Abstract. We present a new technique in parameterized algorithms whose purpose is to solve “resiliency versions” of decision problems. In resiliency problems, the goal is to decide whether an instance remains positive after any (appropriately defined) perturbation applied to it. To tackle this kind of problems, which might be of important practical interest, we introduce a notion of resiliency for Integer Linear Programs (ILP).

Recall that the classical ILP Feasibility problem asks whether there exists an integral assignment of variables $x$ satisfying $Ax \leq b$. In our resiliency version, we introduce resiliency variables $z$ and split the ILP into three parts: $Ax \leq b$, $Cx + Dz \leq e$ and $Fz \leq g$. The ILP Resiliency problem asks whether, for any integral assignment of $z$ satisfying $Fz \leq g$, one can find an integral assignment of $x$ satisfying the remaining constraints. While a celebrated result from Lenstra (Mathematics of Operations Research, 1983) implies that ILP Feasibility is Fixed-Parameter Tractable (FPT) parameterized by the number of variables (i.e. runs in time $O^*(f(n))$, where $n$ is the number of variables, $f$ is some computable function, and $O^*(\cdot)$ suppresses polynomial terms), this tool seems too weak to solve ILP Resiliency in an efficient way.

We first give evidence that testing resiliency of an ILP is computationally harder than simply testing its feasibility, by proving that ILP Resiliency is $\Pi^p_2$-complete (i.e. belongs to the second level of the polynomial hierarchy). We then provide an analog of Lenstra’s theorem but using a larger parameter. Our result uses an adaptation of the Fourier-Motzkin elimination for Linear Programming to the integral case introduced by Williams (J. Comb. Theory, Ser. A, 1976), which might be of independent interest. We believe that our method provides a generic framework for proving that resiliency versions of many parameterized problems remain FPT. To illustrate this fact, we present two applications of our approach to concrete problems.

The first one is called Resiliency Checking Problem (RCP) and was introduced by Li et al. (ACM Trans. Inf. Syst. Sec. 2009). It can be seen as a resiliency version of a generalization of the classical Set Cover problem parameterized by the size of the universe. We prove a result which completes a multivariate complexity classification of the problem we initiated in a previous study.

The other problem is Closest String, a problem arising in computational biology. Closest String is known to be the problem which popularized the use of Lenstra’s algorithm for obtaining FPT results (Gramm et al., Algorithmica, 2003). We solve a resiliency version of this problem (thus generalizing the work of Gramm et al.) motivated by the fact that in practice, some parts of the input might be unreliable.

Gutin’s research was supported by Royal Society Wolfson Research Merit Award.
1 Introduction

The notion of resiliency measures the extent to which a system can tolerate modifications to its configuration and still satisfy given criteria. An organization might, for example, wish to know whether it will still be able to continue functioning, even if some of its staff become unavailable. In the language of decision problems, we would like to know whether an instance is still positive after any (appropriately defined) modification. Intuitively, the resiliency version of a problem is likely to be harder than the problem itself; a naive algorithm would consider every allowed modification of the input, and then see whether a solution exists. In this paper, we introduce a new framework for dealing with this kind of problems, and study their computational complexity through the lens of fixed-parameter tractability. We first define resiliency for Integer Linear Programs (ILP), and confirm the intuition described above by proving that testing resiliency of an ILP is complete for the second level of the polynomial hierarchy. Then, we prove the fixed-parameter tractability (FPT) of this new problem, relying on an adaptation of the Fourier-Motzkin elimination for Linear Programs [4] to the integral case.

Brief Description of Our Approach. Questions of ILP feasibility are typically answered by finding an integral assignment of variables $x$ satisfying $Ax \leq b$. By Lenstra’s theorem [14], this problem can be solved in $O(f(n)LO^{O(1)})$ time and space, where $f$ is a function of the number of variables $n$ only, and $L$ is the size of the ILP (subsequent research has obtained an algorithm of the above running time with $f(n) = n^{O(n)}$ and using polynomial space [7, 12]). In the language of parameterized complexity, this means that ILP Feasibility is Fixed-Parameter Tractable (FPT) parameterized by the number of variables. For more details on this topic, we refer the reader to [3, 6].

We informally introduce resiliency for ILP as follows. We add another set of variables $z$, which can be seen as “resiliency variables”. We then consider the following ILP:\footnote{To save space, we will always implicitly assume that integrality constraints are part of every ILP of this paper. The case of Mixed Integer Linear Programs is discussed in Section 6.}

\begin{align}
Ax & \leq b \\
Cx + Dz & \leq e \\
Fz & \leq g
\end{align}

The idea is that inequalities (1) and (2) represent the intrinsic structure of the problem, among which inequalities (2) represent how the resiliency variables modify the instance. Inequalities (3), finally, represent the structure of the resiliency part. The goal of ILP Resiliency is to decide, for any integral assignment of variables $z$ satisfying inequalities (3), whether there exists an integral assignment of variables $x$ satisfying the whole ILP. We prove that this problem is complete for the second level of the polynomial hierarchy. As our main result, we show that ILP Resiliency is FPT when parameterized by the number of variables, constraints and the maximum value of the coefficients of the ILP. While this parameter may look rather large at the first sight, it will still allow us to apply our result to several concrete problems, as we will see later. Moreover, we stress that, as in the recent result for Integer Quadratic Programming [16], the right-hand sides of the considered ILP are not part of the parameter. These facts make the problem non-trivial from the parameterized point of view, since it may seem necessary to go through every possible assignment of $z$ to solve the resiliency problem, and the number of such assignments generally depends on the values of the right-hand sides of the ILP. Hence, while a naive algorithm would try every possibility for the resiliency variables, we propose in this paper an efficient algorithm assuming that our parameter is small. Our method offers a generic framework to capture many situations. Firstly, it applies to ILP, a general and powerful model for representing many combinatorial problems. Secondly, the resiliency part of each problem can be represented as a whole ILP with its own variables and constraints, instead of, say, a simple additive term. Hence, we believe that our method can be applied to many other problems, as well as many different and intricate definitions of resiliency.

Application of Our Approach. To illustrate the fact that our techniques might be useful in different situations, we apply our framework to two concrete problems.
In previous work [1], we analyzed the parameterized tractability of the Resiliency Checking Problem (RCP), having practical applications in the context of access control [15]. This problem, which can be seen as a resiliency version of a generalization of the Set Cover problem, has five parameters \( n, p, s, d \) and \( t \) (described in more detail in Section 4), among which \( n \) is assumed to be large in practice, relative to the other four parameters [15]. Using well-known tools in parameterized algorithms, we were able to determine the complexity of RCP (FPT, XP, \( W[2] \)-hard, para-NP-hard or para-coNP hard) for all but two combinations of \( p, s, d \) and \( t \) (these two combinations are \( p = p, t \)). However, we could not extend this result to the case of any \( s \), and thus the parameterized complexity of RCP parameterized by \( p \) was left open. We settle this case in this paper by showing that, in general, RCP is FPT parameterized by \( p \). This result gives the complete picture of the parameterized complexity of RCP depending on the considered parameter.

Then, we also define an extension of the Closest String problem, a problem arising in computational biology. Informally, Closest String asks whether there exists a string that is “sufficiently close” to each member of a set of input strings. We modify the problem so that the input strings may be unreliable – due to transcription errors, for example – and show that this resiliency version of Closest String is FPT when parameterized by the number of input strings. This is a generalization of a result due to Gramm et al. for Closest String which was proved using Lenstra’s theorem [8]3.

Organization of the Paper. In Section 2.1 we introduce necessary notation. Section 2.2 is devoted to the proof of an important technical result (Lemma 2) related to Fourier-Motzkin elimination for ILPs. Section 3 introduces ILP resiliency. We prove that this problem is \( \Pi_2^P \)-complete, and provide a fixed-parameter tractable algorithm. We then apply our framework to two concrete problems. We establish the fixed-parameter tractability of RCP parameterized by \( p \) in Section 4. In Section 5, we introduce a resiliency version of Closest String Problem and prove that it is FPT. We conclude the paper in Section 6, where we discuss related literature.

