An Optimization Model for the Extended Role Mining Problem

Emre Uzun, Vijayalakshmi Atluri, Haibing Lu and Jaideep Vaidya

MSIS Department and CIMIC, Rutgers University, USA
{emreu,atluri,haibing,jsvaidya}@cimic.rutgers.edu

Abstract. The primary purpose of Role Mining is to effectively determine the roles in an enterprise using the permissions that have already been assigned to the users. If this permission assignment is viewed as a 0-1 matrix, then Role Mining aims to decompose this matrix into two matrices which represent user-role and role-permission assignments. This decomposition is known as Boolean Matrix Decomposition (BMD). In this paper, we use an Extended BMD (EBMD) to consider separation of duty constraints (SOD) and exceptions, that are common to any security system, in the role mining process. Essentially, in EBMD, we introduce negative assignments. An additional benefit of allowing negative assignments in roles is that, a less number of roles can be used to reconstruct the same given user-permission assignments. We introduce Extended Role Mining Problem and its variants and present their optimization models. We also propose a heuristic algorithm that is capable of utilizing these models to find good decompositions.

1 Introduction

The recent developments in the usage of information technology in many different enterprises facilitate access to data. This situation brings out security issues that must be seriously considered in order to maintain confidentiality. In order to cope with this issue, many enterprises enforce strict access control policies on various data resources that they administer. A typical implementation is to have a 0-1 (Boolean) User-Permission Assignment (UPA) Matrix which indicates whether a particular user has access to a particular resource in the system. An example of this matrix is given in Figure 1.

	Resource 1	Resource 2	Resource 3
User 1	1	1	0
User 2	0	1	1
User 3	1	1	1

Fig. 1. A 0-1 User-Permission Access Control Matrix

Basically, this method can be used in small enterprises with relatively small number of resources. However, administration of this method gets complicated in large enterprises with many resources. Hence, companies seek for a more efficient way of manag-

ing permission assignments. As a result, Role Based Access Control (RBAC) methodologies are developed. The purpose of RBAC is to define roles which can be considered as a set of permissions and assign roles to users in order to grant permissions. This process makes the security administration easier, since the number of roles are significantly smaller than the number of users.

According to Edward Coyne, 'Definition of the roles with their assigned permissions must be accomplished before all the benefits of RBAC can be realized. The goal is to define a set of roles that is complete, correct and efficient' [2]. There are mainly two different approaches in determining roles: Top-down and bottom-up. Top-down approach is to determine the roles by carefully examining the business processes and identifying the potential roles which is in practice, defining job functions from scratch and associating the necessary permissions to the role. However, this method ignores the existing permission assignments and it is costly and labor intensive in large enterprises with large number of business processes and permissions [1],[2]. There are some implementations of top-down approach available in the literature such as [6]. The bottom-up approach, on the other hand utilizes the existing user-permission assignments and tries to aggregate them to obtain potential roles. However, the existing business processes are ignored and as a result of this, the roles that are obtained may not fully represent the existing job functions in the enterprise [8]. Basically, the bottom-up approach is called Role Mining (RM).

There are many different algorithms proposed in RM area. The very first algorithms aim to find a decomposition to a given UPA matrix. CompleteMiner, FastMiner [9] and ORCA [7] are some of these algorithms. After the formalization of the role mining problem (RMP) and its variants by Vaidya et al. [8], many different new algorithms that are capable of handling the new objectives are proposed. Many of these new algorithms are basically an adaptation of the solution procedures of an existing problem. Some examples are: Utilizing Minimum Database Tiling Problem, Discrete Basis Problem, Minimum Biclique Cover Problem and Graph Optimization [11], [1], [8]. Moreover, [4] provides computational tests and comparisons of most of these algorithms.

It is clear that the purpose of RM is to generate a user-to-role (UA) and a role-to-permission (PA) matrix from a given UPA matrix. This is in fact analogous to have a Boolean Matrix Decomposition (BMD) where the UPA matrix is decomposed into two Boolean matrices UA and PA [3]. This decomposition literally means that UPA matrix can exactly be represented by UA and PA matrices using the Boolean Matrix Multiplication operator described by Vaidya et al.[8]. Now, consider that one of the decomposed matrices is allowed to contain -1 in addition to 0 and 1. The purpose of having -1, or namely, negative assignments, is to introduce exception and separation of duty constraints. For instance, suppose that there are three roles in an enterprise: Manager, Auditor and Employee, where Managers have access to all of the permissions that Auditors and Employees have. Now suppose that a new manager, say John, is not allowed to access Auditor's permissions. Such exceptions are quite common to real world policies. This is supported through a negative assignment as it does not make sense to create a new role specifically to John alone. Negative user-role assignments mean that if a role is assigned to a user negatively, the user cannot have access to any permission of that role. The negative user-role assignment is superior to the positive (or

regular) user-role assignment. If the user is already assigned to a permission positively through another role, this assignment is automatically revoked. If the user is assigned to a permission positively in the future, it still does not become effective.

