On Bounded Block Decomposition Problems for Under-Specified Systems of Equations

Matthijs Bomhoff       Walter Kern       Georg Still

December 14, 2010

Abstract

When solving a system of equations, it can be beneficial not to solve it in its entirety at once, but rather to decompose it into smaller subsystems that can be solved in order. Based on a bisimplicial graph representation we analyze the parameterized complexity of two problems central to such a decomposition: The Free Square Block problem related to finding smallest subsystems that can be solved separately, and the Bounded Block Decomposition problem related to determining a decomposition where the largest subsystem is as small as possible. We show both problems to be $W[1]$-hard. Finally we relate these problems to crown structures and settle two open questions regarding them using our results.

keywords: bipartite graph, decomposition, parameterized complexity

1 Introduction

Finding feasible solutions to large systems of equations is a problem that often occurs in technical disciplines. In mechanical engineering design for example, (physical) requirements of an artifact can often be translated into an under-specified system of equations that describes the physically feasible configurations. Another application is that of 3D scene reconstruction in computer vision where systems of hundreds of (possibly non-linear) equations and variables occur naturally for even seemingly small instances (see e.g. [1]). As such systems grow in size, efficiently finding feasible solutions becomes progressively harder: The effort required to find solutions grows superlinearly in the number of variables and equations. In many applications, each equation involves only a relatively small subset of the variables, so decomposition into subsystems that can be solved separately is a promising approach. However, determining a good decomposition, i.e., one in which the largest remaining subsystem is as small as possible, seems to be a hard problem.

In the remainder of this paper, we assume the systems under consideration are consistent and free of redundancy. As an example, consider the following system of equations:

\[
\begin{align*}
    u_1(v_1, v_2, v_4) &= 0 \\
    u_2(v_1, v_2, v_3) &= 0 \\
    u_3(v_3, v_4) &= 0 \\
    u_4(v_3, v_4) &= 0
\end{align*}
\]

This system can be solved by first solving $u_3, u_4, v_3, v_4$ as a subsystem, substituting the values found for $v_3$ and $v_4$ into the remaining equations and subsequently solving $u_1, u_2, v_1, v_2$.
as a subsystem to obtain a complete solution. For systems with an equal number of variables and equations, there is a unique decomposition into minimal subsystems that admit no further decomposition. Such a decomposition of a system of equations is strongly related to the canonical decomposition of bipartite graphs as investigated by Dulmage and Mendelsohn (see e.g. [2, 3, 4]). However, for under-specified systems, the decomposition is no longer unique. Consider for example the following system obtained by adding an additional variable $v_5$:

$$
\begin{align*}
    u_1(v_1, v_2, v_4) &= 0 \\
    u_2(v_1, v_2, v_3, v_5) &= 0 \\
    u_3(v_3, v_4, v_5) &= 0 \\
    u_4(v_3, v_4, v_5) &= 0
\end{align*}
$$

To solve this under-specified system, we could assign a random value to $v_5$, after which we can solve $u_3, u_4, v_3, v_4$ as a subsystem, and finally we can solve $u_1, u_2, v_1, v_2$ as a subsystem to obtain a complete solution. However, if we instead assign a random value to $v_2$ to begin with, the resulting system no longer admits further decomposition and we are forced to solve the remaining four equations as a single subsystem. This simple example shows that the decomposition of under-specified systems is no longer uniquely determined, but instead depends on the choice of free or driving variables. It is clear that an algorithm to determine a good decomposition is crucial for efficient solution techniques. In this paper, we analyze the parameterized complexity of two problems related to finding such a decomposition.

The remainder of this paper is organized as follows: Section 2 explains how the structure of a system of equations can be translated into a bipartite graph and describes the decomposition problem. The third section introduces the free square block approach to the decomposition problem. The subsequent two sections contain proofs of $W[1]$-completeness of both the free square block problem, and the decomposition problem for systems of equations as a whole. The sixth section discusses the relation of our results to crown structures and settles two open questions regarding these. And finally Section 7 contains a summary of the conclusions, as well as some ideas on future investigations into this subject.

## 2 Systems as Bipartite Graphs

In this section, we introduce a common bipartite graph representation for systems of equations and show how the decomposition problem translates to this graph. After going over the notation used, we briefly describe the work of Dulmage and Mendelsohn on the canonical decomposition of bipartite graphs and the related optimization problem.\(^z\)

For a set of vertices $X$ in a graph $G$, we denote by $G[X]$ the subgraph of $G$ induced by $X$ and by $\Gamma_G(X)$ the neighbors of $X$ in $G$; $\Gamma(X)$ is sometimes used when the graph $G$ is clear from the context. For a bipartite graph $G = (U, V, E)$, $U$ and $V$ denote the two classes of vertices and $E$ denotes the set of edges. For notational convenience, if we refer to an element $uv \in E$, we will tacitly assume $u \in U$ and $v \in V$. For a bipartite subgraph $A$ of $G = (U, V, E)$, we denote by $U_A$ and $V_A$ the vertices of $A$ that fall into these respective classes of $G$. If $M$ is a matching in a bipartite graph $G = (U, V, E)$, we denote by $U_M$ and $V_M$ the vertices that are matched in respectively $U$ and $V$. Analogously, for a set of vertices $C \subseteq U \cup V$ such as a vertex cover of $G$, we denote by $U_C$ and $V_C$ the vertices of the set in $U$ resp. $V$. If $X$ and $Y$ are two sets, $X - Y$ is used to denote the set difference. Finally, a non-empty bipartite graph $G = (U, V, E)$ is called square if $|U| = |V|$. By the size of a square bipartite graph, we mean the cardinality of its vertex classes, i.e., $|U|$ (or $|V|$).
Now consider again the following system of equations:

\[
\begin{align*}
    u_1(v_1, v_2, v_4) &= 0 \\
    u_2(v_1, v_2, v_3, v_5) &= 0 \\
    u_3(v_3, v_4, v_5) &= 0 \\
    u_4(v_3, v_4, v_5) &= 0
\end{align*}
\]

We start by constructing an incidence (or occurrence) matrix for this system of equations. The rows of this matrix correspond to the equations and the columns correspond to the variables. A 1-element at position \((i, j)\) in the matrix indicates that variable \(v_j\) occurs in equation \(u_i\). This matrix simply captures the structural dependence between variables and equations while disregarding the other information from the equations. The incidence matrix corresponding to our example system of equations is shown in Figure 1.

\[
\begin{bmatrix}
    1 & 1 & 0 & 1 & 0 \\
    1 & 1 & 1 & 0 & 1 \\
    0 & 0 & 1 & 1 & 1 \\
    0 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

Figure 1: Incidence (biadjacency) matrix of our example system

By interpreting the incidence matrix as the biadjacency matrix of a bipartite graph, we can create such a graph with as vertex classes the equations \(U\) and variables \(V\) of this system. There is an edge between vertices \(v_j\) and \(u_i\) iff variable \(v_j\) occurs in equation \(u_i\). For our example system of equations, this yields the graph in Figure 2. The example decomposition obtained by first assigning a value to \(v_5\) and subsequently solving two smaller subsystems in the graph is shown in Figure 3.

If we have two subsystems \(A\) and \(B\) and we want to solve \(B\) after solving \(A\), the variables in \(B\) may not occur in the equations of \(A\). As the subsystems in our example graph are solved in order from the right to the left, this implies there can be no diagonal lines from the variables in a given subsystem to the equations of other subsystems to the right of it. This structure is immediately obvious from the representation of our example graph.

We also note that a (sub)system can only be solved if its associated bipartite (sub)graph has a (maximum) matching covering all of the equations. For a well-defined system (one with an equal number of equations and variables) this implies the bipartite graph must have a perfect matching. We can now restate our assumption of (structural) consistency as requiring the existence of a matching covering all of the equations in the associated bipartite graph of the original system.

Figure 2: Bipartite graph corresponding to system of equations
Figure 3: Decomposition into subsystems

Figure 4: Dulmage-Mendelsohn decomposition

The bipartite graph representation of a system of equations can be used as a starting point for obtaining a decomposition. We will first describe work by Dulmage and Mendelsohn on a canonical decomposition of bipartite graphs and how it relates to systems of equations. The main subject of this paper will then follow in a natural way.

The Dulmage and Mendelsohn decomposition of a bipartite graph $G = (U, V, E)$ is constructed as follows: first partition the vertices $U \cup V$ into three (possibly empty) sets: $D$ contains the vertices that are each not matched by at least one maximum matching. $A$ consists of all the neighbors of the vertices in $D$ that are not in $D$ themselves, i.e., $A = \Gamma(D) - D$. And finally $C$ contains all vertices not in $D$ or $A$, i.e., $C = (U \cup V) - (A \cup D)$. Using these three sets, we can construct three (possibly empty) induced subgraphs of $G$:

$$G_1 = G[C]$$
$$G_2 = G[U_D \cup V_A]$$
$$G_3 = G[V_D \cup U_A]$$

An example of this construction is shown in Figure 4. This decomposition was originally described by Dulmage and Mendelsohn in terms of vertex covers [5], but is also often described based on the dual concept of matchings (see e.g. [3]). If the bipartite graph represents a system of equations, the three subgraphs $G_1$, $G_2$ and $G_3$ represent respectively the well-constrained, over-constrained and under-constrained parts of the system [2].

Based on the coarse decomposition of the bipartite graph into these three (possibly empty) parts, Dulmage and Mendelsohn also describe how a fine decomposition of the part $G_1$ can be obtained in polynomial time. To this end, a perfect matching of $G_1$ is considered (such a matching always exists) and $G_1$ is turned into a directed bipartite graph by orienting all edges from $V$ to $U$ and adding additional edges from $U$ to $V$ corresponding to the edges in the perfect matching. The strongly connected components in this construction constitute the
fine decomposition of $G_1$. Dulmage and Mendelsohn have shown that this fine decomposition is in fact unique and as such does not depend on the perfect matching used to construct it. An example is shown in Figure 5.

If the bipartite graph represents a system of equations, this fine decomposition of $G_1$ corresponds to a decomposition of the well-constrained part of the system into minimal subsystems that can be solved separately in order [2]. Besides bipartite graphs, both the coarse and fine decomposition have also been investigated in terms of matrices [6, 7].

