博士論文（要約）

The Linear Complementarity Problem: Complexity and Integrality

（線形相補性問題： 計算複雑度と整数性）

Hanna SUMITA

澄田 範奈

Abstract

Since the 1950's, the theory of mathematical programming and game theory have been developed rapidly and extensively. The literature shows the rich theory of linear programming, convex quadratic programming, and bimatrix game, which are fundamental subjects in these areas. As a unifying framework of such problems, the linear complementarity problem was introduced in mathematical programming in the mid 1960's.

The linear complementarity problem is to find, for a square matrix and a vector, a vector satisfying linear constraints and complementarity conditions. The linear complementarity problem has been studied from both theoretical and practical points of view, as it has applications in many areas such as computer science, economics and physics. While the linear complementarity problem is hard to solve in general, each of the above applications reduces to the linear complementarity problem in which input matrices have a certain property. Therefore, there exist many results on the computational complexity of the linear complementarity problem in terms of matrix classes. Furthermore, motivated by applications, finding a solution with a certain property such as an integral solution or a minimum norm solution has been attracting attention.

This thesis aims to study the theory of linear complementarity problems. In particular, we focus on two points: the computational complexity and the integrality of the problem.

To investigate the computational complexity, we focus on sparsity of a given matrix. It is known that we can efficiently find a vector satisfying a system of linear inequalities. Furthermore, if the inequalities have the highest sparsity, that is, if each inequality involves at most two variables, then the problem can be solved in a combinatorial way. In this thesis, we classify the computational complexity of the linear complementarity problem in terms of sparsity of a given matrix. We also give an efficient algorithm based on a procedure for sparse linear inequalities, to solve the sparse linear complementarity problem.

As a further study, we investigate the parameterized complexity of the linear complementarity problem. While the classical complexity theory analyzes the time necessary to solve a problem exclusively in terms of the input data size, the parameterized complexity theory takes a specified parameter into account in addition to the input data. Intuitively, the parameterized complexity theory aims to find out an efficient algorithm when a parameter takes

a small value. There exist results on parameterized complexity of the bimatrix game with some parameters. We analyze the parameterized complexity of the linear complementarity problem by using the idea for the bimatrix game. We note that some existing results cannot be extended to the linear complementarity problem.

We also study the complexity of finding a solution with a certain property to the linear complementarity problem. We introduce the problem of finding a solution to an instance of the linear complementarity problem whose basis is identical to the paired instance. This problem is called the linear complementarity problem with orientation. We present two applications of the problem, and show the computational complexity in terms of matrix classes.

The last part of this thesis is concerned with the existence of integral solutions to the linear complementarity problem. There exist two sufficient conditions for the existence of an integral optimal solution to the linear programming problem. One is total unimodularity of a matrix, and the other is total dual integrality of linear constraints, which means that existence of an integral solution to a dual problem implies that of the primal problem. In the study of the linear complementarity problem, it was shown that a basic solution (a solution of a special form) to the problem with a generalization of totally unimodular matrices is integral. In this thesis, we introduce the notion of total dual integrality to the linear complementarity problem. Then we show that the total dual integrality gives a sufficient condition for the existence of an integral solution. To define total dual integrality, we use the linear complementarity problem with orientation.

Acknowledgment

I would like to express my deepest gratitude to Professor Kazuhisa Makino. My research was greatly supported by his deep insight and wide perspective. This thesis would not be possible without his guidance and encouraging advice.

I am thankful to Professor Kazuo Murota for his support on my research. He has also given me valuable advice on my life from now on as a member of society.

I thank Professor Naonori Kakimura for his enthusiastic support in writing joint papers and preparing presentations. I believe that his support has improved my writing and presentation skills.

My great thanks go to Professor Bernd Gärtner for introducing me the study of the linear complementarity problem. It is not too much to say that this thesis, even my research, is not possible without his suggestions. I am thankful to Professor Komei Fukuda for his kindness to care about me, in particular, during my stay in ETH. I also appreciate Dr. Hiroyuki Miyata and Dr. Lorentz Klaus for sharing interest in the linear complementarity problem.

I am thankful to Professor Hiroshi Hirai and Dr. Yusuke Kobayashi for their supports and valuable comments in the seminar.

I am deeply grateful to the Global COE "The Research and Training Center for New Development in Mathematics," to JST, ERATO, Kawarabayashi Large Graph Project, and to Research Fellowships of the Japan Society for the Promotion of Science for Young Scientists, for financial supports.

My special thanks go to Dr. Yasushi Kawase for giving me emotional support. I thank the members of the Mathematical Informatics 2nd Laboratory, who made my time in this laboratory pleasant and unforgettable.

I greatly appreciate my parents for suggesting me for the choice of going into the doctoral course, and supporting my studies until now.

Contents

Chapter 1

Introduction

1.1 Background

The *linear complementarity problem* is one of the most fundamental problems in mathematical programming due to two characteristics: linearity and complementarity. These two characteristics enable the linear complementarity problem to represent many problems in, for example, computer science, economics, physics, and engineering [5, 27, 52, 73, 93, 112, 133]. The linear complementarity problem was introduced in the mid 1960's, in the natural course of development of mathematical programming and game theory.

1.1.1 Linear and quadratic programming

Modern theory of mathematical programming began to flourish with the emergence of linear programming and the progress of computer performance. Mathematical programming is a study of methods to obtain the best value with respect to some objective in a mathematical model. For example, assume that we wish to go to Hongo Campus of the University of Tokyo to Komaba Campus as fast as possible. Since the railroad network spreads all over Tokyo, we have to find the fastest way among many ways to travel. Mathematical programming helps us solve such a problem. The model is an abstraction of the railroad network, and the objective is to minimize the time to Komaba Campus. *Linear programming* aims to find such a best solution when the objective is represented as a linear function and the model is also represented as a system of linear inequalities. More precisely, the goal is to find a real vector x that archives

$$\text{maximize } c^\top x \quad \text{subject to } Ax \leq b,$$

where A is a real matrix and b and c are real vectors.

A lot of deep research has been done on linear programming since 1940's, and today both theoretical and practical importance of linear programming are well recognized. During about 1940's, Kantorovich and Koopmans independently developed linear programming to solve economic problems, such as how to use available resources to maximize production, or how to distribute income fairly. For the early studies of Kantorovich and Koopmans, see e.g., [77] and [86], respectively. They received the Nobel Prize in Economic Sciences in 1975 for this contribution. This shows that the linear programming had a significant influence on the development of economics. In 1947, Dantzig published a practical efficient method to solve linear programming problems, called the *simplex method*. As Dantzig wrote in [34], before 1947, people were not so interested in optimization because they had no way to compute the optimal solution, but since 1947, linear programming has been used for management of production and transportation. As written in [104], in 1947, von Neumann found the theoretically important duality theorem of linear programming [106]. Gale, Kuhn, and Tucker gave a complete proof for the theorem [58]. Duality means that a problem can be viewed from two perspectives. Danzig noted in [104] that Dantzig also proved the duality theorem in 1948 independently in an unpublished paper.

A study of measuring efficiency of computation, called computational complexity theory, began to develop in the mid 1960's. Around that time, Edmonds [43] argued that we can regard an algorithm "whose difficulty increases only algebraically" with the size of input data as an "efficient" algorithm. In linear programming, it was a big issue to find a polynomial-time algorithm to solve any instance of the linear programming problem. It still remains open whether a simplex method can be a polynomial-time algorithm, while it is efficient in practice. A breakthrough in computation of linear programming was brought by Khachiyan and Karmarkar around 1980. In 1979, Khachiyan showed that the linear programming problem is solvable in polynomial time [83], although Khachiyan's algorithm was not useful in practice. Karmarkar proposed a practical polynomial-time algorithm, called an interior-point method in 1984 [79]. After that, many researchers developed an interior-point method using duality of linear programming so that today an interior-point method solves a large-scale linear programming problem efficiently.

Nevertheless, a simplex method is still interesting in linear programming. An interior-point method is a weakly polynomial-time algorithm, i.e., the running time is bounded by a polynomial in the size of numerical values of input data. It remains open to show the existence of a polynomial-time algorithm whose running time does not depend on numerical values (such an algorithm is called a strongly polynomial-time algorithm). A simplex method is considered to be a candidate of a strongly polynomial-time algorithm for the linear programming problem. A simplex method is said to be a combinatorial algorithm based on *pivoting*. Pivoting is an operation of transforming the data of a problem instance to demonstrate either a solution or its nonexistence, and it is often used in numerical linear algebra. Pivoting should be associated with a rule of transformation, and efficiency of a simplex method depends on the rule. Although most of the well-known rules have bad instances [2, 53, 54, 63, 74, 82], there may exist a good rule with which any instance can be processed in polynomial time. Recently, the behavior of a simplex method when applied to other problems, such as finding an optimal policy of a Markov decision process, is often studied [66, 113, 138].

As the linear programming theory grew up, the theory of *quadratic programming* was also developed rapidly. Quadratic programming is a method to find a vector x which satisfies a linear constraint and maximizes or minimizes a quadratic function of form $\frac{1}{2}x^\top Dx + c^\top x$, where D is a square real matrix, and c is a real vector . A study of quadratic programming has as long history as linear programming. The method of Lagrange multipliers has been known for useful to solve unconstrained (nonlinear) optimization problems. In 1951, Kuhn and Tucker [91] extended this to constrained optimization problems, and presented the first-order necessary conditions for a solution to the nonlinear programming problem to be optimal, called the Karush-Kuhn-Tucker (KKT) conditions. Note that the conditions also appeared in the work of Karush [80]. Frank and Wolfe showed a sufficient condition for the existence of a global minimum in a quadratic programming problem, known as the Frank-Wolfe theorem, in 1956 [51]. Nevertheless, it is not doubtful that advances of linear programming theory made a contribution to revealing the rich theory of quadratic programming. In particular, the linear programming theory had an impact on the study of a natural generalization of the linear programming problem, called the *convex* quadratic programming problem. This is the quadratic programming problem whose objective functions to be maximized (or minimized) have a special property, called convexity. In 1959, Wolfe proposed a simplex-method-like algorithm to solve (nonconvex) quadratic programming problems [137]. Around 1960, Dennis [38] and Dorn [40] studied duality in convex quadratic programming. In 1967, Dantzig and Cottle [33] derived a necessary and sufficient condition for convex quadratic programming problems to have a solution, which includes the results by Dennis and Dorn. In the late 1980's, Kapoor and Vaidya [78], and Ye and Tse [139] extended a variant of an interior point method for linear programming problems to obtain a polynomial-time algorithm for the convex quadratic programming problem.

In fact, Cottle and Dantzig obtained the above result by generalizing the convex quadratic programming problem to the *linear complementarity problem* and studying this generalized problem. Kojima, Mizuno and Yoshise [85] also obtained an interior-point method for the linear complementarity problem.

1.1.2 Bimatrix games

Along with the advances of linear programming, *game theory* also began to develop drastically. Game theory is a study of strategic interactions among agents, or players. In 1928, game theory was established by the work of von Neumann [105] as an area in applied mathematics. In the work, von Neumann dealt with zero-sum game, i.e., the case when the sum of payoffs (or costs) of players is always equal to zero. Then he proved the minimax theorem, which is a fundamental theorem in game theory. Von Neumann and Morgenstern brought game theory into the field of economics, and published a book [107] in 1944. Their works attracted notable mathematicians, and Nash was one such mathematicians. Nash is well known for his work in about 1950 which proved that any finite n-person game has at least one Nash equilibrium, in which no agent has incentive to change only his/her decision while others keep their decisions unchanged [102, 103]. In 1994, Nash, together with two economists, Selten and Harsanyi, received the Nobel Prize in Economic Science for their contribution to the development of game theory and its application to economics. It should be mentioned that Nash's proof in [103] was a nonconstructive proof based on Brouwer fixed-point theorem. Computing equilibria became a matter in computer science, since as [108] says, efficient computability makes it possible to predict agents' equilibrium behavior, while computational intractability leads to a doubt that an equilibrium might not be reached in the real world. Computer scientists established game theory from algorithmic point of view, which is called algorithmic game theory.

As mentioned in [110], the problem of finding a Nash equilibrium is one of the most important complexity problems of this time. The bimatrix game is a simple but important, two-person game in algorithmic game theory. This game has two matrices each of which represents payoffs or costs of one player. It is well known in this area that a Nash equilibrium of a zero-sum bimatrix game can be found by using linear programming. We note that a Nash equilibrium is expressed as a pair of two vectors which represent players' strategies. More generally, the problem of computing a Nash equilibrium of a zero-sum game is also solvable by using linear programming [125]. On the other hand, computability of Nash equilibria (in particular, of bimatrix games) remains unclear. Lemke and Howson proposed a constructive procedure to find a Nash equilibrium of a bimatrix game [94]. This gave also a constructive proof of the Nash theorem. Other methods can be found in [90, 95, 136]. The Lemke-Howson method is efficient in practice, but there exist bad cases which the Lemke-Howson method takes much time to solve [118]. A polynomial-time algorithm is still unknown, and moreover, finding a Nash equilibrium is considered to be hard. It is said to be unlikely that there exist a polynomial-time algorithm, due to the evidence that this problem belongs to the complexity class PPAD. The class PPAD (Polynomial Parity Arguments on Directed graphs), introduced by Papadimitriou in 1994 [109], is the set of search problems which are at least as hard as the problem of finding a Brouwer fixed point. The problem of finding a Nash equilibrium of a bimatrix game is shown to be PPAD-complete in 1995 by Chen and Deng [12], which means that this problem is one of the hardest problems in PPAD. See also [14].

1.1.3 The linear complementarity problem

In the mid 1960's, Lemke [93], Cottle [24], and Cottle and Dantzig [25] introduced the *linear complementarity problem* (LCP), as a unifying problem of the linear and convex quadratic programming, and the problem of finding Nash equilibrium of bimatrix games.

The formal definition of the LCP is as follows: given an $n \times n$ real matrix M and an n-dimensional real vector q, the linear complementarity problem is to find an n-dimensional vector z such that

$$Mz + q \geq 0, \quad z \geq 0, \quad z^\top (Mz + q) = 0. \tag{1.1}$$

The last condition is called the complementarity condition. We denote a problem instance of the LCP with M and q by LCP(M, q). We say that n is the order of the LCP. For any vectors satisfying $Mz + q \geq 0$ and $z \geq 0$, the condition $z^\top (Mz + q) = 0$ is equivalent to the condition that for each index $i = 1, \ldots, n$, at least one of z_i and $(Mz + q)_i$ is equal to zero. Thus a solution to LCP(M, q) must satisfy $Mz + q \geq 0,\ z \geq 0$ and

$$(Mz + q)_i = 0 \ \ (i \in B), \quad z_i = 0 \ \ (i \notin B)$$

for some index set $B \subseteq \{1, \ldots, n\}$. This index set B is called a basis. Once we know a basis of a solution to the LCP, we can find the solution to the LCP easily by using the linear programming. Thus finding a basis is essential for the LCP. However, as we describe later, finding a basis is not considered to be an easy task in the sense of the computational complexity.