We observe that in addition to increasing administration flexibility, negative assignments can help discover alternative representations of UPA matrices. Consider the example of existing user-permission assignments UPA as shown in Figure 2, where $\{u_1, u_2, u_3, u_4\}$ denote users and $\{p_1, p_2, p_3, p_4\}$ denote permissions.

	p_1	p_2	p_3	p_4
u_1	1	0	1	1
u_2	1	0	1	1
u_3	1	1	0	1
u_4	0	1	0	1

Fig. 2. UPA Matrix

In Figure 3 the classical BMD decomposition and in Figure 4, a decomposition with negative role assignments are shown. Clearly, the UPA matrix can be represented by fewer number of roles using negative role assignments.

	r_1	r_2	r_3
u_1	1	0	1
u_2	1	0	1
u_3	1	1	0
u_4	0	1	0

$\otimes$

	p_1	p_2	p_3	p_4
r_1	1	0	0	1
r_2	0	1	0	1
r_3	0	0	1	0

Fig. 3. BMD Decomposition of the sample UPA Matrix in Figure 2

	r_1	r_2
u_1	1	0
u_2	1	0
u_3	1	1
u_4	0	1

$\odot$

	p_1	p_2	p_3	p_4
r_1	1	0	1	1
r_2	0	1	-1	1

Fig. 4. EBMD Decomposition of the sample UPA Matrix in Figure 2

The matrix decomposition with negative assignments is proposed by Lu et al.[5] and called Extended Boolean Matrix Decomposition (EBMD). We use their notion and definitions to utilize Extended Boolean Matrices in Role Mining area and develop Extended Role Mining (ERM), where we allow the UA matrix to contain negative assignments in addition to positive assignments.

RM only aims to decompose the UPA matrix without any objective, which implies any decomposition is indeed a RM task. Vaidya et al. [8] formulate RMP as an optimization problem seeking to minimize the number of roles. Furthermore, they also propose certain variants to RMP with different objectives like minimizing roles given a noise threshold or minimizing noise. In this paper, we propose Extended Role Mining Problem (ERMP) and its variants, in which we optimize the decomposition allowing one of the matrices contain negative assignments.

Since RMP and ERMP and their variants are optimization problems, they can be formulated using Mixed Integer Programming (MIP) techniques. Lu et al. [3] propose a MIP formulation for RMP and its variants. In this paper, we develop MIP formulations for ERMP and its variants. The main advantage of using MIP formulations is that, we can directly adopt many different tools developed for specifically for MIP to obtain a solution, which is guaranteed to be optimal.

Our contributions in this paper are: We define the Extended Role Mining Problem (ERMP) and its variants using EBMD. We propose MIP formulations for these problems. Moreover, we develop a heuristic procedure that seeks to find a good decomposition to a given UPA matrix using the proposed MIP formulations.

The rest of the paper is organized as follows. In the Section 2, a more formal problem definition and some preliminary background information is given. In Section 3, we introduce our Mixed Integer Programming formulations for ERMP and its variants. We present our heuristic algorithm for the ERMP and its variants in Section 4. Finally, conclusions and remarks are noted at Section 6.

2 Problem Definition and Preliminaries

In this section necessary notations and definitions are given.

2.1 Notations and Preliminary Definitions

RBAC

- Let M, K, OPS, and OBJ be the set of users, roles, operations and objects, respectively.
- Let $UA \subseteq M \times K$, be a many-to-many mapping user-to-role assignment relation.
- N(the set of permissions) $\subseteq \{(op, obj) | op \in OPS \text{ and } obj \in OBJ\}$
- Let $PA \subseteq K \times N$ be a many-to-many mapping of role-to-permission assignments.
- Let $UPA \subseteq M \times N$ be a many-to-many mapping of user-to-role assignments.
- Let $assigned\_users(k) = \{m \in M | (m, k) \in UA\}$ be the mapping of role k onto a set of users.
- Let $assigned\_permissions(k) = \{n \in N | (n, k) \in PA\}$ be the mapping of role k onto the set of permissions.

Boolean Matrix Multiplication A Boolean matrix multiplication between Boolean Matrices $A \in \{0,1\}^{m \times k}$ and $B \in \{0,1\}^{k \times n}$ is $A \otimes B = C$ where C is in space $\{0,1\}^{m \times n}$ and

$$c_{ij} = \bigvee_{l=1}^{k} (a_{il} \wedge b_{lj}).$$

Boolean Matrix Decomposition If $A = B \otimes C$, where A, B, C are Boolean matrices, $B \otimes C$ is called the decomposition of A.

