# Parameterized Algorithms for the Matrix Completion Problem

**Robert Ganian** [1] **Iyad Kanj** [2] **Sebastian Ordyniak** [3] **Stefan Szeider** [1]

## Abstract

We consider two matrix completion problems, in which we are given a matrix with missing entries and the task is to complete the matrix in a way that (1) minimizes the rank, or (2) minimizes the number of distinct rows. We study the parameterized complexity of the two aforementioned problems with respect to several parameters of interest, including the minimum number of matrix rows, columns, and rows plus columns needed to cover all missing entries. We obtain new algorithmic results showing that, for the bounded domain case, both problems are fixed-parameter tractable with respect to all aforementioned parameters. We complement these results with a lower-bound result for the unbounded domain case that rules out fixed-parameter tractability w.r.t. some of the parameters under consideration.

## 1. Introduction

**Problem Definition and Motivation**. We consider the matrix completion problem, in which we are given a matrix $\mathbf{M}$ (over some field that we also refer to as the *domain* of the matrix) with missing entries, and the goal is to complete the entries of $\mathbf{M}$ so that to optimize a certain measure. There is a wealth of research on this fundamental problem (Candès & Plan, 2010; Candès & Recht, 2009; Candès & Tao, 2010; Elhamifar & Vidal, 2013; Hardt et al., 2014; Fazel, 2002; Keshavan et al., 2010a;b; Recht, 2011; Saunderson et al., 2016) due to its ubiquitous applications in recommender systems, machine learning, sensing, computer vision, data science, and predictive analytics, among others. In these areas, the matrix completion problem naturally arises after observing a sample from the set of entries of a low-rank matrix, and attempting to recover the missing entries with the goal of optimizing a certain measure. In this paper, we focus our study on matrix completion with respect to two measures (considered separately): (1) minimizing the rank of the completed matrix, and (2) minimizing the number of distinct rows of the completed matrix.

The first problem we consider—matrix completion w.r.t. rank minimization—has been extensively studied, and is often referred to as the low-rank matrix completion problem (Candès & Plan, 2010; Candès & Recht, 2009; Candès & Tao, 2010; Hardt et al., 2014; Fazel, 2002; Keshavan et al., 2010a;b; Recht, 2011; Saunderson et al., 2016). A celebrated application of this problem lies in the recommender systems area, where it is known as the Netflix problem (net). In this user-profiling application, an entry of the input matrix represents the rating of a movie by a user, where some entries could be missing. The goal is to predict the missing entries so that the rank of the complete matrix is minimized.

The low-rank matrix completion problem is known to be **NP**-hard, even when the matrix is over the field $\mathrm{GF}(2)$ (*i.e.*, each entry is 0 or 1), and the goal is to complete the matrix into one of rank 3 (Peeters, 1996). A significant body of work on the low-rank matrix completion problem has centered around proving that, under some feasibility assumptions, the matrix completion problem can be solved efficiently with high probability (Candès & Recht, 2009; Recht, 2011). These feasibility assumptions are: (1) low rank; (2) incoherence; and (3) randomness (Hardt et al., 2014). Hardt *et al.* (2014) argue that feasibility assumption (3), which states that the subset of determined entries in the matrix is selected uniformly at random and has a large (sampling) density, is very demanding. In particular, they justify that in many applications, such as the Netflix problem, it is not possible to arbitrarily choose which matrix entries are determined and which are not, as those may be dictated by outside factors. The low-rank matrix completion problem also has other applications in the area of wireless sensor networks. In one such application, the goal is to reconstruct a low-dimensional geometry describing the locations of the sensors based on local distances sensed by each sensor; this problem is referred to as TRIANGULATION FROM INCOMPLETE DATA (Candès & Recht, 2009). Due to its inherent hardness, the low-rank matrix completion problem has also

*Equal contribution [1]Algorithms and Complexity Group, TU Wien, Vienna, Austria [2]School of Computing, DePaul University, Chicago, USA [3]Algorithms Group, University of Sheffield, Sheffield, UK. Correspondence to: Robert Ganian <rganian@gmail.com>, Iyad Kanj <ikanj@cdm.depaul.edu>, Sebastian Ordyniak <sordyniak@gmail.com>, Stefan Szeider <stefan@szeider.net>.

*Proceedings of the $35^{th}$ International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

been studied with respect to various notions of approximation (Candès & Recht, 2009; Candès & Tao, 2010; Frieze et al., 2004; Hardt et al., 2014; Keshavan et al., 2010a;b; Recht, 2011).

The second problem we consider is the matrix completion problem w.r.t. minimizing the number of distinct rows. Although this problem has not received as much attention as low-rank-matrix completion, it certainly warrants studying. In fact, minimizing the number of distinct rows represents a special case of the SPARSE SUBSPACE CLUSTERING problem (Elhamifar & Vidal, 2013), where the goal is to complete a matrix in such a way that its rows can be partitioned into the minimum number of subspaces. The problem we consider corresponds to the special case of SPARSE SUBSPACE CLUSTERING where the matrix is over GF(2) and the desired rank of each subspace is 1. Furthermore, one can see the relevance of this problem to the area of recommender systems; in this context, one seeks to complete the matrix in such a way that the profile of each user is identical to a member of a known (possibly small) group of users.

In this paper, we study the two aforementioned problems through the lens of *parameterized complexity* (Downey & Fellows, 2013). In this paradigm, one measures the complexity of problems not only in terms of their input size $n$ but also by a certain *parameter* $k \in \mathbb{N}$, and seeks—among other things—fixed-parameter algorithms, i.e., algorithms that run in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some function $f$. Problems admitting such algorithms are said to be *fixed-parameter tractable* (or contained in the parameterized complexity class **FPT**). The motivation is that the parameter of choice—usually describing some structural properties of the instance—can be small in some instances of interest, even when the input size is large. Therefore, by confining the combinatorial explosion to this parameter, one can obtain efficient algorithms for problem instances with a small parameter value for **NP**-hard problems. Problems that are not (or unlikely to be) fixed-parameter tractable can still be solvable in polynomial-time for every fixed parameter value, i.e., they can be solved in time $n^{f(k)}$ for some function $f$. Problems of this kind are contained in the parameterized complexity class **XP**. We also consider randomized versions of **FPT** and **XP**, denoted by $\mathbf{FPT}_R$ and $\mathbf{XP}_R$, containing all problems that can be solved by a randomized algorithm with a run-time of $f(k)n^{\mathcal{O}(1)}$ and $\mathcal{O}(n^{f(k)})$, respectively, with a constant one-sided error-probability. Finally, problems that remain **NP**-hard for some fixed value of the parameter are hard for the parameterized complexity class **paraNP**. We refer to the respective textbooks for a detailed introduction to parameterized complexity (Downey & Fellows, 2013; Cygan et al., 2015). Parameterized Complexity is a rapidly growing field with various applications in many areas of Computer Science, including Artificial Intelligence (Bäckström et al., 2015; van Bevern et al., 2016; Bessiere et al., 2008; Endriss et al., 2015; Ganian & Ordyniak, 2018; Gaspers & Szeider, 2014; Gottlob & Szeider, 2006).

