of edge a include all the neighbors of edge b, it implies that for every edge that edge b can form a biclique with, edge a can also form a biclique with the same edge. As a result, every maximal biclique containing edge b will also contain edge a. In role mining, the resulting roles need to cover all the edges. Therefore, knowing a biclique covering edge b will be selected, allows edge a to be temporarily removed and later added back to the biclique of edge a.

These reductions are variations of the element-domination rule of set cover. They can be performed

by conducting subset checks on either the neighbors or the maximal bicliques of the edges. While predecessors opted for the first approach, we will experiment with the second.

Singleton Rule (Isolated-edge rule) If an access-edge (u, p) has no neighboring edges, i.e., cannot form a biclique with any other edge, then it must (for its inclusion) form a one-edge biclique (a singleton role) itself. The algorithms immediately remove that edge and create a role for it. Equivalently, in set-cover terms, such an edge appears in a single candidate set (biclique), which forces the inclusion of that set.

Interestingly, the neighbor-based approach can only recognize an edge as isolated after it has removed all its neighboring edges due to domination. The other edges within that biclique will dominate an edge that is found in only one maximal biclique because they are part of all the bicliques to which the isolated edge belongs—and potentially more. As the neighbor-based approach does not generate the maximal bicliques, it cannot immediately determine whether an edge is part of just one biclique. This identification occurs only after it removes all the dominating edges and discovers that the edge has become isolated.

In contrast, the biclique-based approach, having generated all the maximal bicliques, can identify isolated edges with a single pass. By recognizing these isolated edges, it can also mark for removal the edges that are part of their respective bicliques. Therefore, with a single pass, both the isolated edges and their associated dominator edges can be immediately identified and removed.

Subset Rule In classic set-cover, if one candidate set S is a subset of another T, one can discard S. In the biclique cover context, maximal bicliques are, by definition, not strict subsets of one another and, therefore, a variation of the subset set cover rule as well.

Recasting these reductions as set cover kernelization rules offers three benefits: (1) theoretical soundness, (2) modularity across implementations, and (3) transparency in computational trade-offs. This common abstraction provides a foundation for combining and comparing reduction strategies more systematically.

4.2. Biclique-Based Reductions

In contrast to neighbor-based approaches, we apply reductions at the biclique level. First, we generate all maximal bicliques (a step necessary for optimality, as shown in Section 4.3.1). Then, we iteratively:

(a) Identify bicliques with unique edges and activate them as roles.

(b) Remove dominating edges based on biclique membership.

(c) Discard bicliques that are strict subsets of others.

See Algorithm 1 for the pseudocode implementation of our biclique-based reduction rules.

Algorithm 1 Iterative Biclique-Based Reductions

Require: Set of maximal bicliques M, edge-to-biclique map $\mathcal{E}$, edge frequency map $\mathcal{F}$
Initialize: `coveredEdges` $\leftarrow \emptyset$, `selectedRoles` $\leftarrow \emptyset$
repeat
    `changed` $\leftarrow$ **false**
    `affectedBicliques` $\leftarrow \emptyset$
                                                ▷ Rule A: Singleton edges (freq = 1)
    for all edges e such that $\mathcal{F}[e] = 1$ **do**
        Activate the unique biclique covering e
        Add its users and permissions to `selectedRoles`
        Mark all contained edges as covered
        Add related bicliques to `affectedBicliques`
        `changed` $\leftarrow$ **true**
    end for
                                                ▷ Rule B: Domination via biclique set inclusion
    for all pairs of uncovered edges (e_1, e_2) **do**
        Let $B_1 \leftarrow \mathcal{E}[e_1]$, $B_2 \leftarrow \mathcal{E}[e_2]$
        if $B_1 \subseteq B_2$ **then**
            Mark e_1 as covered; record domination
            Add B_2 to `affectedBicliques`
            `changed` $\leftarrow$ **true**
        else if $B_2 \subseteq B_1$ **then**
            Mark e_2 as covered; record domination
            Add B_1 to `affectedBicliques`
            `changed` $\leftarrow$ **true**
        end if
    end for
                                                ▷ Update bicliques and frequency maps
    Remove covered edges from all affected bicliques
    Remove bicliques that are strict subsets of others
    Rebuild $\mathcal{E}$ and $\mathcal{F}$
until `changed` = **false**
return `selectedRoles`

4.3. Comparison of Reduction Approaches

Table 4.1 summarizes the differences between the two reduction strategies. We further discuss their respective trade-offs below.

Table 4.1: High-Level Comparison of Reduction Algorithms

Neighbor-Based (Predecessor)	Biclique-Based (Ours)
Data Structures: • Dynamic adjacency maps (up_map, pu_map) • `RemovalMap`, `DominatorMap`	**Data Structures:** • Static set of maximal bicliques M • `edgeToBicliques`, `edgeFreq`
Singleton Detection: • Discovered once left isolated after neighbor removals	**Singleton Detection:** • Immediate via edge frequency = 1
Domination Check: • Subset tests on neighborhoods ($N(e) \supseteq N(f)$)	**Domination Check:** • Subset tests on biclique sets
Graph Updates: • Implicit edge removal (no need to rebuild anything)	**Graph Updates:** • Explicit subset biclique pruning and recalculation of data structures
Trade-Off: • Low startup cost; scalable per iteration	**Trade-Off:** • Higher upfront (/rebuild) cost; fast single pass • Strong dependence on the number of bicliques.

4.3.1. Optimality Requires Maximal Bicliques on original graph

A key requirement for ensuring optimality is that global information, in our case maximal bicliques, must be extracted before reductions. To see why, consider the effect of applying reductions such as domination and singleton edge removal: both involve removing edges from the graph. If one were to enumerate maximal bicliques *after* these reductions, the result would omit valid large bicliques that existed in the original graph but were fragmented by prior edge deletions. In such cases, the missing edges are not structurally absent—they were removed by reduction—yet their removal breaks the biclique property, and the larger structure is no longer visible to the algorithm.

Therefore, our approach begins by enumerating all maximal bicliques upfront. This is not an optional cost—it is required to guarantee that no useful structure is lost during preprocessing.

4.3.2. Iterative Reductions and Maintaining Maximality

As edges are removed, bicliques may become subset of others and cease to be maximal. In iterative settings, we must detect and prune subset bicliques, and optionally re-maximize the remaining structures. This step introduces a quadratic bottleneck in the number of surviving bicliques.

In contrast, neighbor-based reductions implicitly adapt as the graph shrinks, avoiding this recomputation. While they are limited in expressiveness, they scale efficiently across iterations.

Looking Forward. The biclique-based reductions have a strong dependence on the number of bicliques. On the other hand, the neighbor-based reductions have a strong dependence on the number of edges. Its interesting to empirically explore, how these different algorithmic designs affect the performance of the algorithm. Furthermore, even if the experimentation does not favor our version, by presenting this equivalence we can open directions for future research to explore the trade-offs and potentials of each approach further.

5

Methodology

We present an innovative four-level framework for *resource-aware* bottom-up role mining. Each level in the framework represents a progressively more computationally intensive yet accurate strategy. A powerful feature of this framework is its ability to automatically transition to higher levels of reasoning as the size of the underlying graph allows. This adaptive mechanism yields two significant advantages. First, it keeps runtimes reasonable by avoiding the use of computationally intensive methods on large graphs. Second, it avoids making impulsive suboptimal decisions by employing approximate strategies only when the problem size surpasses what a more accurate method can handle. In this way, it protects the quality of the produced solution. In summary, this framework consistently applies the most optimal method feasible, enhancing the scalability and optimality of the role-mining task.

This section introduces the four-level framework, followed by detailed explanations of each level and the corresponding algorithms. Finally, we summarize the main idea and the contributions.

5.1. Proposed Four Level Resource-Aware Framework

We propose a novel, four-level framework for bottom-up role mining:

1. **Step 1: Pure Memory-Light Heuristic**: while all the maximal bicliques of the underlying graph do not fit into memory:
   (a) Sort the nodes in decreasing order of degree.
   (b) Generate the first x maximal bicliques of the graph and select the largest one as the role.
   (c) Remove its edges from the graph.
2. **Step 2: Deterministic Reductions**: Generate all maximal bicliques of the underlying graph. While a fix-point has not been reached:
   (a) Isolated edges: Recognize bicliques with unique edges, activate them as roles, and remove their edges from the rest of the bicliques.
   (b) Dominating edges: Remove dominating edges from the bicliques.
   (c) Subset removal: Remove bicliques that are a subset of another biclique.
3. **Step 3: Greedy Role Selection Heuristic** (with theoretical guarantees): while the maximal bicliques are too many for the constraint solver:
   (a) Pick the (globally) largest biclique.
   (b) Remove its edges from all other bicliques.
4. **Step 4: Solver** (optimal): The remaining bicliques are provided to an exact solver using ILP formulation, which identifies the minimal set of bicliques that collectively cover all edges in the graph.

5.2. Maximal Biclique Enumeration algorithm

Both our pure heuristic approach and our reduction strategies are fundamentally reliant on MBE. In the pure heuristic, we iteratively employ MBE to identify potential roles, whereas in the reductions, we use it to generate all maximal bicliques from the remaining graph. Consequently, the performance of the MBE algorithm we adopt directly affects the overall efficiency of our algorithm.

MBE is a vibrant area of research ([41], [28], [29], [25]); therefore, we consulted the latest studies to identify the most effective algorithm currently available. At [29], we identified a solution that met our research needs: achieving state-of-the-art performance with minimal computational resources and an open-source implementation. We enhanced their implementation to retain all maximal bicliques encountered during the enumeration process. We developed a memory-efficient variant that preserves only the largest biclique discovered and terminates after producing a user-defined number of bicliques.

Importantly, we noted inefficiencies associated with the MBE algorithm utilized in prior works [38] despite employing a combination of three distinct methodologies ([5], [37], [22]), the predecessor's algorithm demonstrates difficulties in scaling to substantial input sizes, specifically above 10000 edges on commodity hardware. To substantiate our claims, we conducted empirical comparisons between the state-of-the-art MBE algorithm we utilize and that of our predecessor, which we present in our experimental evaluations.

5.3. Pure Memory-light Heuristic

When the number of maximal bicliques within an input graph exceeds the available memory capacity, we employ a lightweight heuristic that avoids storing any bicliques. Its algorithm involves iteratively enumerating up to a user-defined threshold x of bicliques and retaining only the largest biclique encountered. This biclique is activated as a role, and its edges are removed from the graph. The process repeats until the number of maximal bicliques in the graph is sufficiently small to transition into Level 2.

This heuristic is designed for scalability and is effective even in massive graphs. However, it is important to acknowledge its inherent sub-optimal nature. First, even though the largest observed biclique is promising, its presence within an optimal solution is not guaranteed. Second, by removing edges early, we may eliminate promising biclique structures that would appear valuable in later stages. Nonetheless, by constraining the threshold x and triggering a transition when the number of total bicliques falls below a threshold y, we minimize non-optimal decisions while maintaining computational feasibility. Note that if the number of maximal bicliques in the input graph is already below the threshold, the pure heuristic will not select any roles, and the algorithm will proceed immediately to the next layer.

5.4. Biclique-Based Reductions

Once we have identified that all maximal bicliques can be loaded into memory and are, in a sense, a manageable amount, we generate all the maximal bicliques and apply our reductions. In contrast to our predecessor's approach, we apply the reductions by observing the maximal bicliques rather than dynamic neighborhoods in the user-permission graph.

No loss of optimality: These reductions apply deterministic choices and, based on the set cover theory, do not affect the optimality of the solution. They allow us to reduce the maximum bicliques, make them more compact, and reach a size that the solver can solve sooner, i.e., with fewer suboptimal decisions.

5.5. Greedy Role Selection Heuristic

The well-established "Greedy" heuristic for the Set Cover problem operates by iteratively selecting the globally largest candidate set, aiming to cover all edges with the least number of subsets. This heuristic is recognized as the most natural and effective heuristic for the set cover problem [36]. By relating our problem to this well-known NP-hard problem, we can effectively utilize its algorithms and theoretical insights.

As a reminder from 2.3, if the optimal solution uses m candidate roles to cover all the $|E|$ edges, then the Greedy Role Selection Heuristic will return a solution that uses at most $m * (ln(|E| + 1)$ candidate

roles. This property was lacking in prior heuristics and presents a significant bound on role set quality, particularly for mid-sized instances.

The main difference between this approach and the pure heuristic is the following: The pure heuristic removes covered edges from the graph itself, which can inadvertently invalidate other promising overlapping bicliques. Here, by working with already generated bicliques, we can still access all the bicliques, even after removing covered edges from them. The pure heuristic may sacrifice global information for the sake of scalability, while the greedy heuristic will not. Our experimental section presents a comparative analysis of the performance of the pure and greedy heuristics, evaluating their respective efficiencies and outcomes.

When the number of bicliques remaining becomes sufficiently small, allowing the solver to find an optimal solution, we transition from the greedy approach to the exact one.