2 Preliminaries

2.1 Notation

Throughout the paper, we write \([d]\) to denote \(\{1, \ldots, d\}\) for any integer \(d \geq 1\). Let \(X\) be a finite set of variables. We say that \(\gamma\) is an atomic formula if it is either a (linear) inequality or congruence relation4 involving variables from \(X\). Thus, an inequality is of the form

\[
\sum_{j=1}^{n} a_j x_j \geq b,
\]

while a congruence relation is of the form

\[
\sum_{j=1}^{n} c_j x_j \equiv d \mod e,
\]

where \(a_j, b, c_j, d \in \mathbb{Z}\) for all \(j \in [n]\), and \(e \in \mathbb{N}\). Then, a formula is a combination of atomic formulas, using binary operators such as disjunction (\(\lor\)), conjunction (\(\land\)) and the unary operator representing negation (\(\neg\)). We define a System of Linear Inequalities and Congruences (SLIC) to be a conjunction of atomic formulas. Given a SLIC \(S\) and using the notation above, the set of \(a_j, c_j, j \in [n]\) from all atomic formulas of \(S\) will be called the coefficient set, the set of \(b, d\) from all atomic formulas of \(S\) will be called the right-hand side set,

\[\footnote{We have also established that certain sub-cases of RCP are FPT using reductions to the Workflow Satisfiability Problem [2].}
\]

\[\footnote{Although not being strictly the first problem proved to be FPT using Lenstra’s theorem (see, [19] for instance), it is considered as the one which popularized this technique [3, 6].}
\]

\[\footnote{All inequalities and congruence relations used in this paper are assumed to be linear.}
\]
and finally the set of \( e \) from all congruence relations of \( S \) will be called the modulus set. An Integer Linear Program (ILP) is a SLIC in which all atomic formulas are inequalities.

Given a set of variables \( X \), an integral assignment of \( X \) is a function \( \sigma : X \rightarrow \mathbb{N} \). Given a formula \( S \) (e.g. a SLIC) defined over the set of variables \( X \), we say that an integral assignment \( \sigma \) of \( X \) satisfies \( S \) if the evaluation of the formula w.r.t. \( \sigma \) is true, in which case we will write \( \sigma \models S \). For the sake of convenience, we will sometimes represent an integral assignment as a set of individual assignments \( \sigma = \{ x_1 \mapsto v_1, \ldots, x_n \mapsto v_n \} \) for \( (v_1, \ldots, v_n) \in \mathbb{N}^n \).

Let \( S \) be a SLIC. Then \(|S|\), \(\text{coef}(S)\) and \(\text{mod}(S)\) denote respectively its number of atomic formulas, the maximum value of its coefficients, and the maximum value of its moduli. Finally, we define \( \kappa(S) = |S| + \text{coef}(S) + \text{mod}(S) \).

### 2.2 Elimination of a Variable Based on the Fourier-Motzkin Elimination

We now describe the key ingredient of our approach, which will be used in order to eliminate a variable from a SLIC. It is based on the Fourier-Motzkin elimination step for (non-integer) linear programs [4]. The approach described below also makes use of work by Williams [20, 21] on the elimination of integer variables. However, Williams did not provide any upper bounds on the size of the system and values of coefficients and moduli, which we require in our FPT analysis. We establish these bounds in Lemma 2. Proof of Lemma 1, however, follows from arguments in [21].

**Lemma 1.** Let \( X \) be a set of variables, and \( x \in X \). Let us consider the following three atomic formulas:

\[
g := (p_1 x \geq s) \tag{4}
\]
\[
\ell := (p_2 x \geq t) \tag{5}
\]
\[
c := (p_3 x \equiv u \mod{k_3}) \tag{6}
\]

where \( s, t \) and \( u \) are linear combinations (with coefficients in \( \mathbb{Z} \)) involving variables from \( X \setminus \{x\} \), and \( p_1, p_2, p_3, k \) are positive integers. For \( h \in \{0, \ldots, k_2 p_2 p_3 - 1\} \), we define:

\[
q^h := (0 \geq p_3(p_2 s + p_1 t) + p_1 h) \tag{7}
\]
\[
c^1_1 := (0 \equiv p_3 t + p_2 u + h \mod{k_2 p_3}) \tag{8}
\]
\[
c^1_2 := (0 \equiv u \mod{k_3}) \tag{9}
\]

Then, for an integral assignment \( \sigma \) of \( X \setminus \{x\} \), we have \( \sigma \models (q^h \land c^1_1 \land c^1_2) \) for some \( h \in \{0, \ldots, k_2 p_2 p_3 - 1\} \) if and only if there exists \( v \in \mathbb{N} \) such that \( \sigma \cup \{x \mapsto v\} \models (g \land \ell \land c) \).

**Proof.** The idea is to combine inequalities (4) and (5) to get an equivalent system:

\[
p_2 p_3 s \leq p_1 p_2 p_3 x \leq -p_1 p_3 t \tag{10}
\]
\[
p_1 p_2 p_3 x \equiv p_1 p_2 u \mod{k_1 p_2 p_3} \tag{11}
\]

By substituting \( p_1 p_2 p_3 x \) by \( y \), we have

\[
p_2 p_3 s \leq y \leq -p_1 p_3 t \tag{12}
\]
\[
y \equiv p_1 p_2 u \mod{k_1 p_2 p_3} \tag{13}
\]
\[
y \equiv 0 \mod{p_1 p_2 p_3} \tag{14}
\]

Then, using the Generalized Chinese Remainder Theorem described in [21, Section 2], we can rewrite congruences (10) and (11) as

\[
y \equiv p_1 p_2 u \mod{k_1 p_2 p_3}
\]
\[
0 \equiv u \mod{p_3}
\]

Taking \( p_1 p_2 u \) from both sides of the inequalities in (12), and setting \( z = y - p_1 p_2 u \), we have

\[
p_2 p_3 s - p_1 p_2 u \leq z \leq -p_1 p_3 t - p_1 p_2 u \tag{15}
\]
\[
z \equiv 0 \mod{k_1 p_2 p_3} \tag{16}
\]
\[
0 \equiv u \mod{p_3}
\]
Then, inequalities in (12) are satisfied by an integral assignment if and only if there exists \( h' \in \{0, \ldots, kp_1p_2p_3 - 1\} \) such that:

\[
\begin{align*}
p_2p_3s - p_1p_2u & \leq -p_1p_3t - p_1p_2u - h' \\
p_1p_3t - p_1p_2u - h' & \equiv 0 \mod kp_1p_2p_3
\end{align*}
\]

(14) 

By setting \( h = \frac{\nu}{p_1} \), atomic formulas (14) and (15) give, for \( h \in \{0, \ldots, kp_2p_3 - 1\} \):

\[
\begin{align*}
q'^h & := (0 \geq p_3(p_2s + p_1t) + p_1h) \\
c'^1 & := (0 \equiv p_3t + p_2u + h \mod kp_2p_3) \\
c'^2 & := (0 \equiv u \mod p_3)
\end{align*}
\]

By construction, given an integral assignment \( \sigma \) of \( X \setminus \{x\} \), one can check that we have \( \sigma \models (q'^h \land c'^1 \land c'^2) \) for some \( h \in \{0, \ldots, kp_2p_3 - 1\} \) if and only if there exists \( v \in \mathbb{N} \) such that \( \sigma \cup \{x \mapsto v\} \models (q \land \ell \land c) \), as desired.