Extended Boolean Matrix Multiplication Given a matrix $C_{k\times n} \in \{0,1\}^{k\times n}$ and a matrix $B_{m\times k} \in \{-1,0,1\}^{m\times k}$, the matrix $A_{m\times n}$ obtained from the operation $B \odot C$ has the following properties:

- If $\exists t_1 : (c_{it_1} = 1 \ \wedge \ b_{t_1 j} = 1) \ \wedge \ \neg\exists t_2 : (c_{it_2} = 1 \ \wedge \ b_{t_2 j} = -1), \ then \ a_{ij} = 1$
- If $\neg\exists t_1 : (c_{it_1} = 1 \ \wedge \ b_{t_1 j} = 1) \ \vee \ \exists t_2 : (c_{it_2} = 1 \ \wedge \ b_{t_2 j} = -1), \ then \ a_{ij} = 0$

where $i \in \{1, .., m\}$ and $j \in \{1, .., n\}$

Extended Boolean Matrix Decomposition Given matrices $A_{m\times n} \in \{0,1\}^{m\times n}$ and $C_{k\times n} \in \{0,1\}^{k\times n}$ and a matrix $B_{m\times k} \in \{-1,0,1\}^{m\times k}$, $A = B \odot C$ is called the EBMD of A, if $A_j = \cup_{b_{ij}=1} C_i \setminus \cup_{b_{ij}=-1} C_i$ where A_j denotes the item subset corresponding to elements of 1 in the j^{th} column of A and C_i denotes similarly.

δ-Consistency A given user-to-role assignment UA, role-to-permission assignment PA and user-to-permission assignment UPA are δ-consistent if and only if

$$||M(UA) \otimes M(PA) - M(UPA)||_1 \leq \delta$$

where $M(UA), M(PA)$ and $M(UPA)$ denote the matrix representation of UA, PA and UPA, respectively.

If negative assignments are allowed in UA, then the condition to be satisfied changes to

$$||M(UA) \odot M(PA) - M(UPA)||_1 \leq \delta$$

where $M(UA), M(PA)$ and $M(UPA)$ denote the matrix representation of UA, PA and UPA, respectively.

L_1 Norm The L_1 Norm of a d-dimensional vector $v \in X^d$, for some set X is,

$$||v||_1 = \sum_{i=1}^{d} |v_i|$$

This definition can be expanded to a distance metric between two vectors v and w as

$$||v - w||_1 = \sum_{i=1}^{d} |v_i - w_i|$$

Furthermore, the definition can be applied to $n \times m$ matrices A and B as

$$||A - B||_1 = \sum_{i=1}^{n} ||a_i - b_i||_1 = \sum_{i=1}^{n} \sum_{i=1}^{m} |a_{ij} - b_{ij}|$$

2.2 Problem Definitions

Vaidya et al. [8] describe the Role Mining Problem (RMP) as follows:

Role Mining Problem (RMP): Given a set of users M, a set of permissions N and a user-permission assignment UPA, find a set of roles $ROLES$, a user-to-role assignment UA and a role-to-permission assignment PA that is 0-consistent with UPA and minimizing the number of roles, k.

The purpose of RMP is to decompose the UPA into PA and UA in such a way that the decomposition exactly describes the UPA and the number of roles are minimized. In theory, enterprises would like to implement RMP to obtain a set of roles. However, obtaining an exact decomposition is not always practical in large UPA matrices. If one allows some amount of "noise" in the decomposition, then the UA and PA matrices obtained from the decomposition do not fully represent the original UPA matrix ($UA \otimes PA = UPA' \neq UPA$), meaning that some of the entries in UPA' matrix are different than the original UPA matrix. Vaidya et al. [8] discuss the situation of having an noised decomposition and define the Minimum Noise RMP.

Minimum Noise RMP: Given a set of users M, a set of permissions N, a user-permission assignment UPA, and the number of roles k, find a set of k roles K, a-user-to-role assignment UA and a role-to-permission assignment PA minimizing

$$||M(UA) \otimes M(PA) - M(UPA)||_1$$

where $M(UA), M(PA)$ and $M(UPA)$ denote the matrix representation of UA, PA and UPA, respectively.

One other variation of RMP is the Edge RMP problem. The difference of Edge RMP is that rather than minimizing the number of roles, minimize the number of user-role and role-permission assignments [3].

Edge RMP: Given a set of users M, a set of permissions N and a user-permission assignment UPA, find a set of roles K, a user-to-role assignment UA and a role-to-permission assignment PA that is 0-consistent with UPA and minimizing $|UA|+|PA|$.

RMP, Minimum Noise RMP and Edge RMP are all NP-Complete problems [8]. These problems are all optimization problems and they only deal with Boolean matrices. Using the Extended Boolean Matrix Decomposition, we now can define the Extended Role Mining Problem and its variants:

Extended Role Mining Problem (ERMP): Given a set of users M, a set of permissions N and a user-permission assignment UPA, find a set of roles K, a user-to-role assignment UA where negative assignments are allowed and a role-to-permission assignment PA that is 0-consistent with UPA and minimizing the number of roles, k.