**Parameterizations**. The parameters that we consider in this paper are: The number of (matrix) rows that cover all missing entries (row); the number of columns that cover all missing entries (col); and the minimum number of rows and columns which together cover all missing entries (comb). Although we do discuss and provide results for the unbounded domain case, i.e, the case that the domain (field size) is part of the input, we focus on the case when the matrix is over a bounded domain: This case is the most relevant from a practical perspective, and most of the related works focus on this case. It is easy to see that, when stated over any bounded domain, both problems under consideration are in **FPT** when parameterized by the number of missing entries, since an algorithm can brute-force through all possible solutions. On the other hand, parameterizing by row (resp. col) is very interesting from a practical perspective, as rows (resp. columns) with missing entries represent the newly-added elements (*e.g.*, newly-added users/movies/sensors, etc.); here, the above brute-force approach naturally fails, since the number of missing entries is no longer bounded by the parameter alone. Finally, the parameterization by comb is interesting because this parameter subsumes (*i.e.*, is smaller than) the other two parameters (*i.e.*, row and col). In particular, any fixed-parameter algorithm w.r.t. this parameter implies a fixed-parameter algorithm w.r.t. the other two parameters, but can also be efficient in cases where the number of rows or columns with missing entries is large.

**Results and Techniques**. We start in Section 3 by considering the BOUNDED RANK MATRIX COMPLETION problem over GF($p$) (denoted $p$-RMC), in which the goal is to complete the missing entries in the input matrix so that the rank of the completed matrix is at most $t$, where $t \in \mathbb{N}$ is given as input. We present a (randomized) fixed-parameter algorithm for this problem parameterized by comb. This result is obtained by applying a branch-and-bound algorithm combined with algebra techniques, allowing us to reduce the problem to a system of quadratic equations in which only few (bounded by some function of the parameter) equations contain non-linear terms. We then use a result by Miura *et al.* (2014) (improving an earlier result by Courtois *et al.* (2002)) in combination with reduction techniques to show that solving such a system of equations is in $\mathbf{FPT}_R$ parameterized by the number of equations containing non-linear terms. In the case where the domain is unbounded, we show that RMC is in **XP** parameterized by either row or col and in $\mathbf{XP}_R$ parameterized by comb.

In Section 4, we turn our attention to the BOUNDED DISTINCT ROW MATRIX COMPLETION problem over both bounded domain ($p$-DRMC) and unbounded domain (DRMC); here, the goal is to complete the input matrix so

that the number of distinct rows in the completed matrix is at most $t$. We start by showing that $p$-DRMC parameterized by comb is fixed-parameter tractable. We obtain this result as a special case of a more general result showing that both DRMC and $p$-DRMC are fixed-parameter tractable parameterized by the *treewidth* (Robertson & Seymour, 1986; Downey & Fellows, 2013) of the compatibility graph, i.e., the graph having one vertex for every row and an edge between two vertices if the associated rows can be made identical. This result also allows us to show that DRMC is fixed-parameter tractable parameterized by row. Surprisingly, DRMC behaves very differently when parameterized by col, as we show that, for this parameterization, the problem becomes **paraNP**-hard.

| | row | col | comb |
|---|---|---|---|
| $p$-RMC | **FPT**(Th. 2) | **FPT**(Cor. 3) | $\mathbf{FPT}_R$(Th. 6) |
| $p$-DRMC | **FPT**(Th. 11) | **FPT**(Th. 11) | **FPT**(Th. 11) |
| RMC | **XP**(Cor. 4) | **XP**(Cor. 4) | $\mathbf{XP}_R$(Cor. 7) |
| DRMC | **FPT**(Th. 12) | **paraNP**(Th. 13) | **paraNP**(Th. 13) |

*Table 1.* The parameterized complexity results obtained for the problems $p$-RMC and $p$-DRMC and their unbounded domain variants RMC and DRMC w.r.t. the parameters row, col, comb.

We chart our results in Table 1. Interestingly, in the unbounded domain case, both considered problems exhibit wildly different behaviors: While RMC admits **XP** algorithms regardless of whether we parameterize by row or col, using these two parameterizations for DRMC results in the problem being **FPT** and **paraNP**-hard, respectively. On the other hand, in the (more studied) bounded domain case, we show that both problems are in **FPT** (resp. $\mathbf{FPT}_R$) w.r.t. all parameters under consideration. Finally, we prove that 2-DRMC remains **NP**-hard even if every column and row contains (1) a bounded number of missing entries, or (2) a bounded number of determined entries. This effectively rules out **FPT** algorithms w.r.t. the parameters: maximum number of missing/determined entries per row or column.

## 2. Preliminaries

For a prime number $p$, let GF($p$) be a field of order $p$; recall that each such field can be equivalently represented as the set of integers modulo $p$. For positive integers $i$ and $j > i$, we write $[i]$ for the set $\{1, 2, \ldots, i\}$, and $i : j$ for the set $\{i, i+1, \ldots, j\}$.

For an $m \times n$ matrix $\mathbf{M}$ (*i.e.*, a matrix with $m$ rows and $n$ columns), and for $i \in [m]$ and $j \in [n]$, $\mathbf{M}[i, j]$ denotes the element in the $i$-th row and $j$-th column of $\mathbf{M}$. Similarly, for a vector $d$, we write $d[i]$ for the $i$-th coordinate of $d$. We write $\mathbf{M}[*, j]$ for the *column-vector* $(\mathbf{M}[1, j], \mathbf{M}[2, j], \ldots, \mathbf{M}[m, j])$, and $\mathbf{M}[i, *]$ for the *row-vector* $(\mathbf{M}[i, 1], \mathbf{M}[i, 2], \ldots, \mathbf{M}[i, n])$. We will also need to refer to submatrices obtained by omitting certain rows or columns from $\mathbf{M}$. We do so by using sets of indices to specify which rows and columns the matrix contains. For instance, the matrix $\mathbf{M}[[i], *]$ is the matrix consisting of the first $i$ rows and all columns of $\mathbf{M}$, and $\mathbf{M}[2 : m, 1 : n-1]$ is the matrix obtained by omitting the first row and the last column from $\mathbf{M}$.

The *row-rank* (resp. *column-rank*) of a matrix $\mathbf{M}$ is the maximum number of linearly-independent rows (resp. *columns*) in $\mathbf{M}$. It is well known that the row-rank of a matrix is equal to its column-rank, and this number is referred to as the *rank* of the matrix. We let $\mathsf{rk}(\mathbf{M})$ and $\mathsf{dr}(\mathbf{M})$ denote the rank and the number of distinct rows of a matrix $\mathbf{M}$, respectively. If $\mathbf{M}$ is a matrix over GF($p$), we call GF($p$) the *domain* of $\mathbf{M}$.

