Parameterized Complexity of Sparse Linear Complementarity Problems

Hanna Sumita[1], Naonori Kakimura[2], and Kazuhisa Makino[3]

1   JST, ERATO, Kawarabayashi Large Graph Project,
    National Institute of Informatics
    Tokyo 101-8430, Japan
    sumita@nii.ac.jp
2   Graduate School of Arts and Sciences, University of Tokyo
    Tokyo 153-8902, Japan
    kakimura@global.c.u-tokyo.ac.jp
3   Research Institute for Mathematical Sciences, Kyoto University
    Kyoto 606-8502, Japan
    makino@kurims.kyoto-u.ac.jp

─── **Abstract** ───

In this paper, we study the parameterized complexity of the linear complementarity problem (LCP), which is one of the most fundamental mathematical optimization problems. The parameters we focus on are the sparsities of the input and the output of the LCP: the maximum numbers of nonzero entries per row/column in the coefficient matrix and the number of nonzero entries in a solution. Our main result is to present a fixed-parameter algorithm for the LCP with all the parameters. We also show that if we drop any of the three parameters, then the LCP is fixed-parameter intractable. In addition, we discuss the nonexistence of a polynomial kernel for the LCP.

1   Introduction

Given a square matrix $M \in \mathbb{R}^{n \times n}$ and a vector $q \in \mathbb{R}^n$, the *linear complementarity problem* (LCP) is to find a vector $z \in \mathbb{R}^n$ such that

$$Mz + q \geq 0, \quad z \geq 0, \quad z^\top(Mz + q) = 0. \tag{1}$$

We denote a problem instance of the LCP with M and q by $\mathrm{LCP}\,(M, q)$. We say that n is the *order* of $\mathrm{LCP}\,(M, q)$. The LCP, introduced by Cottle [9], Cottle and Dantzig [10], and Lemke [24], is one of the most widely studied mathematical programming problems, which, for example, contains linear and convex quadratic programming problems. The decision version of the LCP (i.e., deciding whether (1) has a solution z) is NP-complete [6]. For details of the LCP and related topics, see the books of Cottle, Pang, and Stone [11] and Murty [25].

In this paper, we analyze the parameterized complexity of the LCP. A problem with parameter k is said to be *fixed-parameter tractable* if there exists an algorithm which solves the problem in $f(k) \cdot L^{O(1)}$ time, where f is some computable function and L is the size of a

given instance. An algorithm with such running time is called a *fixed-parameter algorithm*. See e.g., [15, 17] for the detail of the parameterized complexity theory.

The parameterized complexity has been recently attracting attention in the field of mathematical programming, as optimization problems in the real world often have a certain parameterized structure. In particular, sparsities of the input and the output have been investigated, motivated from other fields such as computational biology [12] and coding theory [1]. In this paper, we focus on the sparsities of the input and the output of the LCP.

We consider as parameters the numbers of nonzero entries per row and column in an input matrix. We say that the LCP is *r-row-sparse* (resp. *c-column-sparse*) if the coefficient matrix has at most r nonzero entries per row (resp. at most c nonzero entries per column). The LCP is (r, c)-*sparse* if it is r-row-sparse and c-column-sparse. It is not difficult to see that any LCP instance can be reduced to a 3-row-sparse LCP instance (or a 3-column-sparse LCP instance), and thus it is NP-hard if $r = 3$ or $c = 3$. Moreover, even the 2-row-sparse LCP is NP-hard [27], while the 1-row-sparse LCP is linear-time solvable. Let us remark that this kind of sparsity has been studied for the special classes of the LCP, i.e., the system of linear inequalities [8, 20] and the bimatrix game [4, 5, 7, 13, 19]. The ℓ-*sparse* bimatrix game has payoff matrices each of whose rows and columns has at most ℓ nonzero entries, which can be described as the (ℓ, ℓ)-sparse LCP when a mixed Nash equilibrium has a positive expected payoff. It is also known that the mean payoff game falls into the $(3, c)$-sparse LCP [2], where c is the maximum indegree of a given graph plus two.

Another parameter discussed in this paper is the size of the *support* of a solution, that is, the number of nonzero entries of an output solution. For an integer s, the *s-support* LCP is to find a solution of (1) with support size at most s. Finding a solution with small support size is often useful in applications. For example, in the contact problem of rigid bodies (see e.g., [11]), which is formulated as the LCP, a solution is usually required to have a few nonzero entries when the bodies are in contact at few points on the surfaces. For the bimatrix game, which is also a special class of the LCP, a mixed Nash equilibrium with small support size has been studied recently [16, 19], as such a solution is close to a pure Nash equilibrium.