Extended Minimum Noise Role Mining Problem (Minnoise ERMP): Given a set of users M, a set of permissions N, a user-permission assignment UPA, and the number of roles k, find a set of k roles K, a-user-to-role assignment UA where negative assignments are allowed and a role-to-permission assignment PA minimizing

$$||M(UA) \odot M(PA) - M(UPA)||_1$$

where $M(UA), M(PA)$ and $M(UPA)$ denote the matrix representation of UA, PA and UPA, respectively.

Note that, unlike BMD in which we use the symbol $\otimes$, in EBMD we use the symbol $\odot$ as the matrices contains 0, 1 and -1.

Extended Edge Role Mining Problem (Edge ERMP): Given a set of users M, a set of permissions N and a user-permission assignment UPA, find a set of roles K, a user-to-role assignment UA where negative assignments are allowed and a role-to-permission assignment PA that is 0-consistent with UPA and minimizing $|UA|+|PA|$

3 Mathematical Models for ERMP and its variants

In this section, we present the MIP formulations for the ERMP and its variants. Each of these models utilize an initial decomposition of UPA matrix, which can be obtained using an algorithm proposed in the literature such as FastMiner [9]. The main purpose of using an initial decomposition is due to the fact that the optimization models become non-linear unless one of the matrices UA or PA is known. This is the same approach used by Lu et al. [3] to formulate mathematical models for RMP. Given Boolean matrices UPA and PA, our models try to establish a decomposition consisting of an Extended Boolean UA matrix and a Boolean PA matrix while improving the decomposition in terms of the objective metric. In our analysis, we assume Extended Boolean UA and Boolean PA matrices, and perform our experiments based on this assumption. The opposite case is symmetric and not covered in this paper.

The following models are used to obtain a (0,1,-1) UA matrix given PA and UPA matrices. The decision variables and the parameters used in these models are as follows:

Decision Variables

- Let $x^+_{ik} = \begin{cases} 1, \ if\ the\ user\ i\ is\ positively\ assigned\ to\ role\ k, \ k \in K, \ i \in M \\ 0, \ otherwise \end{cases}$
- Let $x^-_{kj} = \begin{cases} 1, \ if\ the\ user\ i\ is\ negatively\ assigned\ to\ role\ k, \ k \in K, \ i \in M \\ 0, \ otherwise \end{cases}$
- Let $y_k = \begin{cases} 1, \ if\ role\ k\ is\ used \\ 0, \ otherwise \end{cases}$
- Let $t_{ij} \in \{0, 1\}$ be an indicator variable, $i \in M, \ j \in N$
- Let u^+_{ij} denote the amount of noise caused by positively realized x^+_{ik} variables, $i \in M, \ j \in N \ k \in K$
- Let u^-_{ij} denote the amount of noise caused by positively realized x^-_{ik} variables, $i \in M, \ j \in N \ k \in K$

Parameters

- Let a_{ij} denote the entry (i, j) of matrix UPA.
- Let c_{kj} denote the entry (k, j) of matrix PA.

The objective of the ERMP problem is to minimize the total number of roles that are used. On the other hand, Minnoise ERMP seeks to minimize the number of noise in the decomposition given a fixed number of roles and Edge ERMP seeks to find the decomposition that has the least number of role assignments. The primary purpose of using Extended Boolean Matrix Decomposition instead of classic Boolean Matrix Decomposition is to further decrease the size of the resulting matrices (as discussed in Section 1),

hence in our case, decreasing the number of roles. Although Minnoise ERMP and Edge ERMP does not have an objective of minimizing the number of roles, to capture the effect of using Extended Boolean Matrices, we slightly alter the objectives of Minnoise ERMP and Edge ERMP to reflect this property. Hence the objective functions of these problems are composed of two components, one being the sum of the roles.

Other than the objective functions, the feasible region declarations of all of these three models are very similar. Thus, here we give a common explanation to the constraints of each of these models. Constraints 2 and 3, 12 and 13, and 24 and 25 ensure the $\odot$ property of the entries valued 1 in the UPA matrix in ERMP, Minnoise ERMP and Edge ERMP Models, respectively. For each of these entries, both constraints must be satisfied. Constraints 2 and 24 force that there exists at least one positive matching entry in the UA and PA matrices that will satisfy the $\odot$ property. Similarly, Constraints 3 and 25 force that there does not exist any negative matching entries in the UA and PA matrices. The logic in the Constraints 12 and 13 is the same but the main difference is that the $\odot$ property does not have to be satisfied (which implies a noise in the decomposition). Constraints 2 and 3, 12 and 13, and 24 and 25 ensure the $\odot$ property of the entries valued 0 in the UPA matrix in ERMP, Minnoise ERMP and Edge ERMP Models, respectively. The structure of these constraints are similar to the first set of constraints. However the major difference is that for each 0 entry in the UPA matrix, either one of these constraint tuples or both must be satisfied. This is handled using the decision variable t_{ij} which sets at least one of these constraints to be enforced. The constant M in these constraints is a value sufficiently big to make any of these constraints redundant depending on the value of t_{ij}. In constraints 12, 13, 14 and 15, the amount of noise is determined by u^+_{ij} and u^-_{ij} variables. Constraints 6, 16 and 28 ensure that only one of the variables x^+_{ik} and x^-_{ik} can take positive value at the same time (i.e: a cell in the UA matrix cannot take 1 and -1 values at the same time) in ERMP, Minnoise ERMP and Edge ERMP Models, respectively. However, they can both be 0 at the same time which indicates a 0 in the corresponding cell. Constraints 7 and 8, 17 and 18, and 29 and 30 ensure that a role is active whenever there is at least one user assigned either positively or negatively to that role.

3.1 MIP Formulation for ERMP

$$min \sum_{k \in K} y_k \tag{1}$$

$$s.t$$

$$\sum_{k \in K s.t. a_{ij}=1} x^+_{ik} c_{kj} \geq 1,\ \forall i \in M,\ j \in N \tag{2}$$

$$\sum_{k \in K s.t. a_{ij}=1} x^-_{ik} c_{kj} = 0,\ \forall i \in M,\ j \in N \tag{3}$$

$$\sum_{k \in K s.t. a_{ij}=0} x^+_{ik} c_{kj} \leq t_{ij} M,\ \forall i \in M,\ j \in N \tag{4}$$

$$\sum_{k \in K s.t. a_{ij}=0} x^-_{ik} c_{kj} \geq 1 - (1 - t_{ij}) M,\ \forall i \in M,\ j \in N \tag{5}$$

$$x_{ik}^{+} + x_{ik}^{-} \leq 1,\ \forall k, j \tag{6}$$
$$y_k \geq x_{ik}^{+},\ \forall k \in K,\ i \in M \tag{7}$$
$$y_k \geq x_{ik}^{-},\ \forall k \in K,\ i \in M \tag{8}$$
$$t_{ij} \in \{0,1\},\ \forall i \in M, j \in N \tag{9}$$
$$x_{ik}^{+}, x_{ik}^{-} \in \{0,1\},\ \forall k \in K, i \in M \tag{10}$$

3.2 MIP Formulation for Minnoise ERMP

$$min \sum_{i \in M} \sum_{j \in N} u_{ij} + \sum_{k \in K} y_k \tag{11}$$
$$s.t.$$
$$\sum_{k \in K s.t. a_{ij}=1} x_{ik}^{+} c_{kj} + u_{ij}^{+} \geq 1,\ \forall i \in M,\ j \in N \tag{12}$$
$$\sum_{k \in K s.t. a_{ij}=1} x_{ik}^{-} c_{kj} - u_{ij}^{-} = 0,\ \forall i \in M,\ j \in N \tag{13}$$
$$\sum_{k \in K s.t. a_{ij}=0} x_{ik}^{+} c_{kj} - u_{ij}^{+} \leq t_{ij} M,\ \forall i \in M,\ j \in N \tag{14}$$
$$\sum_{k \in K s.t. a_{ij}=0} x_{ik}^{-} c_{kj} + u_{ij}^{-} \geq 1 - (1 - t_{ij}) M,\ \forall i \in M,\ j \in N \tag{15}$$
$$x_{ik}^{+} + x_{ik}^{-} \leq 1,\ \forall k, j \tag{16}$$
$$y_k \geq x_{ik}^{+},\ \forall k \in K,\ i \in M \tag{17}$$
$$y_k \geq x_{ik}^{-},\ \forall k \in K,\ i \in M \tag{18}$$
$$t_{ij} \in \{0,1\},\ \forall i \in M, j \in N \tag{19}$$
$$x_{ik}^{+}, x_{ik}^{-} \in \{0,1\},\ \forall k \in K, i \in M \tag{20}$$
$$u_{ij}^{+}, u_{ij}^{-} \geq 0, \forall i \in M, j \in N \tag{21}$$
$$\tag{22}$$

3.3 MIP Formulation for Edge ERMP

$$min \sum_{i \in M} \sum_{k \in K} x_{ik}^{+} + x_{ik}^{-} + \sum_{k \in K} y_k \tag{23}$$
$$s.t.$$
$$\sum_{k \in K s.t. a_{ij}=1} x_{ik}^{+} c_{kj} \geq 1,\ \forall i \in M,\ j \in N \tag{24}$$
$$\sum_{k \in K s.t. a_{ij}=1} x_{ik}^{-} c_{kj} = 0,\ \forall i \in M,\ j \in N \tag{25}$$

$$\sum_{k \in K s.t. a_{ij}=0} x^+_{ik} c_{kj} \leq t_{ij} M, \ \forall i \in M, \ j \in N \tag{26}$$

