Linear Programming: How It Is Developed and Applied in Decision-Making

Haytham Tang

2021-2022

Advisor: Greg Martin

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I truthfully thank Dr. Martin for his guidance throughout this year. I learn from his experience with researching and dissertation, and I appreciate all his feedback that allows me to improve my work.

This Capstone project is never possible without Mr. Coots. It is he who inspired me to research Linear Programming and make it my topic. Since the beginning, Mr. Coots has been consistently proofreading my chapters and providing his suggestions. I enjoyed consulting with him before I wrote each chapter. I am grateful for having him as my mentor.

I want to deliver my gratitude to my parents and my sister. Though they do not pay much attention to my academics, they care about and support me as always.

Finally, I am thankful for my last three years at Ranney. There had been so much fun as well as difficulties like long-time virtual learning. Mostly, I feel comfortable and happy in this community. I also want to thank my friends and Capstone fellows for supporting me with their positive attitudes whenever I need them.

ABSTRACT

This paper investigates Linear Programming: How It Is Developed and Applied in Decision Making. It has five sections: Introduction of Linear Programming, History of Linear Programming, Methods of Linear Programming, Computer Programs Applying Linear Programming, and Applications of Linear Programming. Linear Programming is a technique that applies Linear inequalities and equations to achieve maximation or minimization. It has progressed since Soviet Mathematician Leonid Kantarovich proposed the idea to American Mathematician George Dantzig devised the Simplex Method to current modern development after the Simplex Method. Linear Programming contains a variety of methods, including the Graphical Method, the Algebraic Method, the Simplex Method with the Big-M Method and the Two-Phase Method, and so forth. Computer programs that apply Linear Programming in languages like Python significantly help calculate the result in a quick time. Linear Programming has been employed in a broad range of fields such as construction scheduling, petroleum industry, agriculture, and so on. This paper is going to focus on the application of Linear Programming in business decision making such as minimizing delivery costs or maximizing manufacturing profits.

METHODOLOGY

For this paper, I first consulted with several people including Mr. Coots and some college students majoring in mathematics or economics to have an overall view of Linear Programming. They provided me with certain directions that I could follow. For example, they showed me some key methods in Linear Programming such as the Simplex Method, which led me to focus on that method in my Methods section.

I spent most of my time researching articles, books, and tutorials related to Linear Programming. I first started researching the history of Linear Programming, in which I concentrated mainly on two figures: Leonid Kantorovich and George Dantzig. In the beginning, I collected some information about them through other people's work. For instance, I learned about Kantorovich's story from Anatoly Vershik, one student of Kantorovich. Later, Dr. Martin suggested I collect some first-hand information about Kantorovich and Dantzig, so I adjusted my direction and found some articles written by Kantorovich and Dantzig, which provide my paper with substantial credibility.

For the Methods section, I researched several detailed tutorials with examples to better understand the methods. I cited the specific steps needed for each method. I personally needed to figure out how the methods work. Then, I explained the method procedures with examples that I made up such as Coots's Pizza and Neill Transportation. The data I chose for the examples are as reasonable as possible. From my experience with these methods, I knew more clearly about these methods' advantages and disadvantages.

My topic initially only included the history, methods, and applications of Linear Programming. However, as I researched deeper and found the necessity of processing large data sets in Linear Programming, I was then interested in coding some computer programs that apply Linear Programming methods since I intended to study Computer Science in college. I researched online for some modules that can apply mathematical features, and I decided to do a Graphical Method program using the *Matplotlib* module and a Simplex Method program using *scipy.optimize* module. When creating these programs, I coded and commented beside the codes to remind readers as well as myself of the purpose of specific lines.

Getting information for the Applications section was the hardest part. I wanted to include both hypothetical applications and real-life applications. For the hypothetical applications, I was able to find some books or articles that contain case studies of applying Linear Programming in decision making. However, for the real-life applications, possibly due to confidentiality, there was not much public information online. I finally picked a pharmaceutical company example with two parts as my real-life application.

INTRODUCTION

The concept of optimization exists everywhere in people's daily lives. People may not even notice when they are working on optimization such as dividing a pie into several pieces for others to share, choosing a cost-efficient subscription plan, and ordering food that meets nutritional needs. Basically, optimization is to utilize resources available to make a decision that can bring as many benefits as possible. All of the examples mentioned above involve maximizing or minimizing something, and these decisions can be made using simple maths such as direct comparisons.

However, if the problem of optimization is much more complex with more variables and requirements, simple comparisons are not helpful in solving the problem. In order to optimize the production or allocation of resources, mathematicians such as Newton, Leibniz, and Fourier applied concepts like Calculus to derive models for optimization through the centuries. Linear Programming, which is a relatively novel technique, has been developed and modified in recent decades by mathematicians around the world. It is an effective technique that models the problem in a linear form and outputs the optimal solution.

This paper will discuss the structure of Linear Programming, the history of Linear Programming, the methods of Linear Programming, the computer programs applying Linear Programming, and the applications of Linear Programming in decision making.

FINDINGS

Chapter 1

What is Linear Programming?

In one's daily life, it is ubiquitous to encounter situations in which one has to make a decision to have as many benefits as possible. This is called optimization. Optimization exists everywhere around the world, such as dividing a pie into several pieces, choosing a cost-efficient subscription plan, and cooking enough food with supplied material, all of which involve the idea of optimization: achieving a maximal profit or a minimal cost. It only requires simple mathematics to yield an optimal value for the scenarios mentioned above. However, most cases requiring optimization are much more complex, requiring techniques that are more sophisticated and efficient. For example, George Dantzig, a prestigious Mathematician in the field of optimization, once proposed the following problem.

Imagine you have to assign 70 people to 70 jobs. Denote x_{ij} as the ith person assigned to the jth job. Denote v_{ij} as the benefit of assigning ith person assigned to the jth job. There are two requirements: every person has to get a job and every job must be filled. So, there could be 70 factorial (70!) ways of assigning the variable x_{ij}. 70! is a gigantic number, which is about 1.198×10^{100}. Suppose you have a computer that can perform one million calculations per nanosecond, and the computer runs without stopping for 365 days a year. It takes $\frac{1.198 \times 10^{100}}{10^{6} \times 10^{9} \times 3600 \times 24 \times 365} = 3.8 \times 10^{77}$ years, which is even longer than the time from the big bang till now (Dantzig).

In this case, a simple comparison of all possible solutions to achieve optimization is impossible. However, it only takes a few moments to figure out the same problem with a technique called Linear Programming and a normal computer.

Humans have unlimited needs and desires that exceed the limited resources existing on Earth. It is a paramount task for economists and mathematicians to achieve the optimization of the allocation of resources, and Linear Programming is one of the techniques that serve to calculate the optimal output. Linear Programming contains three main components: Objective Function, Decision Variables, and Constraints. It is widely used in areas such as business planning, marketing, transportation, and so forth.

The Model of Linear Programming

What exactly is Linear Programming? It is widely misunderstood that Linear Programming is a computer code or program. In fact, Linear Programming is a mathematical model used to achieve optimization. Kenneth J. Arrow, the winner of the Nobel Memorial Prize in Economic Science in 1972, described, “Linear programming is a way of choosing interdependent activities, with inputs and outputs, so as to achieve an optimum in some dimension” (Levy). Linear Programming basically constructs a model that consists of a goal, linear equations, and inequalities to output an optimal value. However, because of the need to process massive data sets, Linear Programming is often applied with computer programs to output a solution in seconds.

Consider a classical carpentry example. Imagine there is a furniture factory that builds only two items: tables and chairs. The factory has twenty-seven kilograms of wood to build the items. Building each table requires three kilograms of wood and building each chair requires two kilograms of wood. There are fifteen workers in this factory. It requires two workers to build one table and one worker to build a chair. The factory can sell each table for six dollars and sell each chair for four dollars. The owner wants to find out how many chairs and tables he should make using his twenty-seven kilograms of wood and fifteen workers in order to have an optimal profit.

This is an optimization problem that can apply Linear Programming to achieve the optimal value. Some experts suggest that there are three major parts of linear programming; others suggest there are four — Decision Variables, Objective Function, Constraints, and, if you ask the others, Non-Negativity Restriction. James Reeb and Scott Leavengood, two professors from Oregon State University, give the following definition of the four components.

Decision Variables are the variables that are manipulated and will decide the optimal value. They are central to determining the goal as well as the constraints. Usually, the Decision Variables are referred to as “the resources available” (Reeb and Leavengood 2). For this furniture case, the number of tables made is a Decision Variable, and the number of chairs made is another Decision Variable. They both vary and cause changes in the output value.

The Objective Function is the goal of making the decisions. In Linear Programming, the Objective Function is always to maximize something like profits or minimize something like costs. “It is to select the best values for the variables (Reeb and Leavengood 2).” For the above example, the Objective Function is the total profits of making and selling the chairs and tables.

Constraints are “limitations on resource availability” (Reeb and Leavengood 2). They are the restrictions that the Linear Programming model binds to for optimization. If there is no Constraint, there is no optimal value since one can have infinite output with an infinite amount of resources. In the furniture example, the Constraints are the availability of wood and labor. There are only twenty-seven kilograms of wood and fifteen workers to make tables or chairs.

Non-Negativity Restriction is the default restriction of the model. It states that all the variables in Linear Programming have to be greater than or equal to zero. This is easy to understand since there is no negative quantity of input or output in reality. For the above example, the Non-Negativity Restriction is that the number of tables made and the number of chairs made have to be greater than or equal to zero. Because the Non-Negativity Restriction is a default restriction that has to be met, some mathematicians include them in the Constraints part while others simply regard it as known to problem-solvers by default. This is the reason for the difference in the number of components as three or four in Linear Programming.

A Linear Programming model consists of the essential components discussed above. To construct a Linear Programming model is just writing out all the components in mathematics.

Construct a Linear Programming Model

According to Hossein Arsham, a Distinguished Professor of Statistics and Management at the University of Baltimore, there are four steps to create a Linear Programming Model. The first step is to identify what the Decision Variables are. One can find the information of Decision Variables by examining what is varying in the problem. The second step is to determine the Objective Function by looking at keywords such as “maximize”, “minimize”, “profit”, and “cost”. The third step is to quantify the Constraints by applying linear equations or linear inequalities. The final step is to mention the Non-Negativity Restriction. The Non-Negativity Restriction can either be stated inside the Constraints or be explicitly stated alone (Arsham).

After determining all the components based on the given information, it is time to utilize mathematical "language" to construct a Linear Programming Model. There are two forms of a Linear Programming Model: the General Form and the Standard Form. Michael Goemans, a Professor of Mathematics at MIT, provides the following definition of these two forms.

The General Form:

In the General Form, the Objective Function of the Linear Programming Model can be maximization or minimization. The Constraints can include equalities or inequalities. Variables in the General Form can be unrestricted in sign (Goemans). The following is an example of the General Form of a Linear Programming Model.

Minimize: $Z = 3a + 2b$

Subject to: $2a + b < 30$

$5a - 7b > 15$

$a \gtrless 0$(The $\gtrless$ sign means a can be positive or negative)

$b > 0$

The Standard Form:

Professor Goemans states that the Objective Function of the Linear Programming Model must be a maximization function in the Standard Form. The Constraints must contain only equalities. Variables in the Standard Form must be restricted to Non-Negativity Restriction. It is feasible to convert a Linear Programming Model in the General Form to the Standard Form by adding negative signs, adding slack or surplus variables, and introducing new decision variables (Goemans). A slack variable is a variable introduced to convert a "smaller than or equal to" inequality to an equality. A surplus variable, also known as a negative slack variable, is a variable introduced to convert a “greater than or equal to” inequality into an equality. The following is an example of converting the General Form of Linear Programming mentioned above to the Standard Form of Linear Programming.

Break a into $a^{+} - a^{-}$. Add slack variables x_3 and x_4. Write the new Linear Programming Model in the Standard Form.

Maximize: $Z = -3a^{+} + 3a^{-} - 2b$

Subject to: $2a^{+} - 2a^{-} + b + x_3 = 30$

$5a^{+} - 5a^{-} - 7b - x_4 = 15$

$a^{+} \geq 0$

$a^{-} \geq 0$

$b \geq 0$

$x_3 \geq 0$

$x_4 \geq 0$

Example of Constructing a Linear Programming Model:

Now, a Linear Programming model can be used for the carpentry case discussed above. The Decision Variables are the number of chairs made and the number of tables made. Denote x_1 as the number of chairs made. Denote x_2 as the number of tables made. Since each table is sold for \$6 and each chair is sold for \$4, and the owner wants to make maximum profit, the Objective

Function can be written as Maximize: $Z = 4x_1 + 6x_2$. Since there are 27 kilograms of wood available, and building each table requires 3 kilograms of wood while building each chair requires 2 kilograms of wood, the Constraint of wood can be written as $2x_1 + 3x_2 \leq 27$. The 15 workers in this factory are assigned to either building tables or chairs, and it requires 2 workers to build one table and 1 worker to build a chair. As a result, the Constraint of labor can be written as $x_1 + 2x_2 \leq 15$. It thus creates the following Linear Programming Model in the General Form.

Maximize: $Z = 4x_1 + 6x_2$

Subject to: $2x_1 + 3x_2 \leq 27$

$x_1 + 2x_2 \leq 15$

$x_1 \geq 0$

$x_2 \geq 0$

Applying the Graphical Method can intuitively and efficiently solve this Linear Programming problem. The Graphical Method will be discussed later in the Methods section. It is employed here to demonstrate how Linear Programming works to achieve an optimal value.

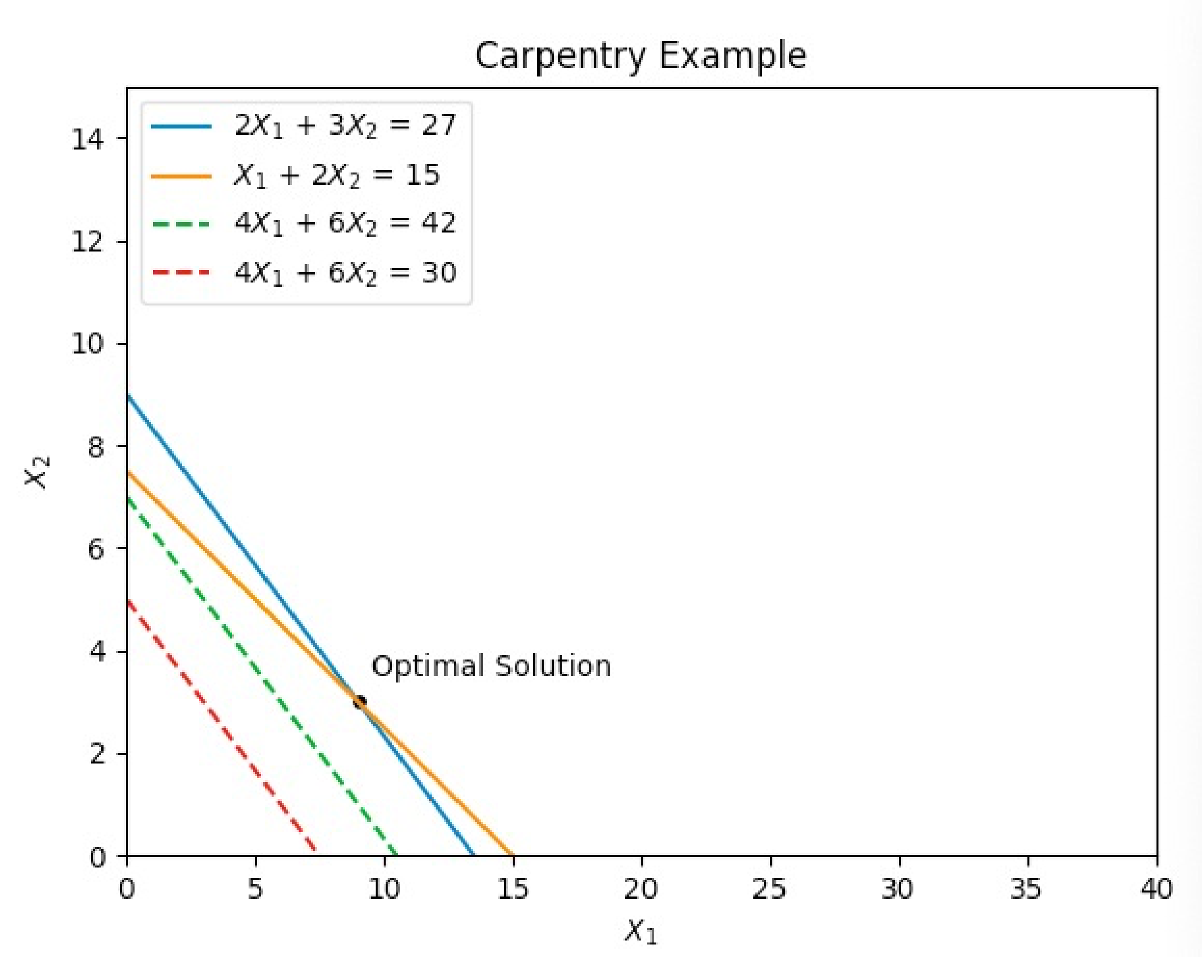

Figure 1.1 —Carpentry Example Graph

Figure 1.1 is plotted using a Python program coded by me applying the Matplotlib modules. The Computer Program section of this project will display and explain the codes. This graph contains only the first quadrant, which satisfies the Non-Negativity Restriction. In this graph, the blue line is the production possibility curve of tables and chairs related to available wood. The area under this line and the points on the line represent the Constraint $2x_1 + 3x_2 \leq 27$. Similarly, the yellow line is the production possibility curve of tables and chairs related to the available labor force. The area under this line and the points on the line represent the Constraint $x_1 + 2x_2 \leq 15$. The area that is under both the blue line and the yellow line is known as the Feasible Zone, and only points in this area are feasible to output a value. The optimal solution is one of the vertices of the Feasible Zone, so it is either the vertical intercept of the yellow line, the horizontal intercept of the blue line, the origin, or the intersection of the yellow line and the blue line. By plotting the dashed lines that represent different Objective values, it is possible to determine the direction of the Objective Function. In this case, the value of the Objective Function increases as the dashed lines shift to the right, so the optimal value should be either the intersection of the yellow line and the blue line or the horizontal intercept of the blue line. The intersection point is (9, 3), and the horizontal intercept point is (13.5, 0). Since the number of tables and chairs must be integers, the only available solution left is the intersection of the yellow line and the blue line. By plugging the data into the Objective Function, the maximal profit is $4 \times 9 + 6 \times 3 = \$54$.