Our main result is to present a fixed-parameter algorithm for the *s*-support (r, c)-sparse LCP with parameters s, r, and c. The running time is $((r + c)rcs)^{\mathrm{O}(cs)} \cdot n^{\mathrm{O}(1)}$ time, where n is the order of a given instance. To obtain the algorithm, we incorporate the idea of Hermelin, Huang, Kratsch and Wahlström [19] for the bimatrix game. We first construct a graph associated with a given LCP instance, and derive a necessary condition on the graph that the support of a solution must satisfy. Our algorithm enumerates all index sets satisfying the necessary condition by traversing the graph. Since the number of candidate index sets is bounded by $((r + c)rcs)^{\mathrm{O}(cs)}$, the algorithm runs in fixed-parameter time.

On the other hand, we show that, if we drop any of the parameters r, c and s, then the LCP becomes fixed-parameter intractable. We first prove the $W[1]$-hardness of the *s*-support *r*-row-sparse LCP (resp., the *s*-support *c*-column-sparse LCP) with parameters s, r (resp., s, c). We remark that the *s*-support LCP with only parameter s is $W[2]$-hard by reducing the problem of finding a Nash equilibrium with support size at most s of the bimatrix game [16].

For the case when we drop the support size s of a solution, the $(2, 2)$-sparse LCP is shown to be NP-hard by extending an NP-hardness proof of the 2-row-sparse LCP [27]. We remark that the 1-row-sparse LCP (resp., the 1-column-sparse LCP) is solvable in linear time [27]. Thus it reveals the time complexity of the (r, c)-sparse LCP with respect to r and c.

Table 1 summarizes our results.

We also show the nonexistence of a polynomial kernel for the *s*-support (r, c)-sparse LCP unless coNP $\subseteq$ NP/poly. A *kernelization* algorithm is a polynomial-time transformation of a

■ **Table 1** The parametrized complexity of sparse LCPs.

Input \ Output	general	support size s
general	NP-hard [6]	$W[2]$-hard [16]
row-sparsity r	NP-hard ($r = 2$) [27]	**$W[1]$-hard** (Theorem 7)
column-sparsity c	**NP-hard ($c = 2$)** (Theorem 10)	**$W[1]$-hard** (Theorem 8)
r and c	**NP-hard ($r = c = 2$)** (Theorem 10)	$((r+c)rcs)^{\mathbf{O(cs)}} \cdot n^{\mathbf{O(1)}}$ (Theorem 6)

problem instance to an equivalent instance whose size and parameter depend only on the parameter of the original instance. The output of a kernelization algorithm is called a *kernel*, and a *polynomial kernel* if the size and the parameter are bounded by a polynomial in the original parameter. Recently, the existence of (polynomial) kernels has been investigated for mathematical programming problems such as a system of sparse linear equations [12] and integer linear programming problems [21, 22, 23]. While the s-support (r, c)-sparse LCP is fixed-parameter tractable, we show that a polynomial kernel does not exist by applying a general framework by Bodlaender, Downey, Fellows and Hermelin [3].

We remark that our hardness results are not implied directly by those on the sparse bimatrix game [4, 5, 19], since a natural reduction to the LCP (see e.g., [11]) destroys the sparsity.

We summarize the notation used in this paper. For a positive integer n, let $[n] = \{1, \ldots, n\}$. For a set $B \subseteq [n]$, we define $\overline{B} = [n] \setminus B$. Let M be an $m \times n$ real matrix, where M has a row index set $[m]$ and a column index set $[n]$. For $S \subseteq [m]$ and $T \subseteq [n]$, we denote by M_{ST} the submatrix of M such that S and T are row and column index sets, respectively. We also define $M_{\cdot T}$ by $M_{\cdot T} = M_{[m]T}$. If $S = \{i\}$, we simply write $M_{i\cdot}$, M_{iT} and M_{ij} instead of $M_{\{i\}[n]}$, $M_{\{i\}T}$ and $M_{\{i\}\{j\}}$, respectively. Let z be a vector in $\mathbb{R}^n$ with index set $[n]$. For index set $B \subseteq [n]$, let z_B denote the subvector of z with entries corresponding to B. For $i \in [n]$, we also denote by z_i the i-th entry of z.

The rest of the paper is organized as follows. Section 2 presents a fixed-parameter algorithm for the parameterized LCP. Section 3 shows the fixed-parameter intractability of the LCP. Section 4 discusses the nonexistence of a polynomial kernel. Some proofs are omitted due to the space limitation.

2 Fixed-parameter algorithm

In the section, we present a fixed-parameter algorithm for the s-support (r, c)-sparse LCP with parameter $s + r + c$. Let LCP (M, q) be an (r, c)-sparse LCP instance of order n.