We now use this lemma to eliminate a variable \( x \in X \) from a SLIC. However, as we can see, Lemma 1 implicitly transforms a SLIC into the disjunction of several SLICs. This explains why we will now consider a disjunction of SLICs. Here again, the proof of the following lemma uses arguments of [21, Section 3]. (As we noted earlier, the main difference is that we now establish the running time of the transformation and size of the obtained systems.)

**Lemma 2.** Let \( S \) be a SLIC over a set of variables \( X \), and let \( x \in X \). We can, in FPT time parameterized by \( \kappa(S) \), compute a disjunction of SLICs \( S' = \hat{S}_1 \lor \cdots \lor \hat{S}_r \) such that:

1. \( \sum_{i=1}^r \kappa(S_i) \) is upper bounded by a function of \( \kappa(S) \) only;
2. \( x \) does not appear in any atomic formula of \( S' \) (i.e. the corresponding coefficients are 0);
3. for an integral assignment \( \sigma \) of \( X \setminus \{x\} \), we have \( \sigma \models \hat{S}_i \) for some \( i \in [r] \) if and only if there exists \( v \in \mathbb{N} \) such that \( \sigma \cup \{x \mapsto v\} \models S \).

**Proof.** Let \( L \) be the conjunction of inequalities of \( S \) in which \( x \) has a negative coefficient, \( R \) be the conjunction of inequalities of \( S \) in which \( x \) has a coefficient different from 0, and finally, let \( O \) be the conjunction of all atomic formulas of \( S \) different from those of \( L, G \) and \( C \). We thus have \( S = O \land L \land R \land C \). We may assume, with a slight modification of \( S \), that \( C = \{c\} \), and that in \( c \), the modulus is a multiple of the coefficient of \( x \). The first modification comes from the application of the Generalized Chinese Remainder Theorem, while the second uses the Extended Euclidean Algorithm, which allows us to replace a congruence relation

\[ vx \equiv w \mod m \]

by the following two relations

\[
\begin{align*}
gcd(v, m)x & \equiv \mu w \mod m \\
0 & \equiv w \mod gcd(v, m)
\end{align*}
\]

(16) 

Here, \( \mu \in \mathbb{Z} \) is such that \( \mu v + \eta m = gcd(v, m) \) for some \( \eta \in \mathbb{Z} \) (\( \mu \) and \( \eta \) can be found by the Extended Euclidean Algorithm). Details of such transformations are described in [21]. Observe that the coefficients and moduli of the transformed SLIC are upper bounded by a function of \( \kappa(S) \) only, and the number of relations eventually gets increased by one. Hence we may assume that the SLIC in the statement of the lemma is as described previously. Let \( L = \ell_1 \land \cdots \land \ell_{|L|} \) and \( G = g_1 \land \cdots \land g_{|G|} \). For all \( i \in \{1, \ldots, |G|\} \) and all \( j \in \{1, \ldots, |L|\} \), we thus have

\[
\begin{align*}
g_i & := (p'_i x \geq s^i) \\
\ell_j & := (-p'_2 x \geq t^j) \\
c & := (p_3x \equiv u \mod kp_3)
\end{align*}
\]

(17) 

(18) 

(19) 

(20) 

where \( s^i, t^j, u \) are linear combinations involving variables from \( X \setminus \{x\} \) and right hand sides, and \( p'_1, p'_2, p_3 \), and \( k \) are positive integers depending on the coefficients of \( S \) only. We now distinguish two cases.
Let $\{h_{ij} \mid i, j \in \{1, \ldots, |G|\}, j \in \{1, \ldots, |L|\}\}$. Using Lemma 1, we construct, for all $h_{ij} \in \{0, \ldots, kp_2^3 p_3 - 1\}$, atomic formulas $q^{h_{ij}}$, $c_1^{h_{ij}}$ and $c_2^{h_{ij}}$ as follows:

$$q^{h_{ij}} := (0 \geq p_3(p_2^2 s + p_1^2 t + p_1^1 h_{ij})$$

$$c_1^{h_{ij}} := (0 = p_3 t + p_2^2 u + h_{ij} \mod kp_2^3 p_3)$$

$$c_2^{h_{ij}} := (0 = u \mod p_3)$$

Then, we define

$$\Omega := O \land \bigwedge_{i \in \{1, \ldots, |G|\}} \left( \bigvee_{j \in \{1, \ldots, |L|\}} (q^{h_{ij}} \land c_1^{h_{ij}} \land c_2^{h_{ij}}) \right)$$

Then, by transforming the term in brackets into disjunctive normal form, we get the equivalent formulation $\Omega'$ of $\Omega$ with

$$\Omega' := O \land \left( \bigvee_{\lambda \in A} \left( \bigwedge_{\gamma \in \Gamma} \omega_{\lambda}^\gamma \right) \right)$$

where $|A|$ and $|\Gamma|$ are functions of $|G| + |L| + \text{coef}(S) + \text{mod}(S)$ only (i.e., a function of $\kappa(S)$ only) and $\omega_{\lambda}^\gamma$ is an atomic formula, for all $\lambda \in A$ and all $\gamma \in \Gamma$. Finally, let us define, for all $\lambda \in A$

$$\hat{S}_\lambda := O \land \left( \bigwedge_{\gamma \in \Gamma} \omega_{\lambda}^\gamma \right)$$

which is, by definition, a SLIC. Also observe that variable $x$ does not appear in $\hat{S}_\lambda$, for all $\lambda \in A$. It only remains to prove condition 3.

Suppose first that $S$ is satisfied by an integral assignment $\sigma$ of $X$. Observe that by construction, it is sufficient to prove that $\Omega$ is satisfied by $\sigma'$ the restriction of $\sigma$ to $X \setminus \{x\}$. Clearly, every atomic formula $o \in O$ is satisfied, since it also appears in $S$. Now, for all $i \in \{1, \ldots, |G|\}$ and all $j \in \{1, \ldots, |L|\}$, we have that $g_i \land \ell_j \land c$ is satisfied. By Lemma 1, it means that there exists $h_{ij} \in \{0, \ldots, kp_2^3 p_3 - 1\}$ such that $q^{h_{ij}} \land c_1^{h_{ij}} \land c_2^{h_{ij}}$ is satisfied as well by $\sigma'$.

Suppose now that $\sigma'$ is an integral assignment of $X \setminus \{x\}$ such that $\hat{S}_\lambda$ is satisfied for some $\lambda \in A$, which implies that $\Omega$ is satisfied. Let $i^* = \arg \max \{\frac{\ell^i}{p_1^i} : i \in \{1, \ldots, |G|\}\}$, and $j^* = \arg \min \{\frac{\ell^j}{p_2^j} : j \in \{1, \ldots, |L|\}\}$. By definition of $\Omega$, there exists $h \in \{0, \ldots, kp_2^3 p_3 - 1\}$ such that $q^{h} \land c_1^{h} \land c_2^{h}$ is satisfied. Hence, by Lemma 1, it means that there exists $\nu \in \mathbb{N}$ such that $\sigma' \cup \{x \rightarrow \nu\}$ satisfies $g_{i^*} \land \ell_{j^*} \land c$. By definition of $i^*$ and $j^*$, observe that this assignment satisfies $g_i$, $\ell_j$ for all $i \in \{1, \ldots, |G|\}$ and all $j \in \{1, \ldots, |L|\}$. Since, as above, $O$ is also satisfied, $\sigma' \cup \{x \rightarrow \nu\}$ satisfies $S$, which concludes this case.