As a result, the carpentry factory owner can have a maximal profit of $54 by manufacturing 9 chairs and 3 tables. The Graphical Method of Linear Programming in this scenario allows one to explicitly achieve the optimization by constructing a graph and testing the vertices.

Application of Linear Programming

The Carpentry problem mentioned above is one of the applications of Linear Programming. Dawn Levy, a science writer for the Stanford University News Service, points out a variety of fields that apply Linear Programming. Manufacturing firms apply Linear

Programming to “price products” and “manage supply chains”; Transportation firms use Linear Programming to determine routes that can deliver the goods within the maximal capacity; Petroleum companies employ Linear Programming to “schedule” and “distribute production”; Iron and steel industries utilize Linear Programming to “evaluate iron cores” and select the most suitable products, and governments use it to determine fiscal and monetary policies. Other applications include “scheduling construction projects”, “controlling water and air pollution”, “selecting advertising media and compensation policies”, “developing bidding strategies”, and so forth (Levy). Linear Programming plays a significant role in these applications to output an optimal solution.

In general, Linear Programming is widely used in different industries such as marketing, investment, transportation, agriculture, and so on. This paper will focus on the application of Linear Programming in the field of business planning and decision making.

Chapter 2

History of Linear Programming

The idea of optimization has evolved through the centuries. Mathematicians such as Newton, Leibniz, and Fourier applied concepts like Calculus to derive models for optimization. Linear Programming, which is a relatively novel technique, has been developed and modified in recent decades by mathematicians all over the world.

Soviet mathematician Leonid Kantorovich proposed the idea of Linear Programming during World War II. He formulated the model of Linear Programming and applied it on a local scale. Unfortunately, he struggled with implementing his idea and gaining recognition for his work. He later shared the Nobel Prize in Economics in 1975 for his contribution to Linear Programming. American mathematician George Dantzig, generally regarded as the father of Linear Programming, devised an efficient and commonly applied method called the Simplex Method. In fact, Dantzig also gave birth to the name “Linear Programming”. After Dantzig, mathematicians such as Leonid Khachiyan in recent decades have developed methods like the Ellipsoid Method to improve the technique of Linear Programming.

Leonid Vitaliyevich Kantorovich

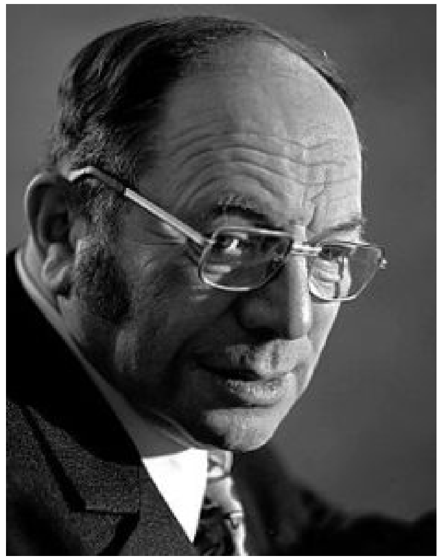

Figure 2.1 — Leonid Vitaliyevich Kantorovich

The idea of applying Linear Programming for optimization started with Soviet mathematician Leonid Vitaliyevich Kantorovich. Leonid Kantorovich was born on January 19, 1912, in Saint Petersburg, Russia. He had demonstrated his talents in mathematics while he was still a child. According to Ivan Boldyrev, a Professor of Economics at Radboud University, and Till Duppe, a Professor of Economics at the University of Quebec in Montreal, Leonid Kantorovich "was dedicated to mathematical training from a very early age. At age fourteen, he enrolled at Leningrad State University with special permission. There he was trained by Grigorii Fichtenholz, a leading mathematician in functional analysis" (Boldyrev and Duppe). Because of his experience with Grigorii Fichtenholz, Kantorovich focused on the study of functional analysis and applied mathematics during his career, which paved the way for his development of Linear Programming techniques.

It was extremely difficult for Kantorovich to formulate the idea of Linear Programming and gain recognition for his work considering his situation in the Soviet Union. Kantorovich started his early career with a "state of terror" during Stalin's Great Purge from 1936 to 1938 when Kantorovich sensed a "repression in mathematics" and hence "stopped publishing abroad" and "withdrew from the international scene" (Boldyrev and Duppe). However, around the same time, the rise of German fascism led Kantorovich back to the study of applied mathematics. Kantorovich remarked years later, "I felt some dissatisfaction with mathematics… because the world was facing a strange menacing brown plague – German fascism… I had a clear perception that the state of economic solutions was a weak spot that was reducing our industrial and economic power" (Kantorovich). Kantorovich perceived that mathematicians like him needed to devise plans for the better arrangement of the economy in preparation for the upcoming war with Germany. Driven by this perception, in 1937, Kantorovich wrote a report titled *On the Allocation of Printed Products*. In this report, he addressed the issues of book shortages and claimed that the way of supplying several kinds of literature was inefficient. He also provided some possible solutions to meet the demand and stop the shortages. The practical measure discussed in the report would contribute to making him the founder of Linear Programming years later (Boldyrev and Duppe).

Kantorovich thus devoted his career to applying mathematical models in order to solve economic problems. In 1938, engineers from the lab of Plywood Trust asked Kantorovich to solve a maximization problem regarding "veneer-cutting machines of different productivities and different materials," Kantorovich applied the duality principle to solve this problem of a set of inequalities, and he later discussed this algorithm in a booklet titled *Mathematical Methods in the*

Organization and Planning of Production in 1939, which became the “first systematic application of Linear Optimization to economic problems” (Boldyrev and Duppe). Kantorovich envisioned the potential of employing the ideas of Linear Programming for optimization and hence to better run the economy of the Soviet Union. Just as he stated, “I began to understand the significance of these models for developing the principles of pricing, estimating effectiveness, at all events, the effectiveness of investments, that is, the basic features of the theory of linear economics for a socialist economy were created” (Kantorovich).

Kantorovich hoped to apply the model of Linear Programming not only on a production unit level but on a national level to “program the USSR” (Kantorovich). Unfortunately, he encountered substantial resistance and struggled to gain recognition for his work because of the Marxist Economy in the Soviet Union. After formulating the idea of Linear Programming, Kantorovich started applying it to solve optimization problems on a local scale. Anatoly Vershik, a Russian mathematician who used to be a student of Leonid Kantorovich, describes Kantorovich's situation. At that time, Kantorovich and his close assistants were trying to apply their theories to local planning such as optimal distribution of the mass transportation systems. Kantorovich’s famous Kantorovich Metric, used for “the optimal value of the objective function in the mass transportation problem”, helped introduce a norm in the space of measures. It was later developed into Kantorovich–Rubinstein metric or Wasserstein Metric. They also attempted to “introduce optimization methods at the Skorokhod Factory, Lianozovo (ex-Egorov) Wagon Plant, Kolomna Locomotive Plant, etc” (Vershik). Recognizing the success of applying Linear Programming to these local production units, Kantorovich saw the possibility of employing Linear Programming on a national scale to make the economy of the Soviet Union more flexible and efficient. Unfortunately, Kantorovich and his colleagues faced much resistance and failed to implement their ideas at that time. Vershik attributes this failure to the reason that the Soviet economic system, namely the command-administrative system, could not accept innovations or economic reforms that could address the problems (Vershik).

It was thus difficult for Kantorovich to propose and implement his theories on a national level. According to Kantorovich himself, in 1943, during the meeting at Gosplan, the State Planning Committee of the Soviet Union, economist and statistician Boris Yastremskii said, “You are talking here about optimum. But do you know who is talking about optimum? The fascist Pareto is talking about optimum” (Kantorovich). At that time, during the war between Nazi Germany and the Soviet Union, it sounded treacherous and insane to be connected with fascism, which discouraged Kantorovich from propagating his theories. Later in 1944, Kantorovich proposed his ideas of Linear Programming again. However, Vladimir Starovsky, the head of Gosplan's Central Statistical Agency, responded, “Having examined your suggestions on the methodology of economic calculation and planning and having discussed them at a conference in which you participated, I consider any practical use of your suggestions in the work of the Gosplan of the USSR to be impossible” (Kantorovich). Although Leonid Kantorovich was awarded the Stalin Prize, the USSR’s highest civilian honor in 1949, it was clear that he won the prize for his contribution to the military. For example, he worked with a group of mathematicians such as Ivan Petrovsky and Andrey Tikhonov to calculate the solutions to differential equations that modeled the explosion of bombs instead of formally for “the totality of his work, with specific mention of the applications of functional analysis in calculation” (Boldyrev and Duppe). Kantorovich’s conflict with the Marxist political economists in charge of the national institutions regarding his theories was exacerbated during the final years of Stalin's regime. He gradually withdrew from the scene of planning and decision-making just as he did during Stalin's Great Purge.

After Stalin's death in 1953 and Khrushchev's Secret Speech in 1956, there seemed to be some hope for economic reforms. Kantarovich started again promoting his ideas of optimization methods, though still facing much pressure from officials. In 1957, during the Fikhtengolts–Kantorovich seminar at the Department of Mathematics and Mechanics of the Leningrad State University, which was a seminar researching functional analysis, an internal veto turned down Kantorovich's ideas concerning the problems of Linear Programming, written in the booklet *Mathematical Methods in the Organization and Planning of Production* (Vershik). After that meeting, Kantorovich still insisted on proposing his ideas of Linear Programming from time to time, though he failed to win recognition for his work. Seeing the obstacle to implementing his theories, Kantorovich commented years later, "Everyone was saying that it was necessary to leave this work for the time being. It was dangerous to continue it – as I subsequently found out, my fear was not unfounded. Of course, this was a severe blow to me as I had put great faith in it. For some time I was even in a state of depression" (Kantorovich). Due to the economic situation in the Soviet Union, Kantorovich was unable to promote his theories of Linear Programming and apply them on a national level to run the country's economy. His career demonstrated how Soviet scientists could "forge utopian visions of socialism by remodeling Marxist ideology and disciplinary structures" and "engaging the rigid institutions of Soviet bureaucracy and the political class" (Boldyrev and Duppe).

Kantarovich has struggled his whole life for the recognition of his theories in the Soviet Union. Despite his struggle, his contributions to the development of Linear Programming were indelible and will always be remembered in history. In 1975, Kantorovich shared the Nobel Prize in Economics with American economist Tjalling C. Koopmans, for their contributions to the theory of the optimum allocation of resources, in which Kantarovich mostly applied Linear Programming ideas to situations in his country (Nobel Prize Press Release). Kantorovich's winning of the Nobel Prize recognized his work and also helped promote science as a mediator in the Cold War. Kantorovich passed away in 1986.

Throughout his whole life, Kantorovich persistently worked to develop and promote the model of Linear Programming for solving optimization problems. Although he was in a difficult situation to gain recognition for his theories, his work laid the foundation for the field of Linear Programming. Just as George Dantzig, another paramount figure in the history of Linear Programming who will be discussed next, remarked, "Kantorovich should be credited with being the first to recognize that certain important broad classes of production problems had well-defined mathematical structures which, he believed, were amenable to practical numerical evaluation and could be numerically solved" (Dantzig).

George Bernard Dantzig

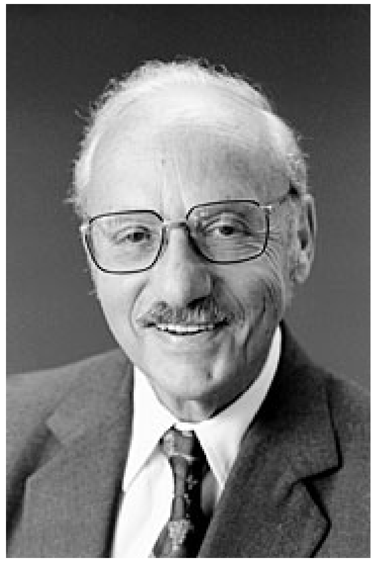

Figure 2.2 — George Bernard Dantzig

This American mathematician is widely regarded as the inventor of the Simplex Method and the father of Linear Programming. His name is George Bernard Dantzig.

Dawn Levy, a science writer for the Stanford University News Service, delineates Dantzig’s early background. Dantzig was born in Portland in 1914. He went to the University of Maryland and received his bachelor's degree in math and physics in 1936. Later, he went to the University of Michigan and got his master's degree in 1938. From 1937 to 1939, Dantzig served as a junior statistician for the U.S. Bureau of Labor Statistics. He went to U.C. Berkeley for a doctorate degree and received his degree in 1946 (Levy). While at Berkeley, George Dantzig was known for solving “two famous unsolved problems in mathematical statistics” once when he arrived late to Jerzy Neyman’s class. He mistook the two problems for his assignments and spent days working on the hard problems. Ultimately, he solved the problems, and he wrote his work later for his doctorate dissertation (Dantzig). According to Amir Ahmadi, a professor at

Princeton, this interesting story of Dantzig inspired the first scene in the movie *Good Will Hunting*, where the main character Will Hunting solves hard problems written on the board while working as a janitor at MIT (Ahmadi).

Dantzig served as the chief of the combat analysis branch of the Army Air Forces from 1941 to 1946 during World War II and was the mathematical advisor to the military for a few years after the war. While serving in the Pentagon, Dantzig developed the Simplex Method and hence formally invented the Linear Programming technique. In 1946, Dantzig was looking for an academic position offered at universities. In order to entice him not to take another job, Dantzig's colleagues at the Pentagon challenged him to "find a way to more rapidly compute a time-staged deployment, training, and logistical supply program" (Dantzig), More specifically, George Dantzig was asked to figure out how the Air Force could mechanize its planning process to accelerate the "computation of deployment of forces and equipment, training and logistical support." (Dantzig). This challenge led to the origin of Linear Programming. Dantzig found that the Air Force wanted a dynamic model with alternative activities, which could not be accomplished by Leontif's input-output model since it was fit for steady-state deployment. So, Dantzig constructed an activity analysis model today known as "a time-staged, dynamic linear program with a staircase matrix structure" (Dantzig).

Dantzig formulated the planning problem as a set of axioms: "first was the set of items being produced or consumed and the second, the set of activities or production processes in which these items would be inputted or outputted in fixed proportions providing these proportions are non-negative multiples of each other. The resulting system to be solved was the minimization of a linear form subject to linear equations and inequalities" (Dantzig). He was wondering how to solve the system and started working. He proposed the idea of moving from one vertex to another along the edge of a polyhedral set. However, he discarded this idea because it seemed to him intuitively inefficient and even impossible to look over all the vertices and edges in a general case. He then proposed another method of checking the geometry of columns instead of looking at one of the rows. This turned out to be exactly the same idea of vertex-edge procedure in a different geometry. George Dantzig thus invented the Simplex Method in 1947 (Dantzig). He then tested the efficiency of the Simplex Method by applying the Simplex Method to address the issue of the minimal cost diet problem.

After many tests, Dantzig and his colleagues found that the Simplex Method was working and was probably more efficient than other possible techniques. He later recognized that most practical planning relations could be reformulated as a system of linear inequalities, which hence founded the model of Linear Programming. Initially, Dantzig named it “programming in a linear structure”, since the military refers to the planning as programs. On June 28, 1948, Dantzig and Tjaling Koopmans took a stroll along the Santa Monica beach. Koopmans said, “Why not shorten ‘programming in a linear structure’ to ‘Linear Programming’?” Dantzig replied, “That’s it! From now on that will be its name” (Dantzig).

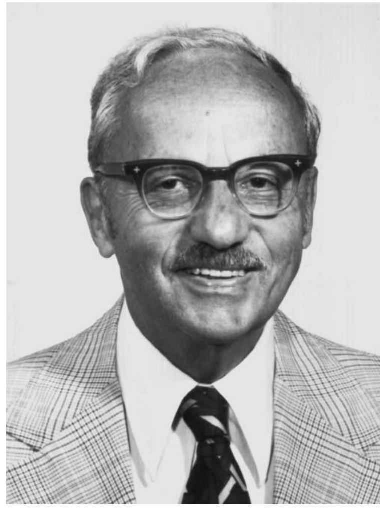

Figure 2.3 — George Bernard Dantzig

After leaving the Air Force, Dantzig continued to further his studies on Linear Programming and mathematical approaches that can be applied to solve real-life problems. According to Saul Gass, a writer for the Institute for Operations Research and the Management Sciences News, in his book *Linear Programming and Extensions* published in 1963, Dantzig delineated a historical discussion of how Linear Programming was developed, the components of Linear Programming, and remarkable applications of Linear Programming (Gass). In 1966, George Dantzig joined the Operations Research department at Stanford to further explore the field of Linear Programming with other esteemed mathematicians.

