Analysis of practical applications using the linear programming model

Xinyan Zhang

The University of Manchester, Manchester, United Kingdom

xinyan.zhang-4@student.manchester.ac.uk

Abstract. A linear programming model is a mathematical optimization technique used to find the best possible outcome in a mathematical model with linear relationships under a set of constraints. This technique has various applications in engineering, management, economic analysis, and military operations. The linear programming model is the most popular technique for determining the best course of action, and it is essential for decision-making and resource allocation in modern society. This research paper will introduce the standard form of the linear programming model and explore the differences and solutions between three linear programming models: 0-1 integer programming, mixed integer programming, and multi-objective programming. In the case of introducing the application and optimization effect of different models in real life, list research and compare the different applications and research scenarios of the three models to derive the strengths, weaknesses, and characteristics of the different models to achieve the effect of providing more targeted models for different real-life cases.

Keywords: Linear programming, 0-1 integer programming, multi-objective programming, mixed integer programming.

1. Introduction

Following Gomory's 1958 proposal of the cutting plane method, integer programming developed into a separate field [1]. Many techniques have been created during the past 30 years to address various issues. Integer programming can be categorized as 0-1 integer programming (0-1 IP). Integer programming issues with only 0 or 1 as the only possible values are known as 0-1 integer programming problems. The 0-1 variable can be used to statistically explain the relationships between discrete variables that are logical, sequential, and mutually exclusive. Any integer programming that uses bounded variables can be processed as 0-1 programming. Numerous industries, including telecommunications, finance, manufacturing, and logistics, can leverage 0-1 integer programming. Researchers have created hybrid approaches integrating 0-1 integer programming with other optimization techniques, including heuristics, meta-heuristics (genetic algorithms), relaxation, and recurrent methods to handle large-scale complex 0-1 integer programming problems. Creating effective cutting planes, branch-and-cut algorithms, parallel computing techniques, and investigation of applications in newly developing fields like data science and machine learning are all part of current research on 0-1 integer programming.

Mixed integer programming (MIP) belongs to the same category of integer programming as 0-1 integer programming. In contrast to 0-1 integer programming, if some of the decision variables are not discrete, the problem is called a mixed-integer programming problem [2]. Because it offers a flexible

DOI: 10.54254/2753-8818/25/20240979

and effective method for resolving substantial, complicated issues and has been employed in various fields, including finance, energy, telecommunications, and bioinformatics, mixed-integer programming is frequently used in systems analysis and optimization. Includes the creation of sophisticated cutting plane techniques, hybrid algorithms that combine Mixed-integer programming with other optimization techniques, and efficient parallel computing strategies for handling large-scale Mixed-integer programming issues.

Multi-objective programming (MOP) problems are mathematical planning issues considering various objective functions [3]. In the middle of the 20th century, multi-objective programming began to develop. Mathematical decision-making and optimization are the roots of multi-objective programming. It entails resolving issues that require simultaneous optimization of several incompatible goals. Genetic algorithms, simulated annealing, particle swarm optimization, and other techniques and algorithms have all been created to handle multi-objective optimization issues. Numerous disciplines use multi-objective programming, including engineering, economics, finance, environmental management, and logistics. For instance, it can be used in engineering to develop products that consider elements like cost, performance, and safety. Multi-objective programming is a crucial tool for supporting decisions. It enