

Article

# Role Minimization Optimization Algorithm Based on Concept Lattice Factor

Tao Wang and Qiang Wu \*

Department of Computer Science and Technology, Shaoxing University, Shaoxing 312000, China; taohit@usx.edu.cn

\* Correspondence: cswq@usx.edu.cn; Tel.: +86-0575-88341283

**Abstract:** Role-based access control (RBAC) is a widely adopted security model that provides a flexible and scalable approach for managing permissions in various domains. One of the critical challenges in RBAC is the efficient assignment of roles to users while minimizing the number of roles involved. This article presents a novel role minimization optimization algorithm (RMOA) based on the concept lattice factor to address this challenge. The proposed RMOA leverages the concept lattice, a mathematical structure derived from formal concept analysis, to model and analyze the relationships between roles, permissions, and users in an RBAC system. By representing the RBAC system as a concept lattice, the algorithm captures the inherent hierarchy and dependencies among roles and identifies the optimal role assignment configuration. The RMOA operates in two phases: the first phase focuses on constructing the concept lattice from the RBAC system's role–permission–user relations, while the second phase performs an optimization process to minimize the number of roles required for the access control. It determines the concept lattice factor using the concept lattice interval to discover the minimum set of roles. The optimization process considers both the user–role assignments and the permission–role assignments, ensuring that access requirements are met while reducing role proliferation. Experimental evaluations conducted on diverse RBAC datasets demonstrate the effectiveness of the proposed algorithm. The RMOA achieves significant reductions in the number of roles compared to existing role minimization approaches, while preserving the required access permissions for users. The algorithm's efficiency is also validated by its ability to handle large-scale RBAC systems within reasonable computational time.



**Citation:** Wang, T.; Wu, Q. Role Minimization Optimization Algorithm Based on Concept Lattice Factor. *Mathematics* **2023**, *11*, 3047. <https://doi.org/10.3390/math11143047>

Academic Editor: Ripon Kumar Chakraborty

Received: 9 June 2023  
Revised: 2 July 2023  
Accepted: 5 July 2023  
Published: 10 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** role-based access control (RBAC); role minimization; intervals; concept lattice factor

**MSC:** 68T30; 68T09

## 1. Introduction

Role-based access control (RBAC) is a widely used security model that provides a structured approach to managing permissions in various domains [1]. In RBAC systems, users are assigned roles, and roles are associated with specific permissions. However, one of the major challenges in RBAC is the efficient assignment of roles to users while minimizing the number of roles (the role mining problem (RMP)) involved.

The proliferation of roles in an RBAC system can lead to administrative complexities, increased maintenance efforts, and potential security vulnerabilities. Therefore, there is a need for effective algorithms that can optimize the role assignment process, reducing the number of roles while ensuring that access requirements are met.

The research field of role minimization optimization algorithms based on the concept lattice factor is still relatively limited but growing. The concept of role minimization in RBAC systems has garnered attention due to the challenges posed by role proliferation and its impact on system complexity and security.

Several studies have explored different approaches for role minimization in RBAC systems [2]. Traditional methods often rely on heuristics, graph-based algorithms, or

mathematical optimization techniques. However, these approaches may face limitations in terms of computational complexity, scalability, and the ability to handle large-scale RBAC systems.

The introduction of the concept lattice factor as a basis for role minimization algorithms has opened up new possibilities for more efficient and effective solutions. The concept lattice, derived from formal concept analysis, provides a structured framework to capture the relationships between roles, permissions, and users [3]. By leveraging the concept lattice factor, we aim to develop algorithms that can exploit the inherent hierarchy and dependencies among roles to minimize their number.

## 2. Related Work

Krra et al. [4] summarized and categorized many methods in recent years to approximate the optimal solutions for role generation and role allocation in access control systems, such as role mining, dynamic user–role assignments, and role refinement.

Role mining was first proposed based on initial clustering of users who were assigned the same privileges [5]. Basic-RMP [6] finds the fewest set of roles from the user rights assignments and provides the user with the role assignments along with the permissions.

Role mining algorithms partially automate the construction of an RBAC policy from an ACL (access control lists) policy and possibly other information, reducing the cost of migration to RBAC [7]. Xu and Stoller [8] proposed algorithms for role mining. The algorithms can easily be used to optimize a variety of policy quality metrics, including metrics based on policy size, metrics based on interpretability of the roles with respect to user attribute data, and compound metrics that consider size and interpretability.

The researchers found that obtaining a workable set of roles to optimize user access mapping to the role mining problem (RMP) is the well-known (NP-hard) problem. Polynomial time approximation algorithms such as greedy and random methods can be used to obtain a feasible set roles. For example, Basic-RMP maps to minimal tiling problems [6] (where each tile corresponds to a role), minimal biclique coverage [9] (where each role corresponds to biclique), and set cover problems [10] (where each subset corresponds to a role). In edge-RMP [11], work has been carried out to minimize the administrative burden by optimizing user–role and permission–role assignments. Since Basic-RMP and Edge RMP prove to be NP hard, a greedy and approximate algorithm is proposed to optimize the edges (i.e., user–role assignments (UR) and permission–role assignments (PR)) in RBAC. Ene et al. [12] also introduced fast graph reductions that allow recovery of the solution from the solution to a problem on a smaller input graph.

An unsupervised role mining method called fast miner [13] is based on permission set enumeration of predefined constraints. The Simple Role Mining Algorithm [14] is a heuristic-based solution for approximating the best set of characters. The user with the fewest privileges will be the initial entry for the role set. This process of selecting the minimum number of permissions is carried out gradually after the individual user's tasks are completed. It maintains subsequent updates to the role set by eliminating roles acquired as a federation of other roles that have been inserted into the role set. Li et al. [15] used operations and resources of permissions as the functional information in role mining algorithm, role mining with functional features (FMiner), to reduce composite roles. The HP Role Minimization Algorithm [7] and Weighted Structure Complexity Optimization [16] are exact variants of RMP because the set of roles is highly compatible with the permissions assigned to users. The process of mining roles is also included in the RBAC extension model, such as Temporary RBAC and Generalized Temporary RBAC. This is known as Temporal RMP [17]. Here, role assignments to users and permissions are enabled only for a set of time intervals. In the constrained role miner [18], the proposed role mining algorithm conforms to various constraints to optimize the role assignment to users and permissions.

When the only information is user–permission relation, roles are discovered whose semantic meaning is based on formal concept lattices [19]. They argue that the theory of formal concept analysis provides a solid theoretical foundation for mining roles from user

permission relation. A dyadic formal context from the triadic security context represents role-based access permission and performs attribute exploration from formal concept analysis (FCA) [20,21]. An FCA construction, by introducing the enrichment of an incidence relation by a set of intervals in a formal context, investigated the approach for lattice-generating interval relations on the context side [22].

The existing algorithms mainly group permissions or users, but for role mining, both users and permissions need to be grouped, so it is necessary to find more effective methods for role mining.

### 3. Preliminaries

RBAC is an access control model that organizes user permissions based on roles. It simplifies access control management by grouping users with similar access requirements into roles, and then assigning permissions to those roles.

In this paper, we follow the basic definitions in NIST standard, which is the most widely known formal description of the RBAC model.

The RBAC model contains the following components:

**User:** An individual or entity that interacts with the system and requires access to resources. Users are assigned roles that define their access rights.

**Role:** A defined set of permissions that represents a specific job function, responsibility, or level of authority within an organization. Roles are associated with users to determine their access privileges.

**Permission:** The rights or actions that users are authorized to perform on resources. Permissions are assigned to roles and determine what actions users can take within the system.

**User–Role Assignment:** The process of associating users with roles based on their job responsibilities, functions, or other attributes. User–role assignments define the roles that each user is authorized to fulfill.

**Role–Permission Assignment:** The process of associating permissions with roles. Role–permission assignments specify the actions that users in a particular role are authorized to perform on resources [23].

The following definitions formalize the above discussion.

$U, R, P$  (users, roles, and permissions).

$UR \subset U \times R$ : a many-to-many user to role assignment relation.

$RP \subset R \times P$ : a many-to-many role to permission assignment relation.

$UP \subset U \times P$ : a many-to-many users to permission assignment relation.

$Pers(r) = \{p \in P \mid (r, p) \in RP\}$ : the permission set owned by role  $r$ .

$PERS(R) = \{p \in P \mid r \in R, (r, p) \in RP\}$ : the permission set owned by the role set  $R$ .

Given  $m$  users,  $n$  permissions, and  $k$  roles, the user–role mapping can be represented as an  $m \times k$  Boolean matrix, where  $a_{ij}$  in cell  $ij$  indicates the assignment of role  $j$  to user  $i$ . Similarly, the role–permission mapping can be represented as a  $k \times n$  Boolean matrix, where a 1 in cell  $ij$  indicates the assignment of permission  $j$  to role  $i$ . Finally, the user–permission mapping can be represented as an  $m \times n$  Boolean matrix, where  $a_{ij}$  in cell  $ij$  indicates the assignment of permission  $j$  to user  $i$ .

**Definition 1.** *Role Mining Problem:* Given an  $m \times n$  access control matrix,  $UP$  is decomposed into sizes of  $m \times k$  and  $k \times n$  two matrices  $UR$  and  $RP$ , and  $k$  is the smallest among all possible matrix decompositions.

**Definition 2.** *A formal context or a dyadic context  $K$  is a triple  $(X, Y, I)$ , where  $X$ , called the universe of discourse, is a nonempty and finite set of objects,  $Y$  is a nonempty finite set of attributes, and  $I \subseteq X \times Y$  is a binary relation between  $X$  and  $Y$ .*

**Definition 3.** *For a formal context  $K$ , operators  $\uparrow: 2^X \rightarrow 2^Y$  and  $\downarrow: 2^Y \rightarrow 2^X$  are defined for every  $A \subseteq X$  and  $B \subseteq Y$  by  $A^\uparrow = \{y \in Y \mid \text{for each } x \in A: \langle x, y \rangle \in I\}$  and  $B^\downarrow = \{x \in X \mid \text{for each } y \in B: \langle x, y \rangle \in I\}$ . The operators  $\uparrow$  and  $\downarrow$  are known as concept-forming operators.*

**Definition 4.** A formal concept of the context  $K = (X, Y, I)$  is a pair  $(A, B)$  of  $A \subseteq X$  and  $B \subseteq Y$ , such that  $A^\uparrow = B$  and  $B^\downarrow = A$ .

We call  $A$  extent and  $B$  intent of the concept  $(A, B)$ . Formal concepts are naturally ordered by partial order " $\leq$ " using a subconcept–superconcept relation, such that, for any two formal concepts  $(A_1, B_1)$  and  $(A_2, B_2)$ ,  $(A_1, B_1) \leq (A_2, B_2)$  if and only if  $A_1 \subseteq A_2$  and  $B_2 \subseteq B_1$ . The objects and attributes are dual in nature, which forms a Galois connection. This connection exhibits closure relation among objects and attributes such that, from any set of formal objects, one can identify all the attributes that they have in common.

**Definition 5.** The collection of all formal concepts of the context  $K = (X, Y, I)$  equipped with subconcept–superconcept partial ordering  $\leq$  is called a concept lattice  $L(K)$ .

According to the definitions of RBAC, a formal context  $K = (U, P, IA)$  corresponds to an access control matrix, where  $U$  is the user set,  $P$  is the permission set, and  $IA$  represents  $UP$ . For  $u \in U, p \in P, (u, p) \in IA$ , it indicates that user  $u$  has permission  $p$ . Therefore, Table 1 can be used to represent the formal context under the RBAC model.

**Table 1.** An example of RBAC formal context.

Users	Permissions						
	a	b	c	d	e	f	g
1	1	0	1	0	1	0	1
2	1	1	0	1	0	0	1
3	0	1	1	1	1	1	1
4	0	0	1	0	0	1	0
5	1	1	0	0	1	1	0

#### 4. Proposed Methodology

On the concept lattice, since all possible roles can be mined and the concepts and roles correspond one-to-one, the problem of solving the minimum set of roles on the access control matrix  $UP$  in the role mining problem can be equivalent to solving the minimum set of role concepts generated by the concept lattice.

**Definition 6.** Minimum Role Concept Set: Let  $K = (U, P, IA)$ , and  $S_m$  be a set of concepts in the concept lattice  $L(K)$  generated by the formal context. If  $S_m$  satisfies the following two conditions, it is called the minimum role concept set on the access control context  $K$ .

Condition 1: The permissions owned by each user in the access control context  $K$  can be represented by the union of the intents of several concepts in the concept set  $S_m$ .

Condition 2 The number of concepts in the concept set  $S_m$  is the smallest.

In the following discussion, we will no longer distinguish between the general formal context and the access control context, and both will be represented by  $K$ .

**Definition 7.** For formal concepts  $(A_1, B_1), (A_2, B_2) \in L(K)$ , the subset  $[(A_1, B_1), (A_2, B_2)] = \{(A, B) \in L(K) \mid (A_1, B_1) \leq (A, B) \leq (A_2, B_2)\}$  is called the interval in  $L(K)$  bounded by  $(A_1, B_1)$  and  $(A_2, B_2)$ .

Furthermore, for  $A \subseteq X$  and  $B \subseteq Y$ , let  $\gamma(A) = (A^{\uparrow\downarrow}, A^\uparrow)$  and  $\mu(B) = (B^\downarrow, B^{\downarrow\uparrow})$ , i.e.,  $\gamma(A)$  and  $\mu(B)$  are the least formal concept in  $L(K)$  whose extent includes  $A$  and the greatest one whose intent includes  $B$ .  $\gamma(\{i\})$  and  $\mu(\{j\})$ , denoted simply by  $\gamma(i)$  and  $\mu(j)$ , are called the object and attribute concept determined by  $i \in X$  and  $j \in Y$ , respectively. We denote  $[A, B] = [\gamma(A), \mu(B)]$ . Clearly, every interval in  $L(K)$  is of this form. Of particular importance are the intervals of the form  $I_{ij} = [\gamma(i), \mu(j)]$ .

**Definition 8.** Assuming that the concept lattice  $L(K)$  with formal context  $K = (X, Y, I)$  has an interval set  $E = \{e_1, e_2, \dots, e_n\}$ , then the factor of  $L(K)$  is a subset  $G = \{(A, B) \mid A \subseteq X, B \subseteq Y\}$ , where  $(A, B) \in L(K)$  is a formal concept. For any  $(A, B), (A', B') \in L(K)$ ,  $(A, B) \in e_i, (A', B') \in e_j$  that satisfies  $e_i \subseteq e_j$ , then  $(A, B)$  must be a formal concept in  $G$ .

**Theorem 1.** If the concept lattice interval  $I_{ij}$  is nonempty and is minimal with respect to  $\subseteq$ , then  $I_{ij}$  is the concept lattice factor.

**Proof.** Note that  $I_{ij} \subseteq I_{i'j'}$  iff  $\gamma(i) \leq \gamma(i')$  and  $\mu(j) \leq \mu(j')$  iff  $\{i\}^\uparrow \subseteq \{i'\}^\uparrow$  and  $\{j\}^\downarrow \subseteq \{j'\}^\downarrow$  and that a nonempty  $I_{ij}$  is minimal with respect to  $\subseteq$  if it does not contain any other  $I_{i'j'}$ , i.e.,  $I_{ij} = I_{i'j'}$  whenever  $I_{ij} \subseteq I_{i'j'}$  for every  $i', j'$ .  $\square$

**Theorem 2.** In the formal context  $K = (U, P, IA)$ , the concept lattice factor is the minimum role concept set.

**Proof.** We prove that the concept lattice factor satisfies two conditions for the minimum role concept set. (1) According to definition 8, concept lattice factors are concepts included in the minimum interval, so all concepts in context  $K = (U, P, IA)$  can be represented by their union of the intents; (2) According to Theorem 1, the concept lattice factor, which is minimal with respect to  $\subseteq$ , satisfies Condition 2.  $\square$

Theorem 1 and Theorem 2 indicate that the optimal set of roles can be determined by determining the concept lattice factor in context  $K = (U, P, IA)$ .

We can first calculate all intervals of the context  $K = (U, P, IA)$  using the algorithm (Algorithm 1) in reference [24].

---

**Algorithm 1** ComputeIntervals [24].

---

```

Input: Boolean matrix IA
Output: Set  $G \subseteq \mathcal{L}(\mathcal{E}(IA))$ 
1  $\mathcal{E} \leftarrow \mathcal{E}(IA)$ ;  $U \leftarrow \{(i, j) \mid \mathcal{E}_{ij} = 1\}$ ;  $G \leftarrow \emptyset$  while  $U \neq \emptyset$  do
2  $D \leftarrow \emptyset$ ;  $s \leftarrow 0$ 
3 while exists  $j \notin D$  with  $\mid ((D \cup \{j\})^\downarrow \mathcal{E})^\uparrow IA^\downarrow IA \times ((D \cup \{j\})^\downarrow \mathcal{E}^\uparrow \mathcal{E})^\downarrow IA^\uparrow IA \cap U \mid > s$  do
4 select  $j$  which maximizes  $\mid ((D \cup \{j\})^\downarrow \mathcal{E})^\uparrow IA^\downarrow IA \times ((D \cup \{j\})^\downarrow \mathcal{E}^\uparrow \mathcal{E})^\downarrow IA^\uparrow IA \cap U \mid$ 
5  $D \leftarrow (D \cup \{j\})^\downarrow \mathcal{E}^\uparrow \mathcal{E}$ ;  $C \leftarrow (D \cup \{j\})^\downarrow \mathcal{E}$ 
6  $s \leftarrow \mid C^\uparrow IA^\downarrow IA \times D^\downarrow IA^\uparrow IA \cap U \mid$ 
7 end
8 add  $(C, D)$  to  $G$ 
9  $U \leftarrow U - C^\uparrow IA^\downarrow IA \times D^\downarrow IA^\uparrow IA$ 
10 end
11 return  $G$ 

```

---

For  $IA \in \{0,1\}^{n \times m}$ , we denote by  $\mathcal{E}(IA)$  the  $n \times m$  Boolean matrix given by  $(\mathcal{E}(IA))_{ij} = 1$  iff  $IA_{ij}$  is nonempty and minimal with respect to  $\subseteq$ .  $G$  is a collection of possibly overlapping groups of essential 1s, i.e., 1s in  $\mathcal{E}(IA)$ .