$$\sum_{k \in K s.t. a_{ij}=0} x^-_{ik} c_{kj} \geq 1 - (1 - t_{ij}) M, \ \forall i \in M, \ j \in N \tag{27}$$

$$x^+_{ik} + x^-_{ik} \leq 1, \ \forall k, j \tag{28}$$

$$y_k \geq x^+_{ik}, \ \forall k \in K, \ i \in M \tag{29}$$

$$y_k \geq x^-_{ik}, \ \forall k \in K, \ i \in M \tag{30}$$

$$t_{ij} \in \{0, 1\}, \ \forall i \in M, j \in N \tag{31}$$

$$x^+_{ik}, x^-_{ik} \in \{0, 1\}, \ \forall k \in K, i \in M \tag{32}$$

4 Heuristic Procedure

In this section, we introduce the heuristic algorithm we propose to find good decompositions to ERMP, Minnoise ERMP and Edge ERMP utilizing the Mixed Integer Programming formulations. Our algorithm is an iterative algorithm which takes a Boolean UPA matrix and a corresponding Boolean PA matrix as an input and tries to improve the decomposition by finding better Extended Boolean UA and Boolean PA matrices at each iteration. The algorithm mainly has two stages: Preprocessing Stage and Iterative Stage. We now explain each stage in detail.

We need a Preprocessing Stage since the MIP formulations that we propose require an initial PA matrix. This PA matrix can be obtained using one of the heuristic Boolean matrix decomposition procedures available in the literature. We use the algorithm described in Vaidya et al. [10] for this purpose. When we implement this algorithm, we get Boolean UA and PA matrices for the corresponding Boolean UPA matrix. Although this PA matrix can be used as the initial PA matrix of our heuristic algorithm, we use RMP formulation described by Lu et al. [3] to further improve it. This RMP formulation takes the UPA and UA matrices as input and constructs the the corresponding optimal PA' matrix, while minimizing the number of roles. This PA' matrix is expected to have smaller (or equal) number of roles when compared to the PA matrix and it is used as the initial matrix of the Iterative Stage of our heuristic procedure. This initial decomposition is not the optimal Boolean Matrix Decomposition of the UPA matrix, rather we obtain a heuristic decomposition and try to improve it as much as we can to get a good starting matrix. Note that none of the matrices used in this stage contains -1 entries.

At each iteration of the Iterative Stage, we either obtain the corresponding optimal Extended Boolean UA matrix given the Boolean PA matrix of the previous iteration, or we obtain the corresponding Boolean PA matrix given the Extended Boolean UA matrix of the previous iteration. The purpose of doing this round-robin operation lies under the fact that in each iteration when we obtain a corresponding optimal UA (PA) matrix using a PA (UA) matrix, the PA (UA) matrix may not be the optimal given the new UA (PA) matrix. Hence we need to do this round-robin operation until we do not observe any improvement in the decomposition. We define the improvement metric and termination criteria later in this section. At an iteration, if a UA matrix is to be obtained given a PA matrix, then one of the ERMP, Minnoise ERMP or Edge ERMP

model is used (This selection is fixed throughout the algorithm). On the other hand, if a PA matrix is to be obtained given a UA matrix, then we need an additional model. Notice that our proposed MIP formulations require a Boolean PA matrix to construct an Extended Boolean UA matrix. However, we cannot use these formulations to obtain a Boolean PA matrix, given an Extended Boolean UA matrix. For this purpose, we develop a Reverse ERMP model as a MIP formulation seeking to minimize number of roles. We do not present the model here since it is very similar to our proposed formulations. See Appendix A for the model formulation. In summary, in the Iterative Stage, we bounce back and forth in a round-robin fashion constructing UA given PA and PA given UA using the selected ERMP formulation and Reverse ERMP formulation, respectively, until we observe N_I consecutive iterations without any improvement or we observe a decomposition which is exactly the same as the minimum solution observed so far (this implies that we are in an infinite loop). Note that, in the Minnoise ERMP case, the solution we obtain may contain some noise, which implies that the resulting UA and PA matrices do not fully represent the UPA matrix. In this case, we cannot use this result to bounce back using the Reverse ERMP Model, because it requires an exact decomposition. So, during the iterative step, if we observe noise in decomposition, we terminate the algorithm at that point. Also note that, although we use MIP formulations and obtain optimal corresponding matrices at each iteration, the overall algorithm is heuristic and may not terminate at a global optimum since we start with a heuristic decomposition and improve only one matrix at a time.