As we can see in [25], the Karush-Kuhn-Tucker condition of a convex quadratic programming problem can be written as an LCP instance. An LCP formulation of finding a Nash equilibrium of a bimatrix game appears in [94]. Intensive study of the LCP began in 1960's, while a special case of the LCP appeared in 1940 [42]. The LCP has two fundamental characteristics in mathematical programming: linearity and complementarity. In fact, the LCP was called the "fundamental problem" in the early works of the LCP [24, 25]. It is revealed later that the LCP has applications in many areas, such as computer science, economics, and physics. An example of such an application in game theory is the mean payoff game. The mean payoff game is significant in complexity theory, because a decision problem (yes-no question) associated with this game belongs to the intersection of the complexity class NP and coNP, where we describe these classes later. Reduction of the mean payoff game to the LCP was shown in [5]. In economics, the problem of finding a market equilibrium is a well-known example of the LCP. A market equilibrium is the state of an economy in which the demands of consumers and the supplies of producers are balanced. The problem of finding the quantity of production and the prices of commodities at the state is formulated as the LCP [133]. In physics, the contact problem can be written as the LCP. The contact problem is to compute the state of two smooth elastic bodies in contact. Roughly speaking, compatibility of deformation is represented as the system of linear inequalities, and contact criterion is written as the complementarity condition. Thus the contact problem is formulated as the LCP. See e.g., [19, 52] for the detail. This application leads to the practical use of the LCP. Indeed, many commercial physics engine, which provides a simulation of physical systems, solves the contact problem via the LCP.

We mention that computation of a solution (to the LCP) with a certain property is desired for some applications. For example, it is known that with a high probability, a randomly chosen bimatrix game has a Nash equilibrium whose vectors have a small number of positive elements [3]. Intuitively, the number of reasonable choices is small in most cases. This implies that LCP instances derived from bimatrix games often have solutions with few positive elements, and finding such a solution to the LCP can save the computation costs. The problem of finding a market equilibrium is another example. When we deal with an indivisible commodity, an integral solution to the LCP is suitable for this application [112]. We will describe this later. There exist results to find a solution to the LCP with a certain property, such as a minimum norm solution [111, 123] or an integral solution [11, 31, 112].

In the theory of the LCP, matrix classes of the input matrix M play a major role, since the LCP shows different properties depending on a class of input matrices M. In [26], it was shown that when M is a positive semi-definite matrix (not necessarily symmetric), the LCP has a solution if the system of linear inequalities in (1.1) is feasible. The LCP instances obtained from the bimatrix game always has a solution, since the bimatrix game has a Nash equilibrium. In addition, each of the above applications of the LCP leads to the LCP instances having matrices with a certain property. In this thesis, we particularly focus on positive semi-definite matrices, $\boldsymbol{P}$*-matrices*, $\boldsymbol{Z}$*-matrices*, $\boldsymbol{K}$*-matrices*, and *sufficient matrices* among many matrix classes concerning the LCP. We note that $\boldsymbol{K}$-matrices are also called $\boldsymbol{M}$-matrices.

Existence theory

The earliest existence result on the LCP was the notable fact that LCP(M, q) has a solution for any real vector q if and only if M is a $\boldsymbol{P}$*-matrix.* This was indirectly shown by Samelson, Thall and Wesler in 1958 whose motivation was in geometry [117]. The notion of $\boldsymbol{P}$-matrices was later introduced in the mid 1960's [59], and properties of $\boldsymbol{P}$-matrices was investigated in [47]. However, the beginning of the systematic study of the existence results was largely due to results on the convex quadratic programming, namely, results by Dorn [40] and Cottle [22, 23]. These dealt with two fundamental classes of positive definite, and positive semi-definite matrices. The class of $\boldsymbol{P}$-matrices is a generalization of positive definite matrices, and is characterized by a property of associated LCP instances. In order to obtain matrix classes which are characterized by properties of the LCP with positive semi-definite matrices, Cottle, Pang and Venkateswaran [28] introduced classes of *row sufficient* matrices and *column sufficient* matrices as a generalization of positive semi-definite matrices. Both a row and column sufficient matrix is called a *sufficient* matrix.

When we focus on matrices with a certain pattern of nonzero entries, we should mention $\boldsymbol{Z}$*-matrices* and the intersection of $\boldsymbol{Z}$-matrices and $\boldsymbol{P}$-matrices called $\boldsymbol{K}$*-matrices*. $\boldsymbol{Z}$-matrices and $\boldsymbol{K}$-matrices have been widely studied, and they appear in physical, economic and geometric applications of the LCP. In 1962, Fiedler and Pták studied systematically these matrices, and they presented 13 equivalent conditions for a $\boldsymbol{Z}$-matrix to be a $\boldsymbol{K}$-matrix [47]. A special case of the LCP with $\boldsymbol{Z}$-matrices appeared in [42], and Cottle and Veinott [29] obtained a more general result. A remarkable characteristic of $\boldsymbol{Z}$-matrices is that the feasible region of linear constraints in (1.1) has a good structure, and this characteristic implies that solving one linear programming problem gives a solution to the LCP with $\boldsymbol{Z}$-matrices. A larger class, called hidden $\boldsymbol{Z}$-matrices, possessing this characteristic was introduced by Mangasarian [96].

Complexity theory

Many methods to solve the LCP have been proposed from the early stage of this literature. In particular, pivoting methods for the LCP can be found in early works on the LCP at around 1960's. In 1965, Lemke [93] extended the Lemke-Howson method for the bimatrix game to an algorithm to solve the LCP. The paper [33] by Dantzig and Cottle in 1967 presented a pivoting algorithm for the LCP with positive semi-definite matrices, called a principal pivoting method. The earliest example of principal pivoting methods was proposed by Zoutendijk in 1960 [140] and Bard in 1972 [4]. The method used in the papers is now called a simple principal pivoting method, which is similar to the simplex method. The simple principal pivoting method is applicable to a certain class of the LCP, including the LCP with $\boldsymbol{P}$-matrices.

Another type of a pivoting method is Chandrasekaran's algorithm for the LCP with $\boldsymbol{Z}$-matrices, proposed in 1970 [9, 116]. If the order of M is n, then the Chandrasekaran's algorithm terminates in at most n iterations.

For the purpose of solving the large-scale LCP instances, pivoting methods may not be suitable for reasons such as round-off errors and data storage, as described in [27]. Round-off errors may lead to incorrect pivoting and erroneous solutions, and such errors accumulate rapidly as the number of iterations increases. In practice, a large-scale problem instance tends to be sparse, which means that input data contains many zeros. Since pivoting methods destroy the sparsity, we need to store all of the data if we use pivoting methods, and this is storage consuming. Therefore, there exist some iterative methods such as a Newton-type method and an interior-point method for the LCP. See [27] for the detail.

In the sense of computational complexity theory, the LCP is considered to be a hard problem. To be more precisely, the LCP is NP-hard, which was shown by Chung in 1989 [15]. In computational complexity theory, the set of decision problems which can be solved in polynomial time is called the class P. The class NP (Nondeterministic Polynomial time) represents a set of decision problems such that if we are given a certificate that the answer of a problem instance is "yes", then we can check it in polynomial time with respect to the size of input data. Note that this says nothing about when we do not have a certificate. When we replace "yes" with "no" in the definition of NP, then it defines the class coNP. The class P is contained in both NP and coNP by definition. Whether or not P coincides with NP is a big issue in this time. The relationship of difficulty of problems is determined by reduction of problems. If problem A can be reduced to problem B (in polynomial time with respect to the input data size of problem A), then we can see that solving problem B is at least as difficult as problem A, since we can solve problem A via problem B. A problem is called NP-hard, if any problem in NP can be reduced to the problem.

However, each application reduces to the LCP having matrices with a certain property as mentioned above, and it is important to identify efficiently solvable classes of the LCP. We present some known results here. In computational complexity theory, it is important to estimate the complexity of a problem when the size of input data is sufficiently large. When we mention the complexity explicitly, we use the notation $f(n) = \mathrm{O}(g(n))$ meaning that there exists a positive number c such that $f(n) \leq c \cdot g(n)$ for sufficiently large numbers n.

It is known that the LCP with positive semi-definite matrices can be solved by an interior-point method [85]. When M is a $\boldsymbol{Z}$-matrix of order n, the LCP can be solved by a pivoting algorithm in $\mathrm{O}(n^3)$ time [9]. We can also solve the LCP with $\boldsymbol{K}$-matrices easily by the simple principal pivoting method with any pivoting rule, which terminates in a linear number of iterations with respect to the order of the LCP [49].

On the other hand, the complexity of the LCP with $\boldsymbol{P}$-matrices is unknown. Megiddo [97] showed that if the LCP with $\boldsymbol{P}$-matrices is NP-hard, then two classes NP and coNP coincide with each other, which is believed to be unlikely. Thus Megiddo's result suggests the existence of a polynomial-time algorithm to solve the LCP with $\boldsymbol{P}$-matrices. There exists a lot of the literature attempting to reveal the complexity of the LCP with $\boldsymbol{P}$-matrices, such as [49, 50, 62, 132]. It is known that the LCP with $\boldsymbol{P}$-matrices of order n can be solved in $\mathrm{O}(c^n)$ $(c < 2)$ time by a pivoting method [132]. We note that the deciding whether or not a matrix is a $\boldsymbol{P}$-matrix was shown to be coNP-complete [30]. This means that recognizing $\boldsymbol{P}$-matrices is a hard task. For a larger class containing $\boldsymbol{P}$-matrices, called $\boldsymbol{P_0}$-matrices, the LCP is known to be NP-hard [84].

1.2 Our contribution

This thesis investigates theoretical aspects of the LCP: the computational complexity and the integrality. Throughout this thesis, the main idea of our study is that the LCP can be regarded as an extension of the system $Mz + q \geq 0,\ z \geq 0$ of linear inequalities. We also make use of the concept introduced into the bimatrix game.

1.2.1 Complexity

The topics can be divided into two parts. The first topic is analysis of the complexity of the LCP in terms of sparsity of the coefficient matrices. The second topic deals with the problem of finding a special solution to the LCP, which we call the LCP with orientation.

Sparse linear complementarity problems

The term "sparsity" means that most entries are zero. Such matrices often appear in application, such as the contact problem. The contact problem is to compute the state of two smooth elastic bodies (called Body 1 and Body 2) in contact. We here present a simple case of the contact problem as an example. See [27] for the detail of this problem.

Suppose that the external force is applied on Body 1, whose value is p, in the vertical direction. Assume that we know n pairs of two points on surfaces of Body 1 and 2 which can come in to contact, and that deformations are small. We also know the distance d_i between the ith pair of points if two bodies can penetrate each other, for $i = 1, \ldots, n$. Since the bodies cannot penetrate each other, deformations of the bodies occur. Let z_i denote the contact stress at the ith pair of points. Let v_i^1 and v_i^2 denote the the deformations (in the vertical direction) at the ith point on Body 1 and 2, respectively. We denote by d the n-dimensional vector whose ith element is d_i. We define z, v^1, and v^2 similarly.

Since the distances between pairs of points are nonnegative, deformations and distances must satisfy

$$v^1 + v^2 + d \geq 0.$$

For each pair of points, if the distance between the points is positive, then the contact stress must be zero. Thus the contact criterion

$$(v_i^1 + v_i^2 + d_i) \cdot z_i = 0 \quad \text{for } i = 1, \ldots, n$$

must be satisfied. In addition, since the deformations are small, we can assume that deformations are given by $v^1 = Az$ and $v^2 = Bz$, where A^1 and A^2 are known matrices. Therefore, it suffices to find the vector z satisfying

$$Az + d \geq 0,\ z \geq 0,\ z^\top (Az + d) = 0,$$

where $A = A^1 + A^2$, and this coincides with LCP(A, d). When, for example, two bodies are relatively solid, each deformation is affected by contact stresses of neighbor points. In this case, the matrix A is sparse.

The rest of this subsection cannot be publicized because a copyright holder's consent has not been obtained.

1.2.2 Integrality

The integrality of the linear programming problem has been studied well. The linear programming problem in which the variables are restricted to integers is called the integer linear programming problem. This problem has many applications, including many NP-hard problems. One direction in the study of the integer linear programming problem is to obtain a good approximation of the optimal value. We focus on another direction, which aims to investigate sufficient conditions that the underlying linear programming problem has an integral optimal solution. In such a case, we can easily obtain an optimal solution to the integer linear programming problem.

In the study of the integrality of the linear programming problem, *totally unimodular* matrices are important. A square integral matrix is called totally unimodular matrix if all square submatrices have determinants 0 or ± 1. Totally unimodular matrices appear in combinatorial optimization. It is well known that a linear programming problem with a totally unimodular matrix and an integral vector has an integral optimal solution, which was shown by Hoffman and Kruskal in 1956 [71]. Later, in 1977, Edmonds and Giles introduced the notion of *total dual integrality* of linear systems [44]. This notion is defined as the dual problems of the linear programming problems associated with a linear system have an integral optimal solution. This notion gives a unified framework for linear programming problems having an integral optimal solution arising in combinatorial optimization. The total dual integrality of linear systems is deeply related with totally unimodular matrices. It is known that an integral matrix A is totally unimodular if and only if the linear system $Ax \leq b,\ x \geq 0$ is totally dual integral for every vector b. For the detail, see e.g., [120].

In this thesis, we focus on integral solutions to the LCP. Integral solutions to the LCP were first considered by Chandrasekaran [10] in the context of the least element theory. A class of the LCP having integral solutions was considered earlier by Pardalos and Nagurney [112], with some applications which need integral solutions. One application is the problem of finding a market equilibrium, whose LCP formulation was pointed out in [133]. We present a simple model here. Assume that there are n economic regions which produces and/or consumes a single indivisible commodity. We know the unit cost c_{ij} of shipping between region i and region j. The problem is to compute the amount of trade and the price of the commodity for each region at an equilibrium. Let t_{ij} denote the number of the commodity shipped from region i to j, and let p_i denote the price of the commodity in region i.