In this paper, we will assume $G_2$ to be empty, such that our system of equations is consistent and has no over-constrained part. The natural question that thus presents itself is whether we can also decompose $G_3$ into subsystems that we can solve separately. It turns out that it is possible to do so, however, this decomposition is no longer unique and depends on the maximum matching used to obtain it. Figures 6 and 7 show an example of this: the matching used in Figure 6 leads to a decomposition into more and smaller subsystems than that of Figure 7.

As a decomposition into smaller blocks is likely to simplify the work of an algorithmic solver, we investigate the complexity of finding an optimal decomposition, i.e., one in which the largest
subsystem is as small as possible. This problem is not new: it has been studied before for
example by Bliek et al.[8]. However, as far as we know, no investigation of its parameterized
complexity has been undertaken before. Even though this problem is only relevant for the
decomposition of $G_3$, we will usually simply consider the more general case of a bipartite graph
$G = (U, V, E)$ with a maximum matching covering $U$. In the remainder of this paper, $G_1$, $G_2$
and $G_3$ will be used to denote other (sub)graphs of $G$ and no longer necessarily correspond to
the parts of the coarse Dulmage-Mendelsohn decomposition.

The decision problem **Bounded Block Decomposition**, related to finding a ‘good’ de-
composition of a system, can be formulated as follows:

**Instance:** A bipartite graph $G = (U, V, E)$, an integer $k$

**Question:** Is there a partition of $U = U_1 \cup U_2 \cdots \cup U_n$ and $V = V_1 \cup V_2 \cdots \cup V_n \cup V_{n+1}$
such that for each $1 \leq i \leq n$, $G_i = G[V_i, U_i]$ is a bipartite graph with a perfect matching,
$\Gamma(V_i) \subseteq \bigcup_{j=1}^i U_j$ and $|U_i| = |V_i| \leq k$?

Note that for any yes-instance of this decision problem, we must have $|U| \leq |V|$ and $G$ contains a matching covering all of $U$. Furthermore, if $G$ is not connected then we may analyze its components separately, so without loss of generality we will assume $G$ is connected. For bipartite graphs $G = (U, V, E)$ with $|U| = |V|$ containing a perfect matching, the solution to the **Bounded Block Decomposition** problem is equal to the fine Dulmage-Mendelsohn decomposition, so **Bounded Block Decomposition** on such graphs is decidable in polynomial
time. Unfortunately, for $|U| < |V|$ we will see that **Bounded Block Decomposition** is in
general a harder problem.

The subsequent section describes some special properties of the part $G_0$, as well as the
relation between this part, the entire decomposition and how it can be used to solve the corre-
sponding system of equations.

3 The Free Square Block Problem

In this section, the concept of a **free square block** is introduced, together with several useful
mathematical properties. We describe the **OPENPLAN** algorithm to construct an optimal de-
composition based on this concept. The section ends with two equivalent problems regarding
hypergraphs and systems of distinct representatives.

A **free square block** $A$ is a non-empty induced subgraph of a bipartite graph $G = (U, V, E)$
such that $|U_A| = |V_A|$, and $\Gamma_G(V_A) \subseteq U_A$. Translated back to the application of systems of
equations, the last requirement states that no variable in $A$ may occur in an equation which is
not part of $A$, so $A$ can be solved after solving the remainder of the system. Figure 8 shows one
of the free square blocks in the example graph of Figure 2.

We proceed by proving several useful properties of free square blocks. In what follows, we
will use two famous theorems regarding matchings in bipartite graphs. The reader is referred
to [3] and [4] for a more in-depth treatment of these theorems, as well as their proofs.

**Theorem 3.1** (König’s Minimax Theorem). In a bipartite graph, the cardinality of a maximum
matching is equal to the cardinality of a minimum vertex cover.

**Theorem 3.2** (P. Hall’s Theorem). Let $G = (U, V, E)$ be a bipartite graph. Then $G$ has a
matching of $V$ into $U$ if and only if $|\Gamma(X)| \geq |X|$ for all $X \subseteq V$.

Using these theorems, we can establish several useful properties of free square blocks.
Theorem 3.3. Let $G = (U, V, E)$ be a connected bipartite graph with $1 \leq |U| \leq |V|$. There exists a non-empty induced subgraph $A \subseteq G$ with a perfect matching, such that $\Gamma_G(V_A) \subseteq U_A$. I.e., $G$ contains a free square block with a perfect matching.

Proof. Consider a minimum vertex cover $C$ of $G$ with $U_C \neq \emptyset$ (the connectedness combined with $|U| \leq |V|$ guarantees such a vertex cover to exist) and a maximum matching $M$ (of equal cardinality by Theorem 3.1). Let $U_A = U_C$ and assume $U_A$ is matched to $V_A \subseteq V$. Now construct the induced subgraph $A = G[U_A \cup V_A]$. By construction, $A$ has a perfect matching. Furthermore, as $C$ is a vertex cover, we must have $\Gamma_G(V_A) \subseteq U_A$.

A minimal free square block $B \subseteq G$ is a free square block of $G$ that contains no smaller free square block.

Corollary 3.4. A minimal free square block has a perfect matching.