The concept lattice interval is actually a set of several formal concepts, so we can use a double loop to check whether each set  $s_i$  is a subset of other sets  $s_j$  in  $G = \{s_1, \dots, s_i, \dots, s_j, \dots, s_n\}$ . If so, then  $s_i$  is not the set we are looking for; otherwise,  $s_i$  may be the set we are looking for. Then, for each possible set  $s_i$ , we need to check if it is a subset of other sets. If  $s_i$  is a subset of other sets, then it is not the set we are looking for; otherwise,  $s_i$  may be one of the sets we are looking for. Finally, for each possible set  $s_i$ , we need to check whether it is the smallest set, that is, whether there is a set smaller than  $s_i$  that can also be a subset of other sets.

Specifically, the algorithm can be implemented as follows (Algorithm 2):

**Algorithm 2** Finding the minimum role concept set algorithm.

---

Input: Concept lattice interval  $G$   
Output: Minimum role concept set  $R_s$

1. Initialize an empty collection result  $R_s$ , representing the final result set.
2.  $is\_subset = 0$  //Initialize a Boolean variable  $is\_subset$  is false, indicating whether  $s_i$  is a subset of other sets.
3. For each set  $s_j$  and  $s_i$ , proceed as follows:
  4. If  $i = j$ , skip this loop.
  5. If  $s_i \subseteq s_j$ ,
  6. Then set  $is\_subset = 1$
  7. jumps out of the loop.
8. If  $s_i$  is not a subset of any set
9. then  $s_i$  is added to the result set result  $R_s$ .
10. For each set  $s_i$  and  $s_j$ , proceed as follows:
  11.  $is\_minimal = 1$  //Initialize a Boolean variable  $is\_minimal$  is true, indicating whether  $s_i$  is the minimum set.
  12. If  $i = j$ , skip this loop.
  13. If  $s_i \subseteq s_j$
  14. then  $is\_minimal = 0$
  15. exit the loop.
16. If  $s_i$  is the smallest set, add  $s_i$  to the result set result  $R_s$ .
17. Returns the result set result  $R_s$ .

---

**5. An Illustrative Example**

To demonstrate the effectiveness of our algorithm, we used the example electronic medical record system in reference [25] as a context instance for role mining and semantic assignment, thereby generating role states with semantic meaning and hierarchical structure.

In this example, user positions are divided into two categories: ordinary positions and management positions. Ordinary positions include registrar (1), surgeon (2), physician (3), gynecologist (4), nurse (5), and pharmacist (6). The management positions include surgical director (7), internal medicine director (8), gynecological director (9), medical department head (10), chief nurse (11), pharmacy director (12), and dean (13). Based on the reading and writing of information in various scenarios and authorized operations for various functions, the permissions used in the system are listed as follows: reading patient basic information (a), writing patient basic information (b), reading hospitalization information (c), writing hospitalization information (d), reading history records (e), reading diagnostic information (f), reading prescriptions (g), reading nurse reports (h), writing internal medicine history records (i), writing surgical history records (j), writing gynecological history records (k), writing internal medicine diagnostic information (l) Write surgical diagnosis information (m), gynecological diagnosis information (n), internal medicine prescription (o), surgical prescription (p), gynecological prescription (q), nurse report (r), physician authorization (s), surgeon authorization (t), gynecologist authorization (u), pharmacist authorization (v), nurse authorization (w). The attributes used in the department and functional information system are as follows: internal medicine (A), surgery (B), gynecology (C), medication (D), registration (E), diagnosis (F), nursing (G), and director (H). The entire system has 13 types of users, 23 types of permissions, and 8 types of attributes. The corresponding relationship between each type of user and permissions is listed in Table 2, and the attributes owned by each type of user are listed in Table 3.

**Table 2.** User–permission relationship.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0
3	1	0	1	0	1	1	1	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0
4	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0
5	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
6	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	1	0	1	1	1	1	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0
8	1	0	1	0	1	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	0	0	0
9	1	0	1	0	1	1	1	1	0	0	1	0	0	1	0	0	1	0	0	0	1	0	0
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	0	0
11	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
12	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

**Table 3.** User–attribute relationship.

	A	B	C	D	E	F	G	H
1	0	0	0	0	1	0	0	0
2	0	1	0	1	0	1	0	0
3	1	0	0	1	0	1	0	0
4	0	0	1	1	1	0	0	0
5	0	0	0	0	0	0	1	0
6	0	0	0	1	0	0	0	0
7	0	1	0	1	0	1	0	1
8	1	0	0	1	0	1	0	1
9	0	0	1	1	0	1	0	1
10	1	1	1	1	1	1	0	1
11	0	0	0	0	0	0	1	1
12	0	0	0	0	0	0	0	1
13	1	1	1	1	1	1	1	1

Step 1: Construct a user permission concept lattice based on the user permission relationships provided in Table 2, mapping it to candidate role states, as shown in Figure 1.

Step 2: Determine  $I_{ji}$  based on  $a_{ij} = 1$  and use the algorithm to determine the concept lattice factor. Establish a correspondence between concepts and reduced concepts to obtain the candidate role states for reduction, as shown in Figure 2.

For example,  $s_{3i} = I_{3i} = [\{\{3,8,10,13\},\{a,c,e,f,g,h,i,l,o\}\}]$ ,  $s_{3e} = I_{3e} = [\{\{3,8,10,13\},\{a,c,e,f,g,h,i,l,o\}\}, \{\{2,3,7,8,9,10,13\},\{a,c,e,f,g,h\}\}]$ ,  $s_{3i} \subseteq s_{3e}$ ,  $s_{3i}$  is a concept lattice factor. All concept lattice factors are marked in red in Figure 2.

Step 3: Generate a user attribute concept set based on the user attribute relationships provided in Table 3, and sort the generated concept set based on the number of users and permissions to obtain an ordered user attribute concept set.

Step 4: In the concept set, for the extension of the corresponding concept for each role, search for its closest expression in order from top to bottom, and assign semantic meaning to each role.

Figures 3 and 4 show the original and minimum roles of the electronic medical record system, respectively.

The role structure mining algorithm in this article has a simple hierarchy and requires fewer allocation relationships to be added. At the same time, the algorithm in this article uses the nearest neighbor expression of user attributes to assign semantic meaning to roles, which is more accurate than assigning semantic meaning to roles based on their permissions, user functions in the system, and actual positions in reference [25].

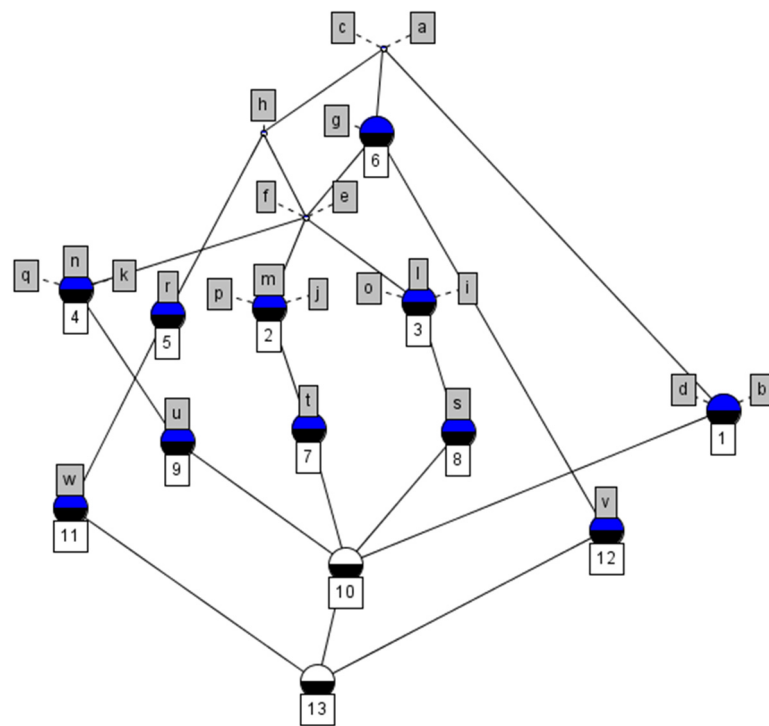


Figure 1. The corresponding role concept lattice in Table 2.

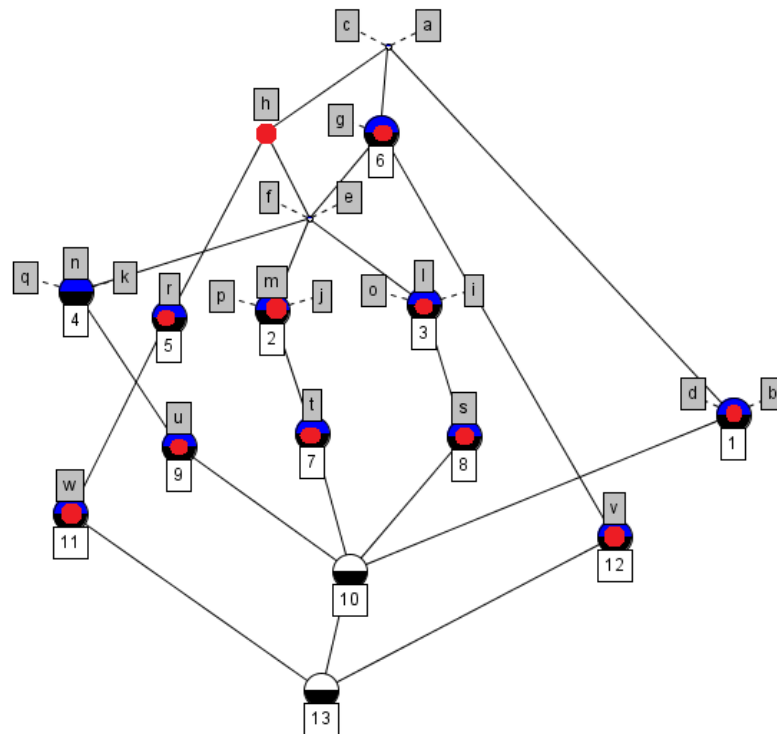


Figure 2. The corresponding role concept lattice factor (nodes marked in red) in Table 2.



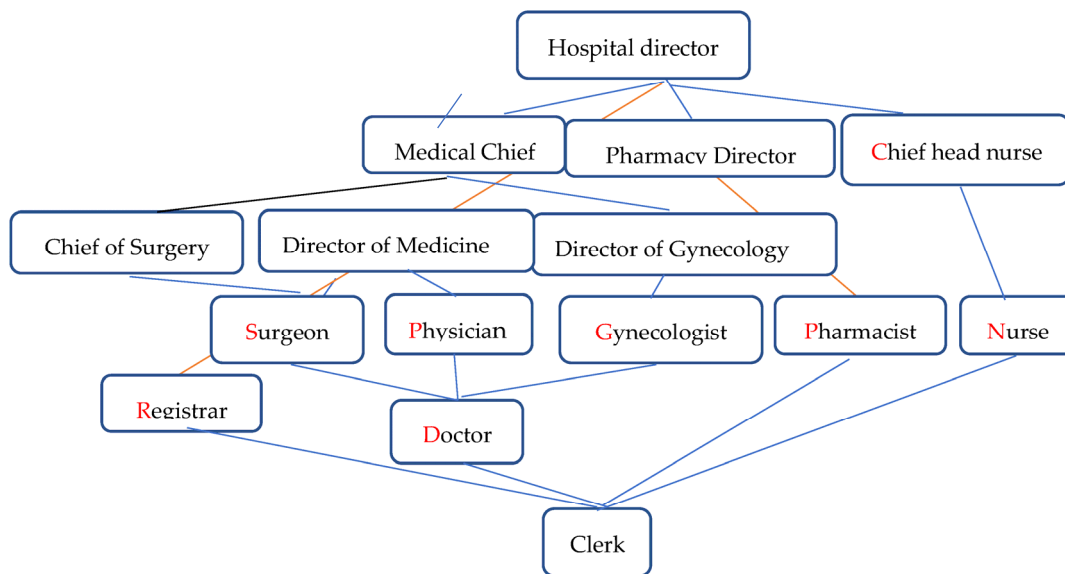


Figure 3. Original position of the electronic medical record system.