An *incomplete matrix* over GF($p$) is a matrix which may contain not only elements from GF($p$) but also the special symbol $\bullet$. An entry is a *missing* entry if it contains $\bullet$, and is a *determined* entry otherwise. A (possibly incomplete) $m \times n$ matrix $\mathbf{M}'$ is *consistent* with an $m \times n$ matrix $\mathbf{M}$ if and only if, for each $i \in [m]$ and $j \in [n]$, either $\mathbf{M}'[i, j] = \mathbf{M}[i, j]$ or $\mathbf{M}'[i, j] = \bullet$.

### 2.1. Problem Formulation

We formally define the problems under consideration below.

BOUNDED RANK MATRIX COMPLETION ($p$-RMC)

Input: An incomplete matrix $\mathbf{M}$ over GF($p$) for a fixed prime number $p$, and an integer $t$.
Task: Find a matrix $\mathbf{M}'$ consistent with $\mathbf{M}$ such that $\mathsf{rk}(\mathbf{M}') \leq t$.

BOUNDED DISTINCT ROW MATRIX COMPLETION ($p$-DRMC)

Input: An incomplete matrix $\mathbf{M}$ over GF($p$) for a fixed prime number $p$, and an integer $t$.
Task: Find a matrix $\mathbf{M}'$ consistent with $\mathbf{M}$ such that $\mathsf{dr}(\mathbf{M}') \leq t$.

Aside from the problem variants where $p$ is a fixed prime number, we also study the case where matrix entries range over a domain that is provided as part of the input. In particular, the problems RMC and DRMC are defined analogously to $p$-RMC and $p$-DRMC, respectively, with the sole distinction that the prime number $p$ is provided as part of the input. We note that 2-RMC is **NP**-hard even for $t = 3$ (Peeters, 1996), and the same holds for 2-DRMC (see Theorem 14). Without loss of generality, we assume that the rows of the input matrix are pairwise distinct.

### 2.2. Treewidth

Treewidth (Robertson & Seymour, 1986) is one of the most prominent decompositional parameters for graphs and has found numerous applications in computer science. A *tree-decomposition* $\mathcal{T}$ of a graph $G = (V, E)$ is a pair $(T, \chi)$,

where $T$ is a tree and $\chi$ is a function that assigns each tree node $t$ a set $\chi(t) \subseteq V$ of vertices such that the following conditions hold: (TD1) for every edge $uv \in E(G)$ there is a tree node $t$ such that $u, v \in \chi(t)$; and (TD2) for every vertex $v \in V(G)$, the set of tree nodes $t$ with $v \in \chi(t)$ forms a non-empty subtree of $T$. The *width* of a tree-decomposition $(T, \chi)$ is the size of a largest bag minus 1. A tree-decomposition of minimum width is called *optimal*. The *treewidth* of a graph $G$, denoted by $\mathsf{tw}(G)$, is the width of an optimal tree decomposition of $G$. We will assume that the tree $T$ of a tree-decomposition is rooted and we will denote by $T_t$ the subtree of $T$ rooted at $t$ and write $\chi(T_t)$ for the set $\bigcup_{t' \in V(T_t)} \chi(t')$.

### 2.3. Problem Parameterizations

One advantage of the parameterized complexity paradigm is that it allows us to study the complexity of a problem w.r.t. several parameterizations of interest/relevance. To provide a concise description of the parameters under consideration, we introduce the following terminology: We say that a $\bullet$ entry at position $[i, j]$ in an incomplete matrix $\mathbf{M}$ is *covered* by row $i$ and by column $j$. In this paper, we study RMC and DRMC w.r.t. the following parameterizations (see Figure 1 for illustration):

- col: The minimum number of columns in the matrix $\mathbf{M}$ covering all occurrences of $\bullet$ in $\mathbf{M}$.
- row: The minimum number of rows in the matrix $\mathbf{M}$ covering all occurrences of $\bullet$ in $\mathbf{M}$.
- comb: The minimum value of $r + c$ such that there exist $r$ rows and $c$ columns in $\mathbf{M}$ with the property that each occurrence of $\bullet$ is covered one of these rows or columns.

$$\begin{pmatrix} 1 & 1 & 1 & 0 & \bullet & 1 \\ 0 & 0 & 1 & 0 & \bullet & 1 \\ 0 & \bullet & \bullet & 0 & \bullet & \bullet \\ 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

*Figure 1.* Illustration of the parameters col, row, and comb in an incomplete matrix. Here $\mathsf{col} = 4$, $\mathsf{row} = 3$, and $\mathsf{comb} = 2$.

For instance, the aforementiond problem TRIANGULATION FROM INCOMPLETE DATA, where a small number of distance-sensors are faulty, would result in matrix completion instances where col and row are both small.

We denote the parameter under consideration in brackets after the problem name (e.g., DRMC[comb]). As mentioned in Section 1, both $p$-RMC and $p$-DRMC are trivially in **FPT** when parameterized by the number of missing entries, and hence this parameterization is not discussed further.

Given an incomplete matrix $\mathbf{M}$, computing the parameter values for col and row is trivial. Furthermore, the parameter values satisfy $\mathsf{comb} \leq \mathsf{row}$ and $\mathsf{comb} \leq \mathsf{col}$. The parameter value for comb can also be computed in polynomial time by reducing the problem to finding a vertex cover in a bipartite graph:

**Proposition 1.** *Given an incomplete matrix $\mathbf{M}$ over GF($p$), we can compute the parameter value for* comb*, along with sets $R$ and $C$ of total cardinality* comb *containing the indices of covering rows and columns, respectively, in time $\mathcal{O}((n \cdot m)^{1.5})$.*

## 3. Rank Minimization

In this section we present our results for BOUNDED RANK MATRIX COMPLETION under various parameterizations.

### 3.1. Bounded Domain: Parameterization by row

As our first result, we present an algorithm for solving $p$-RMC[row]. This will serve as a gentle introduction to the techniques used in the more complex result for $p$-RMC[comb], and will also be used to give an **XP** algorithm for RMC[row].

**Theorem 2.** $p$-RMC[row] *is in* **FPT**.

*Proof Sketch.* Let $R$ be the (minimum) set of rows that cover all occurrences of $\bullet$ in the input matrix $\mathbf{M}$. Since the existence of a solution does not change if we permute the rows of $\mathbf{M}$, we permute the rows of $\mathbf{M}$ so that the rows in $R$ have indices $1, \ldots, k$. We now proceed in three steps.