In order to define the improvement metric in our algorithm, we first need to define certain algorithm parameters:

Let $|UA|$ and $|PA|$ denote the number of nonnegative entries in matrices UA and PA, respectively. Let $cur(|UA|)$ and $cur(|PA|)$ be the current values and $min(|UA|)$ and $min(|PA|)$ be the minimum observed values of $|UA|$ and $|PA|$, respectively and let $cur(k)$ be the current and $min(k)$ be the minimum observed value of the number of roles, k. Then, an improvement occurs iff

$$[cur(|UA|) + cur(|PA|) \leq min(|UA|) + min(|PA|)] \vee cur(k) < min(k)$$

Another parameter is n_i which denotes the current number of iterations in which no improvement occurs. Then, the algorithm terminates iff

$$n_i = N_I \vee [cur(|UA|) = min(|UA|) \wedge cur(|PA|) = min(|PA|) \wedge cur(k) = min(k)]$$

This expression denotes that we terminate the algorithm if we do not observe any improvement in N_I consecutive iterations or we observe the minimum solution again which implies that the algorithm enters an infinite loop.

Now, we give our algorithm to ERMP and its variants:

Algorithm 1 Algorithm for ERMP Problem and its Variants

```
Initialize
Do preprocessing
while n_i < N_I do
  Obtain the corresponding optimal UA matrix
  if There is an improvement then
    Update statistics
  else if Same decomposition observed again then
    break
  else
    Increment n_i
  end if
  Obtain the corresponding optimal PA matrix
  if There is an improvement then
    Update statistics
  else if Same decomposition observed again then
    break
  else
    Increment n_i
  end if
end while
```

5 Computational Experiments and Results

In this section, we present the results of our computational experiments. We code basic structure of our algorithm using C programming language which communicates with CPLEX 12 Optimization Package via CPLEX Callable Library to perform the optimization. We perform our experiments on a Intel Core2Duo 2.00 GHz machine with 2.00 GB memory running 32-bit Windows 7. We have 2 real and 9 randomly generated synthetic data sets with various different sizes. The synthetic data sets can be separated into three groups according to their sizes (There are 3 synthetic data sets with 100 users and 50 permissions; 3 data sets for 200 users and 100 permissions and 3 data sets for 300 users and 150 permissions).

The results are summarized in Table 1. In this table, Size column denotes the number of users (M) and permissions (N). The Initial Decomposition column denotes the statistics of the initial solution, and the other columns state the results of ERMP, Minnoise ERMP and Edge ERMP, respectively. The % column denotes the percentage improvement in the number of roles in each case. In the results, we take the average of 3 synthetic data sets with equal sizes.

According to the results we see that in the Synthetic data sets our algorithm performs better when the problem size increases. Especially, the improvement of the starting solution in terms of the number of roles in the Data Set 3 is significant as we have an improvement of 8%. Furthermore, Edge ERMP performs better when compared to the ERMP and Minnoise ERMP since there is always a decrease in the number of assignments, which is in fact reasonable when we migrate from BMD to EBMD. We believe that the reason for getting small improvements is due to the pure random nature

Data Set	Size	Initial Decompst.			ERMP				Minnoise ERMP				Edge ERMP			
	$(M-N)$	$\|UA\|$	$\|PA\|$	K	$\|UA\|$	$\|PA\|$	K	%	$\|UA\|$	$\|PA\|$	K	%	$\|UA\|$	$\|PA\|$	K	%
Syn.D.1	100 - 50	400.6	59	20	400.6	59	20	0	400.6	59	20	0	315.6	59	20	0
Syn.D.2	200 - 100	767.6	271.6	50.6	751.6	257.3	49.3	2.6	751.6	257	49.3	2.6	611	265	50.6	0
Syn.D.3	300 - 150	1618	903.6	111	1506.6	729	102	8.1	1594.3	864	108.6	2.1	886	911	106.6	3.9
Real D.1	231 - 79	726	152	22	682	233	15	31	625	145	20	10	581	145	20	10
Real D.2	46 - 46	438	381	17	228	317	14	17	354	317	14	17	53	317	14	17

Table 1. Computational Results

of the Synthetic Data Sets. However, since the Real Data Sets are not purely random (i.e, it is reasonable to assume that there can be a pattern in the distribution of the user-permission assignments), the improvement is more significant in terms of the number of roles. For instance, the improvement in Real Data Set 1 for ERMP is 31%.

The limitations of our algorithm is that, since it utilizes MIP formulations, the problem cannot easily be solved for large data sets. CPLEX and other MIP optimizers use Branch and Cut techniques which tend to grow exponentially as the problem size increases. Moreover, although we use MIP formulations and obtain optimal corresponding matrices at each iteration, the overall algorithm is heuristic and may not terminate at a global optimum since we start with a heuristic decomposition and improve only one matrix at a time.

6 Conclusions

The advancements in Role Mining aids in finding better role distributions that will increase effectiveness and efficiency of RBAC systems. Since a basic RBAC scheme is composed of Boolean matrices which represent the user-role assignments, usage of negative assignments in extended Boolean matrices can take into account exceptions and separation of duty constraints while performing role mining. In this paper, we propose the Extended Role Mining Problem and its variants, which allow negative assignments. We present the MIP formulations for each of these problems. We also develop a heuristic procedure which utilizes these formulations to obtain a better decomposition. Our experimental results indicate that EBMD can result in significantly less number of roles when compared to BMD.