▶ **Definition 1.** For a vector x, the *support* of x is the index set $\{i \mid x_i > 0\}$, denoted by S_x.

It is observed that a solution z of LCP (M, q) is a feasible solution to the system of linear inequalities

$$Mz + q \geq 0, \ z \geq 0, \ z_{\overline{S}} = 0, \ \text{and} \ (Mz + q)_S = 0. \tag{2}$$

Conversely, if we have a vector z satisfying (2), then z is a solution of LCP (M, q). Thus if we know the support S of some solution, we can find a solution z with $S_z \subseteq S$ in polynomial

time by solving the system of linear inequalities (2). Hence, enumerating all the index sets S of size at most s, we can find a solution of LCP (M, q) in $n^{O(s)}$ time. In the following, we derive a necessary condition to reduce the number of enumerated index sets.

Note that, if $s \cdot \max(r, c) \geq n$, then the above simple exponential-time algorithm becomes a fixed-parameter one. We may henceforth assume that $s \cdot \max(r, c) < n$.

We denote $T = \{i \mid q_i < 0\}$. It is clear that if $T = \emptyset$, i.e., $q \geq 0$, then $z = 0$ is a solution of LCP (M, q). Thus we may assume that T is nonempty. Moreover, it is observed that for each $i \in T$, any solution z of LCP (M, q) has a positive entry z_j with $M_{ij} > 0$, since $M_i.z \geq -q_i > 0$. This implies that the size of T is not large as follows.

▶ **Lemma 2.** *For any solution z of* LCP (M, q), $|T| \leq c|S_z|$ *holds.*

We define an undirected graph $G = (V, E)$ by $V = [n]$ and $E = \{(i, j) \mid i \neq j$ and $(M_{ij} \neq 0$ or $M_{ji} \neq 0)\}$. For a vertex $i \in V$, let $N_+(i) = \{j \mid j \neq i, \; M_{ij} \neq 0\}$, and $N_-(j) = \{i \mid i \neq j, \; M_{ij} \neq 0\}$. For a vertex set V', we denote $N_-(V') = \bigcup_{j \in V'} N_-(j)$, $N(V') = \bigcup_{i \in V'}(N_+(i) \cup N_-(i))$, and $E_-(V') = \bigcup_{j \in V'}\{(i, j) \mid i \in N_-(j)\}$.

▶ **Definition 3.** A solution z of LCP (M, q) is *minimal* if there exists no solution z' of LCP (M, q) with $S_{z'} \subsetneq S_z$.

A minimal solution has the following property.

▶ **Lemma 4.** *For any minimal solution z of* LCP (M, q), *each connected component of a subgraph $G' = (S_z \cup N_-(S_z), E_-(S_z))$ of G has a vertex in T.*

Proof. Let C be the vertex set of a connected component of G', and we show that $C \cap T$ is nonempty.

It is observed that $C \cap S_z$ is nonempty, since every edge in G' has an end vertex in S_z. Let $D = C \cap S_z$. By appropriately rearranging row and columns, the submatrix $M._{S_z}$ of M is of the form

$$
\begin{array}{c}
\\
S_z \setminus C \\
C \\
N_-(S_z) \setminus C \\
V \setminus (S_z \cup N_-(S_z))
\end{array}
\begin{array}{cc}
S_z \setminus C & D \\
\begin{bmatrix} * & O \\ O & * \\ * & O \\ O & O \end{bmatrix}
\end{array},
$$

where $*$ denotes a matrix of appropriate size, and O is a zero matrix.

To show $C \cap T \neq \emptyset$, suppose to the contrary, that is, $q_C \geq 0$. We define a nonnegative vector z' by setting $z'_D = 0$ and $z'_{\overline{D}} = z_{\overline{D}}$. Then it holds that

$$(Mz' + q)_C = M_{CD}z'_D + q_C = q_C \geq 0,$$

and $(Mz' + q)_{\overline{C}} = (Mz + q)_{\overline{C}} \geq 0$. Thus $Mz' + q \geq 0$ holds. Moreover, since $z'_D = 0$ and $z'_{V \setminus S_z} = 0$, we have

$$(z')^\top (Mz' + q) = (z'_{S_z \setminus C})^\top (Mz' + q)_{S_z \setminus C} = (z_{S_z \setminus C})^\top (Mz + q)_{S_z \setminus C} = 0.$$

Thus z' is a solution of LCP (M, q). However it holds that $S_{z'} \subsetneq S_z$, which contradicts the minimality of z. Therefore, we have $q_C \not\geq 0$, and thus $C \cap T \neq \emptyset$. ◀

Lemma 4 suggests to enumerate all index sets S of size at most s such that

each connected component of $(S \cup N_-(S), E_-(S))$ has a vertex in T, and $T \subseteq S \cup N_-(S)$. (3)

For the purpose, we first split T into two sets T_1 and T_2, and guess that $T_1 \subseteq S$ and $T_2 \subseteq N_-(S)$. Since $T_2 \subseteq N_-(S)$, each vertex in T_2 has to be connected to a vertex of S. We choose one index $p_i \in N_+(i)$ for each $i \in T_2$, and guess p_i's are contained in S. Thus at the beginning, $S_0 = T_1 \cup \{p_i \mid i \in T_2\}$ is supposed to be contained in S.