5.6. Optimal Role Selection approach

This step provides a minimal role set at the cost of potentially high computational. Therefore, if the number of bicliques becomes sufficiently small, we formulate the role-mining problem as a binary integer program and invoke Gurobi to compute an optimal solution. Each variable represents a biclique (candidate role), and the constraints ensure complete edge coverage.

5.7. Overall Strategy and Contribution

On a high level, the inner layers provide better quality solutions but require more computations. Therefore, the more decisions the algorithm makes on the outer layers, the more suboptimal these decisions will be, and the less optimal the solution will also be. Conversely, the more decisions the algorithm makes on the inner layers, the more optimal these decisions will be, and the more optimal the overall solution will be.

The key advantage of this framework lies in its adaptive design: the algorithm consistently utilizes the most optimal functional technique. When input size permits, the algorithm skips outer layers, such as Pure and Greedy, entirely.

This architecture, to the best of our knowledge, is the first role-mining framework that explicitly integrates theoretical guarantees, lossless reductions, and exact optimization within a resource-sensitive pipeline. Importantly, it ensures that every non-optimal decision made is due to necessity, not design, and lays the foundation for delivering the most optimal solution available resources can provide.

We empirically evaluate this framework and its components in the experiments that follow.

6

Experiments & Results

Understanding the empirical behavior of our role-mining framework is critical to validating its practical utility. This chapter presents a systematic evaluation of each component in the proposed four-level pipeline: MBE, biclique-based reductions, greedy and pure heuristics, and the exact solver across multiple datasets. The choice and placement of each algorithm in our pipeline is justified by (1) the computational resources it requires and (2) its performance.

Our experiments are structured to address two core research questions:

1. **Reduction Correctness and Generality:** Do our biclique-based reduction rules perform the same logical eliminations as the edge-centric methods in prior work, and how do their algorithmic dependencies affect performance and applicability?
2. **Heuristic Performance and Generalization:** How effective are our pure and greedy heuristics in practice? How closely do they approximate optimal solutions, and to what extent do their theoretical guarantees translate into real-world performance? Furthermore, how does the pure heuristic mitigate memory bottlenecks while preserving quality?

To this end, we conduct the following empirical investigations:

- **MBE:** We compare the enumeration performance of our implementation against that of prior work, focusing on scalability and completeness.
- **Reduction Analysis:** We verify that our biclique-based reductions yield equivalent results to the neighbor-based rules of predecessors while enabling greater flexibility and more transparent dependency management.
- **Heuristic Evaluation:** We measure the effectiveness and computational requirements of both the greedy and pure heuristics across real and synthetic data. We demonstrate that in practice, both heuristics consistently outperform the worst-case theoretical bounds while achieving near-optimal solutions.
- **End-to-End Pipeline Validation:** Finally, we evaluate the complete framework on 31 corporate datasets. We compare overall runtime, memory usage, and final role set size to a baseline implementation that mimics the structure of earlier state-of-the-art systems.

We begin by detailing our experimental setup, including the datasets, implementations, hardware, and chosen thresholds.

6.1. Experimental Set-up

We now present the datasets, hardware, and implementations used in our experiments.

6.1.1. Datasets

We evaluate our methods on three distinct suites designed to test scalability, realism, and robustness:

- **Real-World**: A proprietary dataset comprising 31 access-control graphs extracted from a large enterprise with approximately 80,000 users. Each graph corresponds to a department or team and reflects authentic user–permission assignments. In the initial teams, we observe a variety of users and permissions, with numbers ranging from tens to a few thousand users. Additionally, the number of edges ranges from a few hundred to approximately ten thousand. In larger datasets, the number of users and permissions reaches several thousand, and the total number of edges exceeds a million.
- **RMP Library** [3]: A curated set of publicly available benchmarks traditionally used in the role mining literature. These include small, medium, and large graphs, enabling comparisons with prior methods and access to ground-truth optimal solutions for smaller instances. The specific characteristics of these datasets are shown in Figure 6.1. Additionally, a dataset called *COMP*, which includes 16 larger graphs, exists, and we leverage it for full framework validation. Information about COMP can be found in the Appendix Table A.3.
- **Synthetic Dataset**: We generated bipartite graphs using two controlled strategies. In the first, we control the number of bicliques and the input size, which are the number of permissions and the number of users. In the second step, a controlled set of seven bipartite graphs is generated, all with a fixed number of bicliques and input size and varying only the edge count. This approach enables precise comparison of specific algorithmic behaviors (especially in MBE and reductions) under different edge densities while holding structural parameters constant.

Table 6.1: RMP data information.

Folder	File	Users	Perm.	Edges
small	1	49	44	600
small	2	50	48	1082
small	3	49	96	1369
small	4	50	88	1932
small	5	99	93	1372
small	6	99	96	2152
small	7	99	193	9371
small	8	100	184	4415
medium	1	499	479	15567
medium	2	500	468	33959
medium	3	500	427	22988
medium	4	499	883	23949
medium	5	499	980	47674
medium	6	500	924	48058
large	1	999	910	60288
large	2	999	992	49579
large	3	999	910	23778
large	4	999	3446	74347
large	5	1000	3522	148067
large	6	999	3545	62292

6.1.2. Implementations and Hardware

All implementations, ours and the re-implemented baselines, are open-sourced and publicly available at: `https://github.com/ckindynis/proper-role-mining`.

Maximal Biclique Enumeration: Our approach (shown as *MBE SOTA* in the plots) uses the sequential variant of Pan et al.'s state-of-the-art MBE algorithm without GPU acceleration. For the pure heuristic stage, we further modified this algorithm to enhance memory efficiency and allow early termination after generating a user-defined number of bicliques. As a baseline for comparison, we include the predecessor algorithm developed by Tripunitara et al. [38] (shown as *MBE Predecessor* in the plots).

Reduction Rules: Our method applies biclique-based reductions (shown as *Biclique-based* in the

plots), as detailed in Section 4.1. For comparison, we evaluate our approach against those of the predecessor's neighbor-based reduction rules, which follow the approach introduced by Ene et al. [9] (shown as *Neighbor-based* in the plots).

Hardware: Experiments were conducted on corporate datasets using a machine equipped with an Intel i5-1245U processor (1.6 GHz) and 16 GB of RAM. For the RMP Library and synthetic datasets, we used an AMD Ryzen 7 4800H processor (2.9 GHz), also with 16 GB of RAM.

Parameters and Thresholds
All thresholds used in our framework were informed by preliminary empirical calibration. Our goal was to identify values that strike a balance between runtime feasibility, memory constraints, and solution quality.

- **Reduction Threshold** $(|B|_{\text{max}}^{\text{reduction}} = 500{,}000)$
  This threshold determines when the algorithm transitions from the pure heuristic stage to the reduction phase. Its value is guided by the maximum number of maximal bicliques that can be safely held in memory on our hardware configuration. Through repeated runs on medium-to-large graphs, we observed that exceeding this threshold resulted in memory usage consistently spiking above 4GB, leading to degraded performance or system-level failures. Thus, the 500,000 cutoff reflects a *hardware-informed upper bound* that balances computational feasibility with role quality.
- **Enumeration Cap per Iteration in Pure Heuristic** $(|B|_{\text{iter}} = 1{,}500{,}000)$
  This threshold defines the number of bicliques we enumerate and consider before selecting one as a role in our pure heuristic. Intuitively, the more we consider, the more informed our selection will be. An intuitively practical lower bound on the value of this threshold is ($|B|_{\text{max}}^{\text{reduction}} = 500,000$). To be observed when we have fewer than $|B|_{\text{max}}^{\text{reduction}}$ bicliques and proceed to our reductions, we must at least enumerate that many.

  In practice, we extend this enumeration limit to 1.5 million bicliques per iteration. We noticed that, due to the MBE's efficiency, the enumeration of these additional bicliques does not introduce a computational overhead; however, it does offer us the chance to view more bicliques before making a selection.
- **Solver Threshold** $(|B|_{\text{max}}^{\text{ILP}} = 200{,}000)$
  When the number of remaining bicliques falls below this cap, we invoke the Gurobi ILP solver to compute an exact minimal role set. We experimented with a range of values and consistently observed that when the candidate biclique set was below 200,000, the solver could reliably return optimal solutions within a short amount of time. For values above this threshold, solution times became highly variable and, in some cases, unbounded within practical time limits that allow multiple experiments. Thus, the 200,000 threshold reflects the empirically observed reliability boundary of the solver under our experimental constraints.
- **Solver Timeout**: 30 seconds
  Due to the variability in Integer Linear Programming (ILP) solution times and the need for consistent execution in our pipeline, we have implemented a 30-second timeout for solver runs. If this time limit is exceeded, the pipeline will automatically revert to the greedy heuristic.

Importantly, we set these values in a way that allows us to conduct various experiments with our computational resources and the time constraints of our research. For deployment in a real-world scenario, we recommend incrementally increasing these thresholds, particularly the solver and reduction limits, while monitoring for solver timeouts and spikes in memory usage. This flexibility allows our framework to serve not only as a research tool but also as a practical engine for effective role mining in real-world RBAC systems.

6.2. Results

This section presents the empirical results of our evaluation, organized around the key components of the framework and the two core research questions: (1) (a) Do our biclique-based reductions produce the same logical effects as predecessor methods? (b) If so, are there cases where they offer more efficient computation? (2) How well do our heuristics, pure and greedy, approximate optimal solutions?

Based on performance and computational requirements, is the placement of the pure heuristic on the outer layer and the greedy on the inner layer justified?

6.2.1. Maximal Biclique Enumeration algorithms

To evaluate the performance of our MBE component, we compared it against the predecessor's implementation across all three dataset suites. The results are presented in Figures 6.1 to 6.5.

Figures 6.1 and 6.2 illustrate the scalability difference between our MBE and the predecessor's, both in terms of time and memory usage in artificial datasets.

Varying input size and number of edges For Figure 6.1 we dynamically increase the input size along with the number of edges. As input size increases, the predecessor's runtime and memory consumption grow steeply, while our method maintains near-constant efficiency. For instance, at input size 300, the predecessor exceeds 850 seconds and 600 MB of memory, whereas our method remains under 1 second and under 10 MB.

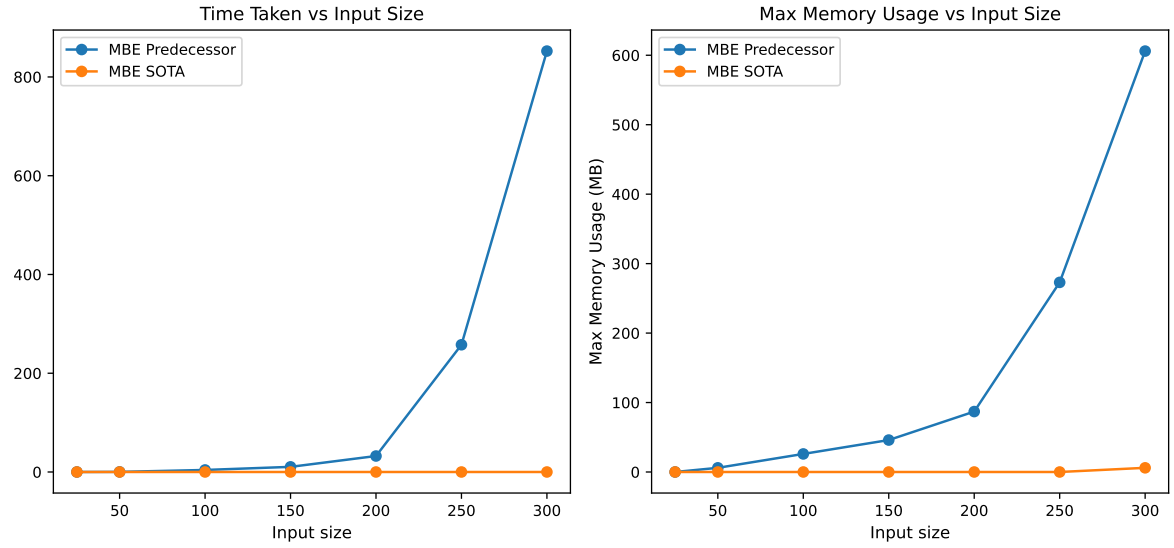

Figure 6.1: Runtime and memory usage comparison of predecessor's MBE and state-of-the-art MBE algorithm we leverage (Number of users = Number of permissions = Input size)

Fixing input size, number of bicliques and increasing only the number of edges A similar performance gain is observed in Figure 6.2 under the more controlled experiments, where fix the number of users (175), the number of permissions (175), and the number of bicliques (122), and vary only the number of edges.