- **Case 2: L (resp. G) is empty.** Here, we simply remove $L$ (resp. $G$), and replace the eventual atomic formula $c \in C$ described as above by $c' := (0 = u \mod p_3)$. Let $S$ be the obtained SLIC and $S' = \hat{S}$. Clearly, conditions 1 and 2 are satisfied. Then, if there exists an integral assignment satisfying $S$, then it also satisfies $\hat{S}$. Finally, let $\sigma$ be an integral assignment of $X \setminus \{x\}$ satisfying $\hat{S}$. We only deal with the case where $L$ is empty, the other case being very similar. Observe that $\sigma \cup \{x \rightarrow \nu\}$ satisfies $S$ with $\nu = \left\lceil \max \{\frac{x}{p_3^3} : i \in \{1, \ldots, |G|\}\} \right\rceil$.

\[\square\]

## 3 Testing Resiliency of SLICs (and ILPs)

Let $S$ be a SLIC with variable set $X$, and $A \subseteq X$. We denote by $F_A$ the conjunction of atomic formulas of $S$ involving only variables from $A$.

**Definition 1.** We say that $S$ is $A$-resilient if, for every integral assignment $\sigma_A$ of $A$ satisfying $F_A$, there exists an integral assignment $\sigma$ of $X \setminus A$ such that $(\sigma_A \cup \sigma) = S$.
We can now define the following problem:

| SLIC Resiliency |
|------------------|
| **Input:** A SLIC $S$ with variables $X$, and $A \subseteq X$. |
| **Question:** Is $S$ $A$-resilient? |

Naturally, the ILP Resiliency problem denotes the special case of SLIC Resiliency where the input SLIC is an ILP. We first prove that this new problem is complete for the second level of the polynomial hierarchy.

**Theorem 1.** ILP Resiliency is $\Pi^P_2$-complete.

**Proof.** We first show that ILP Resiliency is in $\Pi^P_2$. To do so, we show that its complement is in $NP$. Suppose we have an oracle for ILP Feasibility (which is an $NP$-hard problem). We construct a nondeterministic Turing machine which, given an ILP $S$ with variables $X$ and $A \subseteq X$, decides whether $S$ is not $A$-resilient. To that end, the Turing machine nondeterministically computes an integral assignment for variables $A$ satisfying $F_A$ ($F_A$ is the conjunction of atomic formulas of $S$ involving variables from $A$ only), and queries the oracle for ILP Feasibility of the remaining ILP. If the oracle returns “yes”, then our Turing machine returns “no”, and conversely. Clearly this Turing machine solves the complement of ILP Resiliency.

We now turn to the $\Pi^P_2$-hardness, which is proved by a reduction from the following problem.

| ∀∃-3 Dimensional Matching |
|-----------------------------|
| **Input:** Three disjoint sets $A$, $B$, $C$ of the same size $n$, and two disjoint subsets $M_1$ and $M_2$ of $A \times B \times C$. |
| **Question:** For any $S_1 \subseteq M_1$, is there $S_2 \subseteq M_2$ such that $S_1 \cup S_2$ is a matching of size $n$? |

A set $M \subseteq A \times B \times C$ is a matching if, for every $e, e' \in M$, where $e = (x, y, z)$ and $e' = (x', y', z')$, we have $x \neq x'$, $y \neq y'$ and $z \neq z'$ whenever $e \neq e'$ (in that case, we will say that $e$ and $e'$ are disjoint). This problem is $\Pi^P_2$-complete [17]. For an input of ∀∃-3 Dimensional Matching, we set $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_n\}$, $C = \{c_1, \ldots, c_n\}$, $M_1 = \{e_1, \ldots, e_m\}$, $M_2 = \{e_{m+1}, \ldots, e_m\}$. Moreover, for any $j \in [m]$, we set $e_j = (a_{i_j}, b_{j}, c_{j})$, for some $(i_j, j, j) \in [n]^3$.

For all $j \in [m]$, we introduce a variable $\epsilon_j$ which, informally, will represent the fact of taking hyperedge $e_j$ into $S_1$ or $S_2$ (with value 1). Then, for all $j \in [m]$ and $i \in [n]$, we introduce three variables $\alpha^i_j$, $\beta^i_j$ and $\gamma^i_j$. Informally, variable $\alpha^i_j$ (resp. $\beta^i_j$, $\gamma^i_j$) will be set to 1 if and only if $e_j$ is taken in the solution (in $S_1$ or $S_2$) and $a_i$ (resp. $b_i$, $c_i$) belongs to $e_j$. We define the following ILP $L$.

\[
\sum_{j=1}^{m} \epsilon_j \geq n \quad (21)
\]

\[
3\epsilon_j \leq \alpha^i_j + \beta^i_j + \gamma^i_j \quad \text{for all } j \in [m] \quad (22)
\]

\[
\sum_{j=1}^{m} \alpha^i_j = 1 \quad \text{for all } i \in [n] \quad (23)
\]

\[
\sum_{j=1}^{m} \beta^i_j = 1 \quad \text{for all } i \in [n] \quad (24)
\]

\[
\sum_{j=1}^{m} \gamma^i_j = 1 \quad \text{for all } i \in [n] \quad (25)
\]

\[
0 \leq \epsilon_j, \alpha^i_j, \beta^i_j, \gamma^i_j \leq 1 \quad \text{for all } i \in [n], j \in [m] \quad (26)
\]

Finally, we define $Z = \{e_1, \ldots, e_m\}$, and thus, $F_Z$, the conjunction of inequalities involving variables from $Z$ only, only contains inequalities of type (26). Since $L$ can be constructed in polynomial time, it only remains to prove that $L$ is $Z$-resilient if and only if the instance of ∀∃-3 Dimensional Matching is positive.
Suppose $L$ is $Z$-resilient, and let $S_1 \subseteq M_1$. We define an integral assignment $\sigma_Z$ of variables $Z$ as follows: for $j \in [m_1]$, $\sigma_Z(e_j) = 1$ if $e_j \in S_1$, and $\sigma_Z(e_j) = 0$ otherwise. Obviously this assignment satisfies $F_Z$. Hence, there exists an integral assignment $\sigma$ of all remaining variables such that $\sigma_Z \cup \sigma$ satisfies $L$. We define $S_2 = \{e_j : j \in [m_1 + 1, \ldots, m]\}$ such that $\sigma(e_j) = 1$. Thanks to constraint (21), $S_1 \cup S_2$ is a set of at least $n$ hyperedges. Now, suppose by contradiction that there exists $j, j' \in [m], j \neq j'$, $e_j, e_{j'} \in S_1 \cup S_2$ and such that $e_j$ and $e_{j'}$ are not disjoint. W.l.o.g., assume that $e_j$ and $e_{j'}$ agree on the first component which is, say, $a_1$ for both. By constraint (22), we have that $\alpha_j^1 = \alpha_{j'}^1 = 1$. However, this contradicts constraint (23). Hence, $S_1 \cup S_2$ is a matching of size at least $n$.

Suppose now that for any $S_1 \subseteq M_1$, there is $S_2 \subseteq M_2$ such that $S_1 \cup S_2$ is a matching of size $n$, and consider an integral assignment $\sigma_Z$ of variables $Z$ satisfying $F_Z$ (i.e. $\sigma_Z(e_j) \in \{0, 1\}$ for every $j \in [m_1]$). We can thus define $S_1 = \{e_j : j \in [m_1] \mid \sigma_Z(e_j) = 1\}$ which is by definition a subset of $M_1$. Hence, there exists $S_2 \subseteq M_2$ such that $S_1 \cup S_2$ is a matching of size $n$. We now define an integral assignment $\sigma$ for the remaining variables as follows: for any $j \in \{m_1 + 1, \ldots, m\}$, we define:

- If $e_j \in S_2$, then $\sigma(e_j) = 1$, and $\sigma(\alpha_j^1) = \sigma(\alpha_j^2) = \sigma(\alpha_j^3) = 1$,
- If $e_j \notin S_2$, then $\sigma(e_j) = 0$, and $\sigma(\alpha_j^1) = \sigma(\alpha_j^2) = \sigma(\alpha_j^3) = 0$.