The algorithm traverses the graph G starting from $S = S_0$, and augment S keeping the condition (3). During the traversing procedure, each visited vertex is either active or inactive, where only active vertices can be chosen in the iteration.

Algorithm for the s-support (r, c)-sparse LCP

Input: A matrix M, a vector q and a positive integer s.

Step 0: Construct the undirected graph G as described before. Let $T = \{i \mid q_i < 0\}$, and let $\mathcal{S} = \emptyset$.
  If $T = \emptyset$, return a solution $z = 0$. If $|T| > cs$ or $\{j \mid M_{ij} > 0\} = \emptyset$ for some $i \in T$, return that LCP (M, q) has no solution z with $|S_z| \leq s$.

Step 1: Set $\mathcal{T}$ to be the family of all partitions (T_1, T_2) of T. For each $(T_1, T_2) \in \mathcal{T}$, do Step 2.

Step 2: Set $\mathcal{P}$ to be the family of all index sets of size at most $|T_2|$ having one index $p_i \in N_+(i)$ for each $i \in T_2$. For each $P \in \mathcal{P}$, let $S_0 = T_1 \cup P$, and do Step 3.

Step 3: Set each vertex in $S_0 \cup T_2$ to be active.
  While $|S| < s$ and there exists an active vertex, do the following steps.
  (i) take arbitrarily an active vertex i,
  (ii) if $i \notin S$, then either (a) set i to be inactive, or (b) visit a vertex $j \in N_+(i)$, make j active, and add j to S,
  (iii) if $i \in S$, then do exactly one of the following: (a) set i to be inactive, (b) visit a vertex $j \in N(i)$, make j active, and add j to S, or (c) visit a vertex $j \in N_-(i)$ and make j active,
  (iv) set $\mathcal{S} \leftarrow \mathcal{S} \cup \{S\}$ if S is updated.

Step 4: Check whether LCP (M, q) has a solution z with support S for some $S \in \mathcal{S}$. If exists, return the solution z. Otherwise, return that LCP (M, q) has no solution z with $|S_z| \leq s$.

The algorithm enumerates all the index sets satisfying (3) by the construction.

▶ **Claim 5.** $\mathcal{S}$ *contains all index sets satisfying* (3).

In the following, we show the running time of the algorithm. Note that for each vertex i of the graph G, we have $|N_+(i)| \leq r$ and $|N_-(i)| \leq c$ by (r, c)-sparsity.

It is not difficult to see that $|\mathcal{T}| = 2^{|T|}$ and $|\mathcal{P}| \leq r^{|T|}$. At any time in the algorithm, all active or inactive vertices are contained in $S \cup N_-(S)$. Thus in Step 3, there exist at most $(s + cs)$ active vertices, and, for each active vertex, there exist at most $(|N_+(i)| + |N_-(i)|) + |N_-(i)| \leq r + 2c$ choices of unvisited vertices in Step 3 (ii)-(iii). Thus each iteration has at most $(s + cs)(1 + r + 2c)$ branches. Moreover, the procedure is repeated at most $2(s + cs)$ times. Indeed, since the change of the state of vertices is irreversible, at most $s + cs$ vertices are set to be active, and some of them to be inactive. Therefore, the procedure generates at most $((s + cs)(1 + r + 2c))^{2(s+cs)}$ index sets. The existence of a solution of LCP (M, q) with support S can be decided in $n^{O(1)}$ time. Since we have $|T| \leq cs$ by 2, the running time of the algorithm is bounded by

$$2^{cs} \cdot r^{cs} \cdot ((s + cs) \cdot (1 + r + 2c)))^{2(s+cs)} \cdot n^{O(1)} \leq ((r + c)rcs)^{O(cs)} \cdot n^{O(1)}.$$

Then we have proved the following theorem.