We assume that the amount of the demand in region i is represented as $b_i - d_i p_i$, where b_i and d_i are positive numbers. Since the amounts of export and import should be balanced at an equilibrium, variables p_{ij}'s and t_{ij}'s must satisfy

$$(b_i - d_i p_i) + \sum_{j \neq i} t_{ij} = \sum_{j \neq i} t_{ji} \quad \text{for } i = 1, \dots, n.$$

In addition, to ensure that there are no more incentives to trade at an equilibrium, variables p_{ij}'s and t_{ij}'s should satisfy

$$p_i + c_{ij} - p_j \geq 0,\ t_{ij} \cdot (p_i + c_{ij} - p_j) = 0 \quad \text{for all } i, j.$$

We can write the problem of finding variables p_{ij}'s and t_{ij}'s satisfying two conditions as LCP(M, q) by letting z denote the vector consisting of p_{ij}'s and t_{ij}'s, and defining the matrix M and the vector q appropriately. We remark that since the monetary unit is integral, and the commodity is indivisible by assumption, an integral solution to LCP(M, q) is suitable for this model.

The rest of this subsection cannot be publicized because a copyright holder's consent has not been obtained.

1.3 Organization of this thesis

The remainder of this thesis is organized as follows. Chapter 2 defines the linear programming, the bimatrix game, and the integer linear programming, and then it presents existing results on these problems. Chapter 3 describes fundamental properties of the linear complementarity problem and its related works. Chapter 4 shows the time complexity of the k-LCP. The results are also contained in [128, 129]. A part of the results was also mentioned in [127]. Chapter 5 analyzes the parameterized complexity of the LCP. Chapter 6 discusses the applications and the time complexity of the LCP with orientation, whose description is based on [131]. Chapter 7 introduces total dual integrality of the LCP and shows that it certifies integrality of the LCP. The results are also contained in [130]. Chapter 8 summarizes this thesis.

Chapter 2

Preliminaries

In this chapter, we present fundamental notation. Then we define the linear programming, the bimatrix game, and we review previous works.

For a positive integer n, let $[n] = \{1, \ldots, n\}$. We denote by $\mathbb{R}, \mathbb{Z}$, and $\mathbb{N}$ the set of real numbers, integers, and positive integers. A vector or matrix is called *real* (resp. *rational*, *integral*, respectively) if all its entries are real numbers (resp. rational numbers, integers). Let A be an $m \times n$ real matrix, where A has a row index set $[m]$ and a column index set $[n]$. For $S \subseteq [m]$ and $T \subseteq [n]$, we denote by A_{ST} the submatrix of A such that S and T are row and column index sets, respectively. We also define $A_{\cdot T}$ and $A_{S\cdot}$ by $A_{\cdot T} = A_{[m]T}$ and $A_{S\cdot} = A_{S[n]}$, respectively. If $T = \{j\}$, we simply write A_{Sj} and $A_{\cdot j}$ instead of $A_{S\{j\}}$ and $A_{\cdot\{j\}}$, respectively. We similarly define A_{iT} and $A_{i\cdot}$. Let z denote a vector in $\mathbb{R}^n$ with index set $[n]$. In this thesis, we assume that all vectors are column. For index set $B \subseteq [n]$, let z_B denote the subvector of z with elements corresponding to B, i.e., z_B in $\mathbb{R}^B$. We also denote by z_i the ith element of z for i in $[n]$ For a subset $B \subseteq [n]$, let $\overline{B}$ denote the complement of B, i.e., $\overline{B} = [n] \setminus B$. We denote by I the identity matrix, i.e., its diagonal entries I_{ii} are ones, and other entries are zeros. We also denote by $\mathbf{1}, \mathbf{0}$ the vectors whose elements are all one, and zero, respectively.

An *undirected graph* is a pair $G = (V, E)$, where V is a finite set, and E is a family of unordered pairs $\{u, v\}$ of elements u, v of V. Each element of V is called the *vertex* of G, and each elements of E is called the *edge* of G. We say that an edge $\{u, v\}$ *connects* the vertices u and v. The vertices u and v are *adjacent* if there is an edge connecting u and v. The edge $\{u, v\}$ is *incident with* the vertex u and with v, and conversely.

The number of edges incident with a vertex v is called the *degree* of v. The maximum degree of the vertices of G is called the degree of G.

An undirected graph $G' = (V', E')$ is called a *subgraph* of $G = (V, E)$ if V' is a subset of V, and E' is a subset of E. In addition, if E' contains only edges connecting vertices in V', then G' is called an *induced subgraph*.

A *directed graph* is a pair $D = (V, E)$, where V is a finite set, and E is a finite family of ordered pairs (u, v) of elements u, v of V. In a similar way to undirected graphs, we say that elements of V are vertices, and elements of E are edges.

We say that the edge (u, v) *leaves* u and *enters* v. An edge leaving a vertex v is said to be an *outgoing* edge of v. Similarly, an edge entering a vertex v is said to be an *incoming* edge of v. A vertex with no outgoing edges is called a *sink*, and a vertex with no incoming edges is a *source.*

2.1 Linear and quadratic programming

We first introduce some fundamental notation and terminology. For vectors a and b with common index set $[n]$, we write $a \leq b$ if $a_i \leq b_i$ for $i \in [n]$. If A is a matrix, and b, x be vectors, then when we write $Ax = b$ or $Ax \leq b$, we implicitly assume that if A is an $m \times n$ matrix, then the index sets of b and x are $[m]$ and $[n]$, respectively. For a vector a and a real number b, $a^\top x = b$ and $a^\top x \leq b$ are called *linear equation* and *linear inequality*, respectively. For a matrix A and a vector b, we say that

$$Ax = b \quad \text{and} \quad Ax \leq b$$

are a *system of linear equation* and a *system of linear inequalities*, respectively. We also call a system consisting of linear equation and linear inequalities a *linear system.* For a system $Ax \leq b$ of linear inequalities, a system $A'x \leq b'$ is a *subsystem* of $Ax \leq b$ if $A'x \leq b'$ arises from $Ax \leq b$ by deleting some (or none) linear inequalities in $Ax \leq b$. Similarly for a system of linear equations and a linear system.

2.1.1 Linear programming

Let $Ax \leq b$ be a system of linear inequalities. A set P of vectors defined by

$$P = \{x \mid Ax \leq b\}$$

is called a *polyhedron.* The *linear programming problem* is the problem of maximizing or minimizing a linear function over a polyhedron. A normal form of the linear programming problem is represented as

$$\max\{c^\top x \mid Ax \leq b\}, \tag{2.1}$$

where A is a matrix, and b, c are vectors. In this case, a vector x satisfying $Ax \leq b$ is called a *feasible solution*, and the set of feasible solution is called the *feasible region.* If the feasible region is nonempty, then the problem is said to be *feasible*, and otherwise *infeasible.* The function $x \to c^\top x$ is called the *objective function*, and $c^\top x$ is the *objective value* of x. We say that a feasible solution which attains the maximum is an *optimal* solution.

An important characteristic of the linear programming problem is its duality theorem by von Neumann [106], and Gale, Kuhn and Tucker [58].

Theorem 2.1. *Let A be a matrix, and let b and c be vectors. Then it holds that*

$$\max\{c^\top x \mid Ax \leq b\} = \min\{b^\top y \mid A^\top y = c,\ y \geq 0\}$$

if both problems are feasible.

The proof of this theorem is deeply related to the following theorem of alternative [46], so-called Farkas' lemma.

Theorem 2.2 (Farkas' lemma)**.** *Let A be a matrix, and b be a vector. Then there exists a vector $x \geq 0$ with $Ax = b$ if and only if $b^\top y \geq 0$ for each vector y with $A^\top y \geq 0$.*

We say that the minimization problem $\min\{b^\top y \mid A^\top y = c\}$ is the *dual* problem of the maximization problem $\max\{c^\top x \mid Ax \leq b\}$, which is called *primal.* We also say that the maximization problem is the dual problem of the minimization problem. There are several equivalent forms for a linear programming problem. For example, we can transform (2.1) into the form

$$\max\{c^\top x \mid Ax \leq b,\ x \geq 0\}$$

by replacing the matrix and vectors appropriately, and vice versa.

Theorem 2.1 says nothing about the case when either problem is infeasible. In fact, there are four cases:

(a) both maximization and minimization problems are feasible, and the optimal values are equal;

(b) the maximization problem is infeasible, and the minimum is unbounded from below;

(c) the minimization problem is infeasible, and the maximum is unbounded from above;

(d) optimal values of both problems are unbounded.

Optimal solutions to a primal and its dual linear programming problem have the following relation.

Theorem 2.3. *Let A be a matrix, and let b and c be vectors. Assume that* $\max\{c^\top x \mid Ax \leq b\} = \min\{b^\top y \mid A^\top y = c,\ y \geq 0\}$ *is finite, and let x and y be feasible solution to the maximum and the minimum, respectively.*

Then the following are equivalent.

(a) *x and y are optimal solutions.*

(b) $c^\top x = b^\top y$.

(c) $y^\top (b - Ax) = 0$.

The last condition is called *complementary slackness*. More precisely, for each inequality $a^i x \leq b_i$ in $Ax \leq b$, exactly one of $a^i x < b_i$ and $y_i > 0$ holds, since $b - Ax \geq 0$ and $y \geq 0$.

2.1.2 Simplex method

The affine hyperplane $\{x \mid c^\top x = \delta\}$ is called a *supporting hyperplane* of P if c is nonzero and $\delta = \max\{c^\top x \mid x \in P\}$. A subset F of P is called a *face* of P if $F = P$ or F is the intersection of P with some supporting hyperplane of P. Alternatively, F is a face of P if and only if F is nonempty and $F = P \cap \{x \mid A'x = b'\}$ for some subsystem $A'x \leq b'$ of $Ax \leq b$. In particular, a face is *minimal* if it contains no other face. It is known that a minimal face of P can be represented as $\{x \mid A'x = b'\}$ for some subsystem $A'x \leq b'$ of $Ax \leq b$. A zero-dimensional face is called a *vertex*.

A set C of vectors is said to be *convex* if $\lambda x + (1 - \lambda)y$ belongs to C for any $x, y \in C$ and $0 \leq \lambda \leq 1$. For a set X of vectors, we say that the smallest convex set containing X the *convex hull* of X, that is, $\left\{\sum_{i=1}^{t} \lambda_i x^i \mid t \geq 1,\ x^1, \ldots, x^t \in X,\ \lambda_1, \ldots, \lambda_t \geq 0,\ \sum_{i=1}^{t} \lambda_i = 1\right\}$. A polyhedron P is called *integral* if P is the convex hull of the integral vectors in P. A polyhedron P is integral if and only if any face of P contains an integral vector, or equivalently, $\max\{c^\top x \mid x \in P\}$ has an integral optimal solution for each vector c such that the maximum exists.

The oldest practical algorithm for linear programming is Dantzig's simplex method. Suppose that we aim to solve the linear programming problem (2.1). A simplex method is often described as an algorithm that moves on vertices of the polyhedron $P = \{x \mid Ax \leq b\}$ from vertex to vertex until an optimal solution is found. Assuming that we are known a vertex x of P, a basic procedure of a simplex method is as follows.

A simplex method (basic idea)

Input: A matrix A in $\mathbb{R}^{m \times n}$, a vector b in $\mathbb{R}^m$, and a vertex x of $P = \{x \mid Ax \leq b\}$.

Step 0: Choose a set $B \subseteq [m]$ such that $A_{B\cdot}$ is nonsingular and $A_{B\cdot} x = b_B$.

Step 1: Let u be a vector defined by $u_B = (A_{B\cdot}^\top)^{-1} c$ and $u_{\overline{B}} = 0$ ($A^\top u = c$ holds by definition). If $u \geq 0$, then return x as an optimal solution.

Step 2: Choose an index $i \in B$ with $u_i < 0$. Let w be the column of $-(A_{B\cdot})^\top$ with index i (thus $A_{B\cdot} w \lneq 0$). If $Aw \leq 0$, then return w as the certificate that the maximum is unbounded.

Step 3: Let

$$\lambda = \min\left\{ \frac{b_i - A_{i\cdot} x}{A_{j\cdot} w} \mid j \in [m],\ A_{j\cdot} w > 0 \right\}.$$

Let also j the smallest index attaining this minimum.

Step 4: Replace B with $(B \cup \{j\}) \setminus \{i\}$, and x with $x + \lambda w$. Then go to Step 1.

In this procedure, the set B is called a *basis*, and the operation at Step 4 is called *pivoting*. Pivoting means transforming the data of a problem instance to demonstrate either a solution or its nonexistence. This idea is often used in numerical linear algebra. The rule of selecting an index at Step 2 is a *pivoting rule*. There exist many pivoting rules for the simplex method. A simplex method may run into cyclic, but the simplex method with Bland's rule [6] terminates after a finite number of iterations. Unfortunately, the simplex method with Bland's rule is not a polynomial-time algorithm, since Klee and Minty [82], and Avis and Chivátal [2] found bad instances with n variables and $2n$ inequalities where the simplex method with Bland's rule takes 2^n iterations. Most of known pivoting rules have such bad instances [82, 74, 2, 63, 53, 54]. Nevertheless, theoretical studies on a simplex method are still of interesting. A strongly polynomial-time algorithm for the linear programming problem is still unknown, and a simplex method is considered to be a candidate of such an algorithm.

2.1.3 Quadratic programming

The *quadratic programming problem* is the problem of maximizing or minimizing a quadratic function over a polyhedron. A typical formulation is

$$\begin{aligned} \text{max.} \quad & f(x) = \tfrac{1}{2} x^\top Q x + c^\top x \\ \text{s.t.} \quad & Ax \leq b, \end{aligned}$$

where Q is a square matrix, A is a matrix, and b, c are vectors.

A necessary condition for a feasible vector x of the quadratic programming problem (2.2) to be optimal to (2.2) is given by the *Karsh-Kuhn-Tucker condition* [80, 91]: There exists a vector μ satisfying

$$\begin{aligned} Qx - A^\top \mu + c &= 0, \\ -Ax + b &\geq 0, \\ \mu &\geq 0, \\ \mu^\top(-Ax + b) &= 0. \end{aligned} \tag{2.2}$$

If a feasible solution x of (2.2) satisfies the Karush-Kuhn-Tucker condition, then x attains the maximum within its neighbors (i.e., sufficiently close vectors to x). Such a vector is called *local optimal* to (2.2). If a local optimal solution is optimal in the entire feasible region, it is called *global optimal*.