Since $S_1 \cup S_2$ is a set of size $n$, constraint (21) is satisfied. By construction, $e_j = 1$ if and only if $\sigma(\alpha_j^1) = \sigma(\alpha_j^2) = \sigma(\alpha_j^3) = 1$, hence constraint (22) is satisfied. Finally, one can observe that if one of the constraints among (23), (24) or (25) was not satisfied, then it would imply that two hyperedges of $S_1 \cup S_2$ agree on at least one component, which would not be disjoint, a contradiction.

We now prove the main result of our framework, which will be applied in the next sections to two concrete problems.

**Theorem 2.** Let $S$ be a SLIC with variables $X$, and $A \subseteq X$. Deciding whether $S$ is $A$-resilient is FPT parameterized by $\kappa(S) + |X|$.

**Proof.** The first step of our algorithm is to eliminate all variables of $X \setminus A$ from $S$, using Lemma 2. We can thus obtain, in FPT time parameterized by $\kappa(S) + |X|$, a disjunction of SLICs $S' = \tilde{S}_1 \lor \cdots \lor \tilde{S}_r$ such that $\kappa(S')$ is an upper bounded by a function of $\kappa(S)$ only. The crucial observation is that now, by Lemma 2, $S$ is $A$-resilient if and only if for every integral assignment $\sigma_A$ of $A$ satisfying $F_A$, we have $\sigma_A \models \tilde{S}_i$ for some $i \in [r]$ (note that variables of $X \setminus A$ are no longer in $S'$, and thus Definition 1 can be simplified). We now aim at deciding whether $S$ is not $A$-resilient. To that aim, it is sufficient to find a valid assignment $\sigma_A$ of $A$ which satisfies $F_A$ and $\Phi$, where

$$
\Phi := \bigwedge_{i=1}^{r} \neg \tilde{S}_i
$$

Observe that by Lemma 2, $\sum_{i=1}^{r} \kappa(\tilde{S}_i)$ is upper bounded by a function of $\kappa(S) + |X|$ only. Now, for $i \in [r]$, since $\tilde{S}_i$ is a SLIC, its negation is of the form

$$
\neg \tilde{S}_i := \bigvee_{\omega \in \Omega_i} \neg \omega
$$

where $\omega \in \Omega_i$ is an atomic formula (i.e. either an inequality or a congruence relation). Observe that if $\omega$ is an inequality, that is

$$
\omega := \sum_{a \in A} a_\alpha z_\alpha \geq b
$$

(recall that $\Phi$ only involves variables from $A$), then $\neg \omega$ is equivalent to the inequality

$$
\bar{\omega} := \sum_{a \in A} a_\alpha z_\alpha \leq b - 1
$$

If, however, $\omega$ is a congruence relation, that is

$$
\omega := \sum_{a \in A} a_\alpha z_\alpha \equiv b \mod c
$$
then \( \neg \omega \) is equivalent to
\[
\bigvee_{b \in [c] \setminus \{t\}} \left( \sum_{\alpha \in A} a_{\alpha} z_{\alpha} \equiv b \mod c \right)
\]
(27)
Hence, \( \Phi \) is actually equivalent to
\[
\Phi' := \bigwedge_{i=1}^{r} \left( \bigvee_{\omega \in \mathcal{P}_i} \omega \right)
\]
Note that for all \( i \in [r] \), \( |\mathcal{P}_i| \) is upper bounded by a function of \( |\mathcal{S}| + |X| \) only (note that in order to bound the number of atomic formulas in (27), it is crucial that the modulus of the SLIC obtained by Lemma 2 is upper bounded by a function of our parameter only, which is ensured by condition 1 of the lemma). We now rewrite \( \Phi' \) into disjunctive normal form, obtaining
\[
\Phi'' := \bigvee_{\lambda \in A} \left( \bigwedge_{\gamma \in \Gamma_\lambda} \omega^\gamma_\lambda \right)
\]
with \( |A| + |\Gamma_\lambda| \) upper bounded by a function of \( \kappa(S) + |X| \) only. Recall that we aim at finding an integral assignment \( \sigma_A \) of \( A \) satisfying \( F_A \) and \( \Phi'' \). For every \( \lambda \in A \), we set
\[
\Phi_\lambda := F_A \land \left( \bigwedge_{\gamma \in \Gamma_\lambda} \omega^\gamma_\lambda \right)
\]
Then it is sufficient to test the existence of an integral assignment \( \sigma_A \) of \( A \) satisfying \( \Phi_\lambda \) for some \( \lambda \in A \). Finally, observe that for all \( \lambda \in A \), \( \Phi_\lambda \) is a SLIC with \( \kappa(\Phi_\lambda) \) upper bounded by a function of \( \kappa(S) + |X| \).
We are now able to solve these SLICs independently, by eliminating all variables from \( A \) using Lemma 2 once again. We end up, for every \( \lambda \in A \), with a disjunction of SLICs involving only constant terms, whose correctness can be tested in FPT time parameterized by \( \kappa(S) + |X| \). In case of a positive answer for some \( \lambda \in A \), it means, by the previous arguments, that \( S \) is not \( A \)-resilient and vice versa, which concludes the proof. \( \square \)

4 Resiliency Checking Problem
Access control is an important topic in computer security and is typically achieved by enforcing a policy that specifies which users are authorized to access which resources. Authorization policies are frequently augmented by additional policies, articulating concerns such as separation of duty and resiliency. The Resiliency Checking Problem (RCP) was introduced by Li et al. [15] and asks whether it is always possible to allocate authorized users to teams, even if some users are unavailable.

4.1 Definition of the Problem
Given a set of users \( U \) and set of resources \( R \), an authorization policy is a relation \( UR \subseteq U \times R \); we say \( u \) is authorized for resource \( r \) if \( (u, r) \in UR \). For a user \( u \in U \), we define \( N_{UR}(u) = \{ r \in R : (u, r) \in UR \} \), the neighborhood of \( u \); by extension, for \( V \subseteq U \), we define \( N_{UR}(V) = \bigcup_{u \in V} N_{UR}(u) \), the neighborhood of \( V \). Thus \( N_{UR}(u) \) represents the resources for which \( u \) is authorized, and \( N_{UR}(V) \) represents the resources for which the users in \( V \) are collectively authorized. We will omit the subscript \( UR \) if the authorization policy is clear from the context.

Given an authorization policy \( UR \subseteq U \times R \), an instance of the Resiliency Checking Problem (RCP) is defined by a resiliency policy \( \text{res}(P, s, d, t) \), where \( P \subseteq R \), \( s \geq 0 \), \( d \geq 1 \) and \( t \geq 1 \). We say that \( UR \) satisfies \( \text{res}(P, s, d, t) \) if and only if for every subset \( S \subseteq U \) of at most \( s \) users, there exist \( d \) pairwise disjoint subsets of users \( V_1, \ldots, V_d \) such that for all \( i \in \{1, \ldots, d\} \):
\[
V_i \cap S = \emptyset,
\]
(28)
\[
|V_i| \leq t,
\]
(29)
\[
N(V_i) \supseteq P.
\]
(30)
In other words, \( UR \) satisfies \( \text{res}(P, s, d, t) \) if we can find \( d \) disjoint groups of users, even if up to \( s \) users are unavailable, such that each group contains no more than \( t \) users and the users in each group are collectively authorized for the resources in \( P \) (observe that the particular case in which \( s = 0 \) and \( d = 1 \) is equivalent to the well-known Set Cover problem). Thus, we define RCP as follows:

**Resiliency Checking Problem (RCP)**

**Input:** \( UR \subseteq U \times R, P \subseteq R, s \geq 0, d \geq 1, t \geq 1. \)

**Question:** Does \( UR \) satisfy \( \text{res}(P, s, d, t) \)?

In the remainder of this section, we set \( p = |P| \). Given an instance of RCP, we say that a set of \( d \) pairwise disjoint subsets of users \( V = \{V_1, \ldots, V_d\} \) satisfying conditions (29) and (30) is a set of teams. For such a set of teams, we define \( \mathcal{U}(V) = \bigcup_{i=1}^{d} V_i \). Given \( U' \subseteq U \), the restriction of \( UR \) to \( U' \) is defined by \( UR|_{U'} = UR \cap (U' \times R) \). Finally, a set of users \( S \subseteq U \) is called a blocker set if for every set of teams \( V = \{V_1, \ldots, V_d\} \), we have \( \mathcal{U}(V) \cap S \neq \emptyset \). Equivalently, observe that \( S \) is a blocker set if and only if \( UR|_{U \setminus S} \) does not satisfy \( \text{res}(P, 0, d, t) \).