In 1975, Tjalling C. Koopmans shared the Nobel Prize in Economics with Leonid Kantorovich for contributions to the theory of the optimum allocation of resources, which was essentially the idea of Linear Programming. Unfortunately, Dantzig, who contributed much to the development of Linear Programming, was not included in the 1975 award because Dantzig's theory emphasized mathematical and engineering needs. Koopmans found himself so distressed by this pity that he suggested Kantarovich refuse the Nobel Prize together with him, which was a difficult decision that did not occur in the end (Gass).

Nevertheless, George Dantzig won notable recognition for his invention of Linear Programming and the Simplex Method. In 1975, President Gerald Ford awarded Dantzig a National Medal of Science for "inventing Linear Programming and for discovering the Simplex Algorithm that led to wide-scale scientific and technical applications to important problems in logistics, scheduling, and network optimization, and to the use of computers in making efficient use of the mathematical theory". He also won the War Department Exceptional Civilian Service Medal, the John von Neumann Theory Prize, the National Academy of Sciences Award in Applied Mathematics and Numerical Analysis, the Harvey Prize, the Silver Medal of the Operational Research Society (U.K.), the Alfred Coors American Ingenuity Award and a Fellows Award from the Institute for Operations Research and the Management Sciences (Levy).

George Dantzig was a member of the National Academy of Sciences and the National Academy of Engineering and a fellow of the Econometric Society, the Institute of Mathematical Statistics, the American Association for the Advancement of Science, the American Academy of Arts and Sciences, and the Operations Research Society of America. He was also a president of the Institute of Management Science and the first chair of the Mathematical Programming Society (Levy).

"Dantzig shepherded more than fifty graduate students to their doctorates. He had a kindly personality, free of conceit or condescension," said Richard W. Cottle, professor emeritus of management science and engineering at Stanford, "for a man with such an impressive scientific reputation, he had a magical way of putting people at ease. He found just the right way to nurture the potential within them, to further their development and to incorporate their accomplishments into the goals of his large research program" (Levy).

George Dantzig passed away on May 13, 2005, in Stanford, Calif, at age ninety, from diabetes and cardiovascular disease. He devoted his whole life to the development of Linear Programming. Because of his contribution, Linear Programming has become an effective tool to solve optimization problems in contemporary society. As the father of Linear Programming and the Simplex Method, Dantzig will always be remembered in the history of mathematics.

Modern Development

Since George Dantzig invented Linear Programming and the Simplex Method in 1947, mathematicians around the world have studied different methods to improve the Linear Programming technique. Among these methods, the most two representative methods are the Ellipsoid Method and the Interior-Point Method.

The Ellipsoid Method

Figure 2.4 — Leonid Khachiyan

In 1979, Russian American mathematician Leonid Khachiyan demonstrated that using the Ellipsoid Method can solve Linear Programming problems in polynomial time. Margaret Wright, a professor at New York University, explains the Ellipsoid Method, “Polynomiality of the Ellipsoid Method arises from two bounds: an outer bound that guarantees existence of an initial ellipsoid enclosing the solution and an inner bound that specifies how small the final ellipsoid must be to ensure sufficient closeness to the exact solution" (Wright).

The Ellipsoid Method was a theoretical breakthrough. However, its performance in practice was inefficient. Its actual running time when solving real problems was much longer than that of the Simplex Method. Just as Michael Goemans, an MIT professor, comments, "contrary to the simplex algorithm, the ellipsoid algorithm is not very fast in practice; however, its theoretical polynomiality has important consequences for combinatorial optimization problems" (Goemans).

Although the Ellipsoid Method was inefficient in practice, its theoretical inspiration was influential to later methods like the Interior-Point Method.

The Interior-Point Method

Figure 2.5 — Narendra Karmarkar

In 1984, Indian mathematician Narendra Karmarkar invented the Interior-Point Method. Like the Ellipsoid Method, the Interior-Point Method is another polynomial-time algorithm.

Nevertheless, the Interior-Point Method is quite efficient in practice. It is widely used to solve Linear Programming problems nowadays.

What makes the Interior-Point Method so outstanding is its efficiency in the worst-case scenario, which is far better than that of the Simplex Method. According to Margaret Wright, "the worst-case complexity of the Simplex Method is exponential in the problem dimension," while Narendra Karmarkar claimed that his Interior-Point Method was "50 times faster than the Simplex Method," which turned out to be true and formally proved the efficiency of the Interior-Point Method (Wright).

The Interior-Point Method has some unusual properties. For example, it assumes that the Linear Program has a special, nonstandard form. Also, in its description, nonlinear projective geometry is used (Wright). Because of its efficiency, the Interior-Point Method is frequently used to solve Linear Programming problems today.

After decades of development, Linear Programming has contained diverse methods for solving optimization problems. However, it is still a relatively new field, and mathematicians are working on devising better, more efficient methods. It is optimistic that there will be more technological breakthroughs in Linear Programming in the future.

Chapter 3

Methods of Linear Programming

Many different methods are used to solve Linear Programming problems. Each method has its distinct advantages and disadvantages. Some of the methods are more intuitively clear, while some are more efficient in solving problems. Essentially, all the methods work to yield an optimal solution.

The Graphical Method is intuitively useful for solving Linear Programming problems by graphing lines and finding optimal solutions. The Algebraic Method is straightforward to compute an outcome. However, these two methods are only applicable to questions with few variables and small data. In solving more sophisticated problems, the Simplex Method is widely applied to calculate an optimal solution by creating a tableau. The Big-M Method and the Two-Phase Method are also utilized for addressing problems with artificial variables.

The Graphical Method

The Graphical Method is the most intuitive method among Linear Programming methods. It allows one to find an optimal solution by graphing all the equations and inequalities to determine a feasible zone and then achieving the optimal value by comparing all the vertices of the feasible zone. Hossein Arsham, a Distinguished Professor of Statistics and Management at the University of Baltimore, delineates the following series of steps for the Graphical Method.

The first step is to determine whether the question is a Linear Programming problem. A Linear Programming problem has all variables with a power of one, and there is only addition and subtraction between the variables instead of multiplication or division. The constraints are always bounded and closed, and the objective function should be either minimization or maximization. If one of the conditions is not satisfied, it is not a Linear Programming problem and hence the Graphical Method is not applicable (Arsham).

The second step is to determine whether the problem can be solved using the Graphical Method. The Graphical Method is typically used for problems with one or two variables, and it is sometimes applied when there are three variables in three dimensions. However, if the number of variables is greater than three, the Graphical Method will be an inefficient tool to address such problems (Arsham).

Now, after determining that the problem is a Linear Programming problem and that the Graphical Method is applicable, it is time to employ the Graphical Method to output an optimal solution. In this paper, two-dimensional graphs will be displayed to demonstrate how the Graphical Method works since it is easier to plot a two-dimensional graph than a three-dimensional graph. The graphs are all plotted using a computer program coded by me applying Matplotlib in Python.

Applying the Graphical Method, the first step is to identify the axes of the coordinate system. There are two axes in a coordinate system: a horizontal axis and a vertical axis, each representing a decision variable. Identify which axis represents which decision variable and then determine the largest possible value on each axis, which is given through the constraints (Arsham).

The second step is to graph the constraints one by one. First, graph all the constraints that are equations. Then, pretend the inequality constraints as equations and graph the corresponding lines. Each line creates three regions: two sides of the line and the line itself. To determine what regions are feasible, pick a point on either side of the line first. Then, plug the coordinate into the constraints. If the coordinate satisfies the constraints, the side in which the coordinate lies is the feasible zone and the other side is hence the infeasible zone. The feasible zone for equation constraints is all the points on the line (Arsham).

After graphing all the constraints and determining each specific constraint's feasible zone, there should be a convex feasible zone that is bounded by all graphs of the constraints. Now, the third step is to determine the direction of the objective function by creating iso-value lines that represent possible values of the objective function. Usually, pick two or three distinct values of the objective function and draw corresponding iso-value lines. Then, move the lines parallel to see which way the value of the objective function is increasing or decreasing. Through this process, find the corner that yields an optimal solution, which is either a maximum or minimum value based upon the objective function (Arsham).

In general, for feasible zones in the first quadrant, which are the cases in real-life situations as the variables can only be non-negative values, the optimal solution for a maximization problem is usually farther away from the origin point(0, 0), while the optimal solution for a minimization problem is usually closer to the origin point.

Here is an example that demonstrates the application of the Graphical Method to solve an optimization problem. A pizza company called Coots's Pizza sells two kinds of pizzas: regular pizzas and supreme pizzas. The company will make a profit of one dollar for each regular pizza and will make one and a half dollars for each supreme pizza. There are currently 150 pounds of dough mix and 800 ounces of topping mix in stock. Each regular pizza uses one pound of dough mix and four ounces of topping mix. Each supreme pizza uses one pound of dough mix and eight ounces of topping mix. Coots's Pizza can sell at least fifty regular pizzas and at least twenty-five supreme pizzas. The firm wants to determine the number of regular pizzas and the number of supreme pizzas it should produce to have a maximal profit.

Denote the number of regular pizzas produced as X_1 and the number of supreme pizzas produced as X_2. A Linear Programming model can be formulated as the following:

Maximize: $Z = X_1 + 1.5X_2$

Subject to: $X_1 + X_2 \leq 150$

$4X_1 + 8X_2 \leq 800$

$X_1 \geq 50$

$X_2 \geq 25$

$X_1 \geq 0$(Non-negativity constraint)

$X_2 \geq 0$(Non-negativity constraint)

Now, apply the Graphical Method using Matplotlib in Python to draw the graphs. First, start by drawing all the constraints.

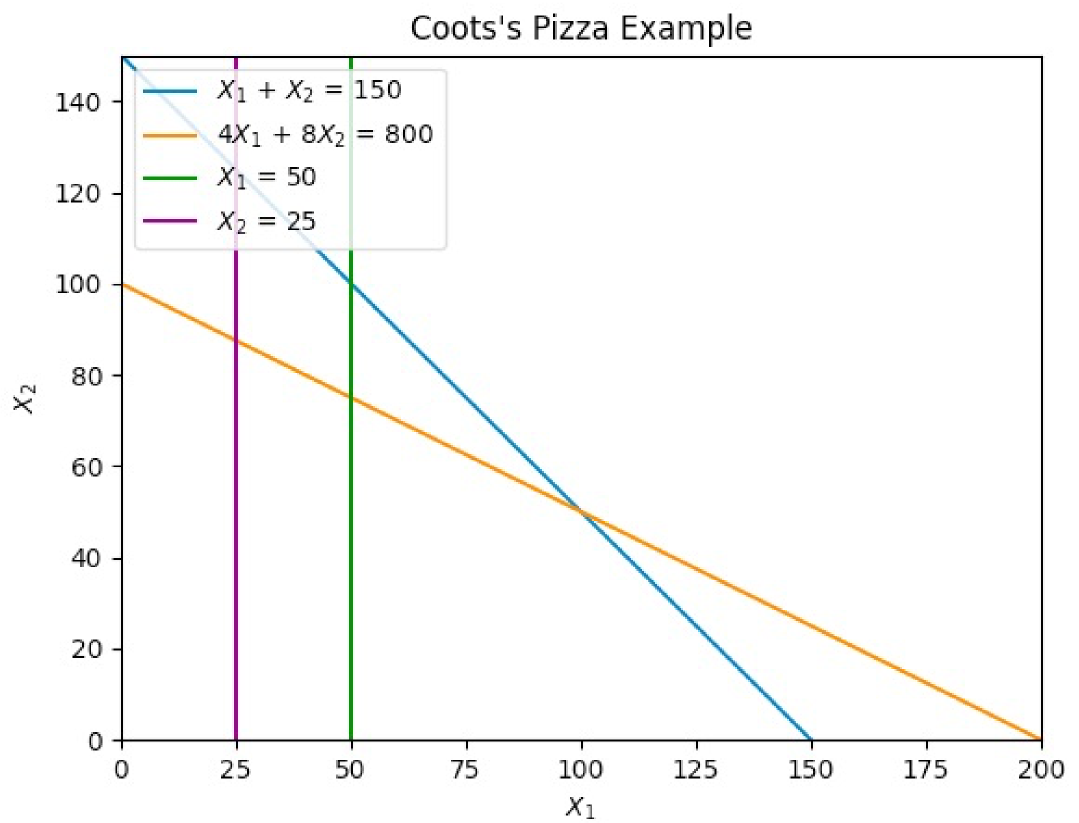

Figure 3.1 — The graph of Coots's Pizza example with all constraints

In the graph above, the blue line represents the constraint of dough mix: $X_1 + X_2 = 150$. The coordinates on and below the blue line are feasible solutions that satisfy the constraint of dough mix. The orange line represents the constraint of toppling mix: $4X_1 + 8X_2 = 800$. The coordinates on and below the orange line meet the constraint of the toppling mix. The green line represents the constraint of the number of regular pizzas: $X_1 \geq 50$. The area to the right of the green line and points on the line are feasible. Similarly, the purple line represents the constraint of the number of supreme pizzas: $X_2 \geq 25$. The area to the right of the purple line and points on the line satisfy that constraint. By examining all constraints' feasible regions, an ultimate feasible region for solutions, which is the intersection part of constraints' feasible regions, can be produced below.

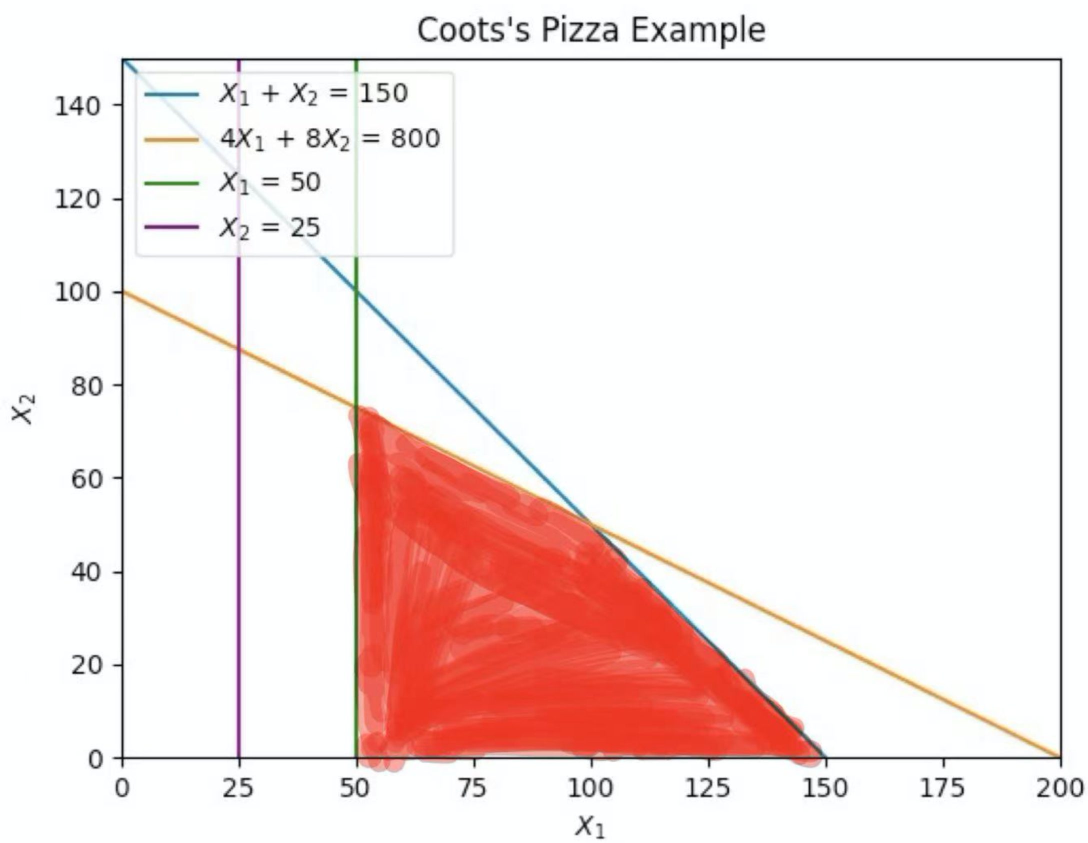

Figure 3.2 — The feasible region graph

The red part drawn in the graph above is the feasible region. All of the solutions that meet the constraints are coordinates inside the region or on the lines. The feasible region has four vertices that are potential optimal solutions. After determining the feasible region, it is time to draw two iso-value lines of the objective function to find out which vertex represents the optimal solution.