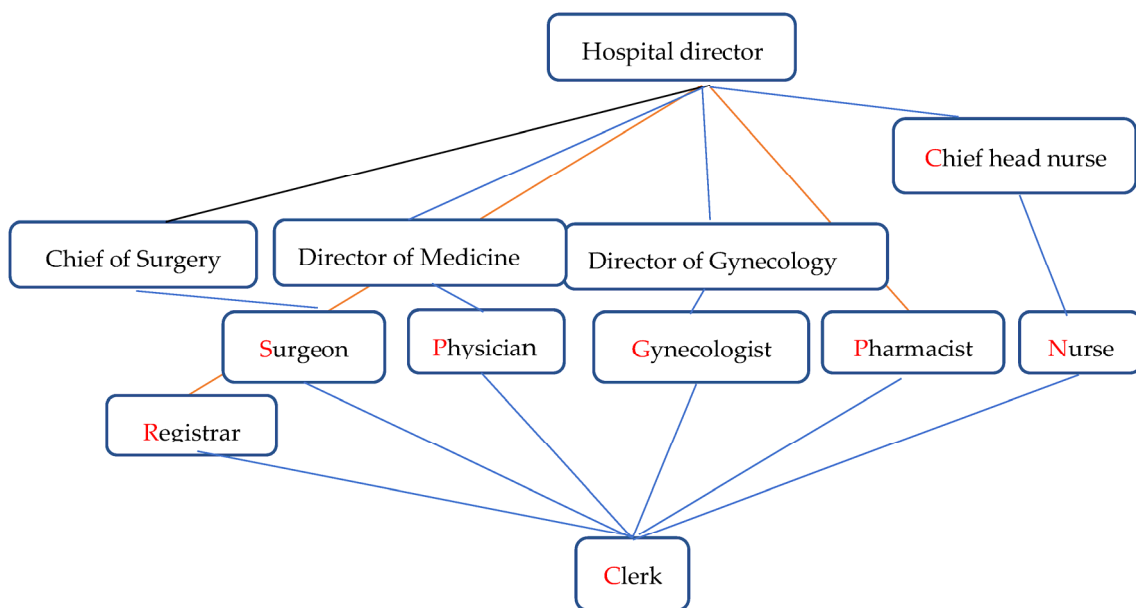


Figure 4. Minimum role concept lattice of the electronic medical record system.

6. Experimental Results

We conducted an experimental study to evaluate our proposed method. The ideal method for evaluating the accuracy of role mining is to use real-world user permission data. However, obtaining such data is extremely difficult, especially those containing complete RBAC states. Therefore, most role mining algorithms use synthesized user permission data as input for evaluation [26]. Similarly, we prepared our input dataset based on the template in reference [27].

To evaluate the performance of our algorithm, we implement the algorithm by Java and run the program on the synthetic dataset. Our experimental platform is a personnel computer with an Intel(R) Core(TM) i5 CPU and 16 GB memory.

In this study, we conducted experiments and analysis on five different datasets, as shown in Table 4. We used the program shown in Algorithm 3 [28] to prepare the dataset. Firstly, we defined a set of roles based on the above template. Then we created multiple users and randomly assigned them to each role, specifying the maximum number of users

for any given role. Then, we set user–permissions based on the roles assigned to each user in the study.

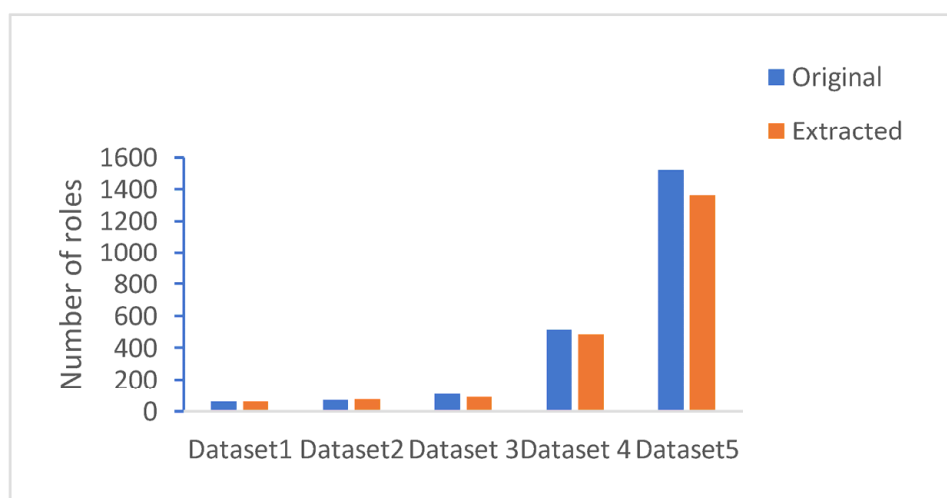
**Algorithm 3** Data preparation algorithm.