Some of the future work can be a better evaluation of the heuristic algorithm with more test runs and using synthetic data where the optimal decomposition is known. Furthermore, the Reverse ERMP model can be improved to cover Minnoise ERMP and Edge ERMP objectives of minimizing noise and assignments rather than only minimizing number of roles in the decomposition.

References

1. A. Ene, W. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R. E. Tarjan. Fast exact and heuristic methods for role minimization problems. *In proceedings of Symposium on Access Control Models and Technologies (SACMAT)*, pages 1–10, 2008.
2. M. Kuhlmann, D. Shohat, and G. Schimpf. Role mining - revealing business roles for security administration using data mining technology. *In Symposium on Access Control Models and Technologies (SACMAT)*, 2003.

3. H. Lu, J. Vaidya, and V. Atluri. Optimal boolean matrix decomposition: Application to role engineering. *In proceedings of International Conference on Data Engineering (ICDE)*, pages 297 – 306, 2008.
4. I. Molloy, N. Li, T. Li, Z. Mao, Q. Wang and J. Lobo. Evaluating Role Mining Algorithms. *In proceedings of ACM Symposium on Access Control Models and Technologies (SACMAT)*, 2009.
5. H. Lu, J. Vaidya, V. Atluri, and Y. Hong. Extended boolean matrix decomposition. *Ninth IEEE International Conference on Data Mining (ICDM)*, pages 317 – 326, 2009.
6. A. Schaad, J. Moffett, and J. Jacob. The role-based access control system of a european bank: A case study and discussion. *In proceedings of ACM Symposisum on Access Control Models and Technologies*, pages 3 – 9, 2001.
7. J. Schlegelmilch and U. Steffens. Role mining with ORCA. *In Symposium on Access Control Models and Technologies (SACMAT)*, 2005.
8. J. Vaidya, V. Atluri, and Q. Guo. The role mining problem: Finding a minimal descriptive set of roles. *In proceedings of Symposium on Access Control Models and Technologies (SACMAT)*, pages 175 – 184, 2007.
9. J. Vaidya, V. Atluri, and J. Warner. Roleminer: mining roles using subset enumeration. *In Proceedings of the ACM conference on Computer and Communications security*, pages 144 – 153, 2006.
10. J. Vaidya, V. Atluri, and Q. Guo. The role mining problem: A formal perspective. *ACM Trans. Inf. Syst. Secur.*, 13(3):1–31, 2010.
11. D. Zhang, K. Ramamohanrao, and T. Ebringer. Role engineering using graph optimisation. *In Symposium on Access Control Models and Technologies (SACMAT)*, pages 139 – 144, 2007.

A Reverse ERMP Model

The following model is used to obtain a Boolean PA matrix given an Extended Boolean UA matrix. The formulation is similar to the ERMP formulation given in the previous section. However, the only difference is that the objective is to minimize the number of roles only.

Decision Variables

- Let $y_k = \begin{cases} 1, \ if\ role\ k\ is\ used \\ 0, \ otherwise \end{cases}$
- Let $x_{kj} = \begin{cases} 1, \ if\ permission\ j\ is\ assigned\ to\ role\ k \\ 0, \ otherwise \end{cases}$
- Let $t_{ij} \in \{0, 1\}$ be an indicator variable, $i \in M, \ j \in N$

Parameters

- Let a_{ij} denote the entry (i, j) of matrix UPA.
- Let b^+_{ik} is 1 if the entry (i, k) of matrix UA is 1, 0 otherwise.
- Let b^-_{ik} is 1 if the entry (i, k) of matrix UA is -1, 0 otherwise.

Then the model is as follows:

$$min \sum_{k \in K} y_k \tag{33}$$

$$s.t.$$

$$\sum_{k \in K s.t. a_{ij}=1} b_{ik}^{+} x_{kj} \geq 1,\ \forall i \in M,\ j \in N \tag{34}$$

$$\sum_{k \in K s.t. a_{ij}=1} b_{ik}^{-} x_{kj} = 0,\ \forall i \in M,\ j \in N \tag{35}$$

$$\sum_{k \in K s.t. a_{ij}=0} b_{ik}^{+} x_{kj} \leq t_{ij} M,\ \forall i \in M,\ j \in N \tag{36}$$

$$\sum_{k \in K s.t. a_{ij}=0} b_{ik}^{-} x_{kj} \geq 1 - (1 - t_{ij}) M,\ \forall i \in M,\ j \in N \tag{37}$$

$$y_k \geq x_{kj},\ \forall k \in K,\ j \in N \tag{38}$$

$$t_{ij} \in \{0, 1\},\ \forall i \in M, j \in N \tag{39}$$

$$x_{kj} \in \{0, 1\},\ \forall k \in K, j \in N \tag{40}$$