Proof. Assume to the contrary that $A$ is a minimal free square block of $G = (U, V, E)$ that contains no perfect matching. Then by Theorem 3.2 there is a strict subset $V' \subset V_A$ such that $|\Gamma(V')| < |V'|$. However, in that case we know from Theorem 3.3 that $G[\Gamma(V') \cup V']$ contains a smaller free square block that is necessarily also a free square block of $G$, contradicting the minimality of $A$.

Minimal free square blocks are of interest for solving systems of equations: They correspond to subsystems that cannot be decomposed further, can be solved after solving the rest of the system, and are consistent due to the existence of a perfect matching. In [8], Blick et al. describe their OpenPlan algorithm to decompose systems of equations using free square blocks. This algorithm finds a smallest (w.r.t. the number of vertices) free square block in the bipartite graph representation of a system of equations and marks it as a subsystem that can be solved last. By iteratively applying this procedure until only variables are left, the algorithm comes up with an optimal decomposition, i.e., one in which the size of the largest subsystem is as small as possible. As finding the smallest free square block in a graph forms the core of this algorithm, we decided to further investigate the tractability of this problem, as well as that of the decomposition problem itself. In the analysis of the following sections, we consider the following natural parameterization of the decision problem regarding the existence of free square blocks of a given size, the Free Square Block problem:

**Instance:** A bipartite graph $G = (U, V, E)$, a positive integer $k$

**Parameter:** $k$

**Question:** Does $G$ contain a free square block of size $k$?
The parameterized complexity of finding the minimum free square block is analyzed in Section 4. To conclude this section, we present two alternative formulations of this problem. The following problem regarding hypergraphs is equivalent to the Free Square Block problem:

**Instance:** A hypergraph $H = (V, E)$, a positive integer $k$

**Parameter:** $k$

**Question:** Is there a subgraph $H' \subseteq H$ with $|V(H')| = |E(H')| = k$?

Another formulation uses the notion of a system of distinct representatives. Let $F = (S_1, \ldots, S_n)$ be a family of subsets of a finite set $S$, a sequence $F = (f_1, \ldots, f_n)$ is called a system of distinct representatives, or SDR, if all elements of $F$ are distinct, and $f_i \in S_i$ for $i = 1, 2, \ldots, n$. In this context, the Free Square Block problem is equivalent to the following decision problem:

**Instance:** A set $S$, a family $F$ of subsets of $S$, a positive integer $k$

**Parameter:** $k$

**Question:** Is there a subset $S' \subseteq S$ and a subset $F' \subseteq F$, such that $|S'| = |F'| = k$ and $\bigcup_{F \in F'} F = S'$ and $F'$ has an SDR with respect to $S'$?

### 4 Free Square Block is $W[1]$-complete

In this section we study the Free Square Block problem to establish its parameterized complexity. The main results are two proofs by reduction that together establish the $W[1]$-completeness of the problem. The Dulmage-Mendelsohn decomposition and related problems have been studied before from a parameterized complexity point of view, for example in the context of variations on vertex cover problems (see e.g. [9, 10]), but the parameterized approach to the specific problems we study seems to be new.

Bliek et al. note in [8] that the smallest free block problem is expected to be NP-hard. In [11], the problem is stated to be NP-hard as being the ‘dual’ of the minimum dense problem (see e.g. [12]), however, this duality is not immediately obvious. Furthermore, NP-completeness is not always the end of the line, as parameterized versions of (decision) problems can sometimes be solved efficiently even though their non-parameterized versions are NP-hard. A nice example of this is given in the introductory chapter of [13] that discusses the (minimum) Vertex Cover problem which is known to be NP-complete and its parameterized version $k$-Vertex Cover that asks if a vertex cover of size $k$ exists. The latter version is fixed parameter tractable, i.e., can be solved in time $O(f(k)\text{poly}(n))$. So the question in our case is: is there an efficient parameterized algorithm to find a small minimal free square block of parameterized (maximum) size? In this section, we show the Free Square Block problem is complete for the $W[1]$ class of decision problems. The proof of $W[1]$-hardness is based on a reduction from $k$-Clique and also shows NP-hardness.

#### Theorem 4.1. Free Square Block is $W[1]$-hard

*Proof.* The proof of $W[1]$-hardness is accomplished by showing how an arbitrary instance $(G, k)$ of the $W[1]$-hard problem $k$-Clique (see e.g. [14]) can be converted in polynomial time into an instance $(G', k')$ of Free Square Block in such a way that the latter is a yes-instance of Free Square Block if and only if the former is a yes-instance of $k$-Clique (a uniform reduction in the sense of [15]). We only prove this for odd values of $k$; any instance of $k$-Clique
Figure 9: Example graph containing a 5-clique

Figure 10: Free square block of size 10 corresponding to a 5-clique

with $k$ even can easily be converted into an equivalent instance of $(k + 1)$-CLIQUE by simply adding one extra vertex and connecting it to all the other vertices.