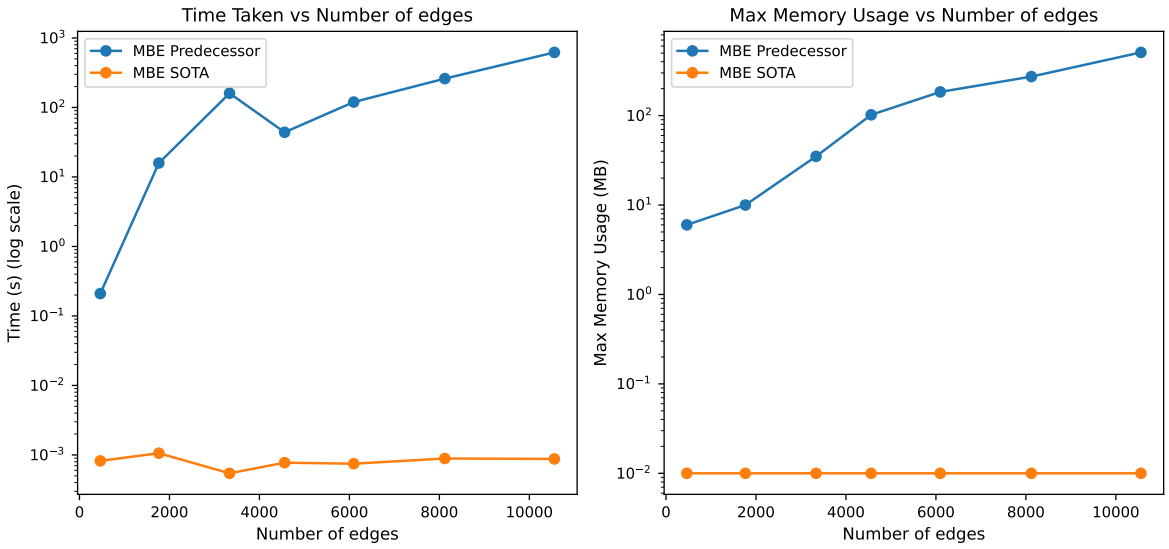

Figure 6.2: Runtime and memory usage comparison of MBE algorithms with increasing edge density (artificial data)

Running the new MBE on a larger scale To further evaluate the scalability of our MBE algorithm, we extend the previous experiment to significantly larger graphs, with an input size of 15,000 and over 10,500 bicliques. The results, shown in Figure 6.3, use a linear (non-logarithmic) scale to highlight absolute performance. Remarkably, the algorithm processes graphs with up to 80 million edges in under 1.75 seconds, demonstrating not just efficiency but exceptional speed.

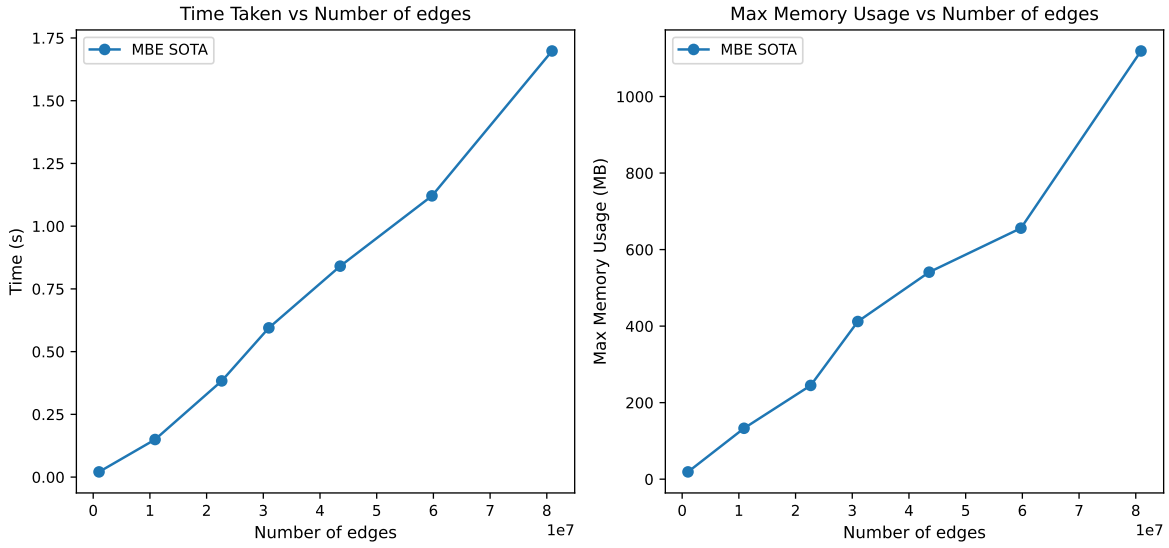

Figure 6.3: Scalability of the new MBE algorithm on large artificial graphs

To step away from any potential caveats associated with controlled artificial datasets, we also verify the difference in performance on the RMP and real-world data.

RMP data In RMP data, we notice the difference in efficiency immediately from the small dataset. In Figure The predecessor's MBE terminates with the 8 hour only for three out of eight instances, while the MBE we leverage successfully terminates in all the datasets in less than a minute. Note that dataset seven is the one with the most number of edges (see Table 6.1) which justifies the increased runtime.

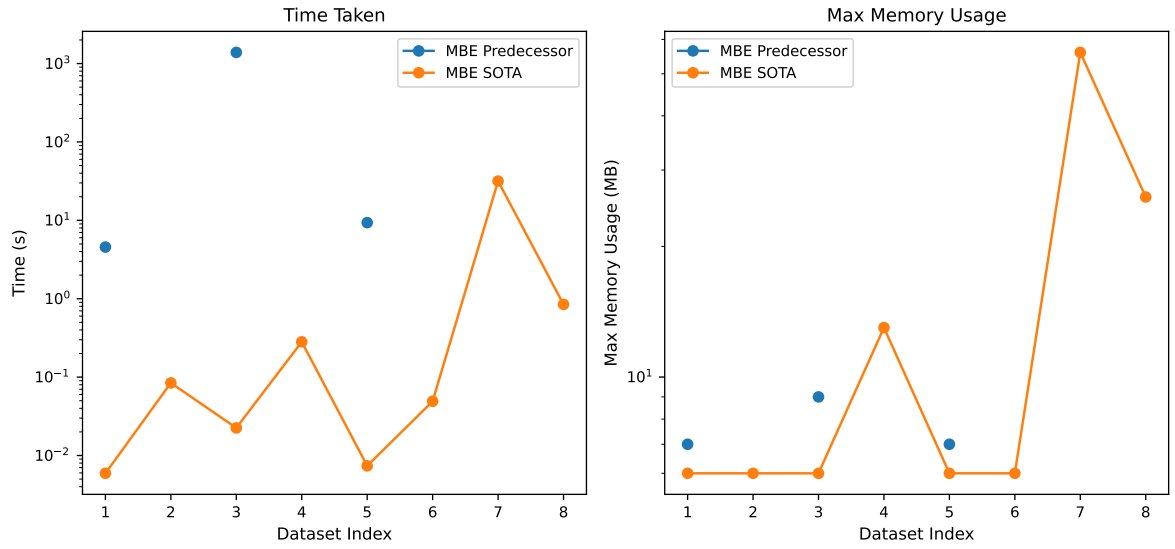

Figure 6.4: Runtime and Memory usage comparison of the new MBE and the predecessor's MBE algorithm on RMP-small datasets (Predecessor's approach time out without providing a solution after eight hours)

Real-world data Figure 6.5 verifies the significant efficient gains of the new MBE algorithm on the real-world datasets as well. Here, our MBE consistently completes in milliseconds, while the predecessor's method quickly becomes impractical, exceeding 10,000 seconds from the fifth in size dataset. Specifically on that dataset, the predecessor required over 14,000 seconds, while our method completed in 0.000033 seconds. Furthermore, the predecessor fails on dataset #8, timing out after eight hours, while our method processes it in just 0.0004 seconds. Notably, the predecessor's MBE approach also timed out for the subsequent datasets, whereas our approach successfully processes even the larger datasets in a matter of seconds. Lastly, we note that the state-of-the-art MBE algorithm we leverage is also more efficient in terms of memory.

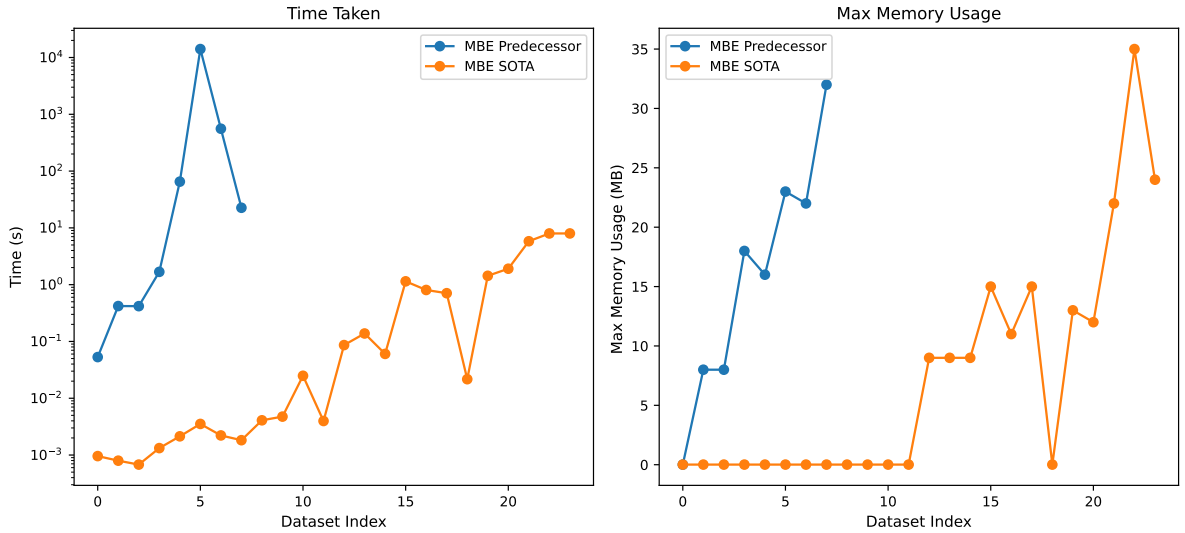

Figure 6.5: Runtime and memory usage comparison of the new MBE algorithm and predecessor's MBE algorithm on real-world data (Predecessor's approach time out without providing a solution after eight hours)

Implications We have incorporated a significantly more efficient MBE algorithm than the one used by our predecessors, which brings three important enhancements to our role-mining framework. We examine these by revisiting how our reduction method and pure heuristic work.

Firstly, before applying the reductions, it is necessary to enumerate all maximal bicliques. Dependence on an inefficient MBE approach could hinder our ability to complete the crucial MBE step in a timely manner for larger graphs. Therefore, while the size of the problem would suggest that a more optimal approach is possible, an MBE inefficiency would hinder our ability to transition to the more accurate method. As a result, we would spend more time on the pure heuristic approach, compromising the quality of our results. Therefore, we establish that selecting a more efficient MBE algorithm ensures we can efficiently and as early as needed transition to the more optimal approaches and, as a result, safeguard the quality of the roles we produce.

The pure heuristic employs the MBE algorithm multiple times, once for each role selection. Importantly, as the first algorithm in our framework, the pure heuristic is essential for managing graphs of any size. Relying on an efficient MBE algorithm directly enhances the scalability of our whole framework.

In each iteration, the pure heuristic enumerates a specific number of maximal bicliques. By employing an efficient MBE algorithm, we can quickly generate a significant number of maximal bicliques. These bicliques are candidate roles from which the pure heuristic can choose. Their increase enables the pure heuristic to make more informed decisions, ultimately enhancing the quality of the roles it produces.

Overall, with such an efficient MBE algorithm, we gain **three significant performance enhancements**:

- We guarantee that the pivotal MBE step, from the pure heuristic to the reductions and more optimal methods, can be reliably and efficiently performed, ensuring that this transition occurs when available, thereby protecting the quality of the results.
- We are confident that our pure heuristic can be efficiently applied to large graphs, ensuring that our framework can handle graphs of any size.
- In each iteration of our pure heuristic, we can enumerate and consider a considerable amount of maximal bicliques (candidate roles). This capability enables it to make well-informed role selections (that are nearly globally promising), ultimately allowing us to achieve near-optimal results (see 6.2.3).

Overall, these results confirm that, through the new MBE algorithm, we are not simply saving computational time; we fundamentally shift the scalability boundaries and improve the quality of results of the entire role-mining framework.

6.2.2. Reduction algorithms

To evaluate our proposed biclique-based reduction strategy against the predecessors' neighbor-based reductions, we conducted a comparative analysis across real-world literature and artificial datasets. This analysis directly addresses **RQ1**, which investigates whether re-framing reduction rules using set cover theory first gives the same reduction and second yields any performance advantages.

From the RMP data, we observed that our reduction algorithm struggled to terminate in medium and large datasets; therefore, we focused on presenting the results for small instances. Additionally, for the seventh file of the small RMP dataset, since the number of maximal bicliques exceeds our threshold, we used our pure heuristic and, therefore, omitted this dataset from the reduction experiments. For the real-world datasets, we utilize 26 datasets for which the pure heuristic was not required, again to ensure a common ground for the two reductions.

Table 6.2 demonstrates that both reduction methods eliminate the same number of edges and identify an equal number of isolated edges. For each isolated edge, a corresponding role is created, and the counts of these roles are presented in the final two columns.