When the matrix Q is *positive semi-definite*, i.e., $x^\top Qx \geq 0$ holds for any vector x, the Karush-Kuhn-Tucker condition (2.2) is also a sufficient condition for a feasible vector to be global optimal. In such a case, the objective function f of the problem (2.2) satisfies that $f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y)$ for any vectors x, y such that $f(x), f(y) < +\infty$, and $0 \leq \lambda \leq 1$. Such a function is called a *convex* function. The problem (2.2) with a positive semi-definite matrix Q is called the *convex* quadratic programming problem, since the objective function and the feasible region are both convex.

2.2 Bimatrix games

A bimatrix game consists of two players, called Player I and Player II. Let K and L be finite sets of strategies of Player I and Player II, respectively, where $K \cap L = \emptyset$. We denote by k and l the sizes of K and L, respectively. A bimatrix game is given two $k \times l$ matrices A and B, where a_{ij} and b_{ij} respectively represent costs to Player I and Player II, when Player I chooses strategy $i \in K$ and Player II chooses strategy $j \in L$. We denote by (A, B) a bimatrix game with Player I's matrix A and Player II's matrix B.

A mixed strategy for Player I is a vector x^* in $\mathbb{R}^k$ where x^*_i represents the probability of choosing strategy $i \in K$. That is, a mixed strategy x^* satisfies that $x^* \geq 0$ and $1^\top x^* = 1$. A mixed strategy y^* for Player II is defined analogously. For mixed strategies x^* and y^* for Players I and II, their expected costs are given by $(x^*)^\top Ay^*$ and $(x^*)^\top By^*$, respectively. The pair (x^*, y^*) of mixed strategies is said to be a *Nash equilibrium* if it satisfies that

$$\begin{aligned} &(x^*)^\top Ay^* \leq {x'}^\top Ay^* && \text{for all } x' \geq 0,\ 1^\top x' = 1, \\ &(x^*)^\top By^* \leq (x^*)^\top By' && \text{for all } y' \geq 0,\ 1^\top y' = 1. \end{aligned} \tag{2.3}$$

In other words, (x^*, y^*) is a Nash equilibrium if neither player can lower his/her expected cost by changing only his/her strategy. More specifically, (x^*, y^*) is a Nash equilibrium if and only if x^* is a best response to y^*, and vice versa. A mixed strategy x^* of Player I is called a *best response* to Player II's mixed strategy y^* if the former condition holds in (2.3). The definition for Player II's mixed strategy is similar.

Theorem 2.4 (von Stengel [125]). *For a bimatrix game (A, B), Player I's mixed strategy x^* is a best response to Player II's strategy y^* if and only if it holds that for each $i \in K$,*

$$x_i > 0 \ \Rightarrow\ (Ay)_i = \max_{j \in L}(Ay)_j.$$

It is well known that every bimatrix game has a Nash equilibrium.

Theorem 2.5 (Nash [102, 103]). *Every bimatrix game has a Nash equilibrium.*

The proof in 1950 was based on Kakutani fixed-point theorem [76], and the one in 1951 uses Brouwer fixed-point theorem [8]. Kakutani fixed-point theorem is a generalization of Brouwer fixed-point theorem: Let B^d be the d-dimensional closed unit ball.

Theorem 2.6 (Brouwer [8]). *Every continuous function $f : B^d \to B^d$ has a fixed point, i.e., a point $x \in B^d$ such that $f(x) = x$.*

The proof of Brouwer's theorem was not constructive, and thus Nash's theorem did not give a constructive procedure to compute a Nash equilibrium of the bimatrix game. Later, algorithms to find a Nash equilibrium were proposed by Vorob'ev [136], Kuhn [90], Mangasarian [96], and Lemke and Howson [94]. However, none of those is known to be a polynomial-time algorithm, and moreover, finding a Nash equilibrium is considered to be hard. This doubt was brought by the work of Cheng and Deng [12] which showed that the problem of finding a Nash equilibrium of the bimatrix game is a complete problem of the complexity class PPAD. See also [14].

The class *PPAD* (Polynomial Parity Arguments on Directed graphs), introduced by Papadimitriou [109], is the set of search problems that can be reduced to the end-of-the-line problem. The *end-of-the-line* problem is, given a directed graph G and a specified unbalanced vertex of G, to find some other unbalanced vertex, where a vertex in a directed graph is said to be *unbalanced* if the number of its incoming edges differs from the number of its outgoing edges. We remark that problems in PPAD are known to have a solution, while finding the solution is not an easy task. The class PPAD contains many problems of finding a fixed point. The problem of finding a fixed point in Brouwer's theorem is PPAD-complete [109]. The problem of finding a Nash equilibrium of the bimatrix game was shown to belong to PPAD in [109], and its completeness was shown in [12]. As described in [35], PPAD contains many problems for which researchers have tried for decades to develop efficient algorithm, and it is still open whether these problems are solvable efficiently. For this reason, PPAD-complete problems are considered to be hard.

2.3 Integer linear programming

In this section, we consider an integral solution to the linear programming problem. Given an integral matrix A in $\mathbb{Z}^{m\times n}$, and integral vectors b in $\mathbb{Z}^m$ and c in $\mathbb{Z}^n$, the *integer linear programming* is to find an integral vector x in $\mathbb{Z}^n$ which attains $\max\{c^\top x \mid Ax \le b,\ x \in \mathbb{Z}^n\}$.

For a polyhedron P, the convex hull of the integral vectors in P is called the *integer hull* of P, denoted by P_I. It is known that the integer hull of a polyhedron is also a polyhedron. We say that a polyhedron $P = \{x \mid Ax \le b\}$ is *rational* if A is a rational matrix and b is a rational vector.

Theorem 2.7 (Meyer [98]). *For any rational polyhedron P, the integer hull P_{I} of P is a rational polyhedron.*

Thus we can write an instance of the integer linear programming as the one of the linear programming problem $\max\{c^\top x \mid x \in P_{\mathrm{I}}\}$, where $P = \{x \mid Ax \leq b\}$, if once we can know the expression $P_{\mathrm{I}} = \{x \mid A'x \leq b'\}$.

It is known that the integer linear programming problem is NP-hard, while the linear programming problem is polynomial-time solvable. This result comes from the NP-hardness (essentially due to Cook [20]) of the problem of finding an integral feasible solution to $Ax \leq b$, where A is a rational matrix, and b is a rational vector. However, for a polyhedron P appearing in a combinatorial optimization, P_{I} often coincides with P. In this case, the integer linear programming problem reduces to the linear programming problem, and hence we can find solve it in polynomial time. Therefore, such conditions under which $P = P_{\mathrm{I}}$ holds have been studied extensively. In the rest of this section, we present those conditions and related work. The following are fundamental results.

Lemma 2.8 (Kronecker [89]). *Let A be a rational matrix, and let b be a rational vector. A linear equation $Ax = b$ has an integral solution if and only if $y^\top b$ is an integer for each rational vector y such that $y^\top A$ is integral.*

Lemma 2.9 (Schrijver [120, Chapter 4]). *Let A be an $n \times k$ integral matrix of full column rank. Then the g.c.d of $k \times k$ subdeterminants of A is one if and only if for any vector x such that Ax is an integral vector, x is integral.*

2.3.1 Integer hull of a polyhedron

Solving the integer linear programming problem is not easy. However, if we know the integer hull is nonempty, then we can know whether or not the integer linear programming problem is finite by considering the corresponding linear programming problem.

Proposition 2.10. *Let A be a rational matrix, let b be a rational vector, and let c be a vector. Assume that the integer hull of $P = \{x \mid Ax \leq b\}$ is nonempty. Then $\max\{c^\top x \mid x \in P\}$ is bounded if and only if $\max\{c^\top x \mid x \in P_{\mathrm{I}}\}$ is bounded.*

For an integral matrix A, we denote by $\Xi(A)$ the maximum absolute value among determinants of submatrices of A. We denote by $||x||_\infty$ the maximum absolute value among elements of x.

Theorem 2.11 (Cook, Gerards, Schrijver and Tardos [21]). *Let A be an integral matrix, and let b, c be vectors. Let $P = \{x \mid Ax \leq b\}$, and suppose that $P_{\mathrm{I}} \neq \emptyset$. If there exists an optimal solution x to $\max\{c^\top x \mid x \in P\}$, then there also exists optimal solution z to $\max\{x \mid x \in P_{\mathrm{I}}\}$ with $||z - x||_\infty \leq n\Xi(A)$.*

By using the above properties, we know the bound on the size of an optimal solution to the integer linear programming problem. We measure the size of numbers, vectors, and matrices as the number of bits in the binary representation. We define $\mathrm{size}(n) = 1 + \lceil \log_2(|n|+1) \rceil$ for an integer n, and $\mathrm{size}(r) = \mathrm{size}(p) + \mathrm{size}(q)$ for a rational number $r = \frac{p}{q}$, where p and q are relatively prime integers. For a rational vector b in $\mathbb{R}^n$, we write $\mathrm{size}(b) = n + \sum_{i=1}^{n} \mathrm{size}(b_i)$. For a rational matrix A in $R^{m\times n}$, we have $\mathrm{size}(A) = mn + \sum_{i,j} \lfloor \mathrm{size}(a_{ij}) \rfloor$. We have the following properties. See e.g. [87] for the detail.

Proposition 2.12. *If $r_1, \ldots, r_n$ are rational numbers, then it holds that $\mathrm{size}(r_1 \cdots r_n) \leq \mathrm{size}(r_1) + \cdots + \mathrm{size}(r_n)$, and $\mathrm{size}(r_1 + \cdots + r_n) \leq 2(\mathrm{size}(r_1) + \cdots + \mathrm{size}(r_n))$.*

Proposition 2.13. *For any rational square matrix A, it holds that $\mathrm{size}(\det A) \leq 2\mathrm{size}(A)$.*

To prove the bound of an optimal solution to the integer linear programming problem, we need a similar result on the linear programming. See e.g. [87] for the proof.

Lemma 2.14. *Let A be a rational matrix, and let b be a rational vector. Assume that $\max\{c^\top x \mid Ax \leq b\}$ has an optimal solution. Then it also has an optimal solution x such that $\mathrm{size}(x_i) \leq 4(\mathrm{size}(A) + \mathrm{size}(b))$ for each element x_i.*

Lemma 2.15. *Let A be an integral matrix, and let b, c be vectors. Let also $P = \{x \mid Ax \leq b\}$. If $\max\{c^\top x \mid x \in P_{\mathrm{I}}\}$ has an optimal solution, then it also has an optimal solution x such that $\mathrm{size}(x_i) \leq 13(\mathrm{size}(A) + \mathrm{size}(b))$ for each element x_i.*

Proof. By Proposition 2.10 and Lemma 2.14, $\max\{c^\top x \mid x \in P\}$ has an optimal solution y such that $\mathrm{size}(y_i) \leq 4(\mathrm{size}(A) + \mathrm{size}(b))$ for each element y_i. Theorem 2.11 implies that there exists an optimal solution x to $\max\{c^\top x \mid x \in P_{\mathrm{I}}\}$ with $||x - y||_\infty \leq n\Xi(A)$. This implies that $x_i \leq y_i + n\Xi(A)$ for each index i. Then we have

$$\begin{aligned}
\mathrm{size}(x_i) &\leq 2\mathrm{size}(y_i) + 2\mathrm{size}(n\Xi(A)) \\
&\leq 8(\mathrm{size}(A) + \mathrm{size}(b)) + 2(\mathrm{size}(n) + 2\mathrm{size}(A)) \\
&\leq 13(\mathrm{size}(A) + \mathrm{size}(b)).
\end{aligned}$$

□

2.3.2 Totally unimodular matrix

We say that a polyhedron P is *integral* if $P = P_{\mathrm{I}}$. One condition for a polyhedron $P = \{x \mid Ax \leq b\}$ to be integral is totally unimodularity of A. A matrix is called *totally unimodular* if all square submatrices have determinants 0 or ± 1. In particular, every entry of a totally unimodular matrix is 0 or ± 1.

Totally unimodular matrices often appear in combinatorial optimization. For example, the shortest-path problem, the max-flow min-cut problem, and the minimum cost flow problem can be naturally formulated as the linear programming problem, and the coefficient matrices of linear constraints are totally unimodular. In fact, the coefficient matrices of the linear programming problem occurring from these problems are incidence matrices of the directed graphs. It is well known that an incidence matrix of any directed graph is totally unimodular. On the other hand, for the undirected case, not every incidence matrix is totally unimodular.

Proposition 2.16. *For an undirected graph G, the incidence matrix of G is totally unimodular if and only if the graph G is bipartite.*

For a totally unimodular matrix A, the following matrices are also totally unimodular:

$$\begin{bmatrix} A & I \end{bmatrix}, \quad \begin{bmatrix} A & -A \end{bmatrix}, \quad \begin{bmatrix} A & O \\ I & I \end{bmatrix}.$$

Totally unimodular matrices are well studied in the literature of the integer linear programming problem, since they characterize integral polyhedra.

Theorem 2.17 (Hoffman and Kruskal [71])**.** *Let A be an integral matrix. Then A is totally unimodular if and only if the polyhedron $\{x \mid Ax \leq b,\ x \geq 0\}$ is integral for each integral vector b.*

2.3.3 Total dual integrality of linear systems

In the previous subsection, we mentioned that a polyhedron associated with a totally unimodular matrix is integral for any constant vector. A weaker concept than total unimodularity for integral polyhedra is total dual integrality, introduced by Edmonds and Giles [44].

We first study integral polyhedra. We say that a hyperplane $H = \{x \mid c^\top x = \delta\}$ is *rational* if c is a rational vector and δ is a rational number.

Proposition 2.18. *Let P be a rational polyhedron. Then the following are equivalent.*

(a) *P is integral.*

(b) *Each face of P contains integral vectors.*

(c) *Each minimal face of P contains integral vectors.*

(d) $\max\{c^\top x \mid x \in P\}$ *is attained by an integral vector for any vector c such that the maximum is finite.*

Proof. It is not difficult to see that (b) $\Leftrightarrow$ (c). We show (a) $\Rightarrow$ (b) $\Rightarrow$ (d) $\Rightarrow$ (a).

(a) $\Rightarrow$ (b): Let F be a face of P, and let x be a vector in F. Note that $F = P \cap H$, where H is a supporting hyperplane of P. Since $P = P_{\mathrm{I}}$, we can see that x can be represented as a convex combination of integral vectors in P, and these vectors belong to H. Thus F contains an integral vector.