### 4.2 Fixed-Parameter Tractability of RCP

In this section we prove that RCP is FPT parameterized by \( p \). To do so, let us introduce some notation. In the following, \( UR \subseteq U \times R, P \subseteq R, s \geq 0, d \geq 1 \) and \( t \geq 1 \) will denote an input of RCP. Without loss of generality, we may assume \( P = R \) and \( N(u) \neq \emptyset \) for all \( u \in U \). For all \( N \subseteq P \), let \( U_N = \{u \in U : N(u) = N\} \) (notice that we may have \( U_N = \emptyset \) for some \( N \subseteq P \)).

Roughly speaking, the idea is that in order to construct a set of teams or a blocker set, it is sufficient to know the size of its intersection with \( U_N \), for every \( N \subseteq P \). We first define the set of configurations.

\[
C = \left\{ \{N_1, \ldots, N_b\} : b \leq t, N_i \subseteq P, i \in [b], \bigcup_{i=1}^{b} N_i = P \right\}.
\]

Then, for any \( N \subseteq P \), we denote the set of configurations involving \( N \) by \( \mathcal{C}_N \). That is

\[
\mathcal{C}_N = \{ c = \{N_1, \ldots, N_b\} : c \in C : N = N_i \text{ for some } i \in [b]\}
\]

Observe that since we assume \( t \leq p \), we have \(|C| = O(2^p)\). The link between sets of teams and configurations comes from the following definition: given a set of teams \( V \), we say that a team \( T \in V \) has configuration \( c \in C \) if \( c = \{N(u), u \in T\} \). In other words, \( c \) represents the distinct neighborhoods of users of \( T \) in \( P \).

We define an ILP \( \mathcal{L} \) over the set of variables \( X \cup Z \), where \( X = \{x_c : c \in C\} \) and \( Z = \{z_N : N \subseteq P\} \), with the following inequalities:

\[
\begin{align*}
\sum_{c \in C} x_c & \geq d & \quad (31) \\
\sum_{N \subseteq P} z_N & \leq s & \quad (32) \\
\sum_{c \in \mathcal{C}_N} x_c & \leq |U_N| - z_N & \text{ for every } N \subseteq P & \quad (33) \\
0 & \leq z_N \leq |U_N| & \text{ for every } N \subseteq P & \quad (34) \\
0 & \leq x_c \leq d & \text{ for every } c \in C & \quad (35)
\end{align*}
\]

Observe that \( \kappa(\mathcal{L}) + |X| + |Z| \) is upper bounded by a function of \( p \) only. The idea behind this model is to represent a set \( S \) of at most \( s \) users by variables \( Z \) (by deciding how many users to take for each set of users \( U_N, N \subseteq P \), and to represent a set of teams by variables \( X \) (by deciding how many teams will have configuration \( c \in C \)). Then, inequalities (33) will ensure that the set of teams does not intersect with the chosen set \( S \). However, while we would be able to solve \( \mathcal{L} \) in FPT time parameterized by \( p \) by using, e.g., Lenstra’s ILP Theorem, the reader might realize that doing so would not solve RCP directly. Nevertheless, the following result establishes the crucial link between this system and our problem.

**Lemma 3.** \( \text{res}(P, s, d, t) \) is satisfiable if and only if \( \mathcal{L} \) is \( Z \)-resilient.
Proof. Let us denote by $L_Z$ the ILP consisting only of inequalities involving variables $Z$, i.e. inequalities (32) and (34). Suppose first that $\text{res}(P, s, d, t)$ is satisfiable, and let $\sigma_Z$ be an integral assignment for $Z$ such that $\sigma_Z \models L_Z$.

We now define a set of users $S$ by picking, in an arbitrary manner, $\sigma_Z(z_N)$ users in $U_N$, for each $N \subseteq P$ (since $\sigma_Z(z_N) \leq \min\{s, |U_N|\}$, such a set $S$ must exist). Since $S$ is a set of at most $s$ users, there exists a set of teams $V = \{T_1, \ldots, T_d\}$ such that $U(V) \cap S = \emptyset$. Then, for each $c \in C$, let $\sigma_X(x_c)$ be the number of teams of $V$ having configuration $c$. Clearly we have $\sigma_X(x_c) \in \{0, \ldots, d\}$ and $\sum_{c \in C} \sigma_X(x_c) = d$, and thus inequalities (31) and (35) are satisfied. Then, for all $N \subseteq P$, we may assume w.l.o.g. that $|T_i \cap U_N| \leq 1$ for all $i \in \{1, \ldots, d\}$. Hence $\sum_{c \in C} \sigma_X(x_c)$ equals $|U(V) \cap U_N|$, which is the number of users of $U_N$ involved in some teams of $V$. Since $U(V) \cap S = \emptyset$, we have $\sum_{c \in C} \sigma_X(x_c) \leq |U_N| - \sigma_Z(z_N)$, and thus inequalities (33) are also satisfied for every $N \subseteq P$. Consequently, $\sigma_X \cup \sigma_Z \models L$.

Conversely, let $S \subseteq U$, $|S| \leq s$. For each $N \subseteq P$, define $\sigma_Z(z_N) = |S \cap U_N|$, which is thus an integral assignment of variables $Z$ satisfying $L_Z$. Hence, there exists a valid assignment $\sigma_X$ such that $\sigma_Z \cup \sigma_X \models L$.

Then, let $C \subseteq U$, $|C| \leq t$, consider a set of users $T$ consisting of a user chosen arbitrarily in $U_N$, for each $i \in [b]$. By definition of a configuration, $T$ is a team. Then, since for all $N \subseteq P$, we have, by inequalities (33), that it is possible to construct $\sigma_X(x_c)$ pairwise disjoint such team for each $c \in C$, each having an empty intersection with $S$. In other words, for every $S \subseteq U$, $|S| \leq s$, there exists a set of teams $V$ (and $V$ contains at least $d$ teams, thanks to inequality (31)) such that $U(V) \cap S = \emptyset$, and thus $\text{res}(P, s, d, t)$ is satisfiable.

Since, as we said earlier, $\kappa(L) + |X| + |Z|$ is bounded by a function of $p$ only, combining the previous lemma with Theorem 2, we obtain the following:

**Theorem 3.** RCP is FPT parameterized by $p$.

## 5 Resiliency in the Context of the Closest String Problem

In the Closest String problem, we are given a collection of $k$ strings $s_1, \ldots, s_k$ of length $L$ over a fixed alphabet $\Sigma$, and a non-negative integer $d$. The goal is to decide whether there exists a string $s$ (of length $L$) such that $d_H(s, s_i) \leq d$ for all $i \in [k]$, where $d_H(s, s_i)$ denotes the Hamming distance between $s$ and $s_i$. If such a string exists, then it will be called a $d$-closest string.