Table 6.2: Number of edges removed and the number of roles promoted by Predecessor's reduction approach (NB (Neighbor-Based)) and Ours (BB (Biclique-Based)

Folder	File	Edges	Rem. Pred	Rem. Ours	Roles Pred	Roles Ours
small	1	600	417	417	4	4
small	2	1082	581	581	1	1
small	3	1369	1369	1369	25	25
small	4	1932	1196	1196	0	0
small	5	1372	1372	1372	49	49
small	6	2152	1108	1108	3	3
small	8	4415	2877	2877	3	3

In Figure 6.6, we present the run time and memory usage of the reduction algorithms for the small instances of the RMP data. In all instances, the predecessor's approach outperforms our implementation in terms of both runtime and memory usage. These results indicate that the balance between the number of edges and the number of bicliques in these datasets does not favor our approach.

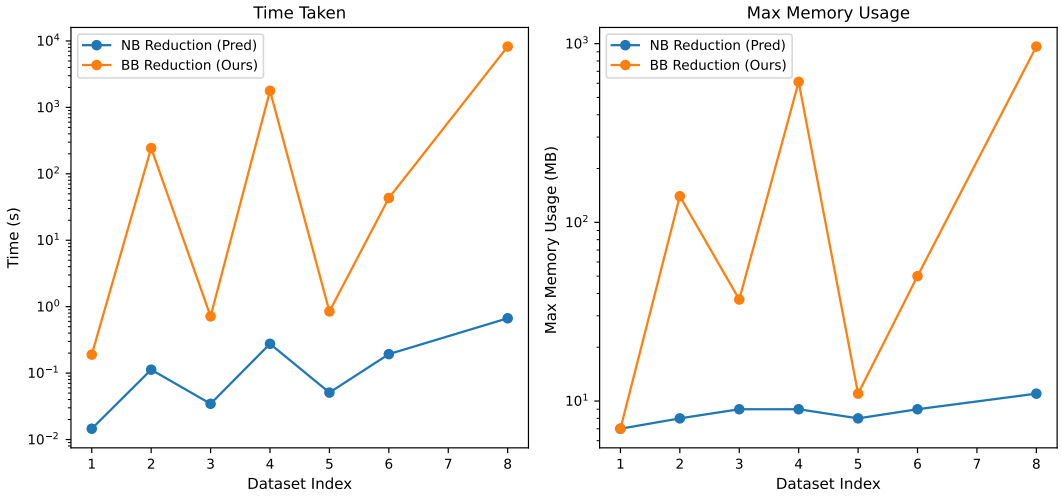

Figure 6.6: Runtime and memory usage comparison of our vs predecessor's reduction algorithms, on RMP small data. A logarithmic scale is used to better capture the differences in execution time, particularly for the predecessor's approach.

In Table 6.3, we notice the **equivalence of the reduction methods** by observing that the two versions have the same effects again, this time in the real-world data. We remove the same number of edges ("Rem.") and promote the same number of roles (through the identification of bicliques with isolated edges).

Additionally, we observed the substantial impact the reduction rules had on these real-world datasets. In the 26 datasets where the pure heuristic was not applied, **not even 1% of the initial edges remained after the reductions**. Thanks to these reductions, we **successfully found the optimal solution for all 26 datasets presented here**; more details on that are provided in Section 6.2.4.

Table 6.3: Number of edges removed and the number of roles promoted by Predecessor's reduction approach (NB (Neighbor-Based)) and Ours (BB (Biclique-Based), along with their effectiveness on real-world data.

File	Edges	Rem. NB	Rem. BB	% Edges Left	Roles NB	Roles BB
0	171	171	171	0.00	7	7
1	306	306	306	0.00	9	9
2	514	514	514	0.00	18	18
3	1471	1471	1471	0.00	20	20
4	2587	2587	2587	0.00	55	55
5	2751	2751	2751	0.00	85	85
6	2810	2791	2791	0.68	52	52
7	4244	4238	4238	0.14	139	139
8	4263	4247	4247	0.38	93	93
9	6215	6205	6205	0.16	109	109
10	12810	12810	12810	0.00	151	151
11	13119	13119	13119	0.00	69	69
12	17573	17543	17543	0.17	262	262
13	27339	27330	27330	0.03	280	280
14	50328	50322	50322	0.01	548	548
15	50618	50606	50606	0.02	376	376
16	52511	52472	52472	0.07	222	222
17	53203	53203	53203	0.00	804	804
18	65172	65165	65165	0.01	551	551
19	66824	66807	66807	0.03	546	546
20	115277	115257	115257	0.02	977	977
21	124130	124014	124014	0.09	1215	1215
22	133217	133087	133087	0.10	1427	1427
23	164213	164132	164132	0.05	1351	1351
25	253750	253363	253363	0.15	890	890
27	437478	437091	437091	0.09	1991	1991

Figure 6.7 compares the runtime and memory usage of our biclique-based reductions with those of the predecessor's neighbor-based approach on real-world data. The results are unambiguous: the predecessor's method is significantly more efficient, with 15 out of 26 datasets showing 10–100× faster execution. This is primarily due to the algorithmic complexity profiles. The predecessor's neighbor-based reductions operate in time cubic in the number of edges ($O(|E^3|)$), while our biclique-based method depends on the number of maximal bicliques ($O(|B|^2)$ for subset checks). On large datasets, the number of maximal bicliques can grow much faster than the number of edges, causing our approach to become memory-bound and computationally expensive, especially as graphs scale.

Despite this, our analysis confirms that both methods achieve identical reductions and produce the same number of roles (Table 6.3), validating the equivalence of our set cover–inspired approach. However, the practical implication is clear: for real-world enterprise data, the predecessor's edge-centric method is preferable for efficiency.

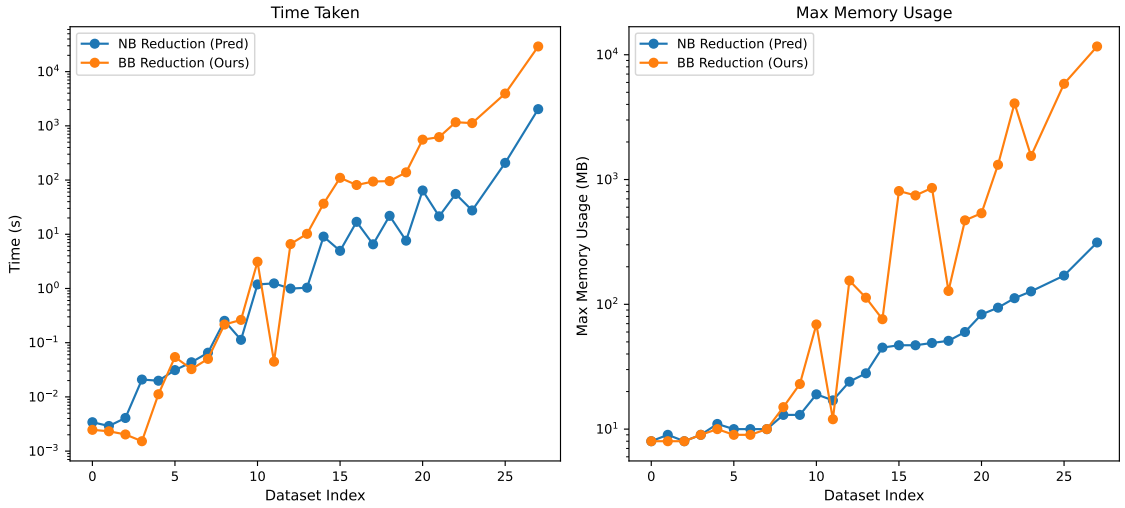

Figure 6.7: Runtime and memory usage comparison of our vs predecessor's reduction algorithm, on real-world data

Implications of the effectiveness of the reductions These impressively effective reductions are a decisive advantage of our approach compared to many recent works. We substantially simplify large input graphs into much smaller and manageable instances, which our optimal methods can easily solve. In fact, we successfully found the optimal solution to all 26 datasets in which the pure heuristic was not used.

This set of deterministic reductions enables us to effectively reduce the input size without compromising optimality. As a result, we can utilize stronger algorithms in larger graphs and achieve better solutions than other role-mining approaches that do not leverage these reductions. This advantage is also why the method we built upon, developed by our predecessor [9], outperformed all other role-mining approaches across multiple datasets, as documented in the comprehensive survey by Mitra et al. [26].

Performance analysis of the two reduction algorithms on artificial datasets To verify the claim that the two reduction versions mainly depend on different parameters, (a) the predecessor is on the number of edges, and (b) ours is on the number of bicliques, we conduct a few experiments on artificial datasets where we control these two parameters. We set the number of users and permissions to the same "input_size" value. We also set the number of maximal bicliques and vary the number of edges. First remark: keeping the number of users, permissions, and bicliques fixed and varying the number of edges is not a trivial task. Therefore, we do not have complete control over the values of these parameters, but we were able to achieve this under six different scales of edge densities. Second remark: We are aware that this approach may not accurately capture common real-world data patterns; however, we are using it to gain clarity on the performance complexities of our algorithms.

Figure 6.8 presents the first results under the configuration of 750 input size, 525 maximal bicliques, and the displayed varying number of edges. We observe that when the number of bicliques is relatively low compared to the number of edges, our algorithm performs better. Even as the number of edges increases, the runtime of our algorithm remains consistently low (below 10 seconds), in contrast to the neighbor-based approach used by previous methods, which can exceed 1000 seconds in runtime. This result validates the design principles behind our algorithms, indicating that the predecessor's approach is more heavily affected by the number of edges than ours.

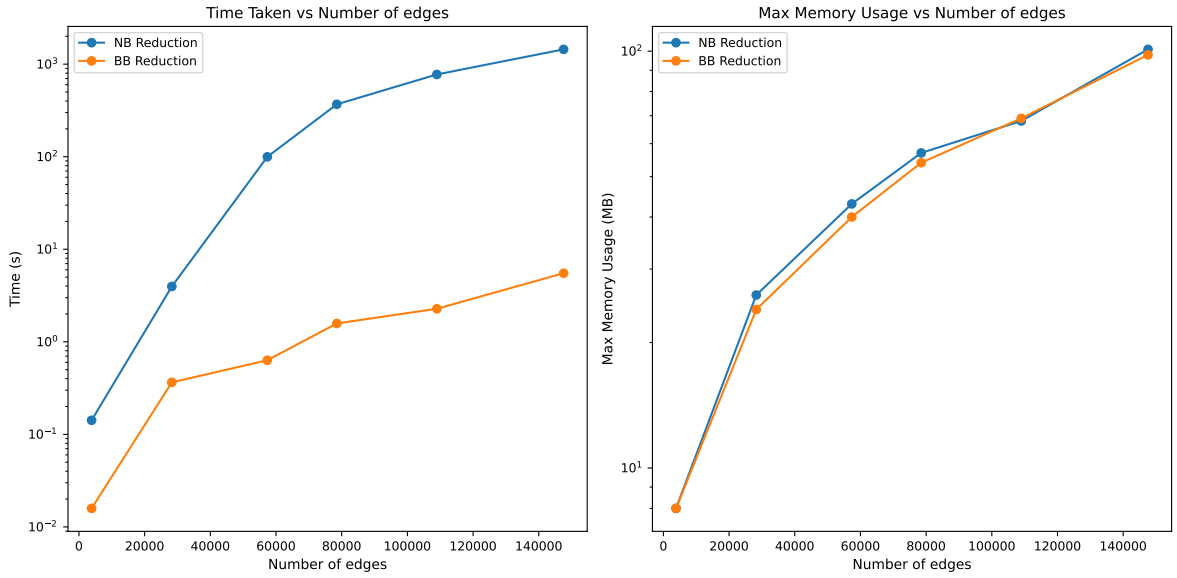

Figure 6.8: Runtime and memory usage comparison of our and predecessor's reduction algorithms on artificial data with low number of bicliques

Implications Our analysis confirms that both reduction approaches achieve identical structural simplifications while preserving solution optimality, validating their theoretical equivalence. This work provides foundational value to the research community in three key areas: First, by establishing the direct connection between classical role mining reductions and set cover kernelization rules, we enable principled extensions to these methods. Second, our observation that biclique-based reductions operate on fixed structures per iteration—unlike the neighbor-based approach with dynamic neighborhoods—reveals promising parallelization pathways worthy of exploration. Third, by open-sourcing both variants, we provide implementation transparency that facilitates direct comparison and future extendability.

The exceptional effectiveness of these reductions, consistently eliminating more than 99% of edges in real-world graphs, demonstrates their critical role in scalable role mining. While the predecessor's edge-centric method demonstrates superior efficiency in our benchmarks, the optimal choice depends on the characteristics of the dataset. Practically, our framework's dual-implementation strategy delivers significant operational advantages: Practitioners can empirically optimize by benchmarking variants against specific RBAC environments, researchers gain extensibility for exploring hybrid or parallelized strategies, and organizations achieve deployment flexibility through dynamic selection of reduction engines based on real-time graph metrics.

6.2.3. Heuristics

This section evaluates the trade-offs between our two heuristic strategies: the Pure Heuristic, which is designed for minimal memory usage and scalability, and the Greedy Heuristic, which is based on a set cover formulation with a provable logarithmic approximation guarantee. The experiments aim to assess how each approach balances solution quality, memory consumption, and runtime performance, thereby addressing **RQ2**.

Pure Heuristic vs Greedy Heuristic

Figures 6.9, 6.10, and 6.11 present results on the RMP benchmark suite. The Greedy Heuristic consistently produces role sets much closer to the optimal. For example, the greedy solution typically falls within 5–15% of the optimal number of roles, while the pure heuristic can deviate more significantly. However, this accuracy comes at a high memory cost. Across all RMP datasets, the greedy variant requires up to 3 orders of magnitude more memory than the pure heuristic 6.11. The minimal memory requirement of the pure heuristic, consistently staying below 100 MB, confirms its suitability for deployment in constrained environments or for handling huge graphs. Figure 6.10 illustrates that both the

Greedy and Pure heuristics surpass the worst-case theoretical upper bound of the Greedy Heuristic, with the greedy achieving that often with an order of magnitude better performance.

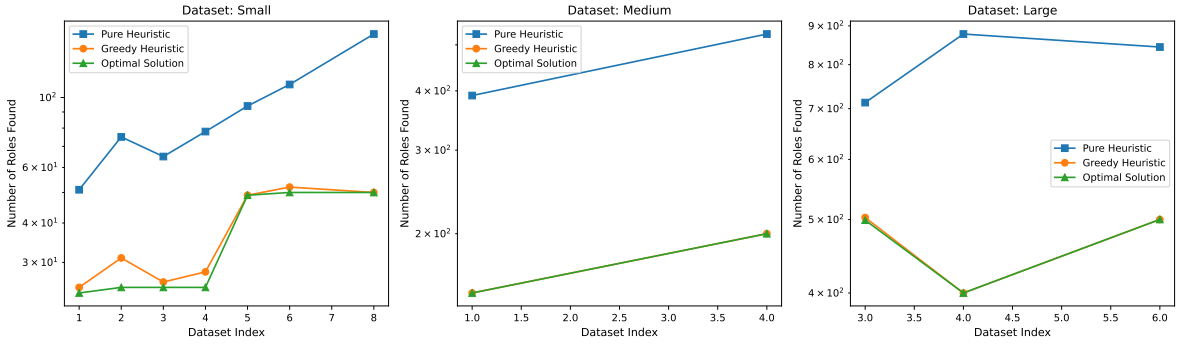

Figure 6.9: Role count comparison of pure heuristic, greedy heuristic, and optimal solutions on RMP data

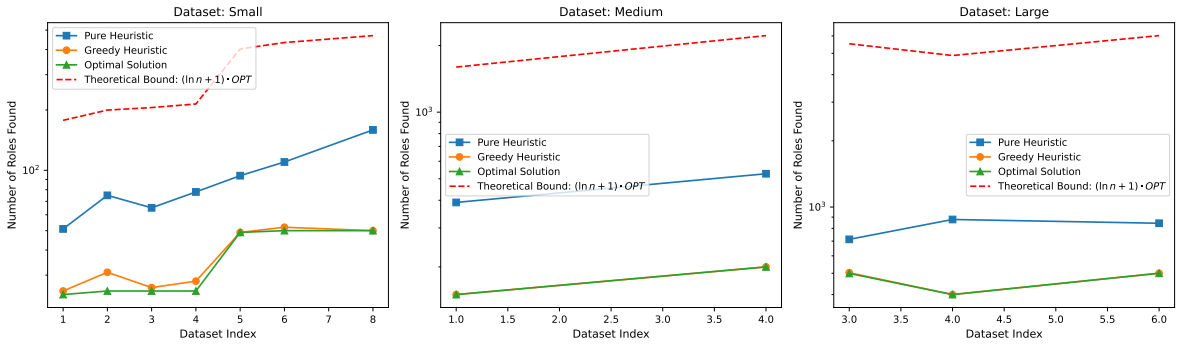

Figure 6.10: Greedy heuristic performance relative to the theoretical upper bound on RMP data

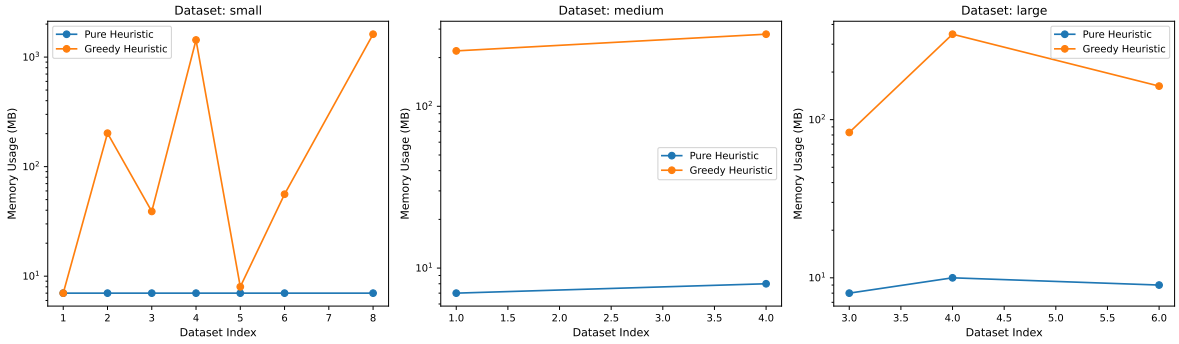

Figure 6.11: Memory usage of pure heuristic versus greedy heuristic on RMP data

A similar argument can be made about the pure heuristic runs on the real-world datasets. We look at the results in Figure 6.12. We immediately notice the minimal memory requirement that the Pure Heuristic comes with again. Additionally, we observe the impressive performance of both of our heuristic methods.

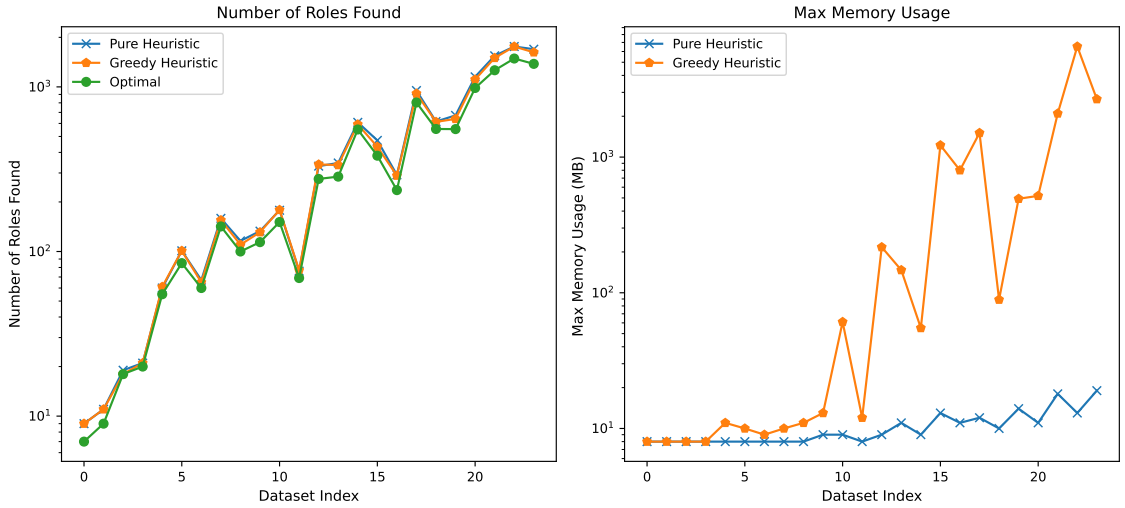

Figure 6.12: Role count comparison and memory usage of pure heuristic and greedy heuristic on real-world data

Implications Thus, the Greedy Heuristic offers **near-optimal results**, with slightly better results than the Pure Heuristic, while the Pure Heuristic provides near-optimal results with **minimal memory requirements**.

Pure Heuristic: Ablation study

To further investigate the Pure Heuristic's behavior, we analyze how node ordering impacts its role quality and progression toward Level 2 reductions. Figures 6.13 and 6.14 show the results of sweeping through three ordering strategies: no reordering, ascending by degree (RInc), and descending by degree (RDec).

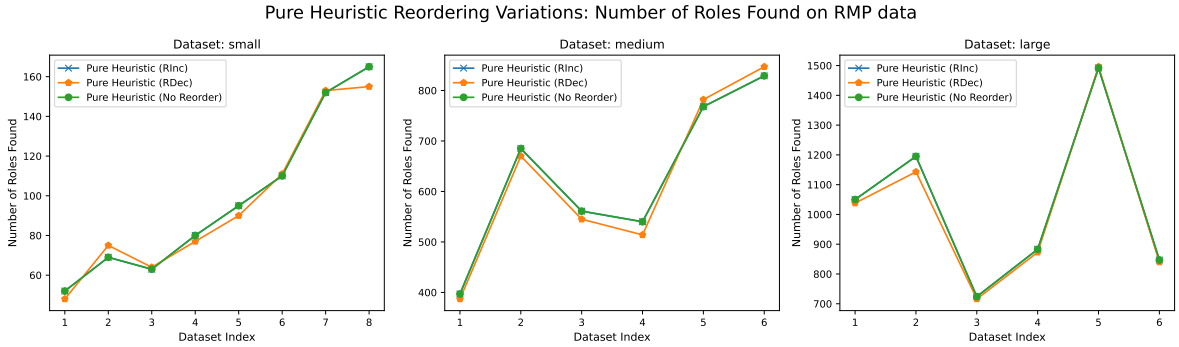

Figure 6.13: Effect of node reordering on pure heuristic role quality (RMP data)

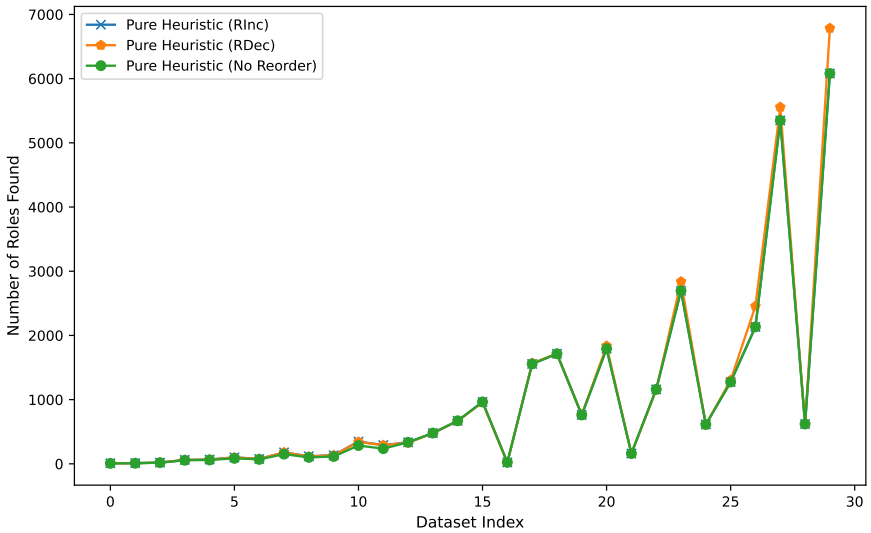

Figure 6.14: Effect of node reordering on pure heuristic role quality (real-world data)

We find that reordering can affect role quality in edge cases, especially when the cap on biclique enumeration is tight. However, for most datasets, the number of roles produced remains stable across reordering strategies, indicating robustness. Notably, ascending degree order (RInc) performs slightly better on large datasets, likely due to the algorithm prioritizing less-connected nodes first and minimizing overlap. This stability allows for flexible deployment: one can trade off between runtime (by skipping reordering) or role quality (by applying the most suitable order), depending on application needs.

6.2.4. Real world validation of the complete framework

The empirical evaluation of our four-level adaptive framework on 31 enterprise datasets and the RMP library directly addresses **RQ3**: *Can we develop a role mining methodology that dynamically decides between heuristic and exact methods to adaptively balance scalability and solution quality?* Our results demonstrate that the framework successfully integrates resource-aware decision-making with formal guarantees, delivering practical benefits in both security and operational efficiency.

Operational Impact: Role Reduction

Our framework consistently produces much more compact RBAC configurations than manually engineered systems, as shown in Table 6.4. Across all datasets, the average role reduction is **53.17%** (ranging from **31.30%** to **89.89%**). This reduction directly translates to:

- **Reduced attack surface**: Fewer roles imply 50% fewer privilege escalation paths, significantly lowering vulnerability risks.
- **Lower administrative overhead**: Compact role sets simplify policy management and reduce manual intervention.
- **Simplified compliance auditing**: Smaller, well-defined roles streamline regulatory compliance and audit processes.

Table 6.4: Percentage of roles after role mining versus initial number of roles

File	Init. Roles	Mined Roles.	% Decrease
0	34	7	79.41
1	89	9	**89.89**
2	40	18	55.00
3	33	20	39.39
4	266	55	79.32
5	190	85	55.26
6	191	60	68.59
7	255	142	44.31
8	167	100	40.12
9	292	114	60.96
10	405	151	62.72
11	202	69	65.84
12	684	276	59.65
13	876	285	67.47
14	802	551	**31.30**
15	739	382	48.31
16	602	236	60.80
17	1650	804	51.27
18	815	555	31.90
19	1119	554	50.49
20	1808	985	45.52
21	3620	1262	65.14
22	3917	1487	62.04
23	3902	1380	64.63
24	830	565	31.93
25	1632	978	40.07
26	3329	1719	48.36
27	4434	2132	51.92
28	6857	4169	39.20
29	13846	7071	48.93
30	6713	4575	31.85
31	6519	4564	29.99

These results validate the practical security and operational advantages of automated role mining in real-world enterprise environments.

Adaptive Behavior Analysis

The framework automatically selects components based on input size and resource constraints. Figures 6.5 and 6.6 show which components were activated per dataset, while tables A.1, A.2 and A.3 in the Appendix A contain additional information.

- **Optimal Execution (26/31 datasets)**: For Files 0–23 and 25–27, reductions alone allowed direct ILP solving, yielding **100% optimal solutions** without heuristic steps.
- **Degraded Execution (5/31 datasets)**: For larger graphs (Files 24, 26, 28–30), the pipeline progressively applied all of its four components. Non-optimal roles came only from heuristic stages, averaging **8.2%** of total roles.

This design indeed prioritizes exact methods when possible and limits heuristics to cases where full solving is infeasible.

6.2.5. Computational Efficiency

Despite running on consumer-grade hardware (16GB RAM), the framework processed all datasets within practical time limits:

- **No solver timeouts** (30s limit),
- **Memory usage under 4GB**,
- **Linear runtime scaling** with problem size (e.g., File 30: 1.3M edges $\rightarrow$ 1.5 hours).

This efficiency contrasts sharply with prior work, which required high-end servers (256GB RAM, 64-core processors) for similar datasets [38]. Our framework's ability to operate within resource constraints is a direct result of its adaptive thresholds and efficient algorithmic components.

Table 6.5: Experimental results showing framework performance across real-world data.

File	Edges	Roles Pure H.	Red. R.	Greedy R.	Gurobi R.	Total Time
0	171	0	7	0	0	00:00:01
1	306	0	9	0	0	00:00:01
2	514	0	18	0	0	00:00:01
3	1471	0	20	0	0	00:00:01
4	2587	0	55	0	0	00:00:01
5	2751	0	85	0	0	00:00:01
6	2810	0	52	0	8	00:00:01
7	4244	0	139	0	3	00:00:01
8	4263	0	93	0	7	00:00:01
9	6215	0	109	0	5	00:00:01
10	12810	0	151	0	0	00:00:02
11	13119	0	69	0	0	00:00:02
12	17573	0	262	0	14	00:00:02
13	27339	0	280	0	5	00:00:02
14	50328	0	548	0	3	00:00:10
15	50618	0	376	0	6	00:00:08
16	52511	0	222	0	14	00:00:19
17	53203	0	804	0	0	00:00:10
18	65172	0	551	0	4	00:00:23
19	66824	0	546	0	8	00:00:10
20	115277	0	977	0	8	00:01:06
21	124130	0	1215	0	47	00:00:29
22	133217	0	1427	0	60	00:01:04
23	164213	0	1351	0	29	00:00:35
24	211329	40	516	0	9	00:07:31
25	253750	0	890	0	88	00:04:04
26	311307	18	1659	0	42	00:04:03
27	437478	0	1991	0	141	00:34:49
28	1090388	378	2044	56	1691	01:18:48
29	1301047	101	6694	0	276	00:43:10
30	1349999	243	3003	0	1329	01:30:42

Table 6.6: Experimental results showing framework performance across RMP datasets (small, medium, large, and comp)

Folder	File	Edges	Roles Pure H.	Red. R.	Greedy R.	Gurobi R.	Total Time
small	1	600	0	4	0	20	00:00:01
small	2	1082	0	1	0	24	00:00:03
small	3	1369	0	25	0	0	00:00:01
small	4	1932	0	0	0	25	00:00:10
small	5	1372	0	49	0	0	00:00:01
small	6	2152	0	3	0	47	00:00:02
small	7	9371	16	0	47	64	00:00:50
small	8	4415	0	3	13	34	00:00:09
medium	1	15567	0	58	0	92	00:00:05
medium	2	33959	48	0	221	172	00:03:09
medium	3	22988	10	0	85	160	00:00:39
medium	4	23949	0	19	0	181	00:00:06
medium	5	47674	56	2	240	283	00:04:34
medium	6	48058	64	0	255	257	00:04:50
large	1	60288	39	5	106	308	00:03:04
large	2	49579	17	1	398	206	00:01:33
large	3	23778	0	33	0	466	00:00:06
large	4	74347	0	69	0	331	00:00:08
large	5	148067	101	2	0	401	00:16:32
large	6	62292	0	58	0	442	00:00:06
comp	1	49283	8	48	46	312	00:00:40
comp	2	60564	14	13	48	347	00:00:57
comp	3	56981	12	56	19	330	00:00:46
comp	4	70122	22	14	43	364	00:01:22
comp	5	278809	57	1251	36	721	00:10:07
comp	6	332985	63	108	64	1848	00:11:21
comp	7	411702	75	1303	0	681	00:11:31
comp	8	494455	184	218	0	1761	00:20:30
comp	9	646399	1005	1314	1138	1390	01:08:20
comp	10	812337	1017	2347	75	503	01:49:29
comp	11	1053246	1281	1667	138	1318	01:45:25
comp	12	1331702	1543	76	2088	2876	03:36:16
comp	13	573065	404	1220	113	2043	00:38:39
comp	14	665217	409	143	216	3127	00:45:10
comp	15	726448	464	1399	0	1906	00:48:10
comp	16	844106	533	3241	0	0	01:07:12

7

Discussion

Having established the empirical analysis of our proposed role-mining framework, we now reflect on its broader implications, practical deployment insights, and remaining limitations. We begin by summarizing the framework's performance across the evaluated tasks, followed by insights gained from real-world applications. We then discuss the avenue for extending this framework to a structure-aware role-mining that enables organizations to encode structural preferences into the algorithmic decision-making process. Finally, we conclude with a discussion of current limitations. Finally, we conclude with a discussion of current limitations.

7.1. Performance Overview of the Framework

Our proposed four-level framework was designed to adaptively balance scalability, memory usage, and solution quality in role mining. Through extensive experimental evaluation, the framework consistently outperforms prior approaches across several critical dimensions:

- **Conclusion on MBE Integration**: The integration of a significantly more efficient MBE algorithm [29] delivers profound benefits to our role-mining framework. Specifically, it enables the framework to (i) transition seamlessly from heuristics to more optimal layers by ensuring that MBE is not a computational bottleneck, (ii) scale effectively to massive graphs, an essential requirement for real-world deployment, and (iii) generate a richer set of candidate roles per iteration in the pure heuristic stage, allowing for more informed and higher-quality role selections. A practical example that quantifies the scalability impact is the experiment on the larger RMP datasets, where the very closely related heuristic of Tripunitara et al. [38] took 4-6 days to terminate, whereas ours terminated within 1-3 hours, while also requiring significantly fewer computational resources. Importantly, these enhancements go beyond reducing runtime. They fundamentally redefine the operational capacity of the framework by enabling it to preserve solution quality under resource constraints while scaling to previously intractable problem sizes. Therefore, the MBE algorithm is not just a performance upgrade; it is a key enabler of the framework's adaptability, effectiveness, and overall competitiveness.
- **Equivalence of the biclique-based reductions with the neighbor-based reductions of the predecessor**: Through our empirical analysis of the reductions, we verify the theoretical assumption we made in 4, that two reduction versions are essentially the same (two variations of the same rules). Real-world deployment of both approaches demonstrated the predecessor's approach as more effective. Nonetheless, through these analyses, we have gained valuable insights into how these reductions can be leveraged, and we open the door for future analysis to explore the potential of these reductions further.
- **Effectiveness of the Reductions**: The integration of reductions into our framework plays a critical role in our ability to minimize non-optimal decisions. Through our real-world deployment, we demonstrated that we can transform any input graph into its reduced version, which contains fewer than 1% of its initial edges, using optimal and deterministic decisions. Without a doubt, any

role-mining pipeline that does not incorporate these or equivalent reduction strategies is likely to miss significant improvements in solution quality and efficiency.

- **Greedy Role Selection Heuristic**: The greedy heuristic's performance confidently surpasses the theoretical upper bound, often achieving near-optimal results in real-world environments as well. Its performance demonstrates that the greedy heuristic is a reliable component of our 4-level resource-aware role mining framework and that we can confidently rely on it for handling the first role promotions in small to medium-sized datasets, knowing that the heuristic choices it makes lead to near-optimal results.
- **Scalable And Effective Pure Heuristic**: Apart from mentioning the heuristic's main component, MBE's impressive efficiency, which allows our heuristic to be used for even massive graphs, we want to explicitly mention the impressive performance the heuristic achieves in terms of the produced mined roles. Even if we run the pure heuristic alone, we still get near-optimal results in real-world settings. This means that this heuristic is not only capable of promoting roles on massive graphs, but the roles it produces are of very high quality. Therefore, due to the MBE's efficiency, we can achieve a heuristic that does not sacrifice quality for efficiency but rather achieves both to an impressive degree. As a result, we are pleased to utilize the pure heuristic as the outer layer of our four-level, resource-aware role-mining framework.
- **Overall**: Overall, we see that our four-level resource-aware framework can be confidently used for user-permission graphs of any size, delivering near-optimal solutions in all cases, as justified by the deployment of this framework in literature, as well as real-world datasets. Notably, the framework's thresholds allow for adjustment to available computational resources, enabling its use even on an ordinary employee laptop, such as the one used in our experiments.

These results validate the practicality of a multi-step decision framework that escalates computational investment based on available resources and problem complexity. The framework successfully navigates the space between fast heuristics and optimal solutions, achieving competitive performance across datasets of varying sizes and structures.

7.2. Observations from Real-World Application

During real-world deployment, we observed two recurring issues in the structure of generated roles:

- Many roles consist of a single user and all their assigned permissions.
- Some roles overlap significantly with others.

The first issue was also noted by Ene et al. [9], who observed that about four-fifths of their roles consist of a single user and all of its permissions. While this may initially appear undesirable, such roles can reveal outliers or miss-configurations in access control policies. These ”orphaned” roles, representing users with entirely unique permission sets, can serve as indicators of exception handling, special privileges, or potential errors in policy assignments. In terms of role mining, these singleton-user roles are unavoidable when covering unique permission sets; a separate role for that user must be constructed, and by the maximal nature of the biclique, it naturally includes all of the user's permissions.

The second issue stems from the inherent nature of maximal bicliques: they often overlap. Although generating only maximal bicliques helps limit the candidate space, overlap between them can introduce redundancy in the final role set. To maintain scalability, we generate roles directly from maximal bicliques. However, post-generation refinement could further reduce unnecessary overlap. We propose this as a direction for future work: a post-processing algorithm that merges or refines overlapping roles without compromising coverage or minimality guarantees.

7.3. Constraint-Aware Role Mining

Deploying our algorithm in a real-world context revealed a key insight: being able to **incorporate structural preferences into the roles** our algorithm generates can significantly increase its value to the organization. Our framework's design, which involves evaluating multiple candidate roles before making a selection, allows the natural integration of domain-specific preferences. This flexibility enables a powerful extension: structure-aware role mining, in which organizations can explicitly define desirable

or undesirable role patterns based on factors such as internal policy, maintainability, or administrative cost.

For example, an organization may prefer roles with at least five users and five permissions, a concept known in the literature as cardinality constraints [16], and penalize singleton-user roles unless the user has truly unique permissions. These structural preferences can be formalized as cost or reward functions associated with each candidate biclique, allowing the algorithm to balance structural quality with coverage. In practice, this could be integrated as follows.

For the *Pure Heuristic*, rather than selecting the largest biclique, the algorithm could be tuned to select the highest-scoring biclique within a ”green zone” of structurally preferred roles while discarding bicliques in a predefined ”red zone” (i.e., roles that will never be adopted). Additionally, the *reductions* remain unchanged, as any role activated through reduction is structurally necessary to ensure coverage and, therefore, cannot be avoided. Then, for the *Greedy Heuristic*, similar to the pure heuristic, biclique selection can be adjusted to account for structure-aware scoring rather than relying purely on size. Lastly, for the *Exact Solver*, the ILP formulation can be extended to minimize the total cost of selected roles (rather than just their count), with each biclique assigned a weight reflecting its desirability.

By incorporating structural preferences into the layers of our pipeline, our framework can be tailored to specific business objectives, offering a scalable and flexible alternative to both rigid automation and costly manual work. Additionally, more constraints, such as Separation of Duties [34], can be modeled using a similar approach. In this way, organizations can transition from inflexible role engineering to a customizable, intelligent framework that enforces security requirements while promoting interpretable, manageable, and contextually appropriate roles.

7.4. Limitations

While our framework demonstrates strong performance and flexibility, certain limitations remain:

Pure Heuristic Threshold: The behavior and quality of the pure heuristic under different biclique enumeration limits and orderings warrant deeper analysis. For instance, how frequently does it discover the optimal or near-optimal biclique within the first k candidates? Understanding this distribution could guide better dynamic cap selection or early termination strategies.

Threshold Calibration: For this research, we established different thresholds with values that enabled us to conduct various experiments. Future research or real-world deployment is advised to continually push the limits of these thresholds, checking for memory usage or solver timeouts to achieve the best results that this framework (and available resources) can provide. Regarding this, future research could also delve deeper into the analysis of how the computational resources and performance of each component scale to illustrate the chosen thresholds clearly and further conceptualize the underlying scalability and optimality trade-off.

Edge-Based Thresholding: Currently, the thresholds are set solely based on the number of bicliques and do not take into account the number of edges in the graph. While this approach works at a fundamental level, the decision to change levels could be more informed by also considering the number of edges. For example, we can have x number of large bicliques that do not fit in memory and x number of small bicliques that do fit in memory. Investigating how the number of edges, together with the number of bicliques, affects the thresholds is an interesting direction for future research.

Resource Limitations: This research did not utilize computational clusters of any kind, and the experiments were conducted on the computational resources of a standard laptop. As a result, the empirical results achieved in the literature data do not surpass those of the predecessor’s work [38] (which leveraged proper computational resources, such as 256GB of RAM or 64-core processors). Future work could empirically analyze the performance of our work in comparison to other recent works under the same settings to empirically showcase the different trade-offs the approaches offer.

Post-processing Step For Overlap Removal Omission: While our method generates roles efficiently, we have not yet implemented a follow-up step to remove overlaps, which could help improve the quality of the produced roles, especially for real-world deployments.

Future work may explore enhancements in these directions to make the framework more dynamic, interpretable, and robust for operational deployment. Future work may explore enhancements in these directions to make the framework more dynamic, interpretable, and robust for operational deployment.

8

Conclusion

Here, we present the summary of our work and point directions for future research.

8.1. Summary

In this work, we introduced a resource-aware, four-level framework for solving the Role Mining Problem. Our design integrates four key components: a memory-light, pure heuristic; biclique-based reductions inspired by set cover theory; a greedy approximation with theoretical guarantees; and an exact ILP solver. The pure heuristic ensures scalability under tight memory budgets, avoiding complete enumeration of maximal bicliques. The reduction phase systematically simplifies the problem space while preserving optimality. The greedy layer leverages set cover approximation theory to deliver near-optimal solutions efficiently. Finally, the ILP layer guarantees exact solutions when feasible.

Through a comprehensive evaluation of real-world, synthetic, and benchmark datasets, we demonstrate that our framework's underlying MBE component outperforms that of previous methods in both runtime and memory usage. Additionally, our heuristics achieve near-optimal results. Our reductions are highly effective, removing more than 99% of the edges in 24 out of 31 real-world datasets, for which we successfully found the globally optimal solution. Lastly, the deployment of the role-mining framework resulted in simplifications of up to 89% across the teams of a real-world organization, illustrating the significant benefit that practical role-mining approaches can bring to real-world applications.

By combining efficiency, scalability, theoretical rigor, and real-world deployment, we present this framework as an effective and scalable solution for role mining. Overall, our approach resulted in a comprehensive toolkit for both researchers and practitioners in the field of role-based access control and policy inference.

Lastly, we have mapped components of our framework to well-known problems, which significantly enhances the expandability of our framework. Currently, the choice of each of these components is extremely promising. However, as research in the SCP, MBE, Constraint Solvers, or role-mining heuristics progresses, more efficient and better-performing algorithms than our current choices may arise. Importantly, as this happens, one can replace the outdated algorithms with more efficient versions and, using this framework, still have a state-of-the-art role-mining algorithm. So, even though the individual components of our four-level role-mining framework may become outdated as research progresses, through the theoretical analysis, mapping to well-known problems, and adaptive nature of our framework, we see this four-level resource-aware biclique-based role-mining framework being a central foundation in the world of role mining, for now, and for the future.

8.2. Future work

While the proposed framework offers a principled and practical solution to the Role Mining Problem, it also opens several promising directions for future research. Below, we outline these promising directions for future work.

- Extend this implementation by introducing a post-processing algorithm that removes overlap between the generated roles.
- Extend this implementation by analyzing how early the largest biclique is found by the pure heuristic and adjusting the thresholds accordingly to save processing time.
- Explore the trade-offs between the two reduction versions further. For example, explore whether a parallelized version of the biclique-based reductions can surpass the corresponding one of the neighbor-based one, e.g., taking into consideration that the biclique-based reductions work on fixed bicliques within each iteration and the potential avenue from parallelization could be higher.
- Extend this implementation by iteratively applying the set cover (biclique-based) reduction rules along with the iterative greedy heuristic. This would allow smaller and fewer bicliques, enabling an earlier transition to the solver, albeit at the cost of introducing more computations. Explore the benefit of this trade-off.
- Extend this implementation by exploring if applying intuitive reduction rules (e.g., removing users having the same permissions and permissions having the same users) to the user-permission graph can improve the speed of the later reductions or enable the earlier transition to more optimal levels.
- Extend this implementation, incorporating cardinality or separation of duty constraints, through a structure-aware approach we suggest in Section 7.3
- Collect implementations of other recent role mining algorithms (e.g., the ones mentioned in Section 3.1.1) and our open-source implementation, and conduct empirical comparisons to conceptualize the strengths and weakness of the different approaches, and potentially illustrate their limitations in regards to our adaptive framework.

References

[1] Simon Anderer, Tobias Kempter, Bernd Scheuermann, and Sanaz Mostaghim. “Dynamic optimization of role concepts for role-based access control using evolutionary algorithms”. In: *SN Computer Science* 4.4 (2023), p. 416.

[2] Simon Anderer, Daniel Kreppein, Bernd Scheuermann, and Sanaz Mostaghim. “The addRole-EA: A New Evolutionary Algorithm for the Role Mining Problem.” In: *IJCCI*. 2020, pp. 155–166.

[3] Simon Anderer, Bernd Scheuermann, Sanaz Mostaghim, Patrick Bauerle, and Matthias Beil. “RMPlib: a library of benchmarks for the role mining problem”. In: *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*. 2021, pp. 3–13.

[4] Edward K Baker, Lawrence D Bodin, William F Finnegan, and Ronny J Ponder. “Efficient heuristic solutions to an airline crew scheduling problem”. In: *AIIE Transactions* 11.2 (1979), pp. 79–85.

[5] Coen Bron and Joep Kerbosch. “Algorithm 457: finding all cliques of an undirected graph”. In: *Commun. ACM* 16.9 (Sept. 1973), pp. 575–577. ISSN: 0001-0782. DOI: 10.1145/362342.362367. URL: https://doi.org/10.1145/362342.362367.

[6] Alberto Caprara, Paolo Toth, and Matteo Fischetti. “Algorithms for the set covering problem”. In: *Annals of Operations Research* 98.1 (2000), pp. 353–371.

[7] Nicos Christofides. “Combinatorial optimization”. In: *A Wiley-Interscience Publication* (1979).

[8] Richard H Day. “On optimal extracting from a multiple file data storage system: An application of integer programming”. In: *Operations Research* 13.3 (1965), pp. 482–494.

[9] Alina Ene, William Horne, Nikola Milosavljevic, Prasad Rao, Robert Schreiber, and Robert E. Tarjan. “Fast exact and heuristic methods for role minimization problems”. In: *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*. SACMAT ’08. Estes Park, CO, USA: Association for Computing Machinery, 2008, pp. 1–10. ISBN: 9781605581293. DOI: 10.1145/1377836.1377838. URL: https://doi.org/10.1145/1377836.1377838.

[10] Uriel Feige. “A threshold of ln n for approximating set cover”. In: *Journal of the ACM (JACM)* 45.4 (1998), pp. 634–652.

[11] David Ferriaolo and Richard Kuhn. “Role-based access controls”. In: *Proceedings of 15th NIST-NCSC National Computer Security Conference*. MD Baltimore. 1992, pp. 554–563.

[12] Brian A Foster and David M Ryan. “An integer programming approach to the vehicle scheduling problem”. In: *Journal of the Operational Research Society* 27.2 (1976), pp. 367–384.

[13] Marina Groshaus and Jayme L Szwarcfiter. “Biclique graphs and biclique matrices”. In: *Journal of Graph Theory* 63.1 (2010), pp. 1–16.

[14] Qi Guo, Jaideep Vaidya, and Vijayalakshmi Atluri. “The role hierarchy mining problem: Discovery of optimal role hierarchies”. In: *2008 annual computer security applications conference (ACSAC)*. IEEE. 2008, pp. 237–246.

[15] *Gurobi Optimization*. URL: https://www.gurobi.com.

[16] Pullamsetty Harika, Marreddy Nagajyothi, John C John, Shamik Sural, Jaideep Vaidya, and Vijayalakshmi Atluri. “Meeting cardinality constraints in role mining”. In: *IEEE transactions on dependable and secure computing* 12.1 (2014), pp. 71–84.

[17] Hejiao Huang, Feng Shang, Jinling Liu, and Hongwei Du. “Handling least privilege problem and role mining in RBAC”. In: *Journal of Combinatorial Optimization* 30 (2015), pp. 63–86.

[18] Hejiao Huang, Feng Shang, Jinling Liu, and Hongwei Du. “Handling least privilege problem and role mining in RBAC”. In: *J. Comb. Optim.* 30.1 (July 2015), pp. 63–86. ISSN: 1382-6905. DOI: 10.1007/s10878-013-9633-9. URL: https://doi.org/10.1007/s10878-013-9633-9.

[19] *IBM ILOG CPLEX Optimizer*. URL: https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer.

[20] Jinsuo Jia, Jianfeng Guan, and Lili Wang. “Role mining: Survey and suggestion on role mining in access control”. In: *International Symposium on Mobile Internet Security*. Springer. 2019, pp. 34–50.

[21] Richard M Karp. “Reducibility among combinatorial problems”. In: *50 Years of Integer Programming 1958-2008: from the Early Years to the State-of-the-Art*. Springer, 2009, pp. 219–241.

[22] Ina Koch. “Enumerating all connected maximal common subgraphs in two graphs”. In: *Theoretical Computer Science* 250.1 (2001), pp. 1–30. ISSN: 0304-3975. DOI: https://doi.org/10.1016/S0304-3975(00)00286-3. URL: https://www.sciencedirect.com/science/article/pii/S0304397500002863.

[23] Haibing Lu, Jaideep Vaidya, and Vijayalakshmi Atluri. “Optimal boolean matrix decomposition: Application to role engineering”. In: *2008 IEEE 24th international conference on data engineering*. IEEE. 2008, pp. 297–306.

[24] Haibing Lu, Jaideep Vaidya, Vijayalakshmi Atluri, and Yuan Hong. “Extended boolean matrix decomposition”. In: *2009 Ninth IEEE International Conference on Data Mining*. IEEE. 2009, pp. 317–326.

[25] Bingqing Lyu, Lu Qin, Xuemin Lin, Ying Zhang, Zhengping Qian, and Jingren Zhou. “Maximum biclique search at billion scale”. In: 13.9 (May 2020), pp. 1359–1372. ISSN: 2150-8097. DOI: 10.14778/3397230.3397234. URL: https://doi.org/10.14778/3397230.3397234.

[26] Barsha Mitra, Shamik Sural, Jaideep Vaidya, and Vijayalakshmi Atluri. “A Survey of Role Mining”. In: 48.4 (Feb. 2016). ISSN: 0360-0300. DOI: 10.1145/2871148. URL: https://doi.org/10.1145/2871148.

[27] Gustaf Neumann and Mark Strembeck. “A scenario-driven role engineering process for functional RBAC roles”. In: SACMAT ’02. Monterey, California, USA: Association for Computing Machinery, 2002, pp. 33–42. ISBN: 1581134967. DOI: 10.1145/507711.507717. URL: https://doi.org/10.1145/507711.507717.

[28] Zhe Pan, Shuibing He, Xu Li, Xuechen Zhang, Rui Wang, and Gang Chen. “Efficient maximal biclique enumeration on gpus”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 2023, pp. 1–13.

[29] Zhe Pan, Shuibing He, Xu Li, Xuechen Zhang, Yanlong Yin, Rui Wang, Lidan Shou, Mingli Song, Xian-He Sun, and Gang Chen. “Enumeration of Billions of Maximal Bicliques in Bipartite Graphs without Using GPUs”. In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis*. SC ’24. Atlanta, GA, USA: IEEE Press, 2024. ISBN: 9798350352917. DOI: 10.1109/SC41406.2024.00106. URL: https://doi.org/10.1109/SC41406.2024.00106.

[30] Simon Parkinson and Saad Khan. “A survey on empirical security analysis of access-control systems: a real-world perspective”. In: *ACM Computing Surveys* 55.6 (2022), pp. 1–28.

[31] Simon Parkinson and Saad Khan. “A survey on empirical security analysis of access-control systems: a real-world perspective”. In: *ACM Computing Surveys* 55.6 (2022), pp. 1–28.