For the first step, we will define the notion of *signature*: A signature $S$ is a tuple $(I, D)$, where $I \subseteq R$ and $D$ is a mapping from $R \setminus I$ to $(I \to \mathrm{GF}(p))$. Intuitively, a signature $S$ specifies a subset $I$ of $R$ which is expected to be independent in $M[k+1:m, *] \cup I$ (i.e., adding the rows in $I$ to $M[k+1:m, *]$ is expected to increase the rank of $M[k+1:m, *]$ by $|I|$); and for each remaining row of $R$, $S$ specifies how that row should depend on $I$. The latter is carried out using $D$: For each row in $R \setminus I$, $D$ provides a set of coefficients expressing the dependency of that row on the rows in $I$. Formally, we say that a matrix $\mathbf{M}'$ that is compatible with the incomplete matrix $\mathbf{M}$ *matches* a signature $(I, D)$ if and only if, for each row (*i.e.*, vector) $d \in R \setminus I$, there exist coefficients $a^d_{k+1}, \ldots, a^d_m \in \mathrm{GF}(p)$ such that $d = a^d_{k+1}\mathbf{M}[k+1, *] + \cdots + a^d_m \mathbf{M}[m, *] + \sum_{i \in I} D(d)(i) \cdot i$. The first step of the algorithm branches through all possible signatures $S$. Clearly, the number of distinct signatures is upper-bounded by $2^k \cdot p^{k^2}$.

For the second step, we fix an enumerated signature $S$. The algorithm will verify whether $S$ is *valid*, *i.e.*, whether there exists a matrix $\mathbf{M}'$ compatible with $\mathbf{M}$ that matches $S$. To do so, the algorithm will construct a system of $|R \setminus I|$ equations over vectors of size $n$, and then transform this into a system $\Upsilon_S$ of $|R \setminus I| \cdot n$ equations over $\mathrm{GF}(p)$ (one equation for each vector coordinate). For each $d \in R \setminus I$, $\Upsilon_S$ contains

one variable for each coefficient $a^d_{k+1}, \ldots, a^d_m$ and one variable for each occurrence of $\bullet$ in the rows of $R$. For instance, the first equation in $\Upsilon_S$ has the following form: $d[1] = a^d_{k+1}\mathbf{M}[k+1,1] + \cdots + a^d_m\mathbf{M}[m,1] + \sum_{i\in I} D(d)(i)\cdot i[1]$, where $a^d_{k+1}, \ldots, a^d_m$ are variables, and $d[1]$ as well as each $i[1]$ in the sum could be a variable or a fixed number. Crucially, $\Upsilon_S$ is a system of at most $(k\cdot n)$ *linear* equations over GF($p$) with at most $m + kn$ variables, and can be solved in time $\mathcal{O}((m+kn)^3)$ by Gaussian elimination. Constructing the equations takes time $\mathcal{O}(m \cdot n)$.

During the second step, the algorithm determines whether a signature $S$ is valid or not, and in the end, after going through all signatures, selects an arbitrary valid signature $S = (I, D)$ with minimum $|I|$. For the final third step, the algorithm checks whether $|I| + \mathsf{rk}(\mathbf{M}[k+1 : m, *]) \leq t$. We note that computing $\mathsf{rk}(\mathbf{M}[k+1 : m, *])$ can be carried out in time $\mathcal{O}(nm^{1.4})$ (Ibarra et al., 1982). If the above inequality does not hold, the algorithm rejects; otherwise it recomputes a solution to $\Upsilon_S$ and outputs the matrix $\mathbf{M}'$ obtained from $\mathbf{M}$ by replacing each occurrence of $\bullet$ at position $[i, j]$ by the value of the variable $i[j]$ in the solution to $\Upsilon_S$. The total running time is $\mathcal{O}((2^k \cdot p^{k^2}) \cdot ((m+kn)^3 + nm^{1.4})) = \mathcal{O}(2^k p^{k^2} \cdot (m+kn)^3)$. □

Since the the transpose of $\mathbf{M}$ has the same rank as $\mathbf{M}$, it follows immediately that $p$-RMC[col] is in **FPT**.

**Corollary 3.** $p$-RMC[col] *is in* **FPT**.

As a consequence of the running time of the algorithm given in the proof of Theorem 2, we obtain:

**Corollary 4.** RMC[row] *and* RMC[col] *are in* **XP**.

### 3.2. Bounded Domain: Parameterization by comb

In this subsection, we present a randomized fixed-parameter algorithm for $p$-RMC[comb] with constant one-sided error probability. Before we proceed to the algorithm, we need to introduce some basic terminology related to systems of equations. Let $\Upsilon$ be a system of $\ell$ equations $\mathrm{EQ}_1, \mathrm{EQ}_2, \ldots, \mathrm{EQ}_\ell$ over GF($p$); we assume that the equations are simplified as much as possible. In particular, we assume that no equation contains two terms over the same set of variables such that the degree/exponent of each variable in both terms is the same. Let $\mathrm{EQ}_i$ be a linear equation in $\Upsilon$, and let $x$ be a variable which occurs in $\mathrm{EQ}_i$ (with a non-zero coefficient). Naturally, $\mathrm{EQ}_i$ can be transformed into an equivalent equation $\mathrm{EQ}_{i,x}$, where $x$ is isolated, and we use $\Gamma_{i,x}$ to denote the side of $\mathrm{EQ}_{i,x}$ not containing $x$, i.e., $\mathrm{EQ}_{i,x}$ is of the form $x = \Gamma_{i,x}$. We say that $\Upsilon'$ is obtained from $\Upsilon$ by *substitution of $x$ in $\mathrm{EQ}_i$* if $\Upsilon'$ is the system of equations obtained by:

1. computing $\mathrm{EQ}_{i,x}$ and in particular $\Gamma_{i,x}$ from $\mathrm{EQ}_i$;
2. setting $\Upsilon' := \Upsilon \setminus \{\mathrm{EQ}_i\}$; and
3. replacing $x$ with $\Gamma_{i,x}$ in every equation in $\Upsilon'$.

Observe that $\Upsilon'$ has size $\mathcal{O}(n \cdot \ell)$, and can also be computed in time $\mathcal{O}(n\cdot\ell)$, where $n$ is the number of variables occurring in $\Upsilon$. Furthermore, any solution to $\Upsilon'$ can be transformed into a solution to $\Upsilon$ in linear time, and similarly any solution to $\Upsilon$ can be transformed into a solution to $\Upsilon'$ in linear time (*i.e.*, $\Upsilon'$ and $\Upsilon$ are equivalent). Moreover, $\Upsilon'$ contains at least one fewer variable and one fewer equation than $\Upsilon$.

The following proposition is crucial for our proof, and is of independent interest.