Input:  $R \leftarrow$  number of roles;  
 $U \leftarrow$  number of users;  
 $P \leftarrow$  number of permissions;  
 $UR \leftarrow$  initialize at zero;  
 $RP \leftarrow$  initialize according to the template;  
Output: Dataset  
1.  $numberUsersPerRole \leftarrow DistFunction(U, R)$ ;  
2. for  $k \leftarrow 1$  to  $R$  do  
3.  $numberUsers \leftarrow numberUsersPerRole [k]$ ;  
4. for  $i \leftarrow 1$  to  $numberUsers$  do  
5.  $user \leftarrow Rand (U)$ ;  
6.  $UR_{user,k} \leftarrow 1$ ;  
7. end for  
8. end for

**Table 4.** Synthesized user permission data.

Dataset	Users	Permissions
Dataset1	1000	42
Dataset2	5000	60
Dataset3	10,000	102
Dataset4	30,212	1178
Dataset5	116,708	4086

Our goal is to achieve a 100% reconstruction rate. Figure 5 illustrates the number of original roles used for preparing the datasets against the number of extracted roles. The number of original roles and extracted roles are indicated by red and blue bars, respectively. Notably, the number of extracted roles among different datasets is close to the number of original roles, indicating that our approach is very close to the optimal solution. More specifically, the number of extracted roles is identical to the number of original roles for Dataset1, i.e., the small-scale dataset. For large datasets, the number of extracted roles is slightly lower than the original number. This is because the concept lattice factor completely eliminates concepts that can be a union of the intents.



**Figure 5.** Comparison between original roles and extracted roles.

### Time Complexity

Consider first Algorithm 1. It first computes  $\mathcal{E}(IA)$ , which may be performed in time  $O(n^2m^2)$ , since it suffices to repeat for every of the  $nm$  entries of  $IA$  the test and since the test may be performed in time  $O(nm)$ . Inside this loop, the most critical is the number of executions of the innermost cycle. The most expensive in that cycle is computing  $((D \cup \{j\})^{\downarrow \mathcal{E}} \uparrow \mathcal{E})^{\downarrow IA} IA$ , which takes time  $O(nm)$ . The outer cycles proceed at most  $m$  times since no more than  $m$  attributes may eventually be added when extending the rectangle under construction. Within the  $j$ th execution of the outer cycle, the inner cycle is executed at most  $m + 1 - j$  times, since this is the number of remaining candidate attributes for extending the so-far computed rectangle  $\langle C, D \rangle$ . Hence, the innermost cycle is executed  $\sum_{j=1}^m (m + 1 - j) = O(m^2)$  times, along with the at most  $O(nm)$  steps within each execution of the innermost cycle. Since  $\max(n, m) \leq \|IA\|$ , the time for ComputeIntervals itself is  $O(n^2m^2) + O(\|IA\|nm^3) = O(\|IA\|nm^3)$ .  $\|IA\| = \sum_{i,j=1}^{m,n} |IA_{ij}|$ .

After Algorithm 1, Algorithm 2 executes at most  $O(nm+nm)$  times the loop 3–9 within which it executes at most  $nm$  times. To sum up, all algorithms have a polynomial upper bound of time complexity, namely,  $O(\|IA\|nm^3)$ .

Our role minimization optimization algorithm is based on the concept lattice factor, which is the formal context matrix factorization. A good factorization algorithm computes a factorization of the input matrix  $IA$  using a reasonably small number of factors in such a way that the first factors have a reasonably good coverage, i.e., they explain a large portion of data. For this purpose, Radim et al. [24] employed the following function of  $A \in \{0, 1\}^{n \times l}$  and  $B \in \{0, 1\}^{l \times m}$ , representing the coverage quality of the first  $l$  factors delivered by the particular algorithm:  $c = 1 - E(IA, A \circ B) / \|IA\|$ . They compared the factorization algorithms. For all datasets, it has the highest coverage by the first few factors, providing the best, almost exact factorizations.

## 7. Conclusions and Future Work

This paper proposes to use operations and resources of the permissions as the function information in role mining and presents a new role mining approach that could reduce composite roles. Our algorithm has two main processes. Firstly, we generate the initial RBAC state that each permission only belongs to a role using formal concept analysis. Secondly, we optimize this RBAC state based on concept lattice factor considering both the user–role assignments and the permission–role assignments, ensuring that access requirements are met while reducing role proliferation.

The algorithm demonstrates effectiveness in handling various optimization tasks by reducing the dimensionality of the problem through concept lattice factorization. By identifying and utilizing the inherent relationships and dependencies among variables, it can efficiently explore the solution space and converge towards optimal or near-optimal solutions.

Our approach is purely data-driven, as all performance metrics are directly associated with the inherent features of the dataset. With this approach, we can quickly set the right goal for role mining before actually running any role mining algorithms.

However, there are areas for further improvement and future work. Firstly, the algorithm's performance could be evaluated and compared against existing state-of-the-art optimization algorithms to assess its competitiveness and scalability. Additionally, conducting comprehensive experimental studies on various benchmark problems and real-world applications would help validate its effectiveness and generalizability.

Furthermore, exploring ways to enhance the algorithm's robustness to handle noisy or uncertain data would be valuable. Investigating the algorithm's behavior on large-scale problems and developing strategies to scale it up effectively would also be beneficial.

Overall, the role minimization optimization by concept lattice factor presents a novel approach to optimization that shows promise [29]. Continued research and development could lead to further advancements, making it a valuable tool for solving complex optimization problems in various domains.

**Author Contributions:** Conceptualization, Q.W. and T.W.; methodology, Q.W.; software, T.W.; validation, T.W.; formal analysis, Q.W.; resources, T.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by the public welfare industry project of Zhejiang Science and Technology Department (LGG18F020012).