It is common to represent an instance of the problem as a matrix $C$ with $k$ rows and $L$ columns (i.e. where each row is a string of the input), hence, in the following, the term column will refer to a column of this matrix. As Gramm et al. [8] observe, as the Hamming distance is measured column-wise, one can identify some columns sharing the same structure. Indeed, let $\Sigma = \{\varphi_1, \ldots, \varphi_{|\Sigma|}\}$. It is shown in [8] that after a simple preprocessing of the instance, we may assume that for every column $c$ of $\Sigma$, $\varphi_i$ is the $i^{th}$ character that appears the most often (in $c$), for $i \in \{1, \ldots, |\Sigma|\}$ (ties broken $w.r.t.$ the considered ordering of $\Sigma$). Such a preprocessed column will be called normalized, and by extension, a matrix consisting of normalized columns will be called normalized. One can observe that after this preprocessing, the number of different columns (called column type) is bounded by a function of $k$ only, namely by the $k^{th}$ Bell number $B_k = O(2^{d \log_2(k)})$.

The set of all column types is denoted by $T$. This observation leads [8] to prove that Closest String is FPT parameterized by $k$, using an ILP with a number of variables depending on $k$ only, and then applying the celebrated Lenstra’s Theorem [14].

### 5.1 Adding Resiliency

There exist several ways in which we might define a notion of resiliency for the Closest String problem. We now describe a general setting which can be solved by our framework. Roughly speaking, we will allow some columns of the instance to change, as well as some columns to be added to the input. The motivation for such a definition of resiliency comes from the introduction of experimental errors, which may change the input strings [18]. While a solution of the Closest String problem tests whether the input strings are consistent, the Resiliency Closest String problem asks whether these strings will remain consistent after some small change. Before defining formally the problem, we need to introduce some notation.
In the following, $C$ denotes a normalized $k \times L$ matrix of elements of $\Sigma$, i.e. an instance of the problem. Let $L_a$ be a non-negative integer, and $A$ be a normalized $k \times L_a$ matrix of elements of $\Sigma$. We denote by $C \oplus A$ the $k \times (L + L_a)$ matrix obtained by appending the columns of $A$ to those of $C$ (in other words, the first $L$ columns of $C \oplus A$ are from $C$, while the $L_a$ last columns are from $A$). Then, suppose that $L_a \leq L$, and let $I = \{l_1, \ldots, l_{L_a}\} \subseteq [L]$. We will denote by $C \otimes_I A$ the matrix obtained by replacing, for each $i \in \{1, \ldots, L_a\}$, the $l_i^{th}$ column of $C$ by the $i^{th}$ column of $A$. We are now introduce our resiliency version of CLOSEST STRING.

**Resiliency Closest String**

- **Input:** A $k \times L$ normalized matrix of elements of $\Sigma$, $d \in \mathbb{N}$, $I \subseteq [L]$, $L_a \in \mathbb{N}$.
- **Question:** For every $I' \subseteq I$, for every $k \times |I'|$ normalized matrix $M$, for every $k \times L_a$ normalized matrix $A$, does $(C \otimes_{I'} M) \oplus A$ admit a $d$-closest string?

### 5.2 Fixed-Parameter Tractability

Given an instance of Resiliency Closest String, we define three sets of variables $X$, $A$ and $M$ as follows:

$$X = \{x_{t,\varphi} : t \in T, \varphi \in \Sigma\},$$

$$A = \{a_t : t \in T\},$$

$$M = \{m^r_t : t \in T\} \cup \{m^a_t : t \in T\}.$$  

Note that the following ILP is an extension of the one used to solve CLOSEST STRING in FPT time parameterized by $k$ [8]. Given $t \in T$ and $\varphi \in \Sigma$, variable $x_{t,\varphi}$ represents the number of columns of type $t$ (in $C$) whose corresponding character in the solution is set to $\varphi$. The idea of our extension is to model the resiliency part by variables $A$ and $M$. More precisely, for a type $t \in T$, $a_t$ will represent the number of columns of type $t$ in $A$, while $m^r_t$ and $m^a_t$ will denote respectively the number of columns of type $t$ in $C$ that will be replaced, and the number of columns of type $t$ in $M$ that will be added instead, from the operation $(C \otimes_{I'} M)$. Given $t \in T$, we denote by $\#_t$ and $\#_t^t$ the number of columns of type $t$ in $C$, and the number of columns of type $t$ in $C$ among those of $I$, respectively. The key observation is that $\#_t - m^r_t + m^a_t + a_t$ is the number of columns of type $t$ in $(C \otimes_{I'} M) \oplus A$. Finally, for $t \in T$ and $i \in [k]$, $\varphi_{t,i}$ denotes the alphabet symbol at the $i^{th}$ entry of column type $t$. Let $\mathcal{L}$ be the following ILP:

$$\sum_{\varphi \in \Sigma} x_{t,\varphi} = \#_t - m^r_t + m^a_t + a_t \text{ for all } t \in T \quad (36)$$

$$\sum_{t \in T} \sum_{\varphi \in \Sigma \setminus \{\varphi_{t,i}\}} x_{t,\varphi} \leq d \text{ for all } i \in [k] \quad (37)$$

$$\sum_{t \in T} m^r_t - m^a_t = 0 \text{ for all } t \in T \quad (38)$$

$$0 \leq m^r_t \leq \#_t^t \text{ for all } t \in T \quad (39)$$

$$\sum_{t \in T} a_t \leq L_a \quad (40)$$

Constraint (36) requires that a solution string can indeed be constructed from an assignment of $X$, and constraint (37) ensures the solution will be a $d$-closest string. Constraints (38) and (39) ensure that we will remove from $C$ columns from $I$ only, while we replace them by the same number of columns (those from $M$). Finally, constraint (40) requires the addition of at most $L_a$ columns (those from matrix $A$).

In the following, we will say that the instance of CLOSEST STRING is *satisfiable* if and only if for every $I' \subseteq I$, for every $k \times |I'|$ normalized matrix $M$, for every $k \times L_a$ normalized matrix $A$, the instance whose matrix is $(C \otimes_{I'} M) \oplus A$ admits a $d$-closest string.

**Lemma 4.** The instance is satisfiable if and only if $\mathcal{L}$ is $(A \cup M)$-resilient.

**Proof.** Observe that the constraints of $\mathcal{L}$ involving only variables from $A \cup M$ are (38), (39) and (40). Let us denote by $\mathcal{L}_{AM}$ the restriction of $\mathcal{L}$ to these constraints.
Suppose first that the instance is satisfiable, and let $\sigma_{AM}$ be an integral assignment for $A \cup M$ such that $\sigma_{AM} \models L_{AM}$. We construct $A$ as a matrix having exactly $\sigma_{AM}(a_t)$ columns of type $t$, for every $t \in T$. Because of constraint (40), $A$ has $L_a$ columns (and $k$ rows). On the other hand, construct $M$ as a matrix having exactly $\sigma_{AM}(m_t^r)$ columns of type $t$, for every $t \in T$. Then, constraint (39) ensures that for every $t \in T$, there exists at least $\sigma_{AM}(m_t^r)$ columns of type $t$ among those of $I$. Hence, it is possible to construct $I' \subseteq I$ as the union, for every $t \in T$, of $\sigma_{AM}(m_t^r)$ column indices of $I$ having type $t$, chosen arbitrarily. Let $C' = (C \otimes I') M \oplus A$. Observe that by constraint (38), matrix $M$ has exactly $|I'|$ columns. Moreover, observe that in $C'$, the number of columns of type $t$ is exactly $\#_t - \sigma_{AM}(m_t^r) + \sigma_{AM}(m_t^r) + \sigma_{AM}(a_t)$, for every $t \in T$. Since the instance is satisfiable, $C'$ admits a $d$ closest string $r$. For $t \in T$ and $\varphi \in \Sigma$, define $\sigma_{X}(x_{t,\varphi})$ as the number of columns of type $t$ in $C'$ whose corresponding character in $r$ is $\varphi$. Since, as said previously, $C'$ has exactly $\#_t - \sigma_{AM}(m_t^r) + \sigma_{AM}(m_t^r) + \sigma_{AM}(a_t)$ columns of type $t$, constraint (36) is satisfied for every $t \in T$. Then, since $r$ is a $d$-closest string, constraint (37) is also satisfied for every $i \in [k]$. We thus have $(\sigma_{AM} \cup \sigma_{X}) \models L$.