▶ **Theorem 6.** *For any (r, c)-sparse LCP instance of order n, a solution with at most s nonzero entries can be found in $((r + c)rcs)^{O(cs)} \cdot n^{O(1)}$ time.*

3 Hardness results

In this section, we present hardness results for parameterized LCPs. First, we show the $W[1]$-hardness of the s-support r-row-sparse LCP with parameter $s + r$ by reducing the weighted 2SAT. Given a Boolean formula in conjunctive normal form ψ with at most two literals per clause and a positive integer s, the weighted 2SAT with parameter s is to decide whether there exists a satisfiable assignment in which at most s variables are set to be true. The problem is $W[1]$-complete with parameter s.

▶ **Theorem 7.** *The s-support 2-row-sparse LCP with parameter s is $W[1]$-hard.*

We also show the following theorem by reducing the s-subset sum problem with parameter s, which is $W[1]$-hard [14]. Given a set of integers $A = \{a_1, \ldots, a_n\}$, and integers t, s, the s-subset sum problem with parameter s is to decide whether there exists a set $S \subseteq [n]$ such that $|S| = s$ and $\sum_{i \in S} a_i = t$.

▶ **Theorem 8.** *The s-support 3-column-sparse LCP with parameter s is $W[1]$-hard.*

In the rest of this section, we show the NP-hardness of the $(2, 2)$-sparse LCP by reducing the *monotone one-in-three 3SAT*. Given a Boolean formula in conjunctive normal form $\psi = \bigwedge_{j=1}^{m}(x_{j_1} \vee x_{j_2} \vee x_{j_3})$ with three positive literals per clause, the monotone one-in-three 3SAT is to decide whether there exists an assignment such that each clause contains exactly one true literal.

This problem was introduced and proved to be NP-complete by Schaefer [26]. The monotone one-in-three 3SAT was also used to prove the NP-hardness of the 2-row-sparse LCP [27]. We obtain a stronger result by improving the proof with the following lemma.

▶ **Lemma 9.** *Let k be a positive integer, and consider an LCP instance*

$$
\begin{aligned}
w_j &= -z_{j+1} + 1 \geq 0, \qquad z_j \geq 0, \qquad w_j z_j = 0 \quad (j = 1, \ldots, k-1), \\
w_k &= -z_1 + 1 \geq 0, \qquad z_k \geq 0, \qquad w_k z_k = 0.
\end{aligned}
\tag{4}
$$

The LCP instance (4) has only two solutions: $z = 0$ and $z = 1$.

▶ **Theorem 10.** *The $(2, 2)$-sparse LCP is NP-hard.*

Proof. Let $\psi = \bigwedge_{j=1}^{m}(x_{j_1} \vee x_{j_2} \vee x_{j_3})$ be a monotone one-in-three 3SAT instance with n literals. We construct an instance of the $(2, 2)$-sparse LCP of order $3mn + 6m$ from ψ as follows: for each literal $i = 1, \ldots, n$, letting $\alpha_i = 3m(i - 1)$, define

$$
\begin{aligned}
w_{\alpha_i + j} &= -z_{\alpha_i + j + 1} + 1 \quad (j = 1, \ldots, 3m - 1), \\
w_{\alpha_i + 3m} &= -z_{\alpha_i + 1} + 1.
\end{aligned}
\tag{5}
$$

Moreover, for each clause $j = 1, \ldots, m$, letting $\beta_j = 3mn + 6(j-1)$ and $\gamma_\ell^j = \alpha_{j_\ell} + 3(j-1)$ ($\ell = 1, 2, 3$), set

$$
\begin{aligned}
w_{\beta_j + 1} &= -z_{\gamma_2^j + 1} - z_{\gamma_3^j + 1} + 1, \\
w_{\beta_j + 2} &= -z_{\gamma_1^j + 2} - z_{\gamma_3^j + 2} + 1, \\
w_{\beta_j + 3} &= -z_{\gamma_1^j + 3} - z_{\gamma_2^j + 3} + 1,
\end{aligned}
\tag{6}
$$

and in addition, set

$$
\begin{aligned}
w_{\beta_j+4} &= z_{\beta_j+1} + z_{\gamma_1^j+1} - 1, \\
w_{\beta_j+5} &= z_{\beta_j+2} + z_{\gamma_2^j+2} - 1, \\
w_{\beta_j+6} &= z_{\beta_j+3} + z_{\gamma_3^j+3} - 1.
\end{aligned}
\tag{7}
$$

We denote by $w = Mz + q$ the system of linear equations consisting of the above constraints (5), (6), and (7). Consider LCP (M, q), i.e., $w = Mz + q \geq 0$, $z \geq 0$, $z^\top w = 0$. Since each constraint has at most two entries of z and each entry of z appears in at most two constraints, this is a $(2, 2)$-sparse LCP instance.

We will show that LCP(M, q) has a solution if and only if the monotone one-in-three 3SAT instance ψ is satisfiable.