**Data Availability Statement:** The data used for this experiment are available in public repository. The detailed information about the data are provided in the result analysis section.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Jaeger, T.; Xiaolan, Z. Policy Management Using Access Control Spaces. *Int. J. ACM Trans.* **2003**, *6*, 327–364. [[CrossRef](#)]
2. Mitra, B.; Sural, S.; Vaidya, J.; Atluri, V. A Survey of Role Mining. *ACM Comput. Surv.* **2016**, *48*, 1–37. [[CrossRef](#)]
3. Mario, F.; Joachim, M.B.; David, B. On the Definition of Role Mining. In Proceedings of the ACM Symposium on Access Control Models and Technologies, Pittsburgh, PA, USA, 9–11 June 2010.
4. Krra, R.R.; Ashalatha, N.; Indranil, G.R.; Yogachandran, R.; Muttukrishnan, R. Role recommender-RBAC: Optimizing user-role assignments in RBAC. *Comput. Commun.* **2021**, *166*, 140–153.
5. Jurgen, S.; Ulrike, S. Role mining with ORCA. In Proceedings of the 10th ACM Symposium on Access Control Models and Technologies, Stockholm, Sweden, 1–3 June 2005.
6. Jaideep, V.; Vijayalakshmi, A.; Qi, G. The role mining problem: Finding a minimal descriptive set of roles. In Proceedings of the 12th ACM Symposium on Access Control Models and Technologies, Sophia Antipolis, France, 20–22 June 2007.
7. Hamid, N.A.; Ahmad, R.; Rahayu, S. Recent Trends in Role Mining Algorithms for Role-Based Access Control: A Systematic Review. *World Appl. Sci. J.* **2017**, *35*, 1054–1058.
8. Xu, Z.; Stoller, S.D. Algorithms for mining meaningful roles. In Proceedings of the 17th ACM symposium on Access Control Models and Technologies, Newark, NJ, USA, 20–22 June 2012; pp. 57–66.
9. Alina, E. *Biclique Covers of Bipartite Graphs: The Minimum Biclique Cover and Edge Concentration Problems*; Princeton University: Princeton, NJ, USA, 2007.
10. Huang, H.; Shang, F.; Liu, J.; Du, H. Handling least privilege problem and role mining in RBAC. *J. Comb. Optim.* **2015**, *30*, 63–68. [[CrossRef](#)]
11. Jaideep, V.; Vijayalakshmi, A.; Qi, G.; Haibing, L. Edge-RMP: Minimizing administrative assignments for role-based access control. *J. Comput. Secur.* **2009**, *17*, 211–235.
12. Ene, A.; Horne, W.; Milosavljevic, N.; Rao, P.; Schreiber, R.; Tarjan, R.E. Fast exact and heuristic methods for role minimization problems. In Proceedings of the 13th ACM Symposium on Access Control Models and Technologies (SACMAT 2008), Estes Park, CO, USA, 11–13 June 2008.
13. Vaidya, J.; Atluri, V.; Warner, J.; Guo, Q. Role engineering via prioritized subset enumeration. *IEEE Trans. Dependable Secur. Comput.* **2010**, *7*, 300–314. [[CrossRef](#)]
14. Carlo, B.; Stelvio, C. A simple role mining algorithm. In Proceedings of the 25th ACM Symposium on Applied Computing, Sierre, Switzerland, 22–26 March 2010.
15. Li, R.; Wang, W.; Ma, X.; Gu, X.; Wen, K. Mining roles using attributes of permissions. *Int. J. Innov. Comput. Inf. Control* **2012**, *8*, 7909–7923.
16. Ian, M.; Hong, C.; Tiancheng, L.; Qihua, W.; Ninghui, L.; Elisa, B.; Seraphin, C.; Jorge, L. Mining roles with multiple objectives. *ACM Trans. Inf. Syst. Secur.* **2010**, *13*, 1–35.
17. Mitra, B.; Sural, S.; Atluri, V.; Vaidya, J. Toward mining of temporal roles. In Proceedings of the 27th International Conference on Data and Applications Security and Privacy, Newark, NJ, USA, 15–17 July 2013.
18. Ye, W.; Li, R.; Gu, X.; Li, Y.; Wen, K. Role mining using answer set programming. *Futur. Gener. Comput. Syst.* **2016**, *55*, 336–343. [[CrossRef](#)]
19. Molloy, I.; Chen, H.; Li, T.; Wang, Q.; Li, N.; Bertino, E.; Calo, S.; Lobo, J. Mining roles with semantic meanings. In Proceedings of the Symposium on Sacmat, Estes Park, CO, USA, 11–13 June 2008.
20. Kumar, C.A. Designing role-based access control using formal concept analysis. *Secur. Commun. Netw.* **2013**, *6*, 373–383. [[CrossRef](#)]
21. Chen, B.; Qiu, J.D.; Chen, M.M. Designing Access Control Policy Using Formal Concept Analysis. *Appl. Mech. Mater.* **2014**, *602–605*, 3822–3825.
22. Koyda, M.; Stumme, G. Factorizing Lattices by Interval Relations. *Int. J. Approx. Reason.* **2023**, *157*, 70–87. [[CrossRef](#)]
23. Haibing, L.; Jaideep, V.; Vijayalakshmi, A. An optimization framework for role mining. *J. Comput. Secur.* **2014**, *22*, 1–31.
24. Belohlavek, R.; Trnecka, M. From-below approximations in Boolean matrix factorization: Geometry and new algorithm. *J. Comput. Syst. Sci.* **2015**, *81*, 1678–1697. [[CrossRef](#)]
25. Zhang, L.; Zhang, H.L.; Han, D.J.; Shen, X.J. Theory and algorithm for roles minimization problem in RBAC based on concept lattice. *Acta Electron. Sin.* **2014**, *42*, 2371–2378. (In Chinese)

26. Ian, M.; Ninghui, L.; Tiancheng, L.; Ziqing, M.; Qihua, W.; Jorge, L. Evaluating role mining algorithms. In Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, Stresa, Italy, 3–5 June 2009.
27. Scott, S.; Ping, Y.; Ramakrishnan, C.R.; Mikhail, G. Efficient policy analysis for administrative role based access control. In *ACM Conference on Computer and Communication Security, CCS*; ACM Press: Alexandria, WV, USA, 2007.
28. Abolfathi, M.; Raghebi, Z.; Jafarian, H.; Banaei-Kashani, F. A Scalable Role Mining Approach for Large Organizations. In Proceedings of the 2021 ACM Workshop on Security and Privacy Analytics, Virtual Event, USA, 28 April 2021; ACM Press: Alexandria, WV, USA, 2021.
29. Blundo, C.; Cimato, S.; Siniscalchi, L. Role Mining Heuristics for Permission-Role-Usage Cardinality Constraints. *Comput. J.* **2022**, *65*, 1386–1411. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.