Let $(G, k)$ with $G = (V, E)$ be an instance of $k$-CLIQUE with $k$ odd and construct a bipartite graph $G' = (U', V', E')$ as follows: Let $V'$ contain one vertex for each of the edges in $E$. Let $U'$ contain $\frac{k-1}{2}$ copies of each of the vertices in $V$. And let $E'$ contain an edge between $u' \in U'$ and $v' \in V'$ if and only if the edge of $G$ corresponding to $v'$ is incident to the vertex corresponding to $u'$. (As each of the vertices of $G$ is duplicated $\frac{k-1}{2}$ times, this means every $v' \in V'$ has degree $k - 1$.)

Free square blocks of $G'$ correspond to subgraphs of $G$ that contain $\frac{k-1}{2}$ times as many edges than vertices. The smallest (in terms of vertices) possible subgraph of $G$ with this ratio is a $k$-clique, so $G'$ contains a free square block of size $k' = \frac{k(k-1)}{2}$ if and only if $G$ contains a clique of size $k$; smaller free square blocks of $G'$ can never exist as $G$ cannot contain smaller subgraphs with this ratio.

We have thus created an instance $(G', k')$ of FREE SQUARE BLOCK that is a yes-instance if and only if the original $(G, k)$ formed a yes-instance of $k$-CLIQUE, proving FREE SQUARE BLOCK to be $W[1]$-hard.

As a free square block of size $k'$ in the constructed instance of the FREE SQUARE BLOCK problem has to be minimum if it exists, this also shows that the MINIMUM FREE SQUARE BLOCK problem determining if a bipartite graph contains a minimum free square block of size $k$ is $W[1]$-hard.

As an example of the construction of $G'$ in this proof, consider the graph $G$ shown in Figure 9. Clearly $G$ contains a 5-clique. We now construct the corresponding bipartite graph $G'$ according to the procedure outlined in the proof of Theorem 4.1 as shown in Figure 10 (every vertex in the original graph is duplicated). The free square block corresponding to the 5-clique is clearly recognizable in the bipartite graph $G'$.

By Corollary 3.4 we know any free square block of size $k'$ in $G'$ has a perfect matching. This can also be seen by observing the following: Consider an Euler walk $v_1 e_1 v_2 e_2 \ldots v_1$ in a $k$-clique.
on an odd number of points. Such a tour contains every $e_i$ exactly once and every $v_i$ exactly $\frac{k-1}{2}$ times. The edges in $G'$ corresponding to $v_1e_1, v_2e_2$ etc. together lead to a perfect matching in the free square block of $G'$ corresponding to the $k$-clique.

After establishing the $\text{W}[1]$-hardness of the $k$-FREE SQUARE BLOCK problem, in essence providing a ‘lower bound’ on its difficulty, we now proceed to show the strictness of this classification by proving it is also a member of $\text{W}[1]$. For the proof, we will use a reduction to the parameterized decision problem $t$-THRESHOLD STABLE SET known to be $\text{W}[1]$-complete (see [13]):

**Instance:** A directed graph $G = (V, A)$, a positive integer $k$

**Parameter:** $k$

**Question:** Does $G$ have a $t$-threshold stable set of size $k$? (A stable set is a set of vertices $S \subseteq V$ such that, with some fixed $t$, for every vertex $v$ of $V - S$, there are fewer than $t$ vertices $u \in S$ with $uv \in A$.)

For our purposes, we will only use $t = 1$, effectively reducing the problem to the following:

**Instance:** A directed graph $G = (V, A)$, a positive integer $k$

**Parameter:** $k$

**Question:** Is there a subset $S \subseteq V$ of size $k$ such that $\Gamma(S) \subseteq S$?

A useful property of a 1-THRESHOLD STABLE SET $S$ is that for a strongly connected component $C \subseteq G$ we have either $C \subseteq G[S]$ or $C \cap S = \emptyset$. Using this, we can prove the following theorem:

**Theorem 4.2.** $k$-FREE SQUARE BLOCK is $\text{W}[1]$-complete.

**Proof.** We construct a uniform reduction from $k$-FREE SQUARE BLOCK to 1-THRESHOLD STABLE SET. Let $p$ and $q$ be two distinct prime numbers each greater than $k$. Given an instance $G = (U, V, E)$ of $k$-FREE SQUARE BLOCK, we construct a directed graph $G'$ that is an instance of 1-THRESHOLD STABLE SET( with parameter value $k'$) as follows: First direct all edges from $V$ to $U$. Then replace each vertex of $U$ by a strongly connected component on $p$ vertices, for example a $p$-cycle. And replace each vertex of $V$ by a strongly connected component on $q$ vertices (e.g., a $q$-cycle). The directed graph $G'$ that is obtained contains a stable set of size $k' = kp + kq$ if and only if $G$ contains a free square block of size $k$. This can be verified as follows: A free square block $A = (U', V', E') \subseteq G$ of size $k$ has $|U'| = |V'| = k$. The union of $U', V'$ and all of the vertices in their strongly connected components form a stable set of size $kp + kq$ as there are no outgoing arrows from this set to the rest of $G'$. Conversely, if we can find a stable set $S$ of size $kp + kq$ in $G'$, then $|S \cap U| = |S \cap V| = k$ and there are no outgoing arrows from $S \cap V$ to $U - (S \cap U)$, showing that $G[S \cup (U \cup V)]$ is in effect a free square block of size $k$. $\square$

Figure 11 shows an example of this construction for a free square block of size $k = 2$ in our example graph from Figure 2.

Finally, we will show that requiring $G$ to have a (maximum) matching covering $U$, as is likely in consistent systems of equations, does not make the problem easier. To this end, consider a graph $G' = (U \cup U', V \cup V', E \cup E')$ where $V'$ contains $|U| + 1$ additional vertices, $U'$ contains a single new vertex, and $E' = \{(uv | u \in U \cup U', v \in V')\}$, i.e., we add $|U| + 1$ vertices to $V$ and
connect each of them to all vertices in $U$ as well as to a new vertex. By this construction, $G'$ contains $K_{|U|+1,|U|+1}$ (a completely connected bipartite graph with $|U|+1$ vertices in each of its classes) as a subgraph, so $G'$ also contains a perfect matching covering $U \cup U'$. However, this construction does not add any new free square blocks of size smaller than $|U|+1$. Passing from $G$ to $G'$ if necessary shows that the above decision problem remains $W[1]$-complete if restricted to instances where $U$ is covered by a maximum matching.

5 Bounded Block Decomposition is $W[1]$-hard

We consider the natural parameterization of the Bounded Block Decomposition problem:

**Instance:** A bipartite graph $G = (U, V, E)$, an integer $k$

**Parameter:** $k$

**Question:** Is there a partition of $U = U_1 \cup U_2 \cdots \cup U_n$ and $V = V_1 \cup V_2 \cdots \cup V_n \cup V_{n+1}$ such that for each $1 \leq i \leq n$, $G_{i} = G[V_i, U_i]$ is a bipartite graph with a perfect matching, $\Gamma(V_i) \subseteq \bigcup_{j=1}^{n} U_j$ and $|U_i| = |V_i| \leq k$?

**Theorem 5.1.** Bounded Block Decomposition is $W[1]$-hard.

For the proof of this theorem, we first require a construction procedure. Given a graph $G = (V, E)$ and an odd $k$ with $k > 1$ consider the graph $\tilde{G} = (\tilde{V}, \tilde{E})$ consisting of $\frac{k-1}{2}$ copies $(G^{(1)}, G^{(2)}, \ldots, G^{(\frac{k-1}{2})})$ of $G$ with edges between all pairs of vertices $u^{(i)}$ and $v^{(j)}$ iff $uv \in E$. So a vertex $v^{(i)}$ is adjacent to its neighbors in $G^{(i)}$ as well as to all the copies of its neighbors (see also Figure 12). Clearly $\Gamma(v^{(i)}) = \Gamma(v^{(j)})$ holds for any two copies $v^{(i)}$ and $v^{(j)}$ of the same vertex $v \in V$.

We now proceed with the proof of Theorem 5.1.

**Proof of Theorem 5.1.** The proof consists of a reduction from $k$-CLIQUE. Let $G = (V, E)$ and $k$ be an instance of $k$-CLIQUE. To avoid a few corner-cases in the reduction, we assume $G$ is connected, $|E| \geq |V|$, $k > 1$ and $k$ is odd. We first construct the graph $\tilde{G} = (\tilde{V}, \tilde{E})$ using $G$ and $k$. This construction can clearly be performed in polynomial time. First, we claim that $\tilde{G}$ contains a clique of size $k$ iff $G$ contains a clique of size $k$. Clearly, any clique of $G$ has a corresponding copy in $\tilde{G}$. Conversely, if $\tilde{S} \subset \tilde{V}$ induces a clique in $\tilde{G}$, then at most a single copy of any vertex $v \in V$ can be in $\tilde{S}$. Replacing every vertex $v^{(i)} \in \tilde{S}$ by $v^{(1)}$ (its copy in $G^{(1)}$), we obtain a clique in $G^{(1)}$, which corresponds to a clique in $G$. We have thus shown that $(\tilde{G}, k)$ and $(G, k)$ are equivalent as instances of $k$-CLIQUE. Using $\tilde{G}$ we now construct a
bipartite graph $G' = (U', V', E')$ as in the proof of Theorem 4.1: $U'$ contains $\frac{k-1}{2}$ copies of each vertex in $\bar{V}$, $V'$ is equal to $\bar{E}$, and $E'$ contains an edge between $u' \in U'$ and $v' \in V'$ iff the edge corresponding to $v'$ in $\bar{G}$ is incident to the vertex in $\bar{G}$ that $u'$ corresponds to. Free square blocks of $G'$ correspond to subgraphs of $\bar{G}$ that contain $\frac{k-1}{2}$ times more edges than vertices. The smallest (in terms of vertices) possible subgraph of $\bar{G}$ with this ratio is a $k$-CLIQUE, so $G'$ contains a free square block of size $\binom{k}{2}$ if and only if $\bar{G}$ contains a clique of size $k$; smaller free square blocks of $G'$ can never exist as $\bar{G}$ cannot contain smaller subgraphs with this ratio. So if $G'$ has a BOUNDED BLOCK DECOMPOSITION with block size bounded by $\binom{k}{2}$, then the first free square block in this decomposition must correspond to a $k$-clique in $\bar{G}$ and thus to a $k$-clique in $G$. For the converse, assume $G$, and thus $\bar{G}$, contains a $k$-clique; pick one such clique. By construction, $G'$ contains $\frac{k-1}{2}$ disjoint free square blocks of size $\binom{k}{2}$ corresponding to this clique in $\bar{G}$. After removing these blocks from $G'$, the remainder of $G'$ can be decomposed into free square blocks of size $\frac{k-1}{2}$ as follows: pick a vertex $v^{(1)} \in \bar{V}$ that is not yet part of the decomposition and has a neighbor $w^{(1)} \in \bar{V}$ that is already part of the decomposition. The $\frac{k-1}{2}$ copies of $v^{(1)}$ in $U'$ and the vertices in $V'$ corresponding to the $\frac{k-1}{2}$ edges connecting $v^{(1)}$ to the copies $w^{(1)}$ of $w^{(1)}$ in $\bar{G}$ together form a free square block of size $\frac{k-1}{2}$ in the remainder of $G'$. Remove this free square block from $G'$ and repeat this operation for $v^{(2)} \ldots v^{(\frac{k-1}{2})}$ in $\bar{G}$. Keep constructing free square blocks in this way until all vertices in $U'$ are part of a decomposition. Due to the connectedness of $\bar{G}$, we can keep picking vertices to induce the next block until the decomposition is complete. All blocks in this decomposition have a size bounded by $\binom{k}{2}$. By transforming an instance $(G, k)$ of $k$-CLIQUE to a corresponding instance $(G', \binom{k}{2})$ of BOUNDED BLOCK DECOMPOSITION, we have shown a uniform reduction from $k$-CLIQUE to BOUNDED BLOCK DECOMPOSITION, proving that BOUNDED BLOCK DECOMPOSITION is $W[1]$-hard.

6 Relation to Crowns

It was pointed out to us by a reviewer of the first draft of this paper that free square blocks seem closely related to a special case of crown structures, a reduction mechanism for kerneliza-
tion of the parameterized \textsc{Vertex Cover} problem \cite{16, 17}. In this section we briefly discuss crown structures and their relation to the free square block problems we have considered in the preceding sections. After showing the relation between the two, we use our results on free square blocks to settle two open questions regarding crowns.

We start by giving a few definitions, adapted from \cite{18}: A \textit{crown} is an ordered pair \((I, H)\) of subsets of vertices from a graph \(G\) that satisfies the following criteria: (1) \(I \neq \emptyset\) is an independent set of \(G\), (2) \(H = \Gamma(I)\), and (3) there exists a matching \(M\) on the edges connecting \(I\) and \(H\) such that all elements of \(H\) are matched. This implies \(|I| \geq |H|\). \(H\) is called the head of the crown. An example of a crown is shown in Figure 13. A crown \((I, H)\) is called a \textit{straight crown} if it satisfies the condition \(|I| = |H|\), otherwise it is called a \textit{flared crown}. A crown that is a subgraph of another crown is called a \textit{subcrown}. The size or order of a crown is the number of vertices in \(I \cup H\). (N.B. this definition of the size of a crown is different from our previous definition of the size of a square block!)

Crowns are used to reduce the size of a problem instance of the \textsc{Vertex Cover} problem by exploiting the fact that if \(G\) is a graph with a crown \((I, H)\), then there is a vertex cover of \(G\) of minimum size that contains all the vertices in \(H\) and none of the vertices in \(I\) \cite{18}. By applying this reduction rule, a smaller instance of \textsc{Vertex Cover} can be solved instead. It has been shown that finding a non-trivial crown in a graph \(G\) can be done in polynomial time. Finding a crown of maximum order is also polynomially solvable \cite{18}.

The remainder of this section is dedicated to establishing the \(W[1]\)-hardness of two parameterized decision problems related to crowns. The first problem we consider is the natural parameterization of the \textsc{Sized Crown} problem previously proven to be \textsc{NP}-complete by Sloper \cite{19}. This decision problem involves determining if a graph contains a crown of a certain size.

\textbf{Instance:} A Graph \(G = (V, E)\), a positive integer \(k\)

\textbf{Parameter:} \(k\)

\textbf{Question:} Does \(G\) contain a crown \((I, H)\) with \(|I \cup H| = k|\)?

The second problem we consider is the parameterized decision problem \textsc{Minimum Crown}, regarding the identification of a crown of minimum order \cite{18}.

\textbf{Instance:} A Graph \(G = (V, E)\), a positive integer \(k\)
Parameter: $k$

Question: Does $G$ contain a minimum crown $(I, H)$ with $|I \cup H| = k$?

The parameterized complexity of these problems is mentioned as an open problem by respectively Sloper [19] and Abu-Khzam et al. [18] and to our knowledge these problems have not been solved before.

To facilitate our discussion, we define a few more terms: We call a crown a minimal crown if it contains no smaller subcrown. A crown with the minimum number of vertices over all crowns is called a minimum crown. The following useful lemma from [18] enables us to consider only straight crowns if we search for crowns of minimum order as it implies that minimal crowns have to be straight.

Lemma 6.1 ([18]). If $(I, H)$ is a flared crown then there is another crown $(I', H)$ that is straight and $I' \subset I$.

We start our analysis by establishing a few additional characteristics of crowns in bipartite graphs and their relationship to free square blocks.