Conversely, suppose that $L$ is $(A \cup M)$-resilient, and let us consider $I' \subseteq I$, a normalized $k \times |I'|$ matrix $M$, and a normalized $k \times L_a$ matrix $A$. Let $C' = (C \otimes I') M \oplus A$. We define, for every $t \in T$, $\sigma_{AM}(a_t)$ as the number of columns of $A$ of type $t$, $\sigma_{AM}(m_t^r)$ as the number of columns of $M$ of type $t$, and finally $\sigma_{AM}(m_t^r)$ as the number of columns of $C$ of type $t$ among those of $I'$. Using similar arguments as previously, we can argue that $\sigma_{AM} \models L_{AM}$. Thus, there exists an integral assignment $\sigma_{X}$ of $X$ such that $(\sigma_{AM} \cup \sigma_{X}) \models L$. We now construct $r$ as a string having, for every column type $t \in T$ in $C'$, $\sigma_{X}(x_{t,\varphi})$ occurrence(s) of character $\varphi$, for every $\varphi \in \Sigma$ (columns chosen arbitrarily among those of type $t$ in $C'$). Because of constraint (36), and since $\#_t - \sigma_{AM}(m_t^r) + \sigma_{AM}(m_t^r) + \sigma_{AM}(a_t)$ is the number of columns of type $t$ in $C'$, $r$ is well defined. Finally, observe that constraint (37) ensures that $r$ is a $d$-closest string of $C'$, which concludes the proof. □

Now, observe that $\kappa(L) + |X| + |A| + |M|$ is bounded by a function of $k$ only (since $|T| = O(2^k \log_2(k))$). Hence, combining the previous lemma with Theorem 2, we obtain the following result.

**Theorem 4.** Resiliency Closest String is FPT parameterized by $k$.

### 6 Discussion

When modeling a problem using (integer) linear programming, a notion related to our notion of resiliency is the one of sensitivity analysis [10], or robustness [13], where one aims to find a solution which will remain close to an optimal one up to some modifications of the data (these modifications might occur in the objective function, the coefficients or the right-hand sides). Hence, this area mainly deals with optimality of solutions and the trade-off between the amount of modifications and the loss of quality of a solution. The notion of resiliency we introduced in this paper is different in at least two aspects. Firstly, we deal with feasibility of an ILP rather than optimality and secondly, the goal of our problem is to decide, for any (carefully defined) modification of the instance, whether there exists a feasible solution, instead of finding a unique solution being robust to some modifications.

For some time, Lenstra’s celebrated theorem was the first and only approach in parameterized algorithms and complexity based on integer programming. Recently other tools based on integer programming have been introduced: the use of Graver bases for the $n$-fold integer programming problem [9], ILP approaches in kernelization [11] and an integer quadratic programming analog of Lenstra’s theorem [16]. Our method is a new addition to this powerful arsenal.

Observe that Lemma 2 of our paper eliminates a variable from an ILP and introduces a disjunction of systems. Such an elimination step could well be implemented in a parallel computation setting, using this disjunction in order to solve subproblems independently. Moreover, note that in the case of a Mixed Integer Linear Program (MILP), the elimination of a non-integral variable is even simpler than what is described in Lemma 2, since it consists in applying the classical Fourier-Motzkin elimination. Hence, our result also holds for MILPs.

A natural question is whether our main algorithm can be improved. First, can its running time be reduced? Then, perhaps a more interesting question, can we use a smaller parameter? One could imagine, for instance, replacing the parameter “number of constraints” by “maximum frequency of a variable”, where the frequency of a variable is the number of constraints a given variable is involved in. Since the principle of
eliminating a variable $x$ is to combine every pair of inequalities involving $x$, we would be able to obtain the same result with this new parameterization. However, note that in an ILP, the total number of constraints is always bounded by the number of variables times the maximum frequency, hence this parameterization is actually not smaller than the one we use. Moreover, as mentioned in the introduction, the problem is already (NP-)hard for ILP with 0-1-coefficients. Thus, the only possible improvement seems to remove “maximum value of coefficients of the ILP” from the list of parameters, and consider instead “number of variables and constraints”, or even “number of variables” only. We conjecture that the problem becomes $W[1]$-hard with these smaller parameterizations. Another way to better understand the parameterized complexity of this kind of problems would be to use the parameterized analog of the polynomial hierarchy introduced by de Haan and Szeider [5].

Acknowledgement. We are thankful to Stéphane Pérennes for suggesting the use of LP Fourier-Motzkin elimination for RCP.

References

1. Jason Crampton, Gregory Gutin, and Rémi Watrigant. A multivariate approach for checking resiliency in access control. In *AAIM’16, Springer LNCS 9778, to appear*, 2016.
2. Jason Crampton, Gregory Gutin, and Rémi Watrigant. Resiliency policies in access control revisited. In *Proceedings of the 21st ACM on Symposium on Access Control Models and Technologies*, SACMAT ’16, pages 101–111. ACM, 2016.
3. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
4. George B Dantzig. Fourier-Motzkin elimination and its dual. Technical Report TR 72-18, Stanford University, 1972.
5. Ronald de Haan and Stefan Szeider. The parameterized complexity of reasoning problems beyond NP. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Fourteenth International Conference, KR 2014*, Vienna, Austria, July 20-24, 2014, 2014.
6. Rod G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
7. András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.
8. Jens Gramm, Rolf Niedermeier, and Peter Rossmanith. Fixed-parameter algorithms for CLOSEST STRING and related problems. *Algorithmica*, 37(1):25–42, 2003.
9. Raymond Hemmecke, Shmuel Onn, and Lyubov Romanchuk. n-fold integer programming in cubic time. *Mathematical Programming*, 137(1):325–341, 2013.
10. John N. Hooker. Integer programming duality. In *Encyclopedia of Optimization, Second Edition*, pages 1657–1667. 2009.
11. Bart M. P. Jansen and Stefan Kratsch. A structural approach to kernels for ILPs: Treewidth and total unimodularity. In *23rd European Symposium on Algorithm - ESA 2015*, pages 779–791, 2015.
12. Ravi Kannan. Minkowski’s convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, August 1987.
13. Panos Kouvelis and Gang Yu. *Robust Discrete Optimization and Its Applications*. Kluwer Academic Publishers, 1997.
14. Hendrik W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
15. Ninghui Li, Mahesh V. Tripunitara, and Qihua Wang. Resiliency policies in access control. *ACM Trans. Inf. Syst. Secur.*, 12(4), 2009.
16. Daniel Lokshtanov. Parameterized integer quadratic programming: Variables and coefficients. *CoRR*, abs/1511.00310, 2015.
17. A. McLoughlin. The complexity of computing the covering radius of a code. *IEEE Transactions on Information Theory*, 30(6):800–804, 1984.
18. Pavel Pevzner. *Computational Molecular Biology: An Algorithmic Approach*. MIT Press, 2000.
19. András Sebő. Integer plane multiflows with a fixed number of demands. *J. Comb. Theory, Ser. B*, 59(2):163–171, 1993.
20. H. Paul Williams. Fourier–Motzkin elimination extension to integer programming problems. *J. Comb. Theory, Ser. A*, 21(1):118–123, 1976.
21. H. Paul Williams. An alternative form of the value function of an integer programme. Faculty of mathematical studies working papers, OR16, Faculty of Mathematical Studies, University of Southampton, Southampton, UK., 1988.