(b) $\Rightarrow$ (d): For each vector c such that $\max\{c^\top x \mid x \in P\}$ is finite, a polyhedron $\max\{y \in P \mid c^\top y = \max\{c^\top x \mid x \in P\}\}$ (i.e., the set of optimal solutions) is a face of P.

(d) $\Rightarrow$ (a): Suppose that there exists a vector $y \in P \setminus P_{\mathrm{I}}$. Then by Theorem 2.7, there exists a linear inequality $\alpha^\top x \leq \beta$ such that $\alpha x \leq \beta$ holds for any vector x in P_{I}, and $\alpha y > \beta$. This contradicts (d), since $\max\{\alpha^\top x \mid x \in P\}$ is finite by Proposition 2.10, and the maximum is attained by a vector in $P \setminus P_{\mathrm{I}}$, which is not integral. $\square$

We have further equivalent conditions for a polyhedron to be integral.

Theorem 2.19 (Edmonds and Giles [44])**.** *A rational polyhedron P is integral if and only if each rational supporting hyperplane of P contains an integral vector.*

Proof. If P is integral, then for each supporting hyperplane H, the intersection of H with P is a face of P, and hence H contains an integral vector.

We assume that each supporting hyperplane contains an integral vector, and show that each minimal face of P contains an integral vector. Let us denote $P = \{x \mid Ax \leq b\}$, where A is a matrix, and b is a vector. Since P is rational, we may assume that A and b are integral. Let$F = \{x \mid A'x = b'\}$ be a minimal face of P , where $A'x \leq b'$ is a subsystem of $Ax \leq b$. If F contains no integral vector, then by Theorem 2.8, there exists a vector y such that $c = (A')^\top y$ is integral, and $\delta = (b')^\top y$ is not an integer. Since A and b are integral, we may assume $y > 0$ by adding a positive integer to elements of y if necessary. Let $H = \{x \mid c^\top x = \delta\}$ be a rational hyperplane. It is observed that since δ is not integer, whereas c is integral, H contains no integral vector, and this contradicts our assumption. Thus it suffices to show that H is a supporting hyperplane by proving $F = P \cap H$. It is not difficult to see $F \subseteq P \cap H$. Conversely, let x be a vector in $P \cap H$. It holds that $y^\top(A')x = c^\top x = \delta = (b')^\top y$, and hence $y^\top(A'x - b') = 0$ holds. Since $y > 0$ and $A'x \leq b'$ by definition, we have $A'x = b'$, which implies $x \in F$. Therefore, H is a supporting hyperplane of P. $\square$

Corollary 2.20. *Let A be a rational matrix, and let b be a rational vector. Then* $\max\{c^\top x \mid Ax \leq b\}$ *is attained by an integral vector for each vector c such that the maximum is finite if and only if* $\max\{c^\top x \mid Ax \leq b\}$ *is integer for each integral vector c such that the maximum is finite*

Proof. If c is integral, then $\max\{c^\top x \mid Ax \leq b\}$ is integral if it has an integral optimal solution. Thus the necessity holds.

Conversely, we assume that for each integral vector c, $\max\{c^\top x \mid Ax \leq b\}$ is an integer if it is finite. Let $H = \{x \mid c^\top x = \delta\}$ be a rational supporting hyperplane of $P = \{x \mid Ax \leq b\}$. By definition, we have$\max\{c^\top x \mid x \in P\} = \delta$. Suppose that H contains no integral vector. Then by Theorem 2.8, there exists a number γ such that γc is an integral vector and $\gamma\delta$ is not an integer. Then we have

$$\max\{(|\gamma|c)^\top x \mid x \in P\} = |\gamma| \max\{c^\top x \mid x \in P\} = |\gamma|\delta \notin \mathbb{Z},$$

and this contradicts our assumption. Thus H contains an integral vector. Therefore, by Proposition 2.18 and Theorem 2.19, $\max\{c^\top x \mid x \in P\}$ is attained an integral vector. □

Note that earlier partial results appeared also in papers of Chvátal [16], Fulkerson [57], Gomory [64], and Hoffman [70].

Corollary 2.20 motivated Edmonds and Giles [44] to define the total dual integrality of linear systems.

Definition 2.21. *Let A be a rational matrix, and let b be a rational vector. A system $Ax \leq b$ is* totally dual integral *if the dual problem of* $\max\{c^\top x \mid Ax \leq b\}$*, that is,*

$$\min\{b^\top y \mid A^\top y = c,\ y \geq 0\} \tag{2.4}$$

has an integral optimal solution for each integral vector c such that (2.4) *is finite.*

By this definition and Theorem 2.17, if A is a totally unimodular matrix, then $Ax \leq b$ is totally dual integral for each rational vector b.

Theorem 2.22 (Edmonds and Giles [44])**.** *Let A be a rational matrix, and let b be an integral vector. If $Ax \leq b$ is a totally dual integral system, then* $\max\{c^\top x \mid Ax \leq b\}$ *has an integral optimal solution for each vector c such that the maximum is finite.*

We have an equivalent formulation:

Theorem 2.23 (Edmonds and Giles [44])**.** *Let A be a rational matrix, and let b be an integral vector. If $Ax \leq b$ is a totally dual integral system, then the polyhedron $\{x \mid Ax \leq b\}$ is integral.*

It is not difficult to see that if a matrix A is totally unimodular, then a linear system $Ax \leq b$ is totally dual integral for every vector b. Hence, we can restate Theorem 2.17 as follows.

Corollary 2.24 (Hoffman and Kruskal [71])**.** *Let A be an integral matrix. Then A is totally unimodular if and only if the system $Ax \leq b,\ x \geq 0$ is totally dual integral for each vector b.*

Chapter 3

The linear complementarity problem

This chapter defines the linear complementarity problem, and also presents related works. In particular, we focus on matrix classes such as positive semi-definite matrices, $\boldsymbol{P}$-matrices, sufficient matrices, $\boldsymbol{Z}$-matrices, and $\boldsymbol{K}$-matrices.

3.1 Definition and equivalent formulations

Given an $n \times n$ real matrix M and a real vector q, the linear complementarity problem is to find a vector z such that

$$Mz + q \geq 0, \quad z \geq 0, \quad z^\top(Mz + q) = 0. \tag{3.1}$$

The last constraint $z^\top(Mz + q) = 0$ is called the *complementarity condition*. Under the assumption of $Mz + q \geq 0$ and $z \geq 0$, we can replace the complementarity condition with

$$z_i = 0 \ \text{ or } \ (Mz + q)_i = 0 \quad \text{for all } i = 1, \dots, n.$$

When we introduce a slack variable $w = Mz + q$, the LCP can be written as the problem of finding two vectors w, z such that

$$w = Mz + q, \quad w, z \geq 0, \quad z^\top w = 0. \tag{3.2}$$

The LCP can be interpreted geometrically. For an $n \times n$ real matrix M (with row and column index sets $[n]$) and index set $B \subseteq [n]$, define an $n \times n$ matrix $C_M(B)$ by

$$C_M(B)_{\cdot i} = \begin{cases} -M_{\cdot i} & \text{if } i \in B, \\ I_{\cdot i} & \text{if } i \notin B, \end{cases}$$

where I is the identity matrix. For a matrix A, let $\operatorname{pos} A$ denote a positive cone spanned by column vectors of A, i.e., $\operatorname{pos} A = \{Ax \mid x \geq 0\}$. $\operatorname{pos} C_M(B)$ is called the *complementary cone* of B relative to M. For example, if $M_1 = \begin{bmatrix} -1 & 1 \\ 1 & 2 \end{bmatrix}$, then the complementary cones are illustrated in Figure 3.1 below. Similarly, the complementary cones relative to $M_2 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}$ are shown in Figure 3.2.

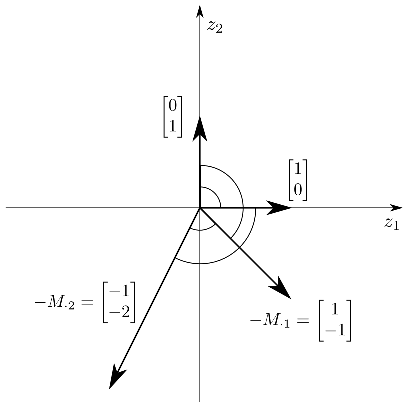

Figure 3.1. Complementary cones relative to M_1.

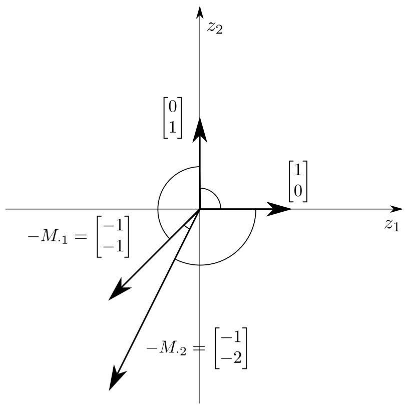

Figure 3.2. Complementary cones relative to M_2.

We say that a solution z to $\mathrm{LCP}(M, q)$ has a *basis* $B \subseteq [n]$ if it holds that $(Mz + q)_B = 0$

and $z_{\overline{B}} = 0$. Note that a vector q in $\mathbb{R}^n$ is contained in $\operatorname{pos} C_M(B)$ for some $B \subseteq [n]$ if and only if LCP(M, q) has a solution z with basis B. To see this, suppose that q is in $\operatorname{pos} C_M(B)$, i.e., there exists a vector x in $\mathbb{R}^n$ with $C_M(B)x = q$ and $x \geq 0$. Then a vector z defined by $z_B = x_B$ and $z_{\overline{B}} = 0$ is a solution to LCP(M, q), since it holds that $z \geq 0, z_{\overline{B}} = 0, (Mz + q)_B = M_{BB}x_B + q_B = 0$, and $(Mz + q)_{\overline{B}} = M_{\overline{B}B}x_B + q_{\overline{B}} \geq 0$. On the other hand, if LCP(M, q) has a solution z with basis B, then a vector x defined by $x_B = z_B$ and $x_{\overline{B}} = M_{\overline{B}B}z_B + q_{\overline{B}}$ is contained in $\operatorname{pos} C_M(B)$. Let $K(M) = \bigcup_{B \subseteq [n]} \operatorname{pos} C_M(B)$. Note that $K(M)$ is not necessarily convex. Figure 3.1 also illustrates such a case.

As we mentioned in the introduction, the LCP is a generalization of the convex quadratic problem. We can also reduce LCP(M, q) to the quadratic problem

$$\begin{aligned} \text{min.} \quad & z^\top (Mz + q) \\ \text{s.t.} \quad & Mz + q \geq 0, \\ & z \geq 0. \end{aligned} \tag{3.3}$$

The objective function $z^\top (Mz+q)$ is bounded below by zero. It is observed that this quadratic problem has the optimal value equal to zero if and only if LCP(M, q) has a solution.

A solution z to LCP(M, q) is called a *basic solution* (*with respect to* B) if z is of the form

$$z_B = -M_{BB}^{-1} q_B, \quad z_{\overline{B}} = 0. \tag{3.4}$$

Let us notice that LCP(M, q) has a basic solution with respect to B if and only if $q \in \operatorname{pos} C_M(B)$ and M_{BB} is nonsingular.

Let LCP(M, q) be an LCP instance. We define two polyhedra associated with LCP(M, q) by

$$\begin{aligned} P(M, q) &= \{z \mid Mz + q \geq 0,\ z \geq 0\}, \text{ and} \\ P_B(M, q) &= \{z \in P(M, q) \mid (Mz + q)_B = 0,\ z_{\overline{B}} = 0\} \text{ for } B \subseteq [n]. \end{aligned}$$

Note that $P_B(M, q)$ is a face of $P(M, q)$. $\bigcup_{B \subseteq [n]} P_B(M, q)$ represents the set of solutions to LCP(M, q). Thus LCP(M, q) is equivalent to finding a nonempty set $P_B(M, q)$ for some B. Recall that a solution z to LCP(M, q) is called a *basic solution* (*with respect to* B) if z is of the form

$$z_B = -M_{BB}^{-1} q_B, \quad z_{\overline{B}} = 0.$$

We remark that LCP(M, q) has a basic solution with respect to B if and only if $q \in \operatorname{pos} C_M(B)$ and M_{BB} is nonsingular. This is equivalent to the condition that $P_B(M, q)$ is a vertex of $P(M, q)$.

3.2 Formulations as the linear complementarity problem

We present reductions of the linear programming problem and the problem of finding a Nash equilibrium of the bimatrix game to the LCP.

3.2.1 Linear programming problem

We consider the linear programming problem

$$\max\{c^\top x \mid Ax \le b,\ x \ge 0\},$$

where A is a matrix, and b, c are vectors. We assume that the maximum is finite. Then its dual problem $\min\{b^\top y \mid A^\top y \ge c,\ y \ge 0\}$ has an optimal solution. Any pair (x, y) of optimal solutions to the primal and dual linear programming problems satisfies

$$-Ax + b \ge 0, \quad x \ge 0, \quad x^\top (A^\top y - c) = 0, \tag{3.5}$$

$$A^\top y \ge c, \quad y \ge 0, \quad y^\top (-Ax + b) = 0. \tag{3.6}$$

If we define

$$M = \begin{bmatrix} O & A^\top \\ -A & O \end{bmatrix}, \quad q = \begin{bmatrix} -c \\ b \end{bmatrix}, \quad z = \begin{bmatrix} x \\ y \end{bmatrix},$$

then system (3.5) and (3.6) coincides with LCP(M, q).

3.2.2 Convex quadratic programming problem

Consider the convex quadratic programming problem

$$\begin{aligned} \text{min.} \quad & \frac{1}{2} x^\top Q x + c^\top x \\ \text{s.t.} \quad & Ax \ge b, \\ & x \ge 0, \end{aligned} \tag{3.7}$$

where Q is a positive semi-definite matrix, A is a matrix, and b, c are vectors. Note that the case $Q = O$ coincides with the linear programming problem. The Karush-Kuhn-Tucker condition is written as

$$\begin{aligned} Qx - A^\top y + c \ge 0, \quad & x \ge 0, \quad x^\top (Qx - A^\top y + c) = 0, \\ Ax - b \ge 0, \quad & y \ge 0, \quad \qquad\qquad y^\top (Ax - b) = 0. \end{aligned}$$

Since Q is positive semi-definite, any vector satisfying the Karush-Kuhn-Tucker condition is an optimal solution to (3.7). The above conditions define LCP(M, q), where

$$M = \begin{bmatrix} Q & -A^\top \\ A & O \end{bmatrix}, \quad q = \begin{bmatrix} c \\ -b \end{bmatrix}.$$

3.2.3 Bimatrix game

We reduce the bimatrix game (A, B) to an LCP instance. We may assume that A and B are positive matrices by adding a large positive number to all entries of A and B, since this modification does not affect a Nash equilibrium. Let S be the set of Nash equilibria of the game (A, B). Then the problem of finding a Nash equilibrium of a bimatrix game is reformulated as the LCP as below (see e.g., [27, Chapter 1]).

Claim 3.1. *The problem of finding* (x^*, y^*) *in* S *is equivalent to finding two vectors* x, y *satisfying*

$$\begin{aligned} Ay - 1 \geq 0, \quad x \geq 0, \quad x^\top(Ay - 1) = 0, \\ B^\top x - 1 \geq 0, \quad y \geq 0, \quad y^\top(B^\top x - 1) = 0. \end{aligned} \tag{3.8}$$

More precisely, for any (x^*, y^*) *in* S*, vectors* x *and* y *defined by*

$$x = x^*/(x^*)^\top B y^*, \ y = y^*/(x^*)^\top A y^* \tag{3.9}$$

satisfy (3.8)*. Conversely, for any vectors* x *and* y *satisfying* (3.8)*,* (x^*, y^*) *defined by*

$$x^* = x/1^\top x, \ y^* = y/1^\top y \tag{3.10}$$

belongs to S*.*

Note that (3.8) coincides with the LCP instance LCP$(M, -1)$, where

$$M = \begin{bmatrix} O & A \\ B^\top & O \end{bmatrix}.$$

3.3 Matrix classes

In the theoretical study of the LCP, the matrix classes play a central role. We define some of well-known matrix classes and present related works on the LCP with those matrices.

3.3.1 Positive semi-definite and positive definite matrices

Recall that a matrix M is called *positive semi-definite* (PSD) if for any vector x, it holds that $x^\top Mx \geq 0$. If $x^\top Mx > 0$ for any vector $x \neq 0$, then M is called *positive definite* (PD). We do not assume that the matrix M is symmetric (i.e., $M = M^\top$). For example, any *skew-symmetric* matrix M (i.e., $M = -M^\top$) is a PSD-matrix. The following matrices are examples of PSD-matrices and PD-matrices, respectively:

$$\begin{bmatrix} 1 & -2 & -2 \\ 2 & 3 & 0 \\ 2 & -2 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & -2 & 0 \\ 2 & 3 & 2 \\ 0 & -2 & 1 \end{bmatrix}.$$

We denote by PSD-LCP and PD-LCP the LCP with PSD-matrices and PD-matrices, respectively. The PSD-LCP arises from the reductions of the linear and convex quadratic programming problems to the LCP.

One fundamental property of the PSD-LCP is the following existence result. We note that fundamental properties of the PSD-LCP and the PD-LCP are collected in the book [27, Chapter 3], and our description is based on this book.

Theorem 3.2. *Let M be a PSD-matrix, and let q be a vector. If linear system $Mz + q \geq 0,\ z \geq 0$ is feasible, then* $\mathrm{LCP}(M, q)$ *has a solution.*

This can be proved by showing that there exists a vector satisfying the Karush-Kuhn-Tucker condition of the quadratic programming problem (3.3):

$$\begin{aligned} (M + M^\top)z - M^\top u + q &\geq 0, \\ z^\top((M + M^\top)z - M^\top u + q) &= 0, \\ u &\geq 0, \\ u^\top(Mz + q) &= 0. \end{aligned} \tag{3.11}$$

In addition, for PD-matrices, we have a stronger result.

Theorem 3.3. *If a matrix M is a PD-matrix, then* $\mathrm{LCP}(M, q)$ *has a unique solution for any vector q.*

Another known property of the PSD-LCP is that the solution set of the PSD-LCP is convex. Recall that the solution set of $\mathrm{LCP}(M, q)$ of order n is represented as the union of convex polyhedra, that is, $\bigcup_{B \subseteq [n]} P_B(M, q)$, and this is not convex in general.

Theorem 3.4. *Let M be a PSD-matrix, and let q be a vector. If* $\mathrm{LCP}(M, q)$ *has a solution, then the solution set of* $\mathrm{LCP}(M, q)$ *can be represented as a polyhedron*

$$\{z \mid Mz + q \geq 0,\ z \geq 0,\ q^\top(z - z^*) = 0,\ (M + M^\top)(z - z^*) = 0\},$$

where z^* is an arbitrarily solution to $\mathrm{LCP}(M, q)$*.*

3.3.2 $\boldsymbol{P}$-matrix

For a square matrix M in $\mathbb{R}^{n\times n}$ and an index set $B \subseteq [n]$, a submatrix M_{BB} of M is called a *principal submatrix*. The determinant of a principal submatrix is a *principal minor*. We say that a square matrix M is a $\boldsymbol{P}$*-matrix* if all principal minors of M are positive. Here is an example of $\boldsymbol{P}$-matrices:

$$M = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 2 & 3 \\ 2 & 1 & 2 \end{bmatrix}.$$

By definition, if a matrix M is a $\boldsymbol{P}$-matrix, then all of its principal submatrices M_{BB} and its transpose $M^\top$ are also $\boldsymbol{P}$-matrices. It is known that a symmetric matrix is PD if and only if it is a $\boldsymbol{P}$-matrix. Every PD-matrix is a $\boldsymbol{P}$-matrix.

Theorem 3.5. *Let M be a square matrix in $\mathbb{R}^{n\times n}$. The following are equivalent.*

(a) *M is a $\boldsymbol{P}$-matrix.*

(b) *For any vector x, it holds that*

$$[z_i(Mz)_i \leq 0\ (\forall i)] \Rightarrow [z = 0].$$

(c) *All real eigenvalues of M and its principal submatrices are positive.*

The notion of $\boldsymbol{P}$-matrices gives a characterization for the class of matrices M such that $\mathrm{LCP}(M, q)$ has a unique solution for any vector q.

Theorem 3.6 (Samelson, Thrall and Wesler [117])**.** *A square matrix M is a $\boldsymbol{P}$-matrix if and only if* $\mathrm{LCP}(M, q)$ *has a unique solution for any vector q.*

This theorem implies Theorem 3.3. Theorem 3.6 originally comes from geometry. In fact, every complementary cone relative to a $\boldsymbol{P}$-matrix M of order n is a full dimensional cone, and does not intersect the interior of other complementary cones, and in addition, $K(M)$ coincides with $\mathbb{R}^n$, and vise versa. Figure 3.2 illustrates such an example.

We say that the LCP with $\boldsymbol{P}$-matrices is the $\boldsymbol{P}$-*LCP*. The $\boldsymbol{P}$-LCP has a combinatorial algorithm called a *simple principal pivoting method* proposed by Zoutendjik [140] and Bard [4]. The idea of this method is the same as the simplex method for the linear programming problem. We use the definition (3.2) of the LCP, that is, finding a pair (w, z) of vectors satisfying (3.2). We assume that a solution (w, z) to the input LCP instance LCP(M, q) has no index i such that $z_i = w_i = 0$. This assumption is not restrictive; we add some perturbation to M and q if necessary. Then a simple principal pivoting method is described as follows.

A simple principal pivoting method

Input: A $\boldsymbol{P}$-matrix M and a vector q.

Step 0: Let $z = 0$, $w = Mz + q$, and let also $B = \emptyset$.

Step 1: If w_B, $z_{\overline{B}} \geq 0$, then return (w, z).

Step 2: Choose an index i such that $z_i < 0$ or $w_i < 0$.

Step 3: Replace B with $B \cup \{i\}$ if $i \notin B$ and $B \setminus \{i\}$ otherwise, and compute w and z by pivoting i. Then go to Step 1.

At Step 2, we need a pivoting rule that defines which index should be selected. Since calculation of w and z at Step 3 can be done in $\mathrm{O}(n^2)$ time, the running time of a simple principal pivoting method depends largely on the number of iterations of Step 3, and this is dominated by the pivoting rule. There exist several pivoting rules [55, 99, 100]. However, it is also known that for most of existing rules, a simple principal pivoting method takes an exponential number of iterations. Thus it still remains open whether a simple principal pivoting method is a polynomial-time algorithm for the $\boldsymbol{P}$-LCP or not.

Settlement of the polynomial-time solvability of the $\boldsymbol{P}$-LCP is significant in the work of the LCP. Many researchers have been tackling this issue, and there are many results which suggest the polynomial-solvability of the $\boldsymbol{P}$-LCP. Megiddo [97] showed that the NP-hardness of the $\boldsymbol{P}$-LCP implies NP=coNP. Papadimitriou showed that the $\boldsymbol{P}$-LCP belongs to PPAD [109], while its completeness is considered to be unlikely [37]. We should remark that although the $\boldsymbol{P}$-LCP always has a solution, the problem of deciding whether or not a matrix is a $\boldsymbol{P}$-matrix was shown to be coNP-complete [30].

Stickney and Watson proposed a combinatorial abstraction of the $\boldsymbol{P}$-LCP based on principal pivoting [126]. Recall that a simple principal pivoting method solves the $\boldsymbol{P}$-LCP by pivoting one index per iteration. For finite sets A, B, we denote $A \triangle B = (A \cup B) \setminus (A \cup B)$. For a $\boldsymbol{P}$-LCP instance LCP(M, q) of order n, we consider a directed graph $G = (V, E)$ given by

$$V = \{B \mid B \subseteq [n]\}, \quad E = \{(B, B') \mid |B \triangle B'| = \{i\},\ (C_M(B)^{-1}q)_i < 0\}.$$

The underlying undirected graph is an n-dimensional hypercube. We note that if a directed edge (B, B') is contained in E, then (B', B) is not contained in E, since we have $(C_M(B')q)_i > 0$ $(i \in B \triangle B')$.

This directed graph G has an important property. Every nonempty subcube, including the whole cube, has a unique sink [126]. A directed hypercube possessing this property is said to have a *unique sink orientation*, whose notion was introduced by Szabó and Welzl [132]. The global sink corresponds the basis of the unique solution to LCP(M, q) by definition.

To resolve the computational complexity of the $\boldsymbol{P}$-LCP, a characterization of unique sink orientations induced by the $\boldsymbol{P}$-LCP has been studied intensively. Gärtner, Morris and Rüst [62] showed that unique sink orientations induced by the $\boldsymbol{P}$-LCP of order n satisfy the Holt-Klee condition [72]: There exist n vertex-disjoint paths from the source to the sink. Foniok,Gärtner, Klaus and Sprecher [50] obtained a result on counting the unique sink orientations induced by the $\boldsymbol{P}$-LCP.

There are algorithms to find a unique sink in a unique sink orientation, assuming that we are given an oracle that receives an arbitrary vertex of the hypercube, and returns the orientations of the incident edges. This problem is also an abstraction of the problem of finding the smallest enclosing ball of a set of points [132]. For unique sink orientations without cycles, Gärtner [60, 61] proposed a randomized procedure to find a unique sink with an expected number of at most $e^{2\sqrt{n}}$ oracle calls. For the general case, Szabó and Welzl [132] presented a deterministic procedure with $\mathrm{O}(1.61^n)$ oracle calls. We can see that the $\boldsymbol{P}$-LCP can be solved in $\mathrm{O}(c^n)$ time $(c < 2)$ by applying their algorithm. Note that there exists a unique sink orientation induced by the $\boldsymbol{P}$-LCP which contains a cycle.

3.3.3 Sufficient matrix

Recall that Theorem 3.2, 3.3 and 3.4 present important characteristics of the PSD-LCP and PD-LCP. Theorem 3.3 is extended to Theorem 3.6 by generalizing PD-matrices to $\boldsymbol{P}$-matrices. Cottle, Pang and Venkateswaran [28] introduced the class of sufficient matrices as a generalization of PSD-matrices, and showed an extension of Theorem 3.2 and 3.4.

A square matrix M is called *column sufficient* if for all vectors x, it holds that

$$[x_i(Mx)_i \leq 0 \ (\forall i)] \Rightarrow [x_i(Mx)_i = 0 \ (\forall i)].$$

A matrix M is *row sufficient* if $M^\top$ is column sufficient, and *sufficient* if it is both column and row sufficient. The class of (column) sufficient matrices contains PSD-matrices and $\boldsymbol{P}$-

matrices. Here is an example of row sufficient matrices:

$$\begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & -1 \\ 2 & -1 & 1 \end{bmatrix}.$$

The row-sufficiency of a matrix gives an extension of Theorem 3.2.

Theorem 3.7 (Cottle, Pang and Venkateswaran [28]). *For a square matrix M, the following are equivalent.*

(a) *M is row sufficient.*

(b) *For each vector q, if there exists a pair (z, u) of vectors satisfying the Karush-Kuhn-Tucker condition* (3.11) *of the quadratic programming problem* (3.3)*, then z is a solution to* $\mathrm{LCP}(M, q)$.

The convexity of the solution set of the LCP is characterized by column sufficient matrices.

Theorem 3.8 (Cottle, Pang and Venkateswaran [28]). *For a square matrix M, the following are equivalent.*

(a) *M is column sufficient.*

(b) *For each vector q,* $\mathrm{LCP}(M, q)$ *has a (possibly empty) convex solution set.*

Sufficient matrices possess the above good properties. In addition, the existence of a solution to the LCP with sufficient matrices is characterized by *duality* of the LCP, introduced by Fukuda and Terlaky [56].

Theorem 3.9 (Fukuda and Terlaky [56]). *Let M be a sufficient matrix, and let q be a vector. Then exactly one of the following holds.*

(a) *There exists a vector z satisfying $Mz + q \geq 0,\ z \geq 0,\ z^\top(Mz + q) = 0$.*

(b) *There exists a vector u satisfying $q^\top u = -1,\ u \geq 0,\ A^\top u \leq 0,\ u^\top A^\top u = 0$.*

This duality theorem implies the duality of the linear programming, since the LCP instances (3.5) and (3.6) have PSD-matrices.

3.3.4 $\boldsymbol{Z}$-matrix

A square matrix is called a $\boldsymbol{Z}$*-matrix* if all off-diagonal entries are nonpositive. For example, the matrix

$$M = \begin{bmatrix} 0 & -1 & -2 \\ -2 & -1 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

is a $\boldsymbol{Z}$-matrix. The class of $\boldsymbol{Z}$-matrices is invariant under transposition. We say that the LCP with $\boldsymbol{Z}$-matrices is the $\boldsymbol{Z}$*-LCP*. The polyhedron associated with the $\boldsymbol{Z}$-LCP has an interesting property.