Lemma 6.2. If $(I, H)$ is a minimal crown of a bipartite graph $G = (U, V, E)$ then either $H \subseteq U$ or $H \subseteq V$.

Proof. Assume to the contrary that $H$ contains vertices from both $U$ and $V$. Due to the existence of a perfect matching between $I$ and $H$, we know that $I$ also contains vertices from both $U$ and $V$. Now let $H_U = H \cap U$ and $I_V = I \cap V$. Clearly, $I_V$ is an independent set of $G$, $H_U = \Gamma(I_V)$, and there exists a perfect matching between $H_U$ and $I_V$. So $(I_V, H_U)$ is a strict subcrown of $(I, H)$ contradicting the minimality of $(I, H)$.

Lemma 6.3. The following properties define the relation between minimal crowns and free square blocks in a bipartite graph $G = (U, V, E)$:

1. A minimal free square block $A$ corresponds to a straight crown $(V_A, U_A)$.
2. A minimal straight crown $(V_A, U_A)$ induces a free square block $A$ of $G$.
3. The number of vertices in a minimum free square block is equal to the number of vertices of the smallest straight crown $(I, H)$ with $H \subseteq U$.

Proof. 1. As $G$ is a bipartite graph, $(1)$ $V_A$ is an independent set. $(2)$ $U_A = \Gamma(V_A)$ and as a minimal free square block has a perfect matching, we have that $(3)$ there exists a matching $M$ on the edges connecting $V_A$ and $U_A$ such that all elements of $U_A$ are matched.

2. A straight crown by definition has $|V_A| = |U_A|$ and $U_A = \Gamma(V_A)$, so $A = G[V_A \cup U_A]$ is a free square block of $G$.

3. Immediate from (1) and (2).

We now come to our main result on crowns.

Proof. Let \( G = (U, V, E) \) and \( k \) be an instance of the Minimum Free Square Block problem and construct a new bipartite graph \( G' \) as follows: Let \( K_{k+1,k+1} = (U^*, V^*, E^*) \) be a complete bipartite graph with \( k + 1 \) vertices in each of its vertex classes. Construct \( G' = (U', V', E') \) as \( U' = U \cup U^*, V' = V \cup V^* \) and \( E' = E \cup E^* \cup \{uv \mid u \in U, v \in V^* \} \). I.e., \( G' \) consists of \( G \) and \( K_{k+1,k+1} \) and an edge between every pair \((u, v)\) with \( u \in U \) and \( v \in V^* \).

Clearly, any crown \((I, H)\) of \( G' \) with \( I \cap (U^* \cup V^*) \neq \emptyset \) must have \( |\Gamma(I)| = |H| \geq k + 1 \). Furthermore, due to its construction, any minimal crown \((I, H)\) in \( G' \) with \( H \subseteq U' \) must have \( V^* \subseteq I \) and thus \( |I| \geq k + 1 \). So this construction introduces no new crowns of size \( k \) or less in \( G' \).

\( G' \) contains a minimum crown of size \( k \) if and only if \( G \) contains a minimum crown of size \( 2k \) with its head in \( U \). Such crowns correspond exactly to minimum free square blocks of \( G \). As the Minimum Free Square Block problem is \( W[1] \)-hard, and the above construction is a uniform reduction in the sense of [15], Minimum Crown is also \( W[1] \)-hard.

As an immediate consequence Sized Crown is \( W[1] \)-hard as well.

7 Conclusion

Our results show that the natural parameterization of the problem of finding a free square block of either a given size or of minimum size is not fixed parameter tractable under the working hypothesis that \( FPT \neq W[1] \) (see [20]). The same holds for finding a decomposition where the size of the largest block is as small as possible.

Based on our results on free square blocks, we have also been able to resolve two open problems regarding crown structures. Due to the relation between free square blocks and crown structures, a reduction from our Free Square Block problem shows finding both minimum crowns and crowns of a given size is \( W[1] \)-hard.

From a practical point of view, an interesting subject for further investigation would be heuristics to find small minimal free square blocks. Bliek et al. discuss one possible heuristic approach in [8], however the performance of this approach has to our knowledge not been analyzed extensively.

Another line of further investigation might lay in additional conditions on the system of equations. It might be possible to construct a good decomposition efficiently if more structural constraints can be placed on the system of equations. We think investigating the existence of such conditions and the corresponding algorithms for decomposition also warrants additional research.

A third direction for further research could be alternative or more extensive parameterizations of the decomposition problem: It may for example be interesting to also include the difference between the number of equations and the number of variables of the entire system as a parameter and see if that could be used to improve the tractability. (In the application of 3D scene reconstruction this would probably not always be of much help, as for example the system of equations mentioned in [1] contains 251 equations and 427 variables.) Another approach might be bounding the degree of the vertices by an additional parameter and trying to improve tractability that way. Further research in both directions is clearly required.

Acknowledgement

We wish to thank the reviewers of the first draft of this paper for their helpful comments and in particular for bringing crown structures to our attention.
Furthermore, we gratefully acknowledge the support of the Innovation-Oriented Research Programme ‘Integral Product Creation and Realization (IOP IPCR)’ of the Netherlands Ministry of Economic Affairs, Agriculture and Innovation.

References