First assume that LCP(M, q) has a solution (w, z). For each $i = 1, \ldots, n$, by applying Lemma 9 to (5), it holds that either

$$
\begin{aligned}
&w_{\alpha_i+1} = \cdots = w_{\alpha_i+3m} = 0 \text{ and } z_{\alpha_i+1} = \cdots = z_{\alpha_i+3m} = 1, \text{ or} \\
&w_{\alpha_i+1} = \cdots = w_{\alpha_i+3m} = 1 \text{ and } z_{\alpha_i+1} = \cdots = z_{\alpha_i+3m} = 0.
\end{aligned}
\tag{8}
$$

Assign each literal x_i *true* if $z_{\alpha_i+1} = 1$, and *false* otherwise. We claim that x is a satisfiable assignment for ψ, that is, each clause has exactly one true literal. Indeed, for each clause j, if $z_{\gamma_1^j+1} = 0$, then $z_{\gamma_1^j+3} = 0$ by (8), and $z_{\beta_j+1} \geq 1 > 0$, $w_{\beta_j+1} = 0$ by (7) and the complementarity, and hence exactly one of $z_{\gamma_2^j+1}$ and $z_{\gamma_3^j+1}$ is equal to one by the first equation in (6). On the other hand, if $z_{\gamma_1^j+1} = 1$, then we have $z_{\gamma_1^j+2} = z_{\gamma_1^j+3} = 1$ by (8), and hence $z_{\gamma_3^j+3} = z_{\gamma_2^j+2} = 0$ by the second and third equations in (6). Thus $z_{\gamma_2^j+1} = z_{\gamma_3^j+1} = 0$, which means each clause has exactly one true literal.

Conversely, assume that $x = (x_1, \ldots, x_n)$ is a satisfiable assignment of ψ. Define $z \in \mathbb{R}^{3mn+6m}$ as follows: For $i = 1, \ldots, n$, set $z_{\alpha_i+1} = \cdots = z_{\alpha_i+3m} = 1$ if x_i is true, and $z_{\alpha_i+1} = \cdots = z_{\alpha_i+3m} = 0$ if x_i is false. For $j = 1, \ldots, m$, set $z_{\beta_j+\ell} = 1 - z_{\gamma_\ell^j+\ell}$ for $\ell = 1, 2, 3$ and $z_{\beta_j+\ell} = 0$ for $\ell = 4, 5, 6$. Define w to be $w = Mz + q$. Then the pair (w, z) satisfies (5), (6), and (7), as (6) follows from that x is a satisfiable assignment. Moreover, $w, z \geq 0$ holds.

In order to prove that z is a solution of LCP(M, q), it remains to show that the pair (w, z) satisfies $w^\top z = 0$. For $i = 1, \ldots, n$, it holds that $w_{\alpha_i+\ell} z_{\alpha_i+\ell} = 0$ ($\ell = 1, \ldots, 3m$) since (8) is satisfied. Let $j \in \{1, \ldots, m\}$. Since the clause j has exactly one true literal, we may suppose by symmetry that $z_{\gamma_1^j+1} = 1$ and $z_{\gamma_2^j+1} = z_{\gamma_3^j+1} = 0$. By (6), it holds that $w_{\beta_j+1} = 1$ and $w_{\beta_j+2} = w_{\beta_j+3} = 0$. On the other hand, we have $z_{\beta_j+1} = 1 - z_{\gamma_1^j+1} = 0$ by definition, which means that $w_{\beta_j+\ell} z_{\beta_j+\ell} = 0$ for $\ell = 1, 2, 3$. For $\ell = 4, 5, 6$, we have $w_{\beta_j+\ell} z_{\beta_j+\ell} = 0$, since $z_{\beta_j+\ell} = 0$. Thus the complementarity condition is satisfied. Therefore, LCP(M, q) has a solution if and only if ψ is satisfiable, which completes the proof. ◀

4 Polynomial kernel

In this section, we discuss the existence of a polynomial kernel for the parameterized LCP. For a matrix M and a vector q, let $\mathrm{size}(M)$ and $\mathrm{size}(q)$ denote the data size of M and q, respectively. For the s-support (r, c)-sparse LCP with parameter $s + r + c$, a polynomial kernelization algorithm is that given an (r, c)-sparse LCP instance LCP (M, q) and an integer s, outputs in $\mathrm{poly}(\mathrm{size}(M) + \mathrm{size}(q) + s)$ time an (r', c')-sparse LCP instance LCP (M', q') and an integer s' such that

- LCP (M, q) has a solution with support size at most s
  $\Leftrightarrow$ LCP (M', q') has a solution with support size at most s',
- $\text{size}(M') + \text{size}(q') + r' + c' + s' \leq \text{poly}(r + c + s)$.