[32] Yury Zhauniarovich (TU Delft) Roberto Moratore (ING Bank) Eduardo Barbaro (ING Bank & TU Delft). “IAM Role Diet: A Scalable Approach to Detecting RBAC Data Inefficiencies”. In: *DSN 2025 The 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (2025).

[33] Haio Roeckle, Gerhard Schimpf, and Rupert Weidinger. “Process-oriented approach for role-finding to implement role-based security administration in a large industrial organization”. In: RBAC ’00. Berlin, Germany: Association for Computing Machinery, 2000, pp. 103–110. ISBN: 158113259X. DOI: 10.1145/344287.344308. URL: https://doi.org/10.1145/344287.344308.

[34] Prasuna Sarana, Arindam Roy, Shamik Sural, Jaideep Vaidya, and Vijayalakshmi Atluri. “Role mining in the presence of separation of duty constraints”. In: *Information Systems Security: 11th International Conference, ICISS 2015, Kolkata, India, December 16-20, 2015. Proceedings 11*. Springer. 2015, pp. 98–117.

[35] *SCIP*. URL: https://www.scipopt.org.

[36] Steven S Skiena. *The algorithm design manual*. Vol. 2. Springer, 2008, pp. 621–624.

[37] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. “The worst-case time complexity for generating all maximal cliques and computational experiments”. In: *Theoretical Computer Science* 363.1 (2006). Computing and Combinatorics, pp. 28–42. ISSN: 0304-3975. DOI: https://doi.org/10.1016/j.tcs.2006.06.015. URL: https://www.sciencedirect.com/science/article/pii/S0304397506003586.

[38] Mahesh Tripunitara. *Minimizing the Number of Roles in Bottom-Up Role-Mining using Maximal Biclique Enumeration*. 2024. arXiv: 2407.15278 [cs.CR]. URL: https://arxiv.org/abs/2407.15278.

[39] Jaideep Vaidya, Vijayalakshmi Atluri, and Qi Guo. “The role mining problem: finding a minimal descriptive set of roles”. In: *Proceedings of the 12th ACM Symposium on Access Control Models and Technologies*. SACMAT ’07. Sophia Antipolis, France: Association for Computing Machinery, 2007, pp. 175–184. ISBN: 9781595937452. DOI: 10.1145/1266840.1266870. URL: https://doi.org/10.1145/1266840.1266870.

[40] Jaideep Vaidya, Vijayalakshmi Atluri, Janice Warner, and Qi Guo. “Role engineering via prioritized subset enumeration”. In: *IEEE Transactions on Dependable and Secure Computing* 7.3 (2008), pp. 300–314.

[41] Chunqi Wu, Jingdong Li, Zhao Li, Ji Zhang, and Pan Tang. “Accelerating Maximal Bicliques Enumeration with GPU on large scale network”. In: *Future Gener. Comput. Syst.* 161.C (Dec. 2024), pp. 601–613. ISSN: 0167-739X. DOI: 10.1016/j.future.2024.07.021. URL: https://doi.org/10.1016/j.future.2024.07.021.

[42] Fubao Zhu, Chenguang Yang, Liang Zhu, Hongqiang Zuo, and Jingzhong Gu. “MFC-RMA (Matrix Factorization and Constraints-Role Mining Algorithm): An Optimized Role Mining Algorithm”. In: *Symmetry* 16.8 (2024), p. 1008.

A

Appendix: Time Spend on different parts of the framework across various datasets

Table A.1: Input, heuristic, and reduction results (Files 0–30)

File	**Edges**	Pure R.	Pure T.	**E. PH**	Bicl. PH	MBE T.	Red. R.	**E. Red.**	Red. T.
0	171	0	00:00:00	171	18	00:00:00	7	0	00:00:00
1	306	0	00:00:00	306	37	00:00:00	9	0	00:00:00
2	514	0	00:00:00	514	33	00:00:00	18	0	00:00:00
3	1471	0	00:00:00	1471	25	00:00:00	20	0	00:00:00
4	2587	0	00:00:00	2587	236	00:00:00	55	0	00:00:00
5	2751	0	00:00:00	2751	626	00:00:00	85	0	00:00:00
6	2810	0	00:00:00	2810	253	00:00:00	52	19	00:00:00
7	4244	0	00:00:00	4244	308	00:00:00	139	6	00:00:00
8	4263	0	00:00:00	4263	911	00:00:00	93	16	00:00:00
9	6215	0	00:00:00	6215	1016	00:00:00	109	10	00:00:00
10	12810	0	00:00:00	12810	1689	00:00:00	151	0	00:00:01
11	13119	0	00:00:00	13119	198	00:00:00	69	0	00:00:01
12	17573	0	00:00:00	17573	9694	00:00:00	262	30	00:00:01
13	27339	0	00:00:00	27339	6683	00:00:00	280	9	00:00:01
14	50328	0	00:00:00	50328	3945	00:00:00	548	6	00:00:09
15	50618	0	00:00:00	50618	40473	00:00:01	376	12	00:00:05
16	52511	0	00:00:00	52511	10406	00:00:00	222	39	00:00:17
17	53203	0	00:00:00	53203	47086	00:00:01	804	0	00:00:07
18	65172	0	00:00:00	65172	2951	00:00:00	551	7	00:00:22
19	66824	0	00:00:00	66824	23439	00:00:01	546	17	00:00:08
20	115277	0	00:00:00	115277	14770	00:00:00	977	20	00:01:04
21	124130	0	00:00:00	124130	75481	00:00:03	1215	116	00:00:21
22	133217	0	00:00:00	133217	122699	00:00:03	1427	130	00:00:55
23	164213	0	00:00:00	164213	66130	00:00:03	1351	81	00:00:28
24	211329	40	00:06:47	44329	421451	00:00:05	516	18	00:00:33
25	253750	0	00:00:00	253750	147635	00:00:12	890	387	00:03:28
26	311307	18	00:03:26	96001	434924	00:00:04	1659	106	00:00:29
27	437478	0	00:00:00	437478	120331	00:00:13	1991	387	00:33:55
28	1090388	378	01:12:16	212975	493146	00:00:05	2044	9250	00:06:22
29	1301047	101	00:40:20	388601	495333	00:00:06	6694	888	00:02:36
30	1349999	243	01:11:37	277335	499367	00:00:03	3003	7556	00:18:55

Note: Pure R. = Roles by Pure Heuristic, Pure T. = Time Pure Heuristic, E. PH = Edges post Pure Heuristic, Bicl. PH = Bicliques post Pure Heuristic, MBE T. = Time MBE, Red. R. = Roles by Reduction Rule A, E. Red. = Edges post Reduction Rule A, Red. T. = Time Reductions (00:00:00 run-times stands for running times of some milliseconds)

Table A.2: Heuristic, solver, and non-optimal roles (Files 0–30)

File	Greedy R.	Greedy T.	Gurobi R.	Gurobi T.	Gurobi F.	Non-Opt.
0	0	00:00:00	0	00:00:00	0	**0**
1	0	00:00:00	0	00:00:00	0	**0**
2	0	00:00:00	0	00:00:00	0	**0**
3	0	00:00:00	0	00:00:00	0	**0**
4	0	00:00:00	0	00:00:00	0	**0**
5	0	00:00:00	0	00:00:00	0	**0**
6	0	00:00:00	8	00:00:00	0	**0**
7	0	00:00:00	3	00:00:00	0	**0**
8	0	00:00:00	7	00:00:00	0	**0**
9	0	00:00:00	5	00:00:00	0	**0**
10	0	00:00:00	0	00:00:00	0	**0**
11	0	00:00:00	0	00:00:00	0	**0**
12	0	00:00:00	14	00:00:00	0	**0**
13	0	00:00:00	5	00:00:00	0	**0**
14	0	00:00:00	3	00:00:00	0	**0**
15	0	00:00:00	6	00:00:00	0	**0**
16	0	00:00:00	14	00:00:00	0	**0**
17	0	00:00:00	0	00:00:00	0	**0**
18	0	00:00:00	4	00:00:00	0	**0**
19	0	00:00:00	8	00:00:00	0	**0**
20	0	00:00:00	8	00:00:00	0	**0**
21	0	00:00:00	47	00:00:00	0	**0**
22	0	00:00:00	60	00:00:00	0	**0**
23	0	00:00:00	29	00:00:00	0	**0**
24	0	00:00:00	9	00:00:00	0	40
25	0	00:00:00	88	00:00:01	0	**0**
26	0	00:00:00	42	00:00:00	0	18
27	0	00:00:00	141	00:00:00	0	**0**
28	56	00:00:00	1691	00:00:01	0	434
29	0	00:00:00	276	00:00:00	0	101
30	0	00:00:00	1329	00:00:01	0	243

Note: Greedy R. = Roles by Greedy Heuristic, Greedy T. = Time Greedy Heuristic, Gurobi R. = Roles by Gurobi Solver, Gurobi T. = Time Gurobi Solve, Gurobi F. = Gurobi Failures, Non-Opt. = Non-Optimal Roles.

Table A.3: Input, heuristic, and reduction results for RMP datasets

Folder	File	Users	Perm	Edges	Roles PH	Time PH	Edges PH	Bicl. PH	Time MBE	Roles RedA	Edges RedA	Time Red
small	1	49	44	600	0	00:00:00	600	1724	00:00:00	4	183	00:00:00
small	2	50	48	1082	0	00:00:00	1082	43260	00:00:00	1	501	00:00:00
small	3	49	96	1369	0	00:00:00	1369	11659	00:00:00	25	0	00:00:00
small	4	50	88	1932	0	00:00:00	1932	137028	00:00:00	0	736	00:00:00
small	5	99	93	1372	0	00:00:00	1372	3608	00:00:00	49	0	00:00:00
small	6	99	96	2152	0	00:00:00	2152	24014	00:00:00	3	1044	00:00:00
small	7	99	193	9371	16	00:00:29	5044	499369	00:00:01	0	2034	00:00:03
small	8	100	184	4415	0	00:00:00	4415	395241	00:00:01	3	1538	00:00:01
medium	1	499	479	15567	0	00:00:00	15567	112678	00:00:00	58	3724	00:00:02
medium	2	500	468	33959	48	00:02:27	20577	491855	00:00:01	0	12980	00:00:21
medium	3	500	427	22988	10	00:00:21	20267	496232	00:00:01	0	11114	00:00:06
medium	4	499	883	23949	0	00:00:00	23949	158645	00:00:00	19	4322	00:00:03
medium	5	499	980	47674	56	00:03:40	30690	497037	00:00:01	2	19838	00:00:29
medium	6	500	924	48058	64	00:04:11	30821	499133	00:00:01	0	16810	00:00:20
large	1	999	910	60288	39	00:02:23	44926	491577	00:00:01	5	21437	00:00:29
large	2	999	992	49579	17	00:00:43	46017	499350	00:00:01	1	41632	00:00:19
large	3	999	910	23778	0	00:00:00	23778	51589	00:00:00	33	14000	00:00:04
large	4	999	3446	74347	0	00:00:00	74347	178486	00:00:00	69	4097	00:00:06
large	5	1000	3522	148067	101	00:16:00	95696	499694	00:00:02	2	8381	00:00:26
large	6	999	3545	62292	0	00:00:00	62292	94785	00:00:00	58	4459	00:00:04
comp	1	996	1647	49283	8	00:00:19	43350	495601	00:00:02	48	9282	00:00:12
comp	2	999	1647	60564	14	00:00:33	52414	497975	00:00:01	13	12780	00:00:16
comp	3	996	1739	56981	12	00:00:25	48675	487753	00:00:01	56	7826	00:00:14
comp	4	999	1739	70122	22	00:00:56	58095	496458	00:00:01	14	14110	00:00:18
comp	5	4993	7528	278809	57	00:09:20	151471	492271	00:00:01	1251	15922	00:00:40
comp	6	4998	7528	332985	63	00:09:49	203117	493134	00:00:01	108	38871	00:01:23
comp	7	4994	8829	411702	75	00:10:27	227420	493991	00:00:01	1303	7479	00:00:59
comp	8	4998	8829	494455	184	00:16:02	273931	498473	00:00:01	218	24841	00:04:23
comp	9	9991	7936	646399	1005	01:03:08	161417	499542	00:00:01	1314	59566	00:04:11
comp	10	9998	7936	812337	1017	01:15:46	251864	498565	00:00:01	2347	5994	00:25:35
comp	11	9994	9436	1053246	1281	01:30:28	244596	499201	00:00:01	1667	20005	00:14:45
comp	12	9998	9436	1331702	1543	03:04:41	326136	499601	00:00:01	76	218904	00:29:09
comp	13	9991	12504	573065	404	00:36:20	282018	499007	00:00:01	1220	45695	00:02:08
comp	14	9997	12504	665217	409	00:42:10	358119	499720	00:00:02	143	82534	00:02:44
comp	15	9993	14427	726448	464	00:46:01	347505	499836	00:00:01	1399	27528	00:02:03
comp	16	9997	14427	844106	533	01:03:03	425336	499443	00:00:01	3241	0	00:04:04

Note: PH = Pure Heuristic, RedA = Reduction Rule A, Bicl. = Bicliques, MBE = Maximal Biclique Enumeration.