**Proposition 5.** *Let $\Upsilon$ be a system of $\ell$ quadratic equations over GF($p$). Then computing a solution for $\Upsilon$ is in* $\mathbf{FPT}_R$ *parameterized by $\ell$ and $p$, and in* $\mathbf{XP}_R$ *parameterized only by $\ell$.*

*Proof.* Let $n$ be the number of variables in $\Upsilon$. We distinguish two cases. If $n \geq \ell(\ell+3)/2$, then $\Upsilon$ can be solved in randomized time $\mathcal{O}(2^\ell n^3 \ell (\log p)^2)$ (Miura et al., 2014). Otherwise, $n < \ell(\ell+3)/2$, and we can solve $\Upsilon$ by a brute-force algorithm which enumerates (all of the) at most $p^n < p^{\ell(\ell+3)/2}$ assignments of values to the variables in $\Upsilon$. The proposition now follows by observing that the given algorithm runs in time $\mathcal{O}(2^\ell n^3 \ell (\log p)^2 + p^{\ell(\ell+3)/2}\ell^2)$. □

**Theorem 6.** $p$-RMC[comb] *is in* $\mathbf{FPT}_R$.

*Proof Sketch.* We begin by using Proposition 1 to compute the sets $R$ and $C$ containing the indices of the covering rows and columns, respectively; let $|R| = r$ and $|C| = c$, and recall that the parameter value is $k = r + c$. Since the existence of a solution for $p$-RMC does not change if we permute rows and columns of $\mathbf{M}$, we permute the rows of $\mathbf{M}$ so that the rows in $R$ have indices $1, \ldots, r$, and subsequently, we permute the columns of $\mathbf{M}$ so that the columns in $C$ have indices $1, \ldots, c$.

Before we proceed, let us give a high-level overview of our strategy. The core idea is to branch over signatures, which will be defined in a similar way to those in Theorem 2. These signatures will capture information about the dependencies among the rows in $R$ and columns in $C$; one crucial difference is that for columns, we will focus only on dependencies in the submatrix $\mathbf{M}[r+1 : m, *]$. In each branch, we arrive at a system of equations that needs to be solved in order to determine whether the signatures are valid. Unlike Theorem 2, here the obtained system of equations will contain non-linear (but quadratic) terms, and hence solving the system is far from being trivial. Once we determine which signatures are valid, we choose one that minimizes the total rank.

For the first step, let us define the notion of signature that will be used in this proof. A *signature* $S$ is a tuple

$(I_R, D_R, I_C, D_C)$ where: 1. $I_R \subseteq R$; 2. $D_R$ is a mapping from $R \setminus I_R$ to $(I_R \to \text{GF}(p))$; 3. $I_C \subseteq C$; and 4. $D_C$ is a mapping from $C \setminus I_C$ to $(I_C \to \text{GF}(p))$.

We say that a matrix $\mathbf{M}'$ compatible with the incomplete matrix $\mathbf{M}$ *matches* a signature $(I_R, D_R, I_C, D_C)$ if:

- for each row $d \in R \setminus I_R$, there exist coefficients $a^d_{r+1}, \ldots, a^d_m \in \text{GF}(p)$ such that $d = a^d_{r+1}\mathbf{M}'[r+1, *] + \cdots + a^d_m \mathbf{M}'[m, *] + \sum_{i \in I_R} D_R(d)(i) \cdot i$; and
- for each column $h \in C \setminus I_c$, there exist coefficients $b^h_{c+1}, \ldots, b^h_n \in \text{GF}(p)$ such that $h[r+1 : m] = b^h_{c+1}\mathbf{M}'[r+1 : m, c] + \cdots + b^h_n \mathbf{M}'[r+1 : m, n] + \sum_{i \in I_C} D_C(h)(i) \cdot i[r+1 : m]$.

The number of distinct signatures is upper-bounded by $2^r \cdot p^{r^2} \cdot 2^c \cdot p^{c^2} \leq 2^k \cdot p^{k^2}$, and the first step of the algorithm branches over all possible signatures $S$. In the second step, for each enumerated signature $S$, we check whether $S$ is valid (*i.e.*, whether there exists a matrix $\mathbf{M}'$, compatible with the incomplete $\mathbf{M}$, that matches $S$) in a similar fashion as in the proof of Theorem 2. Here, this results in a system $\Upsilon_S$ of $|R \setminus I_R| \cdot n + |C \setminus I_C| \cdot (m - r)$ equations which check the dependencies for rows in $R \setminus I_R$ and columns in $C \setminus I_C$. For instance, the first equation in $\Upsilon_S$ for some $d \in R \setminus I_R$ has the following form: $d[1] = a^d_{r+1}\mathbf{M}[r+1, 1] + \cdots + a^d_m \mathbf{M}[m, 1] + \sum_{i \in I_R} D_R(d)(i) \cdot i[1]$, where $a^d_{r+1}, \ldots, a^d_m$ are variables, $D_R(d)(i)$ is a number, and all other occurrences are either variables or numbers. Similarly, the second equation in $\Upsilon_S$ for some $h \in C \setminus I_C$ has the following form: $h[r+2] = b^h_{c+1}\mathbf{M}[r+2, c+1] + \cdots + b^h_n \mathbf{M}[r+2, n] + \sum_{i \in I_C} D_C(d)(i) \cdot i[r+2]$, where $b^h_{c+1}, \ldots, b^h_n$ are variables, $D_C(d)(i)$ is a number, and all other occurrences are either variables or numbers.

Next, observe that the only equations in $\Upsilon_S$ that may contain non-linear terms are those for $d[j]$, where $j \leq c$, and in particular $\Upsilon_S$ contains at most $k^2$ equations with non-linear terms ($k$ equations for at most $k$ vectors $d$ in $R \setminus I_R$). We will now use substitutions to simplify $\Upsilon_S$ by removing all linear equations; specifically, at each step we select an arbitrary linear equation $\text{EQ}_i$ containing a variable $x$, apply substitution of $x$ in $\text{EQ}_i$ to construct a new system of equations with one fewer equation, and simplify all equations in the new system. If at any point we reach a system of equations that contains an invalid equation (*e.g.*, 2=5), then $\Upsilon_S$ does not have a solution, and we discard the corresponding branch. Otherwise, after at most $|R \setminus I_R| \cdot n + |C \setminus I_C| \cdot (m - r) \in \mathcal{O}(kn + km)$ substitutions, we obtain a system of at most $k^2$ quadratic equations $\Psi_S$ such that any solution to $\Psi_S$ can be transformed into a solution to $\Upsilon_S$ in time $\mathcal{O}(kn + km)$. We can now apply Proposition 5 to solve $\Psi_S$ and mark $S$ as a valid signature if $\Psi_S$ has a solution.