We show the following theorem.

▶ **Theorem 11.** *The s-support (r, c)-sparse LCP with parameter $s + r + c$ has no polynomial kernel unless* coNP $\subseteq$ NP/poly.

Bodlaender, Downey, Fellows and Hermelin [3] proposed a general framework to prove the nonexistence of a polynomial kernel. They showed that, for any parameterized problem whose unparameterized version is NP-complete, if it admits an algorithm called a *composition algorithm*, then it has no polynomial kernel unless coNP $\subseteq$ NP/poly. We prove Theorem 11 by using their framework.

Note that the unparameterized version of our problem is the s-support LCP, and the decision version is NP-complete by reduction from the s-support bimatrix game, which is NP-complete [18]. For a square matrix M, we denote by $r(M)$ and $c(M)$ the maximum numbers of nonzero entries per row and column in M, respectively.

▶ **Definition 12.** A *composition algorithm* for the s-support (r, c)-sparse LCP with parameter $s + r + c$ is that given t pairs of an LCP instance LCP (M^i, q^i) and an integer s_i $(i = 1, \ldots, t)$, where $s_i + r(M^i) + c(M^i) = k$ for all i, outputs in $\text{poly}(\sum_{i=1}^{t}(\text{size}(M^i) + \text{size}(q^i)) + k)$ time an (r', c')-sparse LCP instance LCP (M', q') and an integer s' such that
- LCP (M', q') has a solution with support size at most s'
  $\Leftrightarrow$ some LCP (M^i, q^i) has a solution with support size at most s_i,
- $s' + r' + c' \leq \text{poly}(k)$.

In the following, we construct a composition algorithm for the s-support LCP.

▶ **Lemma 13.** *The s-support (r, c)-LCP has a composition algorithm.*

Proof. Suppose that we are given t pairs of an LCP instance LCP (M^i, q^i) and an integer s_i $(i = 1, \ldots, t)$, where $s_i + r(M^i) + c(M^i) = k$ for all i. Let $s = \max_i s_i$, $r = \max_i r(M^i)$, and $c = \max_i c(M^i)$. First, for each $i = 1, \ldots, t$, we define an LCP instance as

$$\begin{bmatrix} M^i & O \\ O & I_{s-s_i} \end{bmatrix} \begin{pmatrix} z \\ x \end{pmatrix} + \begin{bmatrix} q^i \\ -1 \end{bmatrix},$$

where I_{s-s_i} is the identity matrix whose size is $s - s_i$, and O is a zero matrix. The coefficient matrix and the constant vector are denoted by N^i and d^i, respectively. Since any solution of LCP (N^i, d^i) satisfies $x = 1$, it is not difficult to see the following claim.

▶ **Claim 14.** *For each $i = 1, \ldots, t$, it holds that* LCP (M^i, q^i) *has a solution with support size at most s_i if and only if* LCP (N^i, d^i) *has a solution with support size at most s.*

We remark that $\text{size}(N^i)$ is bounded by $\text{poly}(s + \text{size}(M^i))$, and $\text{size}(d^i)$ is bounded by $\text{poly}(s) + \text{size}(q^i)$. Since $r(N^i) \leq r$ and $c(N^i) \leq c$ for each i, every instance LCP (N^i, d^i) is (r, c)-sparse.

The output (M', q', s') of our composition algorithm is constructed as follows. Let $T_i = \{j \mid d_j^i < 0\}$ for $i = 1, \ldots, t$. If $|T_i| > cs$, then Lemma 2 implies that LCP (N^i, d^i) does not have a solution with support size at most s. Thus by removing such instances, we may assume that $|T_i| \leq cs$ for all $i = 1, \ldots, t$. For each $i = 1, \ldots, t$, let ℓ^i be a vector whose j-th

entry is $\min(0, d_j^i)$, and let $u^i = d^i - \ell^i \geq 0$. We define a matrix M', a vector q' and an integer s' as follows:

$$
M' = \begin{bmatrix} \begin{array}{ccccc|c} N^1 & \ell^1 & & & & \\ 0 & -1 & & \Large O & & \\ & & \ddots & & & 0 \\ & & & N^t & \ell^t & \\ & \Large O & & 0 & -1 & \\ \hline 0 & 1 & \cdots & 0 & 1 & 0 \end{array} \end{bmatrix}, \quad q' = \begin{bmatrix} u^1 \\ 1 \\ \vdots \\ u^t \\ 1 \\ \hline -1 \end{bmatrix}, \text{ and } s' = s + 1.
$$

We can construct M', q', s' in time

$$
\text{poly}\left(\sum_{i=1}^{t}(\text{size}(N^i) + \text{size}(d^i)) + (s + r + c) \right) \leq \text{poly}\left(\sum_{i=1}^{t}(\text{size}(M^i) + \text{size}(q^i)) + k \right).
$$

We first claim that the output instance satisfies the second condition of the definition of a composition algorithm. It is observed that M' has at most $r + 1 \leq k + 1$ nonzero entries per row, and $\max_i |T^i| + 2 \leq cs + 2 \leq k^2 + 2$ nonzero entries per column. Thus the parameter $s' + r(M') + c(M')$ of LCP (M', q') is bounded by $(s + 1) + (k + 1) + (k^2 + 2) \leq \text{poly}(k)$.

The rest of the proof shows that the first condition of the definition is satisfied, which completes the proof of the lemma.

▶ **Claim 15.** LCP (M', q') *has a solution with support size at most s' if and only if* LCP (N^i, d^i) *has a solution with support size at most s for some $1 \leq i \leq t$.*

◀

───── **References** ─────

1   V. Arvind, J. Köbler, S. Kuhnert, and J.Torán. Solving linear equations parameterized by Hamming weight. In *Proceedings of the 9th International Symposium on Parameterized and Exact Computation*, pages 39–50, 2014.

2   H. Björklund, O. Svensson, and S. Vorobyov. Linear complementarity algorithms for mean payoff games. Technical Report 2005-05, DIMACS: Center for Discrete Mathematics and Theoretical Computer Science, 2005.

3   H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75:423–434, 2009.

4   X. Chen, X. Deng, and S. Teng. Sparse games are hard. In *Proceedings of the 2nd International Workshop on Internet and Network Economics*, pages 262–273, 2006.

5   X. Chen, X. Deng, and S. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, 56:14:1–14:57, 2009.

6   S. J. Chung. NP-completeness of the linear complementarity problem. *Journal of Optimization Theory and Applications*, 60:393–399, 1989.

7   B. Codenotti, M. Leoncini, and G. Resta. Efficient computation of Nash equilibria for very sparse win-lose bimatrix games. In *Proceedings of the 14th Annual European Symposium on Algorithms*, pages 232–243, 2006.

8   E. Cohen and N. Megiddo. Improved algorithms for linear inequalities with two variables per inequality. *SIAM Journal on Computing*, 23:1313–1347, 1994.

9   R. W. Cottle. The principal pivoting method of quadratic programming. In *Mathematics of Decision Sciences*, Part 1, pages 142–162. American Mathematical Society, Providence R. I., 1968.

10   R. W. Cottle and G. B. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra and Its Applications*, 1:103–125, 1968.

11   R. W. Cottle, J. S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, Boston, 1992.

12   P. Damaschke. Sparse solutions of sparse linear systems: fixed-parameter tractability and an application of complex group testing. *Theoretical Computer Science*, 511:137–146, 2013.

13   C. Daskalakis and C. H. Papadimitriou. On oblivious PTAS's for Nash equilibrium. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 75–84, 2009.

14   R. G. Downey and M. R. Fellows. Fixed-parameter intractability. In *Proceedings of the 7th Annual Structure in Complexity Theory Conference*, pages 36–49, 1992.

15   R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, New York, 1999.

16   V. Estivill-Castro and M. Parsa. Computing Nash equilibria gets harder: new results show hardness even for parameterized complexity. In *Proceedings of the 15th Australasian Symposium on Computing*, pages 83–90, 2009.

17   J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, Berlin, 2006.

18   I. Gilboa and E. Zemel. Nash and correlated equilibria: some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.

19   D. Hermelin, C. Huang, S. Kratsch, and M. Wahlström. Parameterized two-player Nash equilibrium. *Algorithmica*, 65:1–15, 2013.

20   D. S. Hochbaum and J. Naor. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM Journal on Computing*, 23:1179–1192, 1994.

21   S. Kratsch. On polynomial kernels for integer linear programs: covering, packing and feasibility. In *Proceedings of the 21st Annual European Symposium*, pages 647–658, 2013.

22   S. Kratsch. On polynomial kernels for sparse integer linear programs. In *Proceedings of the 30th Symposium on Theoretical Aspects of Computer Science*, pages 80–91, 2013.

23   S. Kratsch and V. A. Quyen. On kernels for covering and packing ILPs with small coefficients. In *Proceedings of the 9th International Symposium on Parameterized and Exact Computation*, pages 307–318, 2014.

24   C. E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management Science*, 11:681–689, 1965.

25   K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming*. Internet Edition, 1997.

26   T. J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, pages 216–226, 1978.

27   H. Sumita, N. Kakimura, and K. Makino. The linear complementarity problems with a few variables per constraint. *Mathematics of Operations Research*, to appear.