Let $Ax \geq b$ be a system of linear inequalities such that each row of A has at most one positive element. Let $F = \{x \mid Ax \geq b\}$ be the feasible region. It is well known, as observed by Veinott [135], that F is closed under the *meet* $\wedge$, i.e., F contains the meet $x \wedge y$ for any $x, y \in F$, where $(x \wedge y)_i = \min(x_i, y_i)$. Indeed, for each inequality $a_i x_i - \sum_{j \neq i} a_j x_j \geq b_k$, where $a_i \geq 0$ and $a_j \geq 0$ $(j \neq i)$, we may assume that $(x \wedge y)_i = y_i$, and it then holds that $a_i (x \wedge y)_i - \sum_{j \neq i} a_j (x \wedge y)_j \geq a_i y_i - \sum_{j \neq i} a_j y_j \geq b_k$, which implies that $x \wedge y \in F$. Then as discussed in [27, Theorem 3.11.5], the following holds. Another proof can be found in [29].

Lemma 3.10. *Let $Ax \geq b$ be a system of linear inequalities such that each row of A has at most one positive element. Let $F = \{x \mid Ax \geq b\}$ be the feasible region. If F is nonempty and bounded below, then there exists a vector $u \in F$ such that $x \geq u$ for any $x \in F$.*

The vector u in this lemma is called the *least element* of F. It is well known that we can find the least element, if it exists, by solving a linear programming problem.

Lemma 3.11. *Let F be the feasible region of a system of linear inequalities. If the least element of F exists, then it is the unique optimal solution to the linear programming problem*

$$\begin{aligned} \text{min.} \quad & 1^\top x \\ \text{s.t.} \quad & x \in F. \end{aligned} \tag{3.12}$$

Let $\mathrm{LCP}(M, q)$ be a $\boldsymbol{Z}$-LCP instance. It is not difficult to see that the polyhedron $P(M, q) = \{z \mid Mz + q \geq 0,\ z \geq 0\}$ is closed under the meet. We can see that $P(M, q)$ is bounded below by 0. Thus by Lemma 3.10, $P(M, q)$ has a least element. In addition, the least element is a solution to $\mathrm{LCP}(M, q)$. See e.g. [27] for the proof.

Theorem 3.12. *Let M be a $\boldsymbol{Z}$-matrix, and let q be a vector. If the polyhedron $P(M, q)$ is not empty, then $P(M, q)$ has a least element u. Moreover, u is a solution to* $\mathrm{LCP}(M, q)$.

Therefore, we can find a solution to the $\boldsymbol{Z}$-LCP by solving a linear programming problem (3.12).

Before it turned out that the linear programming problem can be solved in polynomial time, Chandrasekaran [9] proposed a polynomial-time pivoting algorithm for the $\boldsymbol{Z}$-LCP. More precisely, the algorithm solves the $\boldsymbol{Z}$-LCP of order n in $\mathrm{O}(n)$ steps, and hence the total running time is $\mathrm{O}(n^3)$ time. Saigal [116] showed the greediness of Chandrasekaran's algorithm.

3.3.5 $\boldsymbol{K}$-matrix

If a matrix M is a $\boldsymbol{Z}$-matrix, and also a $\boldsymbol{P}$-matrix, then M is called a $\boldsymbol{K}$*-matrix*. Fiedler and Pták [47] showed as many as 13 equivalent conditions for a $\boldsymbol{Z}$-matrix to be a $\boldsymbol{K}$-matrix. We list particularly useful conditions in the literature of the LCP.

Theorem 3.13 (Fiedler and Pták [47])**.** *Let A be a $\boldsymbol{Z}$-matrix. Then the following are equivalent.*

(a) *A is a $\boldsymbol{K}$-matrix.*

(b) *All leading principal minors of M are positive.*

(c) *The inverse A^{-1} exists and $A^{-1} \geq 0$.*

(d) *There exists a vector $x \geq 0$ such that $Ax > 0$.*

(e) *There exists a vector $x > 0$ such that $Ax > 0$.*

We call the LCP with $\boldsymbol{K}$-matrices the $\boldsymbol{K}$-LCP. By definition, we can solve the $\boldsymbol{K}$-LCP in strongly polynomial time by using Chandrasekaran's algorithm [9] for the $\boldsymbol{Z}$-LCP. In addition, the $\boldsymbol{K}$-LCP possesses a combinatorial structure. Foniok, Fukuda, Gärtner and Lüthi [49] showed that unique sink orientations induced by the $\boldsymbol{K}$-LCP of order n have only paths of length at most $2n$. This implies that a simple principal pivoting method with any pivoting rules can solve the $\boldsymbol{K}$-LCP in a linear number of iterations.

Chapter 4

The linear complementarity problem with a few variables per constraint

The full text of this chapter cannot be publicized because a copyright holder's consent has not been obtained.

Chapter 5

Parameterized complexity of the linear complementarity problem

The full text of this chapter cannot be publicized because a copyright holder's consent has not been obtained.

Chapter 6

The linear complementarity problem with orientation

The full text of this chapter cannot be publicized because a copyright holder's consent has not been obtained.

Chapter 7

Total dual integrality of the linear complementarity problem

The full text of this chapter cannot be publicized because a copyright holder's consent has not been obtained.

Chapter 8

Conclusion

The LCP has been studied since 1960's, and it has many theoretical and practical applications in many areas. Although the LCP is NP-hard, there still exist many things to study, such as identifying polynomial-time solvable cases, and developing efficient algorithms for the LCP with a small parameter. In this thesis, we have studied the LCP from theoretical aspects.

In Chapter 4, we analyzed the computational complexity of the sparse LCP, i.e., the k-LCP in which the coefficient matrix has at most k nonzero entries per row. As a main result, we showed that the 2-LCP is NP-hard, while the sign-balanced 2-LCP is strongly polynomial-time solvable by presenting a combinatorial algorithm based on procedures for solving sign-balanced TVPI systems. This completely classifies the computational complexity of the LCP in terms of sparsity. We note that the algorithm for the sign-balanced 2-LCP matches the currently best-known bound for the feasibility problem of sign-balanced TVPI systems with nonnegativity constraints, which can be reduced to the sign-balanced 2-LCP.

In Chapter 5, we analyzed the parameterized complexity of the linear complementarity problem. Analysis of parameterized computation of a Nash equilibrium of the bimatrix game has been done earlier. We cannot simply apply the previously known results to the LCP, since a natural reduction of the bimatrix game to the LCP destroys the sparsity of the matrix. Nevertheless, the underlying ideas were useful for the LCP. We showed that the problem of finding a solution to the l-sparse LCP of order n with support of size at most s can be solved in time exponential in l and s, but polynomial in n. If we know that parameters l and s are comparatively small, then finding a desired solution is an easy task.

In Chapter 6, we introduced the LCP with orientation to formulate the problem of finding a solution to an LCP instance whose basis is identical to the paired LCP instance. The LCP with orientation is a generalization of the LCP. We analyzed the computational complexity of the LCP with orientation in terms of matrix classes. Then we showed that the LCP with positive semi-definite matrices, $\boldsymbol{P}$-matrices, and $\boldsymbol{Z}$-matrices is NP-hard, while the LCP with skew-symmetric matrices and $\boldsymbol{K}$-matrices can be solved in polynomial time.

In Chapter 7, we discussed integrality of the LCP. We defined total dual integrality of the LCP with the aid of the LCP with orientation. We showed that the total dual integrality of the LCP implies that a basic solution to the LCP instances is always integral. Our result corresponds to the relationship between the total dual integrality of linear systems and integrality of the linear programming problems. For the LCP with sufficient matrices, or rank-symmetric matrices, we showed that the total dual integrality of the LCP guarantees the integrality of the LCP, in the sense that an integral solution exists among solutions with the same basis. Our results gave alternative proofs to the results in [11, 31].

One important open problem in the theory of LCP is the computational complexity of the $\boldsymbol{P}$-LCP. Besides a simple principal pivoting method, there exist algorithms to solve the $\boldsymbol{P}$-LCP based on numerical data of problem instances. Polynomial-time algorithms for the $\boldsymbol{P}$-LCP remains open, and it is also unknown whether it is PPAD-complete or not. However, people believe that the $\boldsymbol{P}$-LCP is unlikely to be hard, relying on Megiddo's result [97].

One of the future works of this thesis is an analysis of the parameterized complexity of the LCP with some restriction on the matrix class. It might be worth studying the $\boldsymbol{P}$-LCP for the purpose of bringing difficulty of the $\boldsymbol{P}$-LCP to light.

The study of the relationships between the LCP with orientation and related problems is also one future work. One of the problems is a generalization of the LCP, called the the oriented matroid complementarity problem. Indeed, the duality theorem of Fukuda and Terlaky [56] (Theorem 3.9) was shown via the oriented matroid theory. It is interesting if some properties of the LCP with orientation can be described by using the oriented matroid theory. For other topics, we mention the generalized LCP and a unique sink orientation.

Constriction of efficient algorithms for the LCP is an important task. An algorithm to find an integral solution of the LCP is one of the future works, as there exist algorithms for the integer linear programming problem such as a cutting-plane method and a branch-and-bound method. A faster fpt-algorithm than the one in Section **??** is also required. Although the fpt-algorithm for the p-s-support l-sparse LCP was constructed by using the idea of the one for the p-l-sparse game with s-support Nash, these two algorithms have a big difference in the running time. We might be able to improve the time complexity of the p-s-support l-sparse LCP by improving the analysis or construction of the underlying graph.

Bibliography

[1] B. Aspvall and Y. Shiloach. A polynomial time algorithm for solving systems of linear inequalities with two variables per inequality. *SIAM Journal on Computing*, **9** (1980), 827–845.

[2] D. Avis and V. Chvátal. Notes on Bland's pivoting rule. *Mathematical Programming Studies*, **8** (1978), 24–34.

[3] I. Bárány, S. Vempala and A. Vetta. Nash equilibria in random games. *Random Structures & Algorithms*, **31** (2007), 391–405.

[4] Y. Bard. An eclectic approach to nonlinear programming. *Optimization, Proc. Sem. Austral. Nat. Univ.*, 1971, 116–128.

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

[6] R. G. Bland. New finite pivoting rules for the simplex method. *Mathematics of Operations Research*, **2** (1977), 103–107.

[7] A. Bouchet. A characterization of unimodular orientations of simple graphs. *Journal of Combinatorial Theory, Series B*, **56** (1992), 45–54.

[8] L. E. J. Brouwer. Über Abbildung von Mannigfaltigkeiten *Mathematische Annalen*, **71** (1911), 97–115.

[9] R. Chandrasekaran. A special case of the complementary pivot problem. *Opsearch*, **7** (1970), 263–268.

[10] R. Chandrasekaran. Integer programming problems for which a simple rounding type algorithm works. *Combinatorial Optimization*, **8** (1984), 101–106.

[11] R. Chandrasekaran, S. N. Kabadi and R. Sridhar. Integer solution for linear complementarity problem. *Mathematics of Operations Research*, **23** (1998), 390–402.

[12] X. Chen and X. Deng. Settling the Complexity of Two-Player Nash Equilibrium. *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006, 261–272.

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

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

[15] S. J. Chung. NP-completeness of the linear complementarity problem. *Journal of Optimization Theory and Applications*, **60** (1989), 393–399.

[16] V. Chvátal. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Mathematics*, **4** (1973), 305–337.

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

[18] E. Cohen and N. Megiddo. Improved algorithms for linear inequalities with two variables per inequality. *SIAM Journal on Computing*, **23** (1994), 1313–1347.

[19] T. F. Conry and A. Seireg. A mathematical programming method for design of elastic bodies in contact. *Journal of Applied Mechanics*, **38** (1971), 387–392.

[20] S. A. Cook. The complexity of theorem-proving procedures. *Proceedings of the 3rd annual ACM symposium on Theory of computing*, 1971, 151–158.

[21] W. Cook, A. M. H. Gerards, A. Schrijver and É. Tardos. Sensitivity theorems in integer linear programming. *Mathematical Programming*, **34** (1986), 251–264.

[22] R. W. Cottle. Symmetric dual quadratic programs. *Quarterly of Applied Mathematics*, **21** (1963), 237–243.

[23] R. W. Cottle. Note on a fundamental theorem in quadratic programming. *Journal of the Society for Industrial and Applied Mathematics*, **12** (1964), 663–665.

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

[25] R. W. Cottle and G. B. Dantzig. Complementary pivot theory of mathematical programming. *Linear Algebra and Its Applications*, **1** (1968), 103–125.

[26] R. W. Cottle and G. B. Dantzig. A generalization of the linear complementarity problem. *Journal on Combinatorial Theory*, **8** (1970), 79–90.

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

[28] R. W. Cottle, J. S. Pang and V. Venkateswaran. Sufficient matrices and the linear complementarity problem. *Linear Algebra Appl.*, **114–115** (1989), 231–249.

[29] R. W. Cottle and A. F. Veinott. Polyhedral sets having a least element. *Mathematical Programming*, **3** (1972), 238–249.

[30] G. E. Coxson. The P-matrix problem is co-NP-complete. *Mathematical Programming*, **64** (1994), 173–178.

[31] W. H. Cunningham and J. F. Geelen. Integral solutions of linear complementarity problems. *Mathematics of Operations Research*, **23** (1998), 61–68.

[32] G. B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity Analysis of Production and Allocation*, (1951), 359–373.

[33] G. B. Dantzig and R. W. Cottle. Positive (semi-)definite programming. *Nonlinear Programming*, (1967), North-Holland, Amsterdam, 55-73.

[34] G. B. Dantzig and M. N. Thapa. *Linear Programming 1: Introduction*, Springer-Verlag, New York, 1997.

[35] C. Daskalakis, P. W. Goldberg and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, **39** (2009), 195–259.

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

[37] C. Daskalakis and C. H. Papadimitriou. Continuous local search. *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, 2011, 790–804.

[38] J. B. Dennis. *Mathematical programming and electrical networks.* Sc.D. thesis, Massachusetts Institute of Technology, 1959.

[39] G. Ding, L. Feng, W. Zang. The complexity of recognizing linear systems with certain integrality properties. *Mathematical Programming*, **114** (2008), 321–334.

[40] W. S. Dorn. Duality in quadratic programming. *Quarterly of Applied Mathematics*, **18** (1960), 155–162.