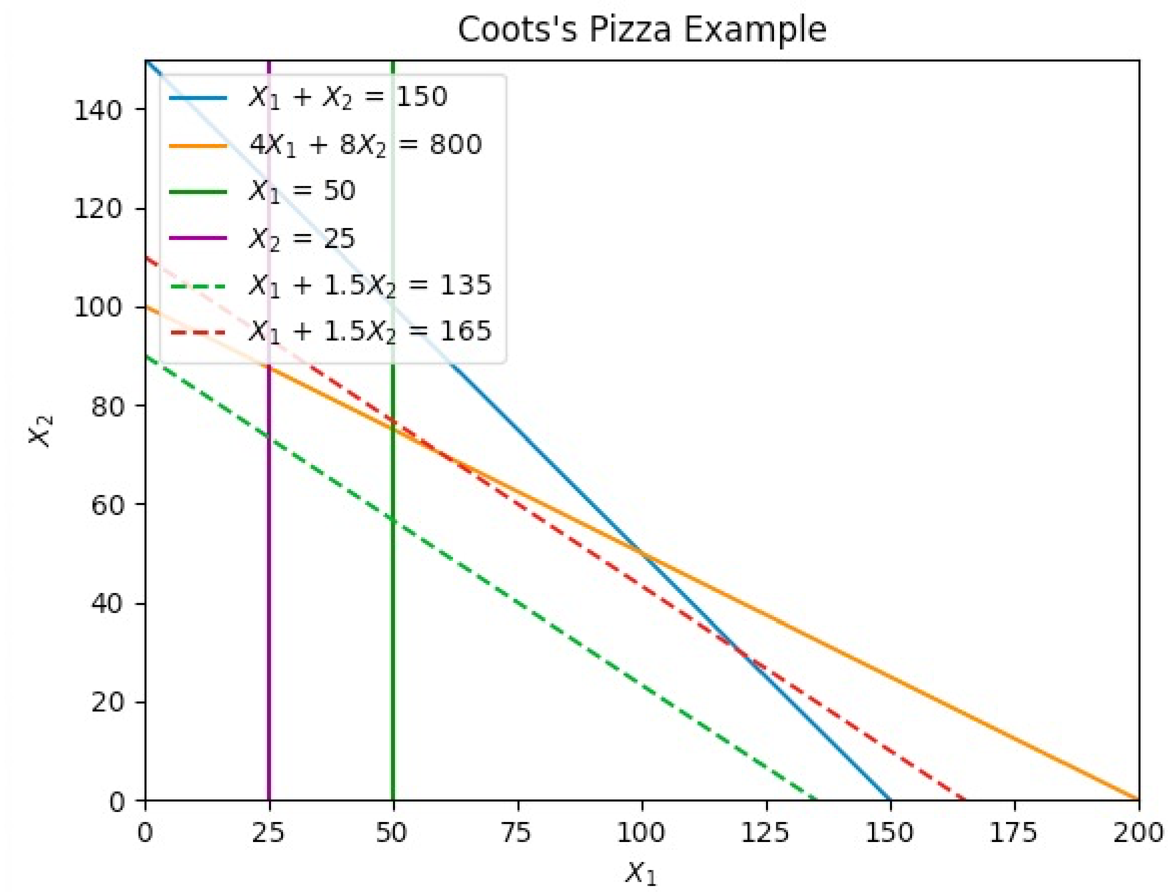

Figure 3.3 — The graph of constraints and iso-value lines of the objective function

In the graph, the green-dashed line represents the case where the objective function has a value of 135, while the red-dashed line represents the case where the objective function has a value of 165. Since the objective function is to maximize profit, the value shall be as great as possible. Based on the graph, the value of the objective function increases as the line shifts to the right, and three of the four vertices of the feasible region are under the red-dashed line, so the optimal solution should be the only vertex left, which is the intersection point of the blue constraint line $X_1 + X_2 = 150$ and the orange constraint line $4X_1 + 8X_2 = 800$. The optimal solution can be calculated and graphed below.

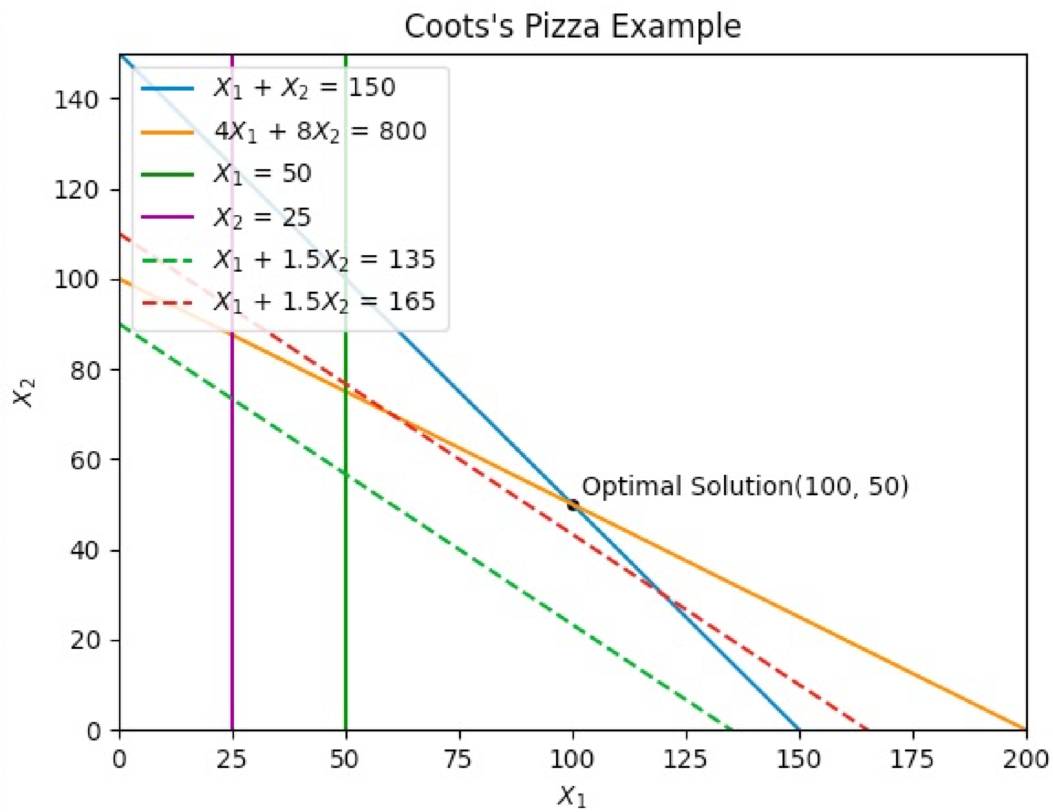

Figure 3.4 — The graph with the optimal solution

By substituting X_2 as $150 - X_1$ and plugging it into $4X_1 + 8X_2 = 800$ to solve for X_1, the intersection point is at $X_1 = 100$ and $X_2 = 50$. $Z = 100 + 1.5 \times 50 = 175$. This means Coots's Pizza can make a maximum profit of 175 dollars by producing one hundred regular pizzas and fifty supreme pizzas.

This example demonstrates how the Graphical Method can visually display the constraints, the feasible zone, and potential values for the objective function. The Graphical

Method intuitively shows how to output an optimal solution for a specific Linear Programming problem. According to Professor Arsham, the Graphical Method “makes the Linear Programming easy to solve.” The solution is one of the vertices of a convex region, and one can calculate all values of the vertices to evaluate the optimal solution since the number of vertices is limited. However, the Graphical Method is also limited to Linear Programming problems mostly with one or two variables. The Graphical Method can also be applied to problems with three variables, but the process will be much tougher and the produced graph will be harder to display. If the number of decision variables is greater than three, it is inefficient and useless to apply the Graphical Method (Arsham).

The Graphical Method is mostly used for simple Linear Programming problems involving two variables, and it is rare to apply the Graphical Method for real-life problems since most problems are more complicated with five or even more variables. Nevertheless, the Graphical Method is one of the most important methods of Linear Programming as it intuitively allows learners to grasp the concept of outputting an optimal solution by checking potential solutions within bounds, which is a key concept in most Linear Programming methods such as the Simplex Method. Moreover, because the Graphical Method contains only simple and straightforward steps, it is a convenient method for one to practice the technique of solving Linear Programming problems.

The Algebraic Method

As its name implies, the Algebraic Method functions by algebraically computing all basic solutions that may satisfy the system of equations of all constraints at binding positions. The idea is to treat all constraints as equations. Then, take certain equations out of the system to calculate a basic solution. After that, check whether the basic solution is feasible by testing it to all the constraints. If a basic solution is feasible, it represents a corner point or vertex of the feasible region. Finally, compare all values of the objective function of the feasible solutions to determine the optimal solution (Arsham).

The following is an example of applying the Algebraic Method to solve a Linear Programming problem. Spencer Luxury is selling two brands of perfumes: Chem-I and Physics-II. The firm can make a profit of five dollars by selling each Chem-I and a profit of three dollars by selling each Physics-II. The budget in total is forty dollars, and purchasing each Chem-I from the manufacturer costs two dollars while purchasing each Physics-II costs one dollar. The company has storage of fifty units for perfumes, and each Chem-I occupies one unit of storage while each Physics-II occupies two units of storage. Spencer Luxury wants to determine the number of Chem-I perfumes and the number of Physics-II perfumes that it should purchase and sell to make a profit as great as possible.

Denote the number of Chem-I perfumes purchased and sold as $X_1$and the number of Physics-II perfume purchased and sold as X_2. A Linear Programming model can be formulated as the following:

Maximize: $Z = 5X_1 + 3X_2$

Subject to: $2X_1 + X_2 \leq 40$

$$X_1 + 2X_2 \leq 50$$

$$X_1 \geq 0$$

$$X_2 \geq 0$$

Here, there are four constraints with two decision variables. It can be treated as a system of equations.

$$2X_1 + X_2 = 40$$

$$X_1 + 2X_2 = 50$$

$$X_1 = 0$$

$$X_2 = 0$$

In this case, there are four equations with two unknowns. By applying the "binomial coefficient" to calculate combinations, there are $C_4^2 = \frac{4!}{2!(4-2)!} = 6$ possible basic solutions. One can list all six following possible scenarios in this case by solving one or two equations and checking whether the solution meets all constraints.

Six Basic Solutions of Spencer Luxury Example

X_1	X_2	$5X_1 + 3X_2$
10	20	110
20	0	100
0	25	75
0	40	Infeasible
50	0	Infeasible

0 0 0

Based on the results, there are four feasible solutions out of six basic solutions. Among the four feasible solutions, the optimal solution is when X_1 is ten and X_2 is twenty, which outputs a maximum value of 110. This indicates that Spencer Luxury should purchase and sell ten Chem-I perfumes and twenty Physics-II perfumes to make a maximum profit of 110 dollars.

The above example demonstrates how the Algebraic Method can be applied to solve simple Linear Programming problems. Similar to the Graphical Method, the Algebraic Method is also pivotally limited to the number of decision variables and constraints. When the number of decision variables and constraints is relatively small, the Algebraic Method can quickly output an optimal solution. However, if the number of decision variables and constraints is big, there will be a large number of possible combinations of basic solutions, and it will be inefficient to calculate all the basic solutions and compare their results. As a result, the Algebraic Method is rarely used for Linear Programming problems.

The Simplex Method

The Graphical Method and the Algebraic Method can intuitively present an optimal solution. However, they are mostly used when the number of decision variables is two. In fact, most real-world problems have more than two or three variables, which makes the Graphical Method and the Algebraic Method ineffective in solving them. As a result, mathematicians prefer to employ more efficient methods such as the Simplex Method for more complex Linear Programming problems.

The Simplex Method devised by George Dantzig in 1947 is regarded as the most widely applied and efficient method to solve Linear Programming problems. According to James Reeb and Scott Leavengood, two professors from Oregon State University, the word "simplex" is a mathematical term with geometric concepts but an algebraic algorithm. "In one dimension, a simplex is a line segment connecting two points. In two dimensions, a simplex is a triangle formed by joining the points. A three-dimensional simplex is a four-sided pyramid having four corners" (Reeb and Leavengood 3).

The idea behind the Simplex Method is similar to that of the Graphical Method. Both the two methods determine an optimal solution by checking the corners, each of which corresponds to a potential-optimal solution of a feasible region. In the Graphical Method, one can find the optimal corner after determining the direction of the objective function. On the other hand, the Simplex Method starts at the corner that has nothing in it. It then moves to a neighboring corner and checks whether the corner improves the solution. It repeats moving to different corners and checking whether there is improvement in the solution until it can no longer improve the solution. When there is no more improvement, the optimal solution is at the most appealing corner (Reeb and Leavengood 3). For a Linear Programming problem with one million possible corners, the Simplex Method will search through one million corners at most. It may sound cumbersome, but in reality, computer programs applying the Simplex Method can quickly output the solution in a few seconds.

A hypothetical instance shown here helps demonstrate the application of the Simplex Method for optimization. Neill Transportation is managing both domestic and international delivery. The company will make a profit of six dollars for delivering each domestic package and will make eight dollars for delivering each international package. There are currently 300 board feet of cardboard and 110 hours of transportation available. Each domestic package requires thirty board feet of cardboard and five hours of transportation. Each international package requires twenty board feet of cardboard and ten hours of transportation. Neill Transportation wants to determine the number of domestic packages and the number of international packages it should deliver to have a maximal profit. Denote the number of domestic packages delivered as X_1 and the number of international packages delivered as X_2. A table of information can be constructed below.

Resource	Domestic(X_1)	International(X_2)	Available
Cardboard(bf)	30	20	300
Transportation(hr)	5	10	110
Unit Profit($)	6	8	

Table 3.1 — Information of Neill Transportation Example

Based on the information, A Linear Programming model can be formulated as the following:

Maximize: $Z = 6X_1 + 8X_2$

Subject to: $30X_1 + 20X_2 \leq 300$

$5X_1 + 10X_2 \leq 110$

$X_1 \geq 0$

$X_2 \geq 0$

After constructing a Linear Programming model, it is time to apply the Simplex Method for solving this problem. The first step of the Simplex Method is to convert all the inequality constraints to equation constraints by adding or subtracting slack variables (Reeb and Leavengood 6). For an inequality constraint that has a less-than-or-equal-to sign, the left side of the constraint adds a slack variable to get an equation. On the other hand, for an inequality constraint that has a greater-than-or-equal-to sign, the left side of the constraint subtracts a slack variable to get an equation. Here, since both of the inequality constraints of this example are less-than-or-equal-to constraints, both of the two constraints add a slack variable to get equations. Denote the slack variable for cardboard constraint as S_c and the slack variable for transportation constraint as S_t. The new Linear Programming model is produced below:

Maximize: $Z = 6X_1 + 8X_2 + 0S_c + 0S_t$

Subject to: $30X_1 + 20X_2 + S_c = 300$

$5X_1 + 10X_2 + S_t = 110$

$X_1 \geq 0$

$X_2 \geq 0$

$S_c \geq 0$

$S_t \geq 0$

Now, there are four variables. The four variables are separated into the basic variable mix and non-basic variables. The basic variables are variables that one intends to solve algebraically, while the non-basic variables are variables given arbitrary values like 0 in order to solve the basic variables. The starting basic variables are usually the slack variables. The second step of the simplex method is to create a simplex tableau that can "evaluate combinations of resources to determine which mix will improve the solution" (Reeb and Leavengood 12). The starting simplex tableau of the transportation example is shown below.

Unit Profit		6	8	0	0	
	Basic Mix	X_1	X_2	S_c	S_t	Solution
	S_c	30	20	1	0	300
	S_t	5	10	0	1	110

Table 3.2 — Starting Simplex Tableau of Neill Transportation

In this tableau, the first row indicates the unit profit of each variable, and all slack variables have a unit profit of 0. The solution mix indicates the values of the basic variables. Since the basic variables here are the slack variables, the other variables are non-basic and will be arbitrarily given a value of zero in this case. As a result, the solution to the first constraint is

$30 * 0 + 20 * 0 + S_c + 0 = 300, S_c = 300$. Similarly, the solution to the second constraint is $5 * 0 + 10 * 0 + 0 + S_t = 110, S_t = 110$. The original coefficients of the variables are called exchange coefficients. They indicate the resources one can exchange for one unit of variable (Reeb and Leavengood 13). For example, 30 in the X_1 column means that thirty board feet of cardboard are exchanged for one domestic package.

The third step of the Simplex Method is to expand the tableau by adding the sacrifice row and improvement row. First, list the unit profit value of the basic variables in the unit-profit column. Since the two basic variables are slack variables, their unit profit values are both 0. The sacrifice row indicates what will be lost in unit profit by making a change in resource allocation, while the improvement row indicates what will be gained by making this change (Reeb and Leavengood 14).

The values in the sacrifice row are determined by the formula: $Unit\ Sacrifice = Unit\ Profit\ Column * Exchange\ Coefficient\ Column$. So, the first value of the sacrifice row is $0 * 30 + 0 * 20 = 0$. Here, the first product is the unit-profit of unused cardboard times the amount of cardboard it takes for one domestic package. The second product is the unit profit of unused transportation hours times the number of transportation hours it takes for one domestic package. The sum of the products is the unit profit sacrifice of delivering one more domestic package. Since the basic variables are slack variables initially and they have unit profit values of zero, the sacrificed profit is thus zero by delivering an extra domestic package. Likewise, the other unit sacrifice values can be calculated in the same way, and all of them are zero. The sacrifice value in the solution column is $0 * 300 + 0 * 110 = 0$, which indicates that the current profit is zero.

On the other hand, the improvement values are calculated by subtracting the unit sacrifice value from the unit profit value. The formula for the improvement row is: $Unit\ Improvement\ =\ Unit\ Profit\ -\ Unit\ Sacrifice$. For example, the first unit improvement value of the X_1 column is $6\ -\ 0\ =\ 0$. Because all values in the sacrifice row are zero, the initial improvement values are equal to the unit profit values.

The expanded simplex tableau with sacrifice and improvement rows is displayed below.

Unit Profit		6	8	0	0	
	Basic Mix	X_1	X_2	S_c	S_t	Solution
0	S_c	30	20	1	0	300
0	S_t	5	10	0	1	110
	Sacrifice	0	0	0	0	0
	Improvement	6	8	0	0	—

Table 3.3 — Simplex Tableau with Sacrifice and Improvement

After completely setting up the simplex tableau, the fourth step is to apply the entry criteria. Start by determining the entering variable. The entering variable is the variable that will optimally improve the solution when it increases from zero. It is called the entering variable because the variable enters the basic mix of the simplex tableau. For a maximization problem, the entering variable is usually the variable with the largest value in the improvement row. Here, the entering variable is X_2 since the objective will increase by eight dollars per unit of international package delivered. Mark the entering variable by placing a downward arrow in the column (Reeb and Leavengood 16).

The fifth step is to determine the exiting variable, which is the variable that will exit the basic mix. The exiting variable is the limiting resource that would run out first. The limiting resource can be determined by choosing the smallest exchange ratio, which is the quotient between the solution and the exchange coefficient in the entering variable column (Reeb and Leavengood 16). Here, the exchange ratios are $300 / 20 = 15$ and $110 / 10 = 11$. Using all 300 board feet of cardboard will make 15 international packages. Using all 110 transportation hours will deliver 11 international packages. Since eleven is smaller, the exiting variable is S_t in this example. Mark the exiting variable by placing an arrow pointing toward S_t. After determining both the entering variable and the exiting variable, highlight the pivot element, which is the intersection of the entering variable column and the exiting variable row (Reeb and Leavengood 17). In this case, the pivot element is ten in the tableau. The pivot value is used after exchanging the X_2 and S_t.

Unit Profit		6	8	0	0	
	Basic Mix	X_1	X_2 ↓	S_c	S_t	Solution
0	S_c	30	20	1	0	300
0	S_t ←	5	10	0	1	110
	Sacrifice	0	0	0	0	0
	Improvement	6	8	0	0	—

Table 3.4 — Simplex Tableau with Entering Variable, Exiting Variable, and Pivot Element

Now, the sixth step is to construct a new simplex tableau and repeat steps three to six until the solution can no longer be improved. There are a series of steps to create the new simplex tableau. First, place the unit profit value of the entering variable in the unit profit column in the exiting variable row. Then, divide all values of the exiting variable row by the pivot element 10. For instance, the intersection element of X_1 column and S_t row divides by 10 yields $5 / 10 = 0.5$.

Unit Profit		6	8	0	0	
	Basic Mix	X_1	X_2	S_c	S_t	Solution
0	S_c					
8	X_2	0.5	1	0	0.1	11
	Sacrifice					
	Improvement					—

Table 3.5 — Second Simplex Tableau with New X_2 Row

After filling the X_2 row, the second step is to fill the S_c row. Start by finding the value in the pivot element column and the S_c row in the original tableau, which is 20. Then, multiply every value in the new X_2 row by 20 and subtract the result from the original value. For example, the first value in the new S_c row should be $30 - 20 \times 0.5 = 20$.

Unit Profit		6	8	0	0	
	Basic Mix	X_1	X_2	S_c	S_t	Solution
0	S_c	20	0	1	-2	80
8	X_2	0.5	1	0	0.1	11
	Sacrifice					
	Improvement					—

Table 3.6 — Second Simplex Tableau with New S_c Row

Now, find the sacrifice row and the improvement of the new Simplex Tableau as before.

Unit Profit		6	8	0	0	
	Basic Mix	X_1	X_2	S_c	S_t	Solution
0	S_c	20	0	1	-2	80
8	X_2	0.5	1	0	0.1	11
	Sacrifice	4	8	0	0.8	88
	Improvement	2	0	0	-0.8	—

Table 3.7 — Second Simple Tableau with Sacrifice and Improvement Row

Here, the profit increases from zero to eighty-eight dollars. The next step is to repeat applying the entering and exiting criteria and constructing new tableaus to improve the solution. The tableau above has an entering variable X_1 and an exiting variable S_c. The pivot element is thus 20.

Unit Profit		6	8	0	0	
	Basic Mix	X_1 ↓	X_2	S_c	S_t	Solution
0	S_c ←	20	0	1	-2	80
8	X_2	0.5	1	0	0.1	11
	Sacrifice	4	8	0	0.8	88
	Improvement	2	0	0	-0.8	—

Table 3.8 — Second Simplex Tableau with Entering and Exiting Variables, and Pivot Element

After determining the entering variable, the exiting variable, and the pivot element, construct a new simplex tableau using previous methods.

Unit Profit		6	8	0	0	
	Basic Mix	X_1	X_2	S_c	S_t	Solution
6	X_1	1	0	0.05	-0.1	4
8	X_2					9
	Sacrifice					
	Improvement					—

Table 3.9 — Third Simplex Tableau with New X_1 Row

Fill the X_2 row using the previous method.

Unit Profit		6	8	0	0	
	Basic Mix	X_1	X_2	S_c	S_t	Solution
6	X_1	1	0	0.05	-0.1	4
8	X_2	0	1	-0.025	0.15	9
	Sacrifice					
	Improvement					—

Table 3.10 — Third Simplex Tableau with New X_2 Row

Now, calculate the sacrifice row and improvement row values.

Unit Profit		6	8	0	0	
	Basic Mix	X_1	X_2	S_c	S_t	Solution
6	X_1	1	0	0.05	-0.1	4
8	X_2	0	1	-0.025	0.15	9
	Sacrifice	6	8	0.1	0.6	96
	Improvement	0	0	-0.1	-0.6	—

Table 3.11 — Third Simplex Tableau with Sacrifice and Improvement Row

After filling the sacrifice row and improvement row, there are no positive values in the improvement row, which indicates that there is no further improvement and the current solution is the optimal output. As a result, the Simplex Method shows that when Neill Transportation delivers four domestic packages and nine international packages, it has a maximum profit of ninety-six dollars.

To sum up, the Simplex Method functions by filling sacrifice and improvement rows, finding the entering variable, exiting variable, and pivot element, eventually constructing a new tableau and repeating the process until there is no available improvement. It solves a sequence of pivot elements in successive tableaus and improves the solution each time a new simplex tableau is constructed.

However, the Simplex Method has some noticeable limitations. According to Sherry Towers, a Canadian statistician and data scientist working at Arizona State University, the Simplex Method can yield a local maximum or minimum instead of a global value based on the starting parameter. Also, "there is no graphical way to determine whether the solution is local or global" and there is no way to reduce the computation time (Towers). What's worse, the complexity of the Simplex Method is $O(nm)$, where m is the number of constraints, and the efficiency of the worst scenario is exponential.

Although the Simplex Method has such disadvantages, it is still one of the most widely used methods since it can efficiently solve complicated problems in most situations, especially with the help of computer data processing. Unlike the Graphical Method or the Algebraic Method, the Simplex Method is a universal method that can solve Linear Programming problems with large numbers of constraints or decision variables, but mathematicians prefer the Graphical Method or the Algebraic Method to the Simplex Method when the problem is relatively simple with few constraints or variables.

Artificial Variables, The Big-M Method, and The Two-Phase Method

In the last section, the constraints are both less-than-or-equal-to constraints, which results in positive slack variables when added to make equations. In this case, since the slack variables have a coefficient of one, they can be used as the starting basic variables. However, when the slack variable has a zero or negative coefficient, there is no basic variable ready for that specific constraint. As a result, artificial variables are added to create basic variables and solve Linear Programming problems

Consider the following problem. Daly Beverage is manufacturing three sodas: A, B, and C. Every day, the shop has a budget of twenty-four dollars, and manufacturing each liter of Soda A costs four dollars, each liter of Soda B costs three dollars, and each liter of Soda C costs one dollar. The shop is planning to use at least thirty kilojoules of energy daily, and producing each liter of Soda A requires two kilojoules, each liter of Soda B requires four kilojoules, and each liter of Soda C requires twelve kilojoules. There are ten hours of labor available in the shop, and it is expected that the hours are efficiently utilized. Producing each liter of Soda A takes two hours of labor, each liter of Soda B takes three hours of labor, and each liter of Soda C takes one hour of labor. The profits of producing each liter of Soda A, Soda B, and Soda C are twenty dollars, six dollars, and nine dollars respectively. Daly Beverage wants to find out the number of different sodas it should produce in order to achieve a maximum profit. A Linear Programming problem can be formulated as the following.

Maximize: $Z = 20A + 6B + 9C$

Subject to: $4A + 3B + C \leq 24$

$2A + 4B + 12C \geq 30$

$2A + 3B + C = 10$

$A \geq 0$

$B \geq 0$

$C \geq 0$

Here, there is one less-than-or-equal-to inequality constraint, one greater-than-or-equal-to inequality constraint, and an equation. Add corresponding slack variables to change the setting.

Maximize: $Z = 20A + 6B + 9C$

Subject to: $4A + 3B + C + s_1 = 24$

$2A + 4B + 12C - s_2 = 30$

$2A + 3B + C = 10$

$A \geq 0$

$B \geq 0$

$C \geq 0$

$s_1 \geq 0$

$s_2 \geq 0$

Now, the slack variable s_1 in the first constraint has a coefficient of positive one, so it can be the starting basic variable for the first constraint. However, the slack variable s_2 in the second constraint has a coefficient of negative one and there is no slack variable in the third constraint (the slack variable in the third constraint has a coefficient of zero), so there would be a basic but infeasible solution for these two constraints. Artificial variables are hence applied to convert the basic but infeasible solution to basic and feasible solutions for the constraints. The new model after adding artificial variables is shown below.

Maximize: $Z = 20A + 6B + 9C$

Subject to: $4A + 3B + C + s_1 = 24$

$2A + 4B + 12C - s_2 + A_2 = 30$

$2A + 3B + C + A_3 = 10$

$A \geq 0$

$B \geq 0$

$C \geq 0$

$s_1 \geq 0$

$s_2 \geq 0$

$A_2 \geq 0$

$A_3 \geq 0$

After adding the artificial variables, there are two methods to solve the Linear Programming: the Big-M Method and the Two-Phase Method.

The Big-M Method

As its name suggests, the Big-M Method gives the coefficients of artificial variables in the objective function with huge numbers. The letter M represents large numbers that replace infinity. It is usually positive infinity for minimization problems and negative infinity for maximization problems. In this example, since it is a maximization problem, the coefficients shall be negative. Instead of infinity, the coefficients, or the unit profit values, of the artificial variables are set to be -1000. After setting the coefficient values, use the same steps of the Simplex Method to solve the Linear Programming problem by constructing several simplex tableaus.

Unit Profit		20	6	9	0	0	-1000	-1000	
	Basic Mix	A	B	C ↓	s_1	s_2	A_2	A_3	Solution
0	s_1	4	3	1	1	0	0	0	24
-1000	A_2 ←	2	4	12	0	-1	1	0	30
-1000	A_3	2	3	1	0	0	0	1	10
	Sacrifice	-4000	-7000	-13000	0	1000	-1000	-1000	-4000
	Improvement	4020	7006	13009	0	-1000	0	0	—

Table 3.12 — First Tableau of the Big-M Method

After an artificial variable leaves the tableau, it will drop from the problem and its values in the column will all become blank.

Unit Profit		20	6	9	0	0	-1000	-1000	
	Basic Mix	A	B ↓	C	s_1	s_2	A_2	A_3	Solution
0	s_1	23/6	8/3	0	1	1/12	—	0	21.5
9	C	1/6	1/3	1	0	-1/12	—	0	2.5
-1000	A_3 ←	11/6	8/3	0	0	1/12	—	1	7.5
	Sacrifice	-1831.8	-2663.7	9	0	-84.1	—	-1000	-7477.5
	Improvement	1851.8	2669.7	0	0	84.1	—	0	—

Table 3.13 — Second Tableau of the Big-M Method

Unit Profit		20	6	9	0	0	-1000	-1000	
	Basic Mix	A ↓	B	C	s_1	s_2	A_2	A_3	Solution
0	s_1	2	0	0	1	0	—	—	14
9	C	-1/16	0	1	0	-3/32	—	—	25/16
6	B ←	11/16	1	0	0	1/32	—	—	45/16
	Sacrifice	57/16	6	9	0	-21/32	—	—	495/16
	Improvement	273/16	0	0	0	21/32	—	—	—

Table 3.14 — Third Tableau of the Big-M Method

Unit Profit		20	6	9	0	0	-1000	-1000	
	Basic Mix	A	B	C	s_1	s_2	A_2	A_3	Solution
0	s_1	0	-32/11	0	1	-1/11	—	—	64/11
9	C	0	1/11	1	0	-1/11	—	—	20/11
20	A	1	16/11	0	0	1/22	—	—	45/11
	Sacrifice	20	329/11	9	0	1/11	—	—	1080/11
	Improvement	0	-263/11	0	0	-1/11	—	—	—

Table 3.15 — Final Tableau of the Big-M Method

Based on the tableau, the optimal solution produced using the Big-M Method is when $A = 45/11$, $B = 0$, and $C = 20/11$. In other words, Daly Beverage should produce 45/11 liters of Soda A, 0 liters of Soda B, and 20/11 liters of Soda C to earn a maximum profit of 1080/11 dollars.

In order for the algorithm to work, the M, which is the coefficient of artificial variables in the objective function, is usually several times larger than those of the decision variables. The large difference will cause the algorithm to be unstable or lead to huge round-off errors in computer programs. As a result, another more practical algorithm is preferred over the Big-M Method: the Two-Phase Method.

The Two-Phase Method

The Two-Phase Method essentially does the same thing as the Big-M Method: it converts the infeasible solution into a feasible solution first and then finds the best feasible solution. The goal of Phase 1 is to change the basic but infeasible solution into a basic and feasible solution as well as remove the artificial variables from the problem. Phase 2 begins when artificial variables are removed from the tableau and repeat the same steps of the Simplex Method to find out the optimal solution.

The same Daly Beverage problem shown above can be solved using the Two-Phase Method.

Phase 1

In Phase 1, all decision variables are set to have a unit profit value (the coefficients in the objective function) of zero, while all artificial variables have a unit profit value of negative one. In this way, the artificial variables will leave the basis at the beginning of the algorithm and output an initial basic and feasible solution.

Unit Profit		0	0	0	0	0	-1	-1	
	Basic Mix	A	B	C ↓	s_1	s_2	A_2	A_3	Solution
0	s_1	4	3	1	1	0	0	0	24
-1	A_2 ←	2	4	12	0	-1	1	0	30
-1	A_3	2	3	1	0	0	0	1	10
	Sacrifice	-4	-7	-13	0	1	-1	-1	-40
	Improvement	4	7	13	0	-1	0	0	—

Table 3.16 — First Tableau of Phase 1

Unit Profit		0	0	0	0	0	-1	
	Basic Mix	A	B ↓	C	s_1	s_2	A_3	Solution
0	s_1	23/6	8/3	0	1	1/12	0	21.5
0	C	1/6	1/3	1	0	-1/12	0	2.5
-1	A_3 ←	11/6	8/3	0	0	1/12	1	7.5
	Sacrifice	-11/6	-8/3	0	0	-1/12	-1	-7.5
	Improvement	11/6	8/3	0	0	1/12	0	—

Table 3.17 — Final Tableau of Phase 1

Phase 2

The basic feasible solution at the end of Phase 1 will be used as the initial basic feasible solution at the beginning of Phase 2. Repeat the same process of the Simplex Method to find out the optimal solution. Set all the unit profit values of the decision variables as they are in the objective function.

Unit Profit		20	6	9	0	0	
	Basic Mix	A ↓	B	C	s_1	s_2	Solution
0	s_1	2	0	0	1	0	14
9	C	-1/16	0	1	0	-3/32	25/16
6	B ←	11/16	1	0	0	1/32	45/16
	Sacrifice	57/16	6	0	0	-21/32	495/16
	Improvement	273/16	0	0	0	21/32	—

Table 3.18 — First Tableau of Phase 2

Unit Profit		20	6	9	0	0	
	Basic Mix	A	B	C	s_1	s_2	Solution
0	s_1	0	-32/11	0	1	-1/11	64/11
9	C	0	1/11	1	0	-1/11	20/11
20	A	1	16/11	0	0	1/22	45/11
	Sacrifice	20	329/11	9	0	1/11	1080/11
	Improvement	0	-263/11	0	0	-1/11	—

Table 3.19 — Optimal Tableau of Phase 2

As it demonstrates, the tableaus of Phase 2 are exactly the same as those of the Big-M Method after artificial variables leave the basis. The results of the two methods are hence the same since both algorithms are essentially the extension of the Simplex Method with artificial variables. However, the Two-Phase Method is more convenient and stable compared to the Big-M Method as it does not involve numbers with incredibly large magnitudes.

Chapter 4

Computer Programs Applying Linear Programming

Because of the need to process large data, computer programs that code for the algorithms of Linear Programming methods are frequently employed to solve for optimal solutions.

This section will display and explain two computer programs coded by me in Python. The first program utilizes the *matplotlib* module to create a graphical function that can draw lines applying the Graphical Method. The graphs created by the program have been exhibited in previous sections. The second program imports *linprog* function from *scipy.optimize* module to numerically output an optimal solution using Linear Programming methods, which will be the Simplex Method in this case.

The Graphical Method Computer Program

There are three parts in the program for the Graphical Method: import, function, and inputs. The green comment lines in the code explain the goal and logic of the specific codes.

Import

```
#Import
from cProfile import label
import matplotlib.pyplot  as plt
import numpy as np
import math

from numpy.core.fromnumeric import size
```

Figure 4.1 — Import Part of The Graphical Method Program

The most important *import* is module *matplotlib.pyplot*, which is the main tool used in the Python program for drawing graphs. One needs to install the *matplotlib* module before coding this program. Importing this module as *plt* allows it easier to call the object in the later sections. Another important import is the *numpy* module. The *np* will help give an x variable for drawing lines.

Function

```
#Function
def graphical(titleOFProblem, x1_ax, x2_ax, numberOfConstraints, bmList, constraintLines, objSlope, objIntercept, objLines):
    plt.title(titleOFProblem)
    plt.axis([0, x1_ax, 0, x2_ax])
    x1 = np.linspace(0, x1_ax)
    plt.ylabel('$X_{2}$')
    plt.xlabel('$X_{1}$')
    for i in range(0,numberOfConstraints):
        slope = bmList[i][0]
        intercept = bmList[i][1]
        plt.plot(x1, slope * x1 + intercept, label= constraintLines[i])
    plt.axvline(x= 50, label='$X_1$ = 50', color= 'green')
    plt.axvline(x=25, label = '$X_2$ = 25', color = 'purple')
    plt.plot(x1, objSlope * x1 + objIntercept[0], '--', label= objLines[0])
    plt.plot(x1, objSlope * x1 + objIntercept[1], '--', label= objLines[1])
    plt.legend(loc='upper left')
```

Figure 4.2 — Function Part of The Graphical Method Program Overview

The figure above is the overall code of the function part. It has two subparts: set-up and graphing. The following explains each subpart separately.

- **Set-Up**

```
def graphical(titleOFProblem, x1_ax, x2_ax, numberOfConstraints, bmList, constraintLines, objSlope, objIntercept, objLines):
    #titleOfProblem: the title of the graph
    #x1_ax: the horizontal maximum value
    #x2_ax: the vertical maximum value
    #numberOfConstraints: the number of constraints excluding non-negativity constriants
    #bmList: the slopes of constraint lines
    #objSlope: the slope of objective function
    #objIntercept: the vertical intercepts of possible objective function lines
    #objLines: label of possible objective function lines
```

Figure 4.3 — Function and Parameters

Here shows the function and specific parameters of the defined function. There are eight parameters that require users to input. *titleOfProblem* is a string parameter that will show the title of the graph at the top. *x1_ax* and *x2_ax* are two numbers that represent the maximum of horizontal and vertical axes respectively. This can be calculated based on constraints and will help scale the graph appropriately. *numberOfConstraints* is an integer parameter that represents how many lines of constraints will be graphed. It omits the non-negativity constraints. *bmList* is a list parameter including the slopes of all the different constraint lines. The slopes need to be calculated by users based on the given information. *objSlope* is the slope of the objective function, which also needs to be calculated by the user based on the objective function given. *objIntercept* is a list of vertical intercepts chosen by the user for potential objective function lines. The intercepts need to be chosen appropriately to better show the direction of the objective function. *objLines* is a list parameter that contains the string labels for the potential objective function lines.

```
plt.title(titleOFProblem) #Set the title of the graph
plt.axis([0, x1_ax, 0, x2_ax]) #Set the window of the graph based on horizontal and vertical maximum values
x1 = np.linspace(0, x1_ax) #Set x as the horizontal variable

#Set the name of the axises
plt.ylabel('$X_{2}$')
plt.xlabel('$X_{1}$')
```

Figure 4.4 — Set-up Codes

The first step is to name the graph. The *plt.title()* function will set the title of the graph according to the given parameter. The *plt.axis()* function will scale the graph based upon the horizontal and vertical maximum values input by the user. Because of the non-negativity constraint, both the horizontal and vertical axis starts with zero. The first two parameters in the *plt.axis()* function represent the minimum and maximum value of the horizontal axis, while the last two parameters represent the minimum and maximum value of the vertical axis. The *np.linspace()* will give a variable that represents the horizontal or vertical variable. The *plt.ylabel()* function will set the name of the vertical axis while the *plt.xlabel()* function will set the name of the horizontal axis.

- **Graphing**

```
    #Draw each constraint
    for i in range(0,numberOfConstraints):
        slope = bmList[i][0]
        intercept = bmList[i][1]
        plt.plot(x1, slope * x1 + intercept, label= constraintLines[i])

    #Draw the vertical lines
    plt.axvline(x= 50, label='$X_1$ = 50', color= 'green')
    plt.axvline(x=25, label = '$X_2$ = 25', color = 'purple')

    #Draw the potential objective function lines
    plt.plot(x1, objSlope * x1 + objIntercept[0], '--', label= objLines[0])
    plt.plot(x1, objSlope * x1 + objIntercept[1], '--', label= objLines[1])

    #Display the Legend of the graph
    plt.legend(loc='upper left')
```

Figure 4.5 — Graphing Codes

The graphing part mainly consists of drawing the constraint lines and the potential objective functions lines. The constraints are drawn using a *for* loop. The *for* loop will traverse the *bmList* to extract each constraint's slope and the vertical intercept value. Then, the *plt.plot()* function will draw the lines in slope-intercept form ($y = mx + b$). The parameters of the *plot()* function include x-variable, the line in the slope-intercept form, and also the label of the constraint line. For constraints that are vertical lines such as $x = 25$, the *plt.axvline()* function is applied to plot the lines. To draw the potential objective function lines, instead of using a *for* loop, it draws two lines directly since the number of potential objective function lines is two by default. In the end, call the *plt.legend()* function to display the legend with the given location.

Input

```
#Input
b_m = [[-1, 150], [-1/2, 100]] #Slope and y-intercept of constraint lines
constraints = ['$X_{1}$ + $X_{2}$ = 150', '4$X_{1}$ + 8$X_{2}$ = 800'] #Constraint labels
inter = [90, 110] #y-intercepts of potential objective function lines
lines = ['$X_{1}$ + 1.5$X_{2}$ = 135', '$X_{1}$ + 1.5$X_{2}$ = 165'] #Potential objective function line labels
graphical("Coots's Pizza Example", 200, 150, 2, b_m, constraints, -2/3, inter, lines) #Call the graphing function

plt.scatter(100,50, color='black', s = 15) #Draw the intersection point
plt.annotate('Optimal Solution(100, 50)', xy=(102,52)) #Annotate the intersection point
plt.show() #Display the window of graph
```

Figure 4.6 — Input Part Codes

The input part is basically setting values for the graphing function. The *b_m* is a list variable containing lists of constraints, each inner list contains the slope and y-intercept of the constraint line. The values are calculated based on the given information. The *constraints* is a list of strings that label the corresponding constraints. The *inter* is a list of y-intercepts of potential objective function. The user sets the values appropriately to show the direction of the objective function. The *lines* is a list of strings that label the corresponding potential objective function lines. Then, the *graphical()* function is called to plot the graph with the given parameters. The *plt.scatter()* function will draw the intersection point, which is the optimal solution corner and is calculated by the user based on the intersecting lines. The *plt.annotate()* function will mark the intersection point. Notice that the coordinate-point in the *annotate()* function is slightly different from the actual intersection point. This is to avoid overlapping with the intersection point and to better label the intersection point. In the end, the *plt.show()* function is called to display the window of the graph.

The graph below is the graph of the Coots's Pizza example mentioned in the Methods section using the inputs shown in the pictures above.

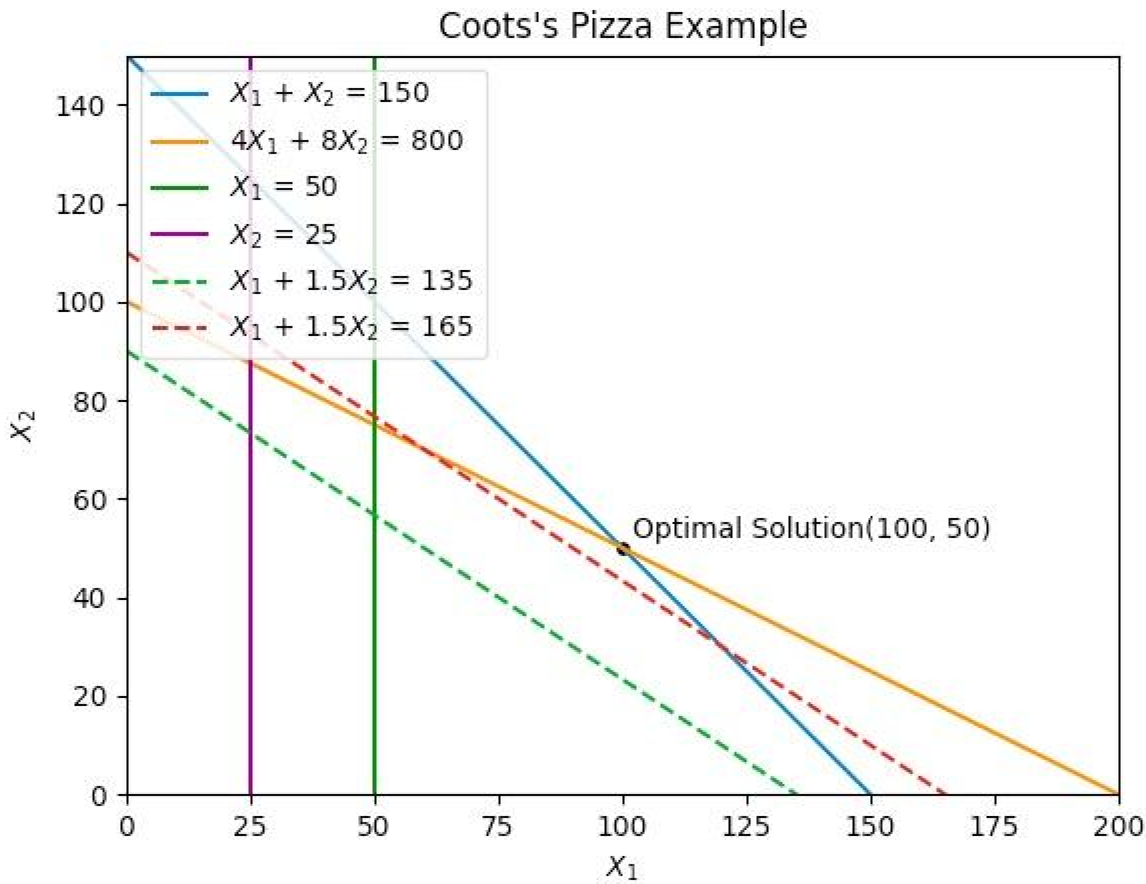

Figure 4.7 — Coots's Pizza Example Graph

The Simplex Method Computer Program

Like the Graphical Method Computer Program, the Simplex Method Computer Program also has three parts: import, function, and input.

Import

```
from scipy.optimize import linprog
import math
```

Figure 4.8 — Import Part of The Simplex Method Computer Program

The import part of the Simplex Method Program is straightforward. Import *linprog()* function from *scipy.optimize*. This will be the most crucial function in later parts of the program as the *linprog()* function automatically outputs the optimal value.

Function

```
def simplex(numOfVar, objCoef, inEqualityLeft, inEqualityRight, bound, containsEquality, isMax, equalityLeft = [], equalityRight = []):

    #Create corresponding variables, which are in the form of list
    objective = objCoef
    left_ineq = inEqualityLeft
    right_ineq = inEqualityRight


    #Check whether there is equality.
    if(containsEquality):
        #Set the left/right equality accordingly
        left_eq = equalityLeft
        right_eq = equalityRight
        #Create an object representing the outcome accordingly with the equality parameter
        outcome = linprog(c=objective, A_ub=left_ineq, b_ub=right_ineq, A_eq=left_eq, b_eq=right_eq, bounds=bound, method="revised simplex")

    #Create an object representing the outcome without the equality parameter
    else:
        outcome = linprog(c=objective, A_ub=left_ineq, b_ub=right_ineq, bounds=bound, method="revised simplex")

    #Print information of the outcome
    print("Whether there is an optimal solution: " + str(outcome.success))
    print()
    if(isMax):
        print("The value of optimal solution: " + str(outcome.fun/-1))
    else:
        print("The value of optimal solution: " + str(outcome.fun))
    print()
    print("The corresponding values of decision variables: " + str(outcome.x))
```

Figure 4.9 — Function Part of The Simplex Method Program

There are nine parameters for the function. The *numOfVar* represents the number of decision variables in the problem. The *objCoef* is a list consisting of the coefficients of the decision variables in the objective function. The *inEqualityLeft* is a 2D list that contains lists of the coefficients on the left side of inequality constraints. On the other hand, the *inEqualityRight* is a one-dimensional list containing the right side number of inequality constraints. The *bound* is a list parameter consisting of tuples, and each tuple represents the range of values of the decision variables. The *containsEquality* parameter is a boolean variable. If *True*, it contains equality. If *False*, it does not contain equality. The *isMax* parameter checks whether it is a maximization problem or a minimization problem. The *equalityLeft* and *equalityRight* are similar to the *inEqualityLeft* and the *inEqualityRight* respectively, but they are set to contain null lists by default unless there is an equation constraint.

The function first copies the values of the *objCoef*, *inEqualityLeft*, and *inEqualityRight* to newly created variables. Then, the function checks whether there is an equation constraint based on the value of the *containsEquality* variable. If there is, it will do the same thing to copy values of the *equalityLeft* and the *equalityRight*. Then, the function will call the *linprog()* function to automatically generate the optimal solution object. The *c* inputs the coefficients of decision variables in the objective function. The *A_ub* inputs the coefficients on the left side of inequality constraints, and the *b_ub* inputs the coefficients on the right side of inequality constraints. If there is an equation constraint, the *A_eq* and *b_eq* input the coefficients on the left and right side of equation constraints respectively. The *bound* is set to be the bounds of the decision variables, and the *method* for Linear Programming is set to be the Revised Simplex Method here. Since the *linprog()* function treats the problem as a minimization problem by default, the value of the optimal solution, *outcome.fun*, needs to be divided by negative one to have the expected value. The function will also print whether there is an optimal solution and the decision variables for the optimal solution.

Input

```
#Input
obj = [-1,-1.5]
left = [[1,1],[4,8]]
right = [150,800]
bounds = [(0,200), (0,140)]
simplex(2, obj, left, right, bounds, False, True)
```

Figure 4.10 — Input Part of the Simplex Method Program

For the inputs, most of them can be directly taken from the given information of the problem. However, since the *linprog()* function calculates the optimal solution as a minimization problem, the coefficients of a maximization problem need to divide negative one to become a minimization problem. For example, the coefficients in $Z = 1X_1 + 1.5X_2$ will become -1 and -1.5 respectively.

```
pc@Hebe Capstone LP % /usr/local/bin/python3 "/Users/pc/Capstone LP/LP_Simplex.py"
Whether there is an optimal solution: True

The value of optimal solution: 175.0

The corresponding values of decision variables: [100.  50.]
pc@Hebe Capstone LP %
```

Figure 4.11 — Result of the Simplex Method Program

Above is the result of running this program to solve the Coots's Pizza example problem. Which is the same result produced by the graphical method. The Simplex Method Program can be extremely useful when calculating large data sets using the computer program.

Chapter 5

Applications of Linear Programming

As a tool for optimization, Linear Programming has been widely used to discover the complex relationships between profits and costs with linear equations or linear inequalities. It is usually helpful for determining the maximum profits or minimum costs. The fields that apply Linear Programming include agriculture, transportation, marketing, investment planning, advertising, and so forth.

This section focuses on the application of Linear Programming in the field of business planning and decision making. It covers several hypothetical case studies as well as real-life examples.

Case Studies of Applying Linear Programming

Because Linear Programming addresses linear equations or linear inequality constraints, which are not always the case in real-life situations, most applications that purely involve Linear Programming are hypothetical examples. Still, these applications resemble some real-life problems that would employ the Linear Programming technique to achieve optimization. Here, two case studies are shown to illustrate the application of Linear Programming.

1. Marketing Case Study

This is a hypothetical case study provided by J.K. Sharma, author of the book *Operations Research: Theory and Application*. It is a hypothetical application of Linear Programming in the field of marketing.

A businessperson is opening a new restaurant and has $800,000 available for advertising next month. There are four types of advertising that he can choose: "(i) 30-second television commercials; (ii) 30-second radio commercials; (iii) Half-page advertisements in a newspaper; (iv) Full-page advertisements in a weekly magazine that will appear in four times during the coming month." The businessman wants to advertise to families with incomes over $50,000 and families with incomes below $50,000 using different strategies. He wants to maximize the number of exposures by determining the number of each type of advertisement. The data on costs and exposure of each type of advertisement are given below.

Media	*Cost of Advertisement (Rs) Rs 50,000 (a)*	*Exposure to Families with Annual Income Over Rs 50,000 (b)*	*Exposure to Families with Annual Income Under*
Television	40,000	2,00,000	3,00,000
Radio	20,000	5,00,000	7,00,000
Newspaper	15,000	3,00,000	1,50,000
Magazine	5,000	1,00,000	1,00.000

Figure 5.1 — Information about Marketing Example

The businessperson has also identified four constraints: (i) The number of television advertisements shall not exceed four; (ii) The number of advertisements in magazines shall not exceed four; (iii) The percent of advertisements in newspapers and magazines shall be no more than sixty percent of all advertisements; (iv) The exposure to families with annual incomes over \$50,000 must be at least 45,00,000 (Sharma). Let X_1, X_2, X_3, X_4 denote the number of advertisements on television, radio, newspaper, and magazine, respectively. The problem can be formulated as the following model:

Maximize: $Z = 500,000X_1 + 1,200,000X_2 + 450,000X_3 + 200,000X_4$

Subject to: $40,000X_1 + 20,000X_2 + 15,000X_3 + 5,000X_4 \leq 800,000$

$X_1 \leq 4$

$X_4 \leq 4$

$\frac{X_3+X_4}{X_1+X_2+X_3+X_4} \leq 0.6$ or $-0.6X_1 - 0.6X_2 + 0.4X_3 + 0.4X_4 \leq 0$

$200,000X_1 + 500,000X_2 + 300,000X_3 + 100,000X_4 \geq 4,500,000$

$X_1 \geq 0$

$X_2 \geq 0$

$X_3 \geq 0$

$X_4 \geq 0$

Now, plug the data into the Python program to calculate the optimal solution using the simplex method. The result is shown below.

```
Whether there is an optimal solution: True

The value of optimal solution: 48000000.0

The corresponding values of decision variables: [ 0. 40.  0.  0.]
```

Figure 5.2 — Results of Marketing Case Study

In this case, the optimal solution is when the businessperson chooses forty times of advertisements on the radio, which will result in a maximum number of exposures of 48,000,000.

2. **Transportation Case Study**

Transportation firms often employ Linear Programming to maximize the amount that can be transported or minimize the routine costs. This is a hypothetical example of minimizing busing costs in a district provided by Catherine Lewis, a Senior Actuary at Cambia Health Solutions.

One middle school in the district of Springfield school board has been closed next year. The board wants to assign all the sixth, seventh, and eighth-grade students from that school to the three remaining schools. The school district wants to reassign the students in a way that minimizes the transportation costs. The costs for transporting each student from six residential areas are shown in the figure below. 0 indicates that there is no cost and dash indicates that it is an infeasible management. School 1 can accept up to 900 students. School 2 can accept no more than 110 students, and school 3 can accept up to 1000 students. There is also a restriction that each grade must constitute between thirty and thirty-six percent of each school's population (Lewis).

Area	No. of Students	Number in 6th Grade	Number in 7th Grade	Number in 8th Grade	Busing Cost per Student School 1	School 2	School 3
1	450	144	171	135	\$300	0	\$700
2	600	222	168	210	-	\$400	\$500
3	550	165	176	209	\$600	\$300	\$200
4	350	98	140	112	\$200	\$500	-
5	500	195	170	135	0	-	\$400
6	450	153	126	171	\$500	\$300	0

Figure 5.3 — Information on Bussing Costs

Let x_{ij} indicates the number of sixth-grade students from the area i to school j. Let y_{ij} indicates the number of seventh-grade students from the area i to school j. Let z_{ij} indicates the the number of eighth-grade students from the area i to school j. The domain of i is between 1 and 6, while the domain of j is between 1 and 3. The problem can be formulated as the following Linear Programming model.

Minimize: $$Z = 300(x_{11} + y_{11} + z_{11}) + 700(x_{13} + y_{13} + z_{13})$$

$$+ 400(x_{22} + y_{22} + z_{22}) + 500(x_{23} + y_{23} + z_{23})$$

$$+ 600(x_{31} + y_{31} + z_{31}) + 300(x_{32} + y_{32} + z_{32})$$

$$+ 200(x_{33} + y_{33} + z_{33}) + 200(x_{41} + y_{41} + z_{41})$$

$$+\ 500(x_{42} + y_{42} + z_{42})\ +\ 400(x_{53} + y_{53} + z_{53})$$

$$+\ 500(x_{61} + y_{61} + z_{61})\ +\ 300(x_{62} + y_{62} + z_{62})$$

Subject to: Number Constraints:

$x_{11} + y_{11} + z_{11} + x_{31} + y_{31} + z_{31} + x_{41} + y_{41} + z_{41} + x_{51} + y_{51} + z_{51} + x_{61} + y_{61} + z_{61} \leq 900$

$x_{12} + y_{12} + z_{12} + x_{22} + y_{22} + z_{22} + x_{32} + y_{32} + z_{32} + x_{42} + y_{42} + z_{42} + x_{62} + y_{62} + z_{62} \leq 110$

$x_{13} + y_{13} + z_{13} + x_{23} + y_{23} + z_{23} + x_{33} + y_{33} + z_{33} + x_{53} + y_{53} + z_{53} + x_{63} + y_{63} + z_{63} \leq 1000$

Sixth Grade Constraints:

$x_{11} + x_{12} + x_{13} = 144$

$x_{22} + x_{23} = 222$

$x_{31} + x_{32} + x_{33} = 165$

$x_{41} + x_{42} = 98$

$x_{51} + x_{53} = 195$

$x_{61} + x_{62} + x_{63} = 153$

Seventh Grade Constraints:

$y_{11} + y_{12} + y_{13} = 171$

$y_{22} + y_{23} = 168$

$y_{31} + y_{32} + y_{33} = 176$

$y_{41} + y_{42} = 140$

$y_{51} + y_{53} = 170$

$y_{61} + y_{62} + y_{63} = 126$

Eigth Grade Constraints:

$z_{11} + z_{12} + z_{13} = 135$

$z_{22} + z_{23} = 210$

$z_{31} + z_{32} + z_{33} = 209$

$z_{41} + z_{42} = 112$

$z_{51} + z_{53} = 135$

$z_{61} + z_{62} + z_{63} = 171$

Sixth Grade Percentage Constraints:

$.64x_{11} + .64x_{31} + .64x_{41} + .64x_{51} + .63x_{61} - .36y_{11} - .36y_{31}$
$-.36y_{41} - .36y_{51} - .36y_{61} - .36z_{11} - .36z_{31} - .36z_{41} - .36z_{51} - .36z_{61} \leq 0$

$.7x_{11} + .7x_{31} + .7x_{41} + .7x_{51} + .7x_{61} - .3y_{11} - .3y_{31}$
$-.3y_{41} - .3y_{51} - .3y_{61} - .3z_{11} - .3z_{31} - .3z_{41} - .3z_{51} - .3z_{61} \geq 0$

$.64x_{12} + .64x_{22} + .64x_{32} + .64x_{42} + .63x_{62} - .36y_{12}$
$- .36y_{22} - .36y_{32} -.36y_{42} - .36y_{62} - .36z_{12} - .36z_{22} - .36z_{32} - .36z_{42} - .36z_{62} \leq 0$

$.7x_{12} + .7x_{22} + .7x_{32} + .7x_{42} + .7x_{62} - .3y_{12} - .3y_{22} - .3y_{32}$

$-.3y_{42} - .3y_{62} - .3z_{12} - .3z_{22} - .3z_{32} - .3z_{42} - .3z_{62} \geq 0$

$.64x_{13} + .64x_{23} + .64x_{33} + .64x_{53} + .64x_{63} - .36y_{13} - .36y_{23} - .36y_{33}$

$-.36y_{53} - .36y_{63} - .36z_{13} - .36z_{23} - .36z_{33} - .36z_{53} - .36z_{63} \leq 0$

$.7x_{13} + .7x_{23} + .7x_{33} + .7x_{53} + .7x_{63} - .3y_{13} - .3y_{23} - .3y_{33}$

$-.3y_{53} - .3y_{63} - .3z_{13} - .3z_{23} - .3z_{33} - .3z_{53} - .3z_{63} \geq 0$

Seventh Grade Percentage Constraints:

$.64y_{11} + .64y_{31} + .64y_{41} + .64y_{51} + .64y_{61} - .36x_{11} - .36x_{31}$

$-.36x_{41} - .36x_{51} - .36x_{61} - .36z_{11} - .36z_{31} - .36z_{41} - .36z_{51} - .36z_{61} \leq 0$

$.7y_{11} + .7y_{31} + .7y_{41} + .7y_{51} + .7y_{61} - .3x_{11} - .3x_{31}$

$-.3x_{41} - .3x_{51} - .3x_{61} - .3z_{11} - .3z_{31} - .3z_{41} - .3z_{51} - .3z_{61} \geq 0$

$.64y_{12} + .64y_{22} + .64y_{32} + .64y_{42} + .64y_{62} - .36x_{12} - .36x_{22} - .36x_{32}$

$-.36x_{42} - .36x_{62} - .36z_{12} - .36z_{22} - .36z_{32} - .36z_{42} - .36z_{62} \leq 0$

$.7y_{12} + .7y_{22} + .7y_{32} + .7y_{42} + .7y_{62} - .3x_{12} - .3x_{22} - .3x_{32}$

$-.3x_{42} - .3x_{62} - .3z_{12} - .3z_{22} - .3z_{32} - .3z_{42} - .3z_{62} \geq 0$

$.64y_{13} + .64y_{23} + .64y_{33} + .64y_{53} + .64y_{63} - .36x_{13} - .36x_{23} - .36x_{33}$

$-.36x_{53} - .36x_{63} - .36z_{13} - .36z_{23} - .36z_{33} - .36z_{53} - .36z_{63} \leq 0$

$.7y_{13} + .7y_{23} + .7y_{33} + .7y_{53} + .7y_{63} - .3x_{13} - .3x_{23} - .3x_{33}$

$-.3x_{53} - .3x_{63} - .3z_{13} - .3z_{23} - .3z_{33} - .3z_{53} - .3z_{63} \geq 0$

Eighth Grade Percentage Constraints:

$.64z_{11} + .64z_{31} + .64z_{41} + .64z_{51} + .64z_{61} - .36x_{11} - .36x_{31}$

$-.36x_{41} - .36x_{51} - .36x_{61} - .36y_{11} - .36y_{31} - .36y_{41} - .36y_{51} - .36y_{61} \leq 0$

$.7z_{11} + .7z_{31} + .7z_{41} + .7z_{51} + .7z_{61} - .3x_{11} - .3x_{31}$

$-.3x_{41} - .3x_{51} - .3x_{61} - .3y_{11} - .3y_{31} - .3y_{41} - .3y_{51} - .3y_{61} \geq 0$

$.64z_{12} + .64z_{22} + .64z_{32} + .64z_{42} + .64z_{62} - .36x_{12} - .36x_{22} - .36x_{32}$

$-.36x_{42} - .36x_{62} - .36y_{12} - .36y_{22} - .36y_{32} - .36y_{42} - .36y_{62} \leq 0$

$.7z_{12} + .7z_{22} + .7z_{32} + .7z_{42} + .7z_{62} - .3x_{12} - .3x_{22} - .3x_{32}$

$-.3x_{42} - .3x_{62} - .3y_{12} - .3y_{22} - .3y_{32} - .3y_{42} - .3y_{62} \geq 0$

$.64z_{13} + .64z_{23} + .64z_{33} + .64z_{53} + .64z_{63} - .36x_{13} - .36x_{23} - .36x_{33}$

$-.36x_{53} - .36x_{63} - .36y_{13} - .36y_{23} - .36y_{33} - .36y_{53} - .36y_{63} \leq 0$

$.7z_{13} + .7z_{23} + .7z_{33} + .7z_{53} + .7z_{63} - .3x_{13} - .3x_{23} - .3x_{33}$

$-.3x_{53} - .3x_{63} - .3y_{13} - .3y_{23} - .3y_{33} - .3y_{53} - .3y_{63} \geq 0$

Using the computer program to solve, the minimum busing cost is $555555.60.

Real-Life Examples of Applying Linear Programming

Linear Programming is also applied in real-life situations for optimizations. Most companies utilize Linear Programming to maximize profits or minimize costs with a given set of constraints. Because Linear Programming is still a relatively new technique, and companies are confidential about their decision-making process, the real-life examples that can be found publicly are few. Here, a pharmaceutical company application with two different parts illustrates the application of Linear Programming in real cases.

Amit Kumar Jain and other researchers conduct this real-life application. It is about a pharmaceutical company called MASCOT HERBALS PVT. LTD. and ASHWINI HERBAL PHARMACY in India. This example has two parts: maximizing profits and minimizing the transportation costs for medicines.

1. Maximizing Profits

The pharmaceutical is manufacturing two medicines: Type A and Type B. The company can make a profit of twenty-seven dollars for each packet of Type A and a profit of forty-eight dollars for each packet of Type B. There are some rules the firm needs to follow. The number of packets of Type A can not exceed 400, and the number of packets of Type B can not exceed 200. The total number of packets of Type A and Type B can be no more than 450. Packing every 100 packets of Type A takes one hour while packing every 100 packets of Type B takes three hours. The total hours that can be used for packing are no more than ten hours. The ingredient information for each medicine is shown in the table below. The company wants to make maximal profits by deciding the proper amounts of Type A and Type B medicines (Jain et al.).

Ingredients	Amount per packet of Type A (gm)	Amount Per Packet of Type B (gm)	Available(gm)
Bala Mool Extract	12.5	25	10000
Madhu Malini Vasant	2	4	1500
Pushp Dhanva Ras	2	4	1400
Madhu Yashthi Extract	3	6	1500
Kam Doodha Ras	2.5	5	1350
Sanshamani Vati	2.5	5	1200
Amalki Churn	5	10	2800
Arjun ki Chhal Extract	2.5	5	2600
Pipal ki Chhal Extract	2.5	5	1100
Abhrak Bhasm 100 putty	0.5	1	300
Shatawari churn	15	30	8250

Table 5.1 — Information of Ingredients for Pharmaceutical Company Example

Let A denote the number of packets of Type A produced and let B denote the number of packets of Type B produced. The problem can be formulated into the following Linear Programming model.

Maximize $Z = 27A + 48B$

Subject to $12.50A + 25B \leq 10000$

$2A + 4B \leq 1500$

$2A + 4B \leq 1400$

$3A + 6B \leq 1500$

$2.5A + 5B \leq 1350$

$$2.5A + 5B \leq 1200$$

$$5A + 10B \leq 2800$$

$$2.5A + 5B \leq 2600$$

$$2.5A + 5B \leq 1100$$

$$0.5A + B \leq 300$$

$$15A + 30B \leq 8250$$

$$A + B \leq 450$$

$$A + 3B \leq 1000$$

$$A \leq 400$$

$$B \leq 200$$

$$A \geq 0$$

$$B \geq 0$$

Using the computer program, the result is produced below.

```
Whether there is an optimal solution: True

The value of optimal solution: 11760.0

The corresponding values of decision variables: [400.  20.]
```

Figure 5.4 — Results of Maximizing Profits

Based on the results, the company can make a maximum profit of 11760 dollars by producing 400 packets of Type A and 20 packets of Type B.

2. Minimizing Costs

The second part of this pharmaceutical company example is to minimize the transportation costs for the medicines. The firm is delivering cough syrup from two factories P and Q to three different customers X, Y, and Z. The information on the specific cost and quantity of syrup packets are included in the table below. The company wants to minimize the transportation costs by deciding the amount of cough syrup packets produced by each company as well as the amount of cough syrup delivered to different customers (Jain et al.).

Factory	Costs for X per packet	Costs for Y per packet	Costs for Z per packet	Available Packet
P	70	120	100	120
Q	50	140	160	80
Demand	60	90	50	

Table 5.2 — Information of Costs and Demand for Each Customer

Let x_1 denote the number of syrup packets delivered from P to X, x_2 denote the number of syrup packets delivered from P to Y, x_3 denote the number of syrup packets delivered from P to Z, x_4 denote the number of syrup packets delivered from Q to X, x_5 denote the number of syrup packets delivered from Q to Y, x_6 denote the number of syrup packets delivered from Q to Z. Apply Linear Programming model to solve this problem.

Minimize: $Z = 70x_1 + 120x_2 + 100x_3 + 50x_4 + 140x_5 + 160x_6$

Subject to: $x_1 \leq 60$

$x_2 \leq 90$

$x_3 \leq 50$

$x_4 \leq 60$

$x_5 \leq 90$

$x_6 \leq 50$

$x_1 + x_2 + x_3 = 120$

$x_4 + x_5 + x_6 = 80$

$x_1 + x_4 = 60$

$x_2 + x_5 = 90$

$x_3 + x_6 = 50$

$x_1 \geq 0$

$x_2 \geq 0$

$x_3 \geq 0$

$x_4 \geq 0$

$x_5 \geq 0$

$x_6 \geq 0$

The result is shown below.

```
Whether there is an optimal solution: True

The value of optimal solution: 19200.0

The corresponding values of decision variables: [ 0. 70. 50. 60. 20.  0.]
```

Figure 5.5 — Results of Minimizing Costs

According to the results of Linear Programming, the firm will have a minimum cost of 19200 dollars when delivering zero packets from factory P to X, seventy packets from factory P to Y, fifty packets from factory P to Z, sixty packets from factory Q to X, twenty packets from factory Q to Y, and zero packets from factory Q to Z.

Above are examples of utilizing Linear Programming to achieve optimization. Linear Programming has been applied frequently in business planning and decision making. With the advancement of methods of Linear Programming, it is optimistic that more and more applications of Linear Programming can be seen in real-life situations.

CONCLUSION

Humans have unlimited needs and desires that exceed the limited resources existing on Earth. It is a paramount task for economists and mathematicians to achieve the optimization of the allocation of resources, and Linear Programming is one of the techniques that serve to calculate the optimal output.

Linear Programming has three main components: the objective function, decision variables, and constraints. There are two forms of Linear Programming: the General Form and the Standard Form. One can convert the model from the General Form to the Standard Form.

The development of Linear Programming involves three main stages. Soviet Mathematician Leonid Kantorovich first proposed the idea of Linear Programming. He formulated the model of Linear Programming and applied it on a local scale. Unfortunately, he struggled with implementing his idea and gaining recognition for his work. American Mathematician George Dantzig, regarded as the father of Linear Programming, devised an efficient method called the Simplex Method. In fact, Dantzig also gave birth to the name Linear Programming. After Dantzig, Mathematicians such as Leonid Khachiyan and Narendra Karmarkar in recent decades develop methods like the Ellipsoid Method and the Interior-Point Method to improve the technique of Linear Programming.

There are several methods to solve a Linear Programming problem. Each method has its distinct advantages and limitations. Among them, the Graphical Method is intuitive enough for users to solve problems, but it is limited to the number of variables of the problem. The Simplex Method constructs simplex tableaus to efficiently improve the solution step by step. When there are greater than or equal to constraints, the artificial variables are used through the Big-M Method and the Two-Phase Method to solve the problem.

Computer programs that apply Linear Programming help visually display the problem or solve problems with massive data sets. Using the *Matplotlib* module in python, the computer program can apply the Graphical Method to draw the constraints and find out the optimal vertex. On the other hand, applying the *scipy.optimize* module in python allows one to quickly output the solution using the Simplex Method.