After all signatures have been processed, the algorithm selects a valid signature $S = (I, D)$ that has the minimum value of $|I_R| + |I_C|$, checks whether $|I_R| + |I_C| + \mathsf{rk}(\mathbf{M}[r+1 : m, c : 1+n]) \leq t$, and either uses this to construct a solution (similarly to the proof of Theorem 2), or outputs "no". The theorem now follows by observing that the total running time of the algorithm is obtained by combining the branching factor of branching over all signatures ($\mathcal{O}(2^k \cdot p^{k^2})$) with the run-time of Proposition 5 for $k^2$ many quadratic equations ($\mathcal{O}(3^{k^2} n^3 (\log p)^2 + p^{k^4})$). In particular, we obtain a running time of $\mathcal{O}(3^{k^2} \cdot p^{k^4} \cdot n^3)$. □

As a consequence of the running time of the algorithm given in the proof of Theorem 6, we obtain:

**Corollary 7.** RMC[comb] *is in* $\mathbf{XP}_R$.

## 4. Bounded Distinct Row Matrix Completion

Let $(p, \mathbf{M}, t)$ be an instance of DRMC. We say that two rows of $\mathbf{M}$ are *compatible* if whenever the two rows differ at some entry then one of the rows has a • at that entry. The *compatibility graph* of $\mathbf{M}$, denoted by $G(\mathbf{M})$, is the undirected graph whose vertices correspond to the row indices of $\mathbf{M}$ and in which there is an edge between two vertices if and only if their two corresponding rows are compatible. See Figure 2 for an illustration.

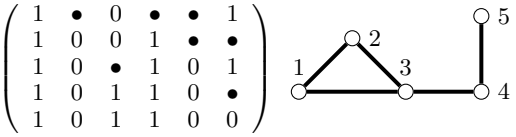


*Figure 2.* Illustration of a matrix and its compatibility graph. The vertex label indicates the corresponding row number.

We start by showing that DRMC (and therefore $p$-DRMC) can be reduced to the Clique Cover problem, which is defined as follows.

| Clique Cover (CC) | |
|---|---|
| Input: | An undirected graph $G$ and an integer $k$. |
| Task: | Find a partition of $V(G)$ into at most $k$ cliques, or output that no such partition exists. |

**Lemma 8.** *An instance $\mathcal{I} = (p, \mathbf{M}, t)$ of* DRMC *has a solution if and only if the instance $\mathcal{I}' = (G(\mathbf{M}), t)$ of* CC *does. Moreover, a solution for $\mathcal{I}'$ can obtained in polynomial-time from a solution for $\mathcal{I}$ and vice versa.*

*Proof Sketch.* The lemma follows immediately from the observation that a set $R$ of rows in $\mathbf{M}$ can be made identical if and only if $G(\mathbf{M})[R]$ is a clique. □

**Theorem 9.** CC *is in* **FPT** *when parameterized by the treewidth of the input graph.*

*Proof Sketch.* We show the theorem via a standard dynamic programming algorithm on a tree-decomposition of the input graph (Bodlaender & Koster, 2008). Namely after computing an optimal tree-decomposition $(T, \chi)$ of the input graph $G$, which can be achieved in **FPT**-time w.r.t. the treewidth of $G$ (Kloks, 1994; Bodlaender, 1996; Bodlaender et al., 2016), we compute a set $\mathcal{R}(t)$ of tuples via a bottom-up dynamic programming algorithm for every $t \in V(T)$. In our case $(\mathcal{P}, c) \in \mathcal{R}(t)$ if and only if $\mathcal{P}$ is a partition of $G[\chi(t)]$ into cliques and $c$ is the minimum number such that $G[\chi(T_t)]$ has a partition $\mathcal{P}'$ into $c$ cliques with $\mathcal{P} = \{ P' \cap \chi(t) \mid P' \in \mathcal{P}' \} \setminus \{\emptyset\}$. □

Note that the above theorem also implies that the well-known COLORING problem is **FPT** parameterized by the treewidth of the complement of the input graph. The theorem below follows immediately from Lemmas 8 and 9.

**Theorem 10.** DRMC *and* $p$-DRMC *are in* **FPT** *when parameterized by the treewidth of the compatibility graph.*

### 4.1. $p$-DRMC

**Theorem 11.** $p$-DRMC[comb] *is in* **FPT**.

*Proof.* Let $(\mathbf{M}, t)$ be an instance of $p$-DRMC, and let $k$ be the parameter comb. By Proposition 1, we can compute a set $R_\bullet$ of rows and a set $C_\bullet$ of columns, where $|R_\bullet \cup C_\bullet| \leq k$, and such that every occurrence of $\bullet$ in $\mathbf{M}$ is either contained in a row or column in $R_\bullet \cup C_\bullet$. Let $R$ and $C$ be the set of rows and columns of $\mathbf{M}$, respectively. Let $\mathcal{P}$ be the unique partition of $R \setminus R_\bullet$ such that two rows $r$ and $r'$ belong to the same set in $\mathcal{P}$ if and only if they are identical on all columns in $C \setminus C_\bullet$. Then $|P| \leq (p+1)^k$, for every $P \in \mathcal{P}$, since two rows in $P$ can differ on at most $|C_\bullet| \leq k$ entries, each having $(p+1)$ values to be chosen from. Moreover, any two rows in $R \setminus R_\bullet$ that are not contained in the same set in $\mathcal{P}$ are not compatible, which implies that they appear in different components of $G(\mathbf{M}) \setminus R_\bullet$ and hence the set of vertices in every component of $G(\mathbf{M}) \setminus R_\bullet$ is a subset of $P$, for some $P \in \mathcal{P}$. It is now straightforward to show that $\mathsf{tw}(G(\mathbf{M})) \leq k + (p+1)^k$, and hence, $\mathsf{tw}(G(\mathbf{M}))$ is bounded by a function of the parameter $k$. The theorem now follows from Theorem 10. □

### 4.2. DRMC

The proof of the following theorem is very similar to the proof of Theorem 11, i.e., we mainly use the observation that the parameter row is also a bound on the treewidth of the compatibility graph and then apply Theorem 10.

**Theorem 12.** DRMC[row] *is in* **FPT**.

For the remainder of this section, we will introduce the PARTITIONING INTO TRIANGLES (PIT) problem: Given a graph $G$, decide whether there is a partition $\mathcal{P}$ of $V(G)$ into triangles. We will often use the following easy observation.

**Observation 1.** *A graph $G$ that does not contain a clique with four vertices has a partition into triangles if and only if it has a partition into at most $|V(G)|/3$ cliques.*

**Theorem 13.** DRMC[col] *is* **paraNP**-*hard.*

*Proof Sketch.* We will reduce from the **NP**-complete 3-SAT-2 problem (Berman et al., 2003): Given a propositional formula $\phi$ in conjunctive normal form such that (1) every clause of $\phi$ has exactly three distinct literals and (2) every literal occurs in exactly two clauses, decide whether $\phi$ is satisfiable. To make our reduction easier to follow, we will divide the reduction into two steps. Given an instance (formula) $\phi$ of 3-SAT-2, we will first construct an equivalent instance $G$ of PIT with the additional property that $G$ does not contain a clique on four vertices. We note that similar reductions from variants of the satisfiability problem to PIT are known (and hence our first step does not show anything new for PIT); however, our reduction is specifically designed to simplify the second step, in which we will construct an instance $(\mathbf{M}, |V(G)|/3)$ of DRMC such that $G(\mathbf{M})$ is isomorphic to $G$ and $\mathbf{M}$ has only seven columns. By Observation 1 and Lemma 8, this proves the theorem since $(\mathbf{M}, |V(G)|/3)$ has a solution if and only if $\phi$ does.

Let $\phi$ be an instance of 3-SAT-2 with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$. We first construct the instance $G$ of PIT such that $G$ does not contain a clique of size four. For every variable $x_i$ of $\phi$, let $G(x_i)$ be the graph illustrated in Figure 3, and for every clause $C_j$ of $\phi$, let $G(C_j)$ be the graph illustrated in Figure 4. Let $f : [m] \times [3] \to \{ x_i^o, \bar{x}_i^o \mid 1 \leq i \leq n \wedge 1 \leq o \leq 2 \}$ be any bijective function such that for every $j$ and $r$ with $1 \leq j \leq m$ and $1 \leq r \leq 3$, it holds that: If $f(j,r) = x_i^o$ (for some $i$ and $o$), then $x_i$ is the $r$-th literal of $C_j$; and if $f(j,r) = \bar{x}_i^o$, then $\bar{x}_i$ is the $r$-th literal of $C_j$.

The graph $G$ is obtained from the disjoint union of the graphs $G(x_1), \ldots, G(x_n), G(C_1), \ldots, G(C_m)$ after applying the following modifications: (1) For every $j$ and $r$ with $1 \leq j \leq m$ and $1 \leq r \leq 3$ add edges forming a triangle on the vertices $l_{j,r}^1$, $l_{j,r}^2$, $f(j,r)$; and (2) for every $i$ with $1 \leq i \leq 2n - m$, add the vertices $g_i^1, g_i^2$ and an edge between $g_i^1$ and $g_i^2$. Finally we add edges forming a complete bipartite graph between all vertices in $\{ g_i^o \mid 1 \leq i \leq 2n - m \wedge 1 \leq o \leq 2 \}$ and all vertices in $\{ h_i^o \mid 1 \leq i \leq n \wedge 1 \leq o \leq 2 \}$.

This completes the construction of $G$. The following claim concludes the first step of our reduction.

**Claim 1.** *$\phi$ is satisfiable if and only if $G$ has a partition*

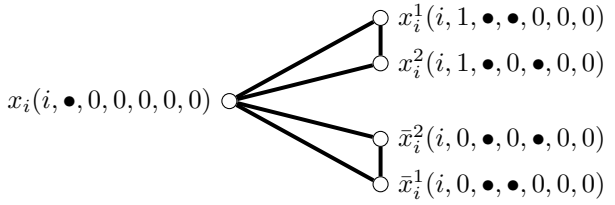


*Figure 3.* An illustration of the gadget $G(x_i)$ introduced in the reduction of Theorem 13. The label of each vertex $v$ indicates the row vector $R(v)$.

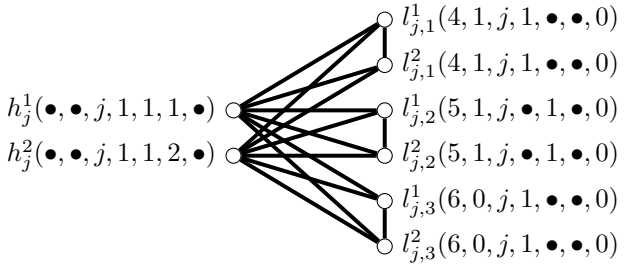


*Figure 4.* An illustration of the gadget $G(C_j)$ introduced in the reduction of Theorem 13. The label of each vertex $v$ indicates the row vector $R(v)$; here we assume that $f(j,1) = x_4^1$, $f(j,2) = x_5^2$, and $f(j,3) = \bar{x}_6^1$.

*into triangles. Moreover, $G$ does not contain a clique of size four.*

We will now proceed to the second (and final) step of our reduction, i.e., we will construct an instance $(\mathbf{M}, |V(G)|/3)$ of DRMC such that: $G(\mathbf{M})$ is isomorphic to $G$ and $\mathbf{M}$ has only seven columns.

$\mathbf{M}$ contains one row $R(u)$ for every $u \in V(G)$. The definition of $R(u)$ for every vertex $u$ that is part of some gadget $G(x_i)$ or $G(C_j)$ is illustrated in Figures 3 and 4. Additionally, we set $R(g_j^o) = (\bullet,\bullet,\bullet,\bullet,\bullet,\bullet,j)$ for every $j$ and $o$ with $1 \leq j \leq 2n-m$ and $1 \leq o \leq 2$. Using an exhaustive case analysis, one can show that $G(\mathbf{M})$ is indeed isomorphic to $G$, which concludes the proof of the theorem. □

We conclude this section with a hardness result showing that 2-DRMC remains **NP**-hard when the number of missing or known entries in each column/row is bounded.

**Theorem 14.** *The restriction of* 2-DRMC *to instances in which each row and each column contains exactly three missing entries is* **NP***-hard. The same holds for the restriction of* 2-DRMC *to instances in which each row and each column contains at most 4 determined entries.*

*Proof Sketch.* Consider the problem of (properly) coloring a graph on $n$ vertices, having minimum degree $n-4$ and no independent set of size 4, by $n/3$ colors, where $n$ is divisible by 3; denote this problem as $(n/3)$-COLORING$^{\delta=n-4}$. This problem is **NP**-hard via a reduction from the PARTITION INTO TRIANGLES problem on $K_4$-free cubic graphs. The **NP**-hardness of the latter problem follows from the **NP**-hardness of the PARTITION INTO TRIANGLES problem on planar cubic graphs (Cerioli et al., 2008), since a $K_4$ in a cubic graph must be isolated, and hence can be removed from the start. Finally, using Observation 1, PARTITION INTO TRIANGLES on $K_4$-free cubic graphs is polynomial-time reducible to $(n/3)$-COLORING$^{\delta=n-4}$, via the simple reduction that complements the edges of the graph.

Now we reduce from $(n/3)$-COLORING$^{\delta=n-4}$ to $p$-DRMC by mimicking a standard reduction from 3-coloring to rank minimization (Peeters, 1996). Given an instance $G$ of $(n/3)$-COLORING$^{\delta=n-4}$, we construct an $n \times n$ matrix $\mathbf{M}$ whose rows and columns correspond to the vertices in $G$, as follows. The diagonal entries of $\mathbf{M}$ are all ones. For an entry at row $i$ and column $j$, where $i \neq j$, $\mathbf{M}[i,j] = 0$ if $ij \in E(G)$, and is $\bullet$ otherwise. Finally, we set $r = n/3$. Observe that since each vertex in $G$ has $n-4$ neighbors, the number of missing entries in any row and any column of $\mathbf{M}$ is 3.

It is not difficult to show that $G$ is a yes-instance of $(n/3)$-COLORING$^{\delta=n-4}$ if and only if $(\mathbf{M}, n/3)$ is a yes-instance of 2-DRMC.

The second statement in the theorem follows via a reduction from 3-COLORING on graphs of maximum degree at most 4 (Garey et al., 1976), using similar arguments. □

## 5. Conclusion

We studied the parameterized complexity of two fundamental matrix completion problems under several parameterizations. For the bounded domain case, we painted a positive picture by showing that the two problems are in **FPT** (resp. $\mathbf{FPT}_R$) w.r.t. all considered parameters. For the unbounded domain case, we characterized the parameterized complexity of DRMC by showing that it is in **FPT** parameterized by row, and **paraNP**-hard parameterized by col (and hence by comb). For RMC, we could show its membership in **XP** (resp. $\mathbf{XP}_R$) w.r.t. all considered parameters. Three immediate open questions ensue:

- Is it possible to obtain a deterministic algorithm for $p$-RMC and RMC parameterized by comb?
- Can we improve our **XP** (resp. $\mathbf{XP}_R$) results for RMC to **FPT** (resp. $\mathbf{FPT}_R$) or show that the problems are **W**[1]-hard?
- Does a hardness result, similar to the one given in Theorem 14 for $p$-DRMC, hold for $p$-RMC?

**Acknowledgments.** Robert Ganian is also affiliated with FI MU, Czech Republic.

## References


Matrix completion and the Netflix challenge. See https://en.wikipedia.org/wiki/Matrix_completion.

Bäckström, C., Jonsson, P., Ordyniak, S., and Szeider, S. A complete parameterized complexity analysis of bounded planning. *J. of Computer and System Sciences*, 81(7): 1311–1332, 2015.

Berman, P., Karpinski, M., and Scott, A. D. Approximation hardness of short symmetric instances of MAX-3SAT. *Electronic Colloquium on Computational Complexity (ECCC)*, (049), 2003.

Bessiere, C., Hebrard, E., Hnich, B., Kiziltan, Z., Quimper, C., and Walsh, T. The parameterized complexity of global constraints. In Fox, D. and Gomes, C. P. (eds.), *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pp. 235–240. AAAI Press, 2008.

Bodlaender, H. L. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25 (6):1305–1317, 1996.

Bodlaender, H. L. and Koster, A. M. C. A. Combinatorial optimization on graphs of bounded treewidth. *The Computer Journal*, 51(3):255–269, 2008.

Bodlaender, H. L., Drange, P. G., Dregi, M. S., Fomin, F. V., Lokshtanov, D., and Pilipczuk, M. A $O(c^k n)$ 5-approximation algorithm for treewidth. *SIAM J. Comput.*, 45(2):317–378, 2016.

Candès, E. J. and Plan, Y. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Candès, E. J. and Tao, T. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Information Theory*, 56(5):2053–2080, 2010.

Cerioli, M. R., Faria, L., Ferreira, T. O., Martinhon, C. A. J., Protti, F., and Reed, B. A. Partition into cliques for cubic graphs: Planar case, complexity and approximation. *Discrete Applied Mathematics*, 156(12):2270–2278, 2008.

Courtois, N., Goubin, L., Meier, W., and Tacier, J. Solving underdefined systems of multivariate quadratic equations. In *Public Key Cryptography, 5th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2002, Paris, France, February 12-14, 2002, Proceedings*, volume 2274 of *Lecture Notes in Computer Science*, pp. 211–227. Springer, 2002.

Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., and Saurabh, S. *Parameterized Algorithms*. Springer, 2015.

Downey, R. G. and Fellows, M. R. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer Verlag, 2013. ISBN 978-1-4471-5558-4, 978-1-4471-5559-1.

Elhamifar, E. and Vidal, R. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11):2765–2781, 2013.

Endriss, U., de Haan, R., and Szeider, S. Parameterized complexity results for agenda safety in judgment aggregation. In Weiss, G., Yolum, P., Bordini, R. H., and Elkind, E. (eds.), *Proceedings of AAMAS 2015, the 14th International Conference on Autonomous Agents and Multiagent Systems*, pp. 127–136. IFAAMAS/ACM, 2015.

Fazel, M. *Matrix rank minimization with applications*. PhD thesis, Stanford University, 2002.

Frieze, A. M., Kannan, R., and Vempala, S. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, 2004.

Ganian, R. and Ordyniak, S. The complexity landscape of decompositional parameters for ILP. *Artificial Intelligence*, 257:61 – 71, 2018.

Garey, M. R., Johnson, D. S., and Stockmeyer, L. J. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

Gaspers, S. and Szeider, S. Guarantees and limits of preprocessing in constraint satisfaction and reasoning. *Artificial Intelligence*, 216:1–19, 2014. doi: 10.1016/j.artint.2014.06.006.

Gottlob, G. and Szeider, S. Fixed-parameter algorithms for artificial intelligence, constraint satisfaction, and database problems. *The Computer Journal*, 51(3):303–325, 2006. Survey paper.

Hardt, M., Meka, R., Raghavendra, P., and Weitz, B. Computational limits for matrix completion. In *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *JMLR Workshop and Conference Proceedings*, pp. 703–725. JMLR.org, 2014.

Ibarra, O. H., Moran, S., and Hui, R. A generalization of the fast LUP matrix decomposition algorithm and applications. *J. Algorithms*, 3(1):45–56, 1982.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from a few entries. *IEEE Trans. Information Theory*, 56(6):2980–2998, 2010a.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–2078, 2010b.

Kloks, T. *Treewidth: Computations and Approximations*. Springer Verlag, Berlin, 1994.

Miura, H., Hashimoto, Y., and Takagi, T. Extended algorithm for solving underdefined multivariate quadratic equations. *IEICE Transactions*, 97-A(6):1418–1425, 2014.

Peeters, R. Orthogonal representations over finite fields and the chromatic number of graphs. *Combinatorica*, 16(3): 417–431, 1996.

Recht, B. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

Robertson, N. and Seymour, P. D. Graph minors. II. Algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.

Saunderson, J., Fazel, M., and Hassibi, B. Simple algorithms and guarantees for low rank matrix completion over $F_2$. In *Proceedings of The IEEE International Symposium on Information Theory*, pp. 86–90. IEEE, 2016.

van Bevern, R., Komusiewicz, C., Niedermeier, R., Sorge, M., and Walsh, T. H-index manipulation by merging articles: Models, theory, and experiments. *Artif. Intell.*, 240:19–35, 2016.