[41] R. G. Downey and M. R. Fellows. *Parameterized Complexity*, Springer, New York, 1999.

[42] P. Du Val. The unloading problem for plane curves. *American Journal of Mathematics*, **62** (1940), 307–311.

[43] J. Edmonds. Paths, trees, and flowers. *Canadian Journal of Mathematics*, **17** (1965), 449–467.

[44] J. Edmonds and R. Giles. A min-max relation for submodular functions on graphs. *Annals of Discrete Mathematics*, **1** (1977), 185–204.

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

[46] G. Farkas. A Fourier-féle mechanikai elv alkalmazásai. *Mathematikai és Természettudományi Értesitö*, **12** (1894), 457–472.

[47] M. Fiedler and V. Pták. On matrices with non-positive off-diagonal elements and positive principal minors. *Czechoslovak Mathematical Journal*, **12** (1962), 382–400.

[48] J. Flum and M. Grohe. *Parameterized Complexity Theory*, Springer, Berlin, 2006.

[49] J. Foniok, K. Fukuda, B. Gärtner and H. Lüthi. Pivoting in linear complementarity: two polynomial-time cases. *Discrete & Computational Geometry*, **42** (2009), 187–205.

[50] J. Foniok, B. Gärtner, L. Klaus and M. Sprecher. Counting unique-sink orientations. *Discrete Applied Mathematics*, —bf 163 (2014), 155–164.

[51] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, **3** (1956), 95–110.

[52] V. M. Fridman and V. S. Chernina. An iteration process for the solution of the finite-dimensional contact problem. *U.S.S.R. Computational Mathematics and Mathematical Physics*, **7** (1967), 210—-214.

[53] O. Friedmann. A Subexponential lower bound for Zadeh's pivoting rule for solving linear programs and games. *Integer Programming and Combinatoral Optimization*, Lecture Notes in Computer Science Volume 6655, 2011, 192–206.

[54] O. Friedmann, T. D. Hansen and U. Zwick. Subexponential lower bounds for randomized pivoting rules for the simplex algorithm. *Proceedings of the 43rd annual ACM symposium on Theory of computing*, 2011, 283–292.

[55] K. Fukuda and M. Namiki. On extremal behaviors of Murty's least index method. *Mathematical Programming*, **64** (1994), 365–370.

[56] K. Fukuda and T. Terlaky. Linear complementarity and oriented matroids. *J. Oper. Res. Soc. Japan*, **35** (1992), 45–61.

[57] D. R. Fulkerson. Blocking and anti-blocking pairs of polyhedra. *Mathematical Programming*, **1** (1971), 168–194.

[58] D. Gale, H. W. Kuhn and A. W. Tucker. Linear programming and the theory of games. *Activity Analysis of Production and Allocation*, (1951), 317–329.

[59] D. Gale and H. Nikaido. The Jacobian matrix and global univalence of mappings. *Mathematische Annalen*, **159** (1965), 81–93.

[60] B. Gärtner. Combinatorial linear programming: Geometry can help. *Proceedings of the 2nd International Workshop on Randomization and Approximation Techniques in Computer Science*, 1998, 82–96.

[61] B. Gärtner. The random-facet simplex algorithm on combinatorial cubes. *Random Structures & Algorithms*, **20** (2002), 353-—381.

[62] B. Gärtner, W. D. Morris and L. Rüst. Unique sink orientations of grids. *Algorithmica*, **51** (2008), 200–235.

[63] D. Goldfarb and W. Y. Sit. Worst case behavior of the steepest edge simplex method. *Discrete Applied Mathematics*, **1** (1979), 277–285.

[64] R. E. Gomory. An algorithm for integer solutions of linear programs. *Recent Advances in Mathematical Programming*, 1963, McGraw-Hill, New York, 207–226.

[65] C. D. Ha and S. C. Narula. Tolerance approach to sensitivity analysis in linear complementarity problems. *Journal of Optimization Theory and Applications*, **73** (1992), 197–203.

[66] T. D. Hansen, H. Kaplan and U. Zwick. Dantzig's pivoting rule for shortest paths, deterministic MDPs, and minimum cost to time ratio cycles. *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2014, 847–860.

[67] D. Hermelin, C. Huang, S. Kratsch and M. Wahlström. Parameterized two-player Nash equilibrium. *Algorithmica*, **65** (2013), 1–15.

[68] D. S. Hochbaum, N. Megiddo, J. Naor and A. Tamir. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Mathematical Programming*, **62** (1993), 69–83.

[69] 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** (1994), 1179–1192.

[70] A. J. Hoffman. A generalization of max flow-min cut. *Mathematical Programming*, **6** (1974), 352–359.

[71] A. J. Hoffman and J. B. Kruskal. Integral boundary points of convex polyhedra. *Linear Inequalities and Related Systems*, Annals of Mathematics Studies 38 (1956), Princeton University Press, 223–246.

[72] F. Holt and V. Klee. A proof of the strict monotone 4-step conjecture. *Advances in Discrete and Computational Geometry*, **223** (1999), 201–216.

[73] J. T. Howson, Jr.. Equilibria of polymatrix games. *Management Science*, **18** (1972), 312–318.

[74] R. G. Jeroslow. Cutting-planes for complementarity constraints. *SIAM Journal on Control and Optimization*, **16** (1978), 56–62.

[75] N. Kakimura. Sign-solvable linear complementarity problems. *Linear Algebra and Its Applications*, **429** (2008), 606–616.

[76] S. Kakutani. A generalization of Brouwer's fixed point theorem. *Duke Mathematical Journal*, **8** (1941), 457–459.

[77] L. V. Kantorovich. *Mathematical Methods of Organizing and Planning Production.* Publication House of the Leningrad State University, 1939. [Translated in English in *Management Science*, **6** (1960), 366–422.]

[78] S. Kapoor and P. M. Vaidya. Fast algorithms for convex quadratic programming and multicommodity flows. *Proceedings of the 18th annual ACM symposium on Theory of computing*, 1986, 147–159.

[79] N. Karmarker. A new polynomial-time algorithm for linear programming. *Combinatorica*, **4** (1984), 373–395.

[80] W. Karush. *Minima of Functions of Several Variables with Inequalities as Side Conditions*, Master thesis, University of Chicago, 1939.

[81] L. G. Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, **20** (1979), 191–194.

[82] V. Klee and G. J. Minty. How good is the simplex algorithm? *Inequalities III*, Academic Press, New York, 1972, 159–175.

[83] L. G. Khachiyan. Polynomial algorithms in linear programming. *U.S.S.R. Computational Mathematics and Mathematical Physics*, **20** (1980), 53–72.

[84] M. Kojima, N. Megiddo, T. Noma and A. Yoshise. A unified approach to interior point algorithms for linear complementarity problems. *Lecture Notes in Computer Science*, **538**, Springer, Berlin, 1991.

[85] M. Kojima, S. Mizuno and A. Yoshise. A polynomial-time algorithm for a class of linear complementarity problems. *Mathematical Programming*, **44** (1989), 1–26.

[86] T. Koopmans. *Tanker freight rates and tankship building : an analysis of cyclical fluctuations.* De erven F. Bohn, 1939.

[87] B. Korte and J. Vygen. *Combinatorial Optimization: Theory and Algorithms.* Fifth Edition, Springer-Verlag, Berlin, Heidelberg, 2012.

[88] M. K. Kozlov, S. P. Tarasov and L. G. Khachiyan. Polynomial solvability of convex quadratic programming. *Soviet Mathematics Doklady*, **20** (1979), 1108–1111.

[89] L. Kronecker. Näherungsweise ganzzahlige Auflösung linearer Gleichungen. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*, 1884, 1179–1193, 1271–1299.

[90] H. W. Kuhn. An algorithm for equilibrium points in bimatrix games. *Academy of Sciences of the United States of America*, **47** (1961), 1657–1662.

[91] H. W. Kuhn and A. W. Tucker. Nonlinear programming. *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, 1951, University of California Press, Berkeley, 481–492.

[92] J. C. Lagarias. The computational complexity of simultaneous Diophantine approximation problems. *SIAM Journal on Computing*, **14** (1985), 196–209.

[93] C. E. Lemke. Bimatrix equilibrium points and mathematical programming. *Management Science*, **11** (1965), 681–689.

[94] C. E. Lemke and J. T. Howson, Jr. Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, **12** (1964), 413–423.

[95] O. L. Mangasarian. Equilibrium points of bimatrix games. *Journal of the Society for Industrial and Applied Mathematics*, **12**, (1964), 778–780.

[96] O. L. Mangasarian. Linear complementarity problems solvable by a single linear program. *Mathematical Programming*, **10** (1976), 263–270.

[97] N. Megiddo. A note on the complexity of P-matrix LCP and computing an equilibrium. *Technical report*, IBM Almaden Research Center, 1988.

[98] R. R. Meyer. On the existence of optimal solutions to integer and mixed-integer programming problems. *Mathematical Programming*, **7** (1974), 223–235.

[99] W. D. Jr. Morris. Randomized pivot algorithms for P-matrix linear complementarity problems. *Mathematical Programming*, **92** (2002), 285–296.

[100] K. G. Murty. Note on a Bard-type scheme for solving the complementarity problem. *Opsearch*, **11** (1974), 123–130.

[101] K. G. Murty. *Linear Complementarity, Linear and Nonlinear Programming.* Internet Edition, 1997.

[102] J. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the United States of America*, **36** (1950), 48–49.

[103] J. Nash. Non-cooperative games. *The Annals of Mathematics*, **54** (1951), 286–295.

[104] E. D. Nering and A. W. Tucker. *Linear Programs and Related Problems.* Academic Press, Boston, 1993.

[105] J. von Neumann. Zur Theorie der Gesellschaftsspiele [On the theory of games of strategy]. *Mathematische Annalen*, **100** (1928), 295–320.

[106] J. von Neumann. *Discussion of a maximum problem*, unpublished working paper, Institute for Advanced Study, Princeton, 1947. [Reprinted in *John von Neumann, Collected Works, Vol. VI*, Pergamon Press, Oxford, 1963, 89–95.]

[107] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, Princeton, 1944.

[108] N. Nisan, T. Roughgarden, É. Tardos and V. V. Vazirani. *Algorithmic Game Theory.* Cambridge University Press, Cambridge, 2007.

[109] C. H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, **48** (1994), 498–532.

[110] C. H. Papadimitriou. Algorithms, games, and the internet. *Proceedings of the 33rd annual ACM symposium on Theory of computing*, 2001, 749–753.

[111] P. M. Pardalos, S. Ketabchi and H. Moosaei. Minimum norm solution to the positive semidefinite linear complementarity problem. *Optimization*, **63** (2014), 359–369.

[112] P. M. Pardalos and A. Nagurney. The integer linear complementarity problem. *International Journal of Computer Mathematics*, **31** (1990), 205–214.

[113] I. Post and Y. Ye. The simplex method is strongly polynomial for deterministic Markov decision processes. *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2013, 1465–1473.

[114] G. Ramalingam, J. Song, L. Joskowicz and R. E. Miller. Solving systems of difference constraints incrementally. *Algorithmica*, **23** (1999), 261–275.

[115] C. Roth-Korostensky. *Algorithms for Building Multiple Sequence Alignments and Evolutionary Trees.* Ph. D. thesis, ETH Zürich, 2000.

[116] R. Saigal. A note on a special linear complementarity problem. *Opsearch*, **7** (1970), 175–183.

[117] H. Samelson, R. M. Thrall and O. Wesler. A partition theorem for Euclidean n-space. *Proceedings of the American Mathematical Society*, **9** (1958), 805–807.

[118] R. Savani and B. Stengel. Hard-to-solve bimatrix games. *Econometrica*, **74** (2006), 397–429.

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

[120] A. Schrijver. *Theory of Linear and Integer Programming.* John Wiley & Sons, Toronto, 1986.

[121] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency.* Springer-Verlag, Berlin, 2003.

[122] R. Shostak. Deciding linear inequalities by computing loop residues. *Journal of the ACM*, **28** (1981), 769–779.

[123] M. Shang, C. Zhang and N. Xiu. Minimal zero norm solutions of linear complementarity problems. *Journal of Optimization Theory and Applications*, **163** (2014), 795–814.

[124] U. Stege. *Resolving Conflicts in Problems from Computational Biology.* Ph. D. thesis, ETH Zürich, 2000.

[125] B. von Stengel. Computing equilibria for two-person games. *Handbook of Game Theory with Economic Applications*, **3** (2002), 1723–1759.

[126] A. Stickney and L. Watson. Digraph models of Bard-type algorithms for the linear complementarity problem. *Mathematics of Operations Research*, **3** (1978), 322–333.

[127] H. Sumita. *On the linear complementarity problem and its generalizations* (In Japanese). Master thesis, University of Tokyo, 2012.

[128] H. Sumita, N. Kakimura and K. Makino. Sparse linear complementarity problems. *Proceedings of the 8th International Conference on Algorithms and Complexity*, 2013, 358–369.

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

[130] H. Sumita, N. Kakimura and K. Makino. Total dual integrality of the linear complementarity problem. *Mathematical Engineering Technical Reports METR 2014-25*, Department of Mathematical Informatics, University of Tokyo (2014).

[131] H. Sumita, N. Kakimura and K. Makino. On the complexity of a generalization of the linear complementarity problem, in preparation.

[132] T. Szabó and E. Welzl. Unique sink orientations of cubes. *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, 2000, 547–555.

[133] T. Takayama and G. G. Judge. *Spatial and Temporal Price Allocation Models.* North-Holland, Amsterdam, 1971.

[134] H. Väliaho. Criteria for sufficient matrices. *Linear Algebra and Its Applications*, **233** (1996), 109–129.

[135] A. F. Veinott. Representation of general and polyhedral subsemilattices and sublattices of product spaces. *Linear Algebra and Its Applications*, **114–115** (1989), 681–704.

[136] N. N. Vorob'ev. Equilibrium points in bimatrix games. *Theory of Probability & Its Applications*, **3** (1958), 297–309.

[137] P. Wolfe. The simplex method for quadratic programming. *Econometrica*, **27** (1959), 382–398.

[138] Y. Ye. The simplex and policy-iteration methods are strongly polynomial for the Markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, **36** (2011), 593–603.

[139] Y. Ye and E. Tse. An extension of Karmarkar's projective algorithm for convex quadratic programming. *Mathematical Programming*, **44** (1989), 157–179.

[140] G. Zoutendijk. *Methods of feasible directions: A study in linear and non-linear programming.* Elsevier, Amsterdam, 1960.