Linear Programming has been widely used to optimize profits with linear equations or linear inequalities. The fields that apply Linear Programming include agriculture, transportation, marketing, investment planning, advertising, and so forth. Case studies like minimizing busing costs apply Linear Programming to address the problem. In real-life situations, pharmaceutical companies like MASCOT HERBALS PVT. LTD. and ASHWINI HERBAL PHARMACY in India apply Linear Programming to maximize profits and minimize transportation costs.

ANNOTATED BIBLIOGRAPHY

Aboelmagd, Yasser. "Linear Programming Applications in Construction Sites - ScienceDirect." *ScienceDirect.Com | Science, Health and Medical Journals, Full Text Articles and Books.*, ScienceDirect.com, 28 Nov. 2018, https://www.sciencedirect.com/science/article/pii/S1110016818302126#:~:text=The%20Applications%20of%20Linear%20Programming,(6)%20Traveling%20Salesman%20Problem.

This one covers an introduction of linear programming and a hypothetical application of a Project Management Office when deciding which project the office should bid on with the available resources.

I will include the Project Management part as an example in my application section. This one is a really interesting example with some other features besides normal constraints and objective function.

Ahmadi, Amir. "Linear Programming." *Princeton University, Operations Research and Financial Engineering*, Princeton.edu, https://www.princeton.edu/~aaa/Public/Teaching/ORF363_COS323/F20/ORF363_COS323_F20_Lec11.pdf. Accessed 8 Oct. 2021.

This source is a lecture article delivered by Amir Ahmadi, who is a professor in the Department of Operations Research and Financial Engineering at Princeton University. In this source, Ahmadi discusses the history, the graphical method, the simplex method, and some sophisticated proof.

I mainly cite this source's history part. This source allows me to include a little development of Linear Programming after George Dantzig in my History Section.

Arshman, Hossein. "Linear Optimization." *University of Baltimore Home Page Web Services*, Ubalt.edu, 25 Feb. 1994, http://home.ubalt.edu/ntsbarsh/opre640a/partviii.htm.

This article covers several topics related to Linear Optimization. However, mainly focused on the part related to Linear Programming. The part presents a few hypothetical linear programming problems and introduces two methods: the graphical solution method and the algebraic method.

I use this source to learn about some features of linear programming and how to use graphical methods and algebraic methods. I won't necessarily cite it in my paper, but this article provides me with steps and ideas to employ the graphical method and algebraic method to solve LP problems. Also, I get a few ideas of hypothetical applications of linear programming from the source, which allows me to construct similar examples while describing methods.

Bala, Shujit, et al. "Application of Linear Programming Approach for Determining Optimum Production Cost." *Asian Business Review*, Abc.us.org, 7 June 2020, https://abc.us.org/ojs/index.php/abr/article/view/466/911.

This is an article discussing the application of Linear Programming in business planning to minimize costs. The article contains a real-life example of a bicycle company that tries to minimize the cost.

I will cite the part of the source that discusses modeling the real-life application of Linear Programming in a bicycle company planning. This is valuable to be included in my project.

Boldyrev, Ivan, and Till Duppe. “Programming the USSR: Leonid V. Kantorovich in Context | The British Journal for the History of Science | Cambridge Core.” *Cambridge Core*, Cambridge University Press, 8 Apr. 2020, https://www.cambridge.org/core/journals/british-journal-for-the-history-of-science/article/programming-the-ussr-leonid-v-kantorovich-in-context/4BF0F0D89079DD94AF595EA25A991299.

This is an article about Leonid Kantorovich's whole career in the Soviet Union. It discusses Kantorovich's ideas and situations in which he studied Linear Programming. It also relates Kantorovich with the whole Soviet Mathematicians and Economists at that particular time.

I will cite this source for my history part of Leonid Kantorovich. Some of the information is overlapping and irrelevant to my topic, so I would selectively choose what I will need.

Dantzig, George. *Impact of Linear Programming on Computer Development*. Department of Operations Research, Stanford University, June 1985, https://apps.dtic.mil/dtic/tr/fulltext/u2/a157659.pdf.

This is a first-hand article written by George Dantzig. In the first half of this article, George Dantzig describes his story of coming up with the Simplex Method. In the second half of this article, George Dantzig discusses how the

newly invented computers applied the Simplex Method for calculation at that time.

I mainly focus on the history of the Simplex Method. Some of the historical backgrounds have been roughly mentioned by other articles. However, this article contains some more details and Dantzig's genuine thoughts as a first-hand article.

---. *LINEAR PROGRAMMING*. Duke.edu, https://courses.cs.duke.edu/spring07/cps296.2/papers/LinearProgramming_article.pdf. Accessed 30 Sept. 2021.

Dantzig, who is a paramount figure in the history of linear programming writes this article. In this article, Dantzig describes his experience and thoughts about linear programming. He discusses his experience of working for the Air Force, which led him to develop linear programming. He also discusses things like the origin of the name “linear programming”.

This source actually does not tell many things about Dantzig’s background or knowledge about linear programming, but as a first-hand source written by Dantzig, it is valuable for me to cite information about Dantzig’s experience and ideas in my Capstone project’s part about the history of linear programming.

Gale, David. “Linear Programming and the Simplex Method.” *American Mathematical Society*, Ams.org, Mar. 2007, https://www.ams.org/notices/200703/fea-gale.pdf.

This is a source that describes the Simplex Method in terms of vectors and linear algebra, which requires sophisticated mathematics. It demonstrates how the Simplex Method works in its standard form.

Since I am only focusing on applying the Simplex Method with a tableau that does not involve vectors, I won't cite this source much. This source gives me some extra details on the Simplex Method, which I can incorporate into my essay.

Gass, Saul. “NAE Website - GEORGE B. DANTZIG 1914–2005.” *NAE Website*, 2008, https://www.nae.edu/187662/GEORGE-B-DANTZIG-19142005.

This is an article discussing the biography of George Dantzig. This book has some overlapping information that I have already collected, but it has some more details in terms of accomplishments done by George Dantzig.

I will cite some details from this article. It's overlapping but still has valuable information.

---. *The Life and Times of the Father of Linear Programming*. Informs.org, 1 Aug. 2005, https://pubsonline.informs.org/do/10.1287/orms.2005.04.15/full/.

This article talks about the biography of Dantzig, the father of linear programming. It reveals Dantzig’s personal life, contributions to linear programming, and awards. It describes Dantzig’s ideas about linear programming.

I have three articles about Dantzig, and this article is one of them. Each article contains overlapping information as well as unique information about Dantzig. I will cite some information from this article or combine the information with those from the other two articles to discuss Dantzig’s experience with linear programming as a part of the history of linear programming.

Goemans, Michael. “Linear Programming.” *Massachusetts Institute of Technology*, 17 Mar. 2015, https://math.mit.edu/~goemans/18310S15/lpnotes310.pdf.

This source presents lots of information about linear programming, some of which is about advanced mathematics such as how linear programming is used in matrix form. However, I mainly focus on the first part of this article: the introduction and mathematical formulation of linear programming, including the components of linear programming models and how they are written.

I will use this source in the introduction part of my Capstone project discussing What is Linear Programming. I will cite directly some information from this source, but I mainly will describe them in my own words. This article is useful for me when discussing the forms of linear programming formulations.

---. " Lecture Notes on the Ellipsoid Algorithm." *Massachusetts Institute of Technology*, Math.mit.edu, 4 May 2019, http://www-math.mit.edu/~goemans/18433S09/ellipsoid.pdf.

This source is a discussion of the Ellipsoid Method, which is a method that requires sophisticated mathematics. It contains a brief history of the ellipsoid method and the interior-point method.

Since I won't include the ellipsoid method and the interior-point method in my method section, I would only cite the little discussion of the ellipsoid method and the interior-point method in my history section.

Hunter, John, et al. "Matplotlib.Pyplot.Plot — Matplotlib 2.1.2 Documentation." *Matplotlib: Python Plotting — Matplotlib 3.4.3 Documentation*, Matplotlib.org, https://matplotlib.org/2.1.2/api/_as_gen/matplotlib.pyplot.plot.html. Accessed 19 Oct. 2021.

This is a source published by Matplotlib that discusses detailed attributes and parameters of Matplotlib.pyplot.plot method. It discusses the components of the method.

I cite this source because it gives me more details of graphing using pyplot.plot in my project.

---. "Pyplot Tutorial — Matplotlib 3.4.3 Documentation." *Matplotlib: Python Plotting — Matplotlib 3.4.3 Documentation*, Matplotlib.org, https://matplotlib.org/stable/tutorials/introductory/pyplot.html#sphx-glr-tutorials-introductory-pyplot-py. Accessed 19 Oct. 2021.

This is another Pyplot tutorial published by Matplotlib. This source discusses how to use the pyplot method of Matplotlib, which allows you to draw graphs using Python.

I cite this source because I learned how to apply the Matplotlib.pyplot method in Python to create graphs for my project.

Jain, Amit, et al. "APPLICATION OF LINEAR PROGRAMMING FOR PROFIT MAXIMIZATION OF A PHARMA COMPANY." *Advance Scientific Research*, Jcreview.com, 22 May 2020, https://drive.google.com/file/d/1Eobezn2m1lisD5Y8w-llWKDeGob63etW/view.

This article presents a real example of a Pharma Company in India applying Linear Programming to maximize profits in one case and minimize costs in another case. This article contains a concrete table of data for each case.

I will use both the maximation part as well as the minimization part of this source. It's a powerful real-life example that I should include when discussing the applications of Linear Programming in business decision-making.

Kantarovich, Leonid. *My Journey in Science*. Russian Math. Surveys (1987) 42(2), 1987, pp. 233–70, https://sophie.huiberts.me/files/kantorovich-my-journey-in-science-1987.pdf.

This is a section of a report written by Leonid Kantorovich. This is a first-hand source written by him. He discusses his feelings and situations throughout his career and his theory on Linear Programming.

I will cite this source for my History part discussing Leonid Kantorovich. Some of the information is not necessary to me, so I selectively read and cite.

Levy, Dawn. "George B. Dantzig, Operations Research Professor, Dies at 90." *Stanford University*, Stanford.edu, 25 May 2005, https://news.stanford.edu/news/2005/may25/dantzigobit-052505.html.

This article is from Stanford University after Dantzig passed away in 2005. This article mainly delineates background information about Dantzig. Indeed, this article also talks about linear programming while discussing Dantzig. It presents Dantzig's life path, honors, contributions, and awards.

As I mentioned above, I have three articles about Dantzig, and this is one of them. I will cite information about Dantzig from this article when discussing the history of linear programming in my project. Moreover, some parts of this article are related to the essence of linear programming, which can be cited by me in my introduction.

Lewis, Catherine. *Linear Programming: Theory and Applications*. Whitman.edu, 11 May 2008, https://www.whitman.edu/Documents/Academics/Mathematics/lewis.pdf.

This source is a comprehensive article that covers the introduction, method, and examples of using Linear Programming. However, I will only focus on the Case Study part of this article.

There are several related Case Studies in this source. I am going to use the first Case Study, which is about assigning students to different schools that can minimize the busing cost.

Linear Programming. Ukey.edu, https://www.uky.edu/~dsianita/300/online/LP.pdf. Accessed 30 Sept. 2021.

This article is a supplement material about linear programming. It covers information about LP from introduction to applications. So far, I focus on the parts about the simplex method, artificial variables, Big-M method, and two-phase method of this source. It discusses the steps to use the methods mentioned above.

I will cite some information about those methods from this article, but I will majorly apply the ideas learned from this source while discussing the methods in my project without direct citation. This source has illustrations of the simplex method and Big-M method, which allows me to understand how they work.

"Linear Programming | Applications Of Linear Programming." *Analytics Vidhya*, https://www.facebook.com/AnalyticsVidhya/, 28 Feb. 2017, https://www.analyticsvidhya.com/blog/2017/02/lintroductory-guide-on-linear-programming-explained-in-simple-english/.

This article covers several topics related to linear programming. What I focus on in this article is the introduction of linear programming, which discusses what linear programming is used for and how optimization exists in our world.

I will cite information about components of linear programming and some brief steps from this article for my introduction. Mostly, I get the idea of what linear programming is in general terms from this source.

"Optimization and Root Finding (Scipy.Optimize) — SciPy v1.7.1 Manual." *Numpy and Scipy Documentation — Numpy and Scipy Documentation*, Docs.Scipy.org, https://docs.scipy.org/doc/scipy/reference/optimize.html#module-scipy.optimize. Accessed 19 Oct. 2021.

This is an API reference of the Scipy.optimize module published by the Scipy community. It discusses the attributes and parameters of the Scipy.optimize module.

I anticipate that I won't use this source in my project. I cite this because I am using Scipy.optimize.linprog method and I think I just cite the API reference in case.

Reeb, James, and Scott Leavengood. *Using the Simplex Method to Solve Linear Programming Maximization Problems.* Oregonstate.edu., Oct. 1998, https://catalog.extension.oregonstate.edu/sites/catalog/files/project/pdf/em8720.pdf.

This source contains information about the simplex method. It indicates the overview of the simplex method, how it is compared to the graphical method, and steps to employ the simplex method. It also has some shortcuts using the simplex method as well as a little summary about the simplex method.

I will apply the ideas about the simplex method learned from this source. I won't necessarily cite much from this source, but I will cite a few key components like major steps or shortcuts. Furthermore, this article reviews the graphical method, which provides me with ideas of how to use the graphical method and what its deficiencies are.

Sharma, J. K. "Linear Programming: Applications and Model Formulation." *Operations Research: Theory and Applications*, by J.K. Sharma, Macmillan Publishers India Limited, 2009, pp. 26–55, http://www.nitjsr.ac.in/course_assignment/CA02CA3103%20RMTLPP%20%20Formulation.pdf.

This source is a chapter from the book Operations Research: Theory and Applications. This chapter is about Linear Programming and its applications. It gives an overview of the Linear Programming techniques and examples of using Linear Programming.

I will mainly use the hypothetical examples provided by this group. The examples cover several topics including Production, Marketing, Personnel, Agriculture, and so forth. This will be a helpful source in my Application part.

Sniedovich, Moshe. *Two Phase Method*. Edu.eu, 15 Feb. 2000, https://ifors.ms.unimelb.edu.au/tutorial/simplex/two_phase.html.

This article discusses the scenario of using the two-phase method when there is an artificial variable. It covers steps to construct a two-phase tableau and when and what to do with the phases.

I won't cite direct information from this source, but this source gives me ideas and illustrations of the two-phase method, which I will discuss in the methods section of my project.

Stojiljkovic, Mirko. "Hands-On Linear Programming: Optimization With Python – Real Python." *Python Tutorials – Real Python*, Real Python, 22 June 2020, https://realpython.com/linear-programming-python/.

This is a source discussing how to use Python to apply the simplex method. It gives a brief description of Linear Programming and the Simplex Method, and then it presents several ways of using the Scipy module in python to use the Simplex Method.

I cite the part of the source where it discusses the use of Scipy.optimize to apply the Simplex Method for calculations in Python.

The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel 1975. Nobelprize.org, 14 Oct. 1975, https://www.nobelprize.org/prizes/economic-sciences/1975/press-release/.

This article is a Press Release for the Nobel Prize. It says that the Nobel Prize in Economics of 1975 is given to Kantarovich and Koopmans for their contribution to the theory of the optimum allocation of resources.

I cite a photo of Kantarovich from this source, which will be presented when discussing Kantarovich. Also, the information about the Nobel Prize is significant to mention in my parts about Kantarovich in the section on the history of linear programming.

Towers, Sherry. “Simplex Method | Polymatheia.” *Polymatheia*, Sherrytowers.com, 14 July 2014, http://sherrytowers.com/2014/07/14/simplex-method.

This article is relatively short. It talks about the features and disadvantages of the simplex method without much elaboration.

The key information I will cite from this article is about the disadvantages of the simplex method.

“Two Phase Method, Linear Programming, Minimization Example.” *IGNOU MBA Assignments, MBA Projects, Ignou MCA BCA MBA Assignment*, Universalteacherpublications.com, http://www.universalteacherpublications.com/univ/ebooks/or/Ch3/twophase.htm. Accessed 30 Sept. 2021.

This article gives an illustration of constructing a tableau using the two-phase method. It demonstrates how to use the two-phase method when there is a linear programming problem with artificial variables.

I may cite some direct quotes from this article, but I mostly will apply the procedure I learned from this article with my own example and formulation when discussing the two-phase method in my project.

Vershik, Anatoly. *L.V.Kantorovich and Linear Programming*. Researchgate.net, Aug. 2007, https://www.researchgate.net/publication/1897118_LVKantorovich_and_Linear_Programming/citation/download.

This source is an article written by Anatoly Vershik, who is a student of Kantarovich. This article discusses background information about Kantarovich as well as his theories of optimization. It discusses Kantarovich's experience and his struggle for recognition of his theory related to linear programming in the Soviet Union.

Some information from this article is subjective, so I will only cite certain objective information from the article. This source is valuable for me when discussing Kantarovich's experience and his theory in the section on the history of linear programming.

Wright, Margaret. "THE INTERIOR-POINT REVOLUTION IN OPTIMIZATION: HISTORY, RECENT DEVELOPMENTS, AND LASTING CONSEQUENCES." *American Mathematical Society*, Ams.org, 21 Sept. 2004, https://www.ams.org/journals/bull/2005-42-01/S0273-0979-04-01040-7/S0273-0979-04-01040-7.pdf.

This is an article that discusses the Interior-Point method. It contains some history and how the method works. This article contains sophisticated mathematics and case studies.

Again, since I don't include the Interior-Point method in my Method Section, I only cite this history part of the Interior-Point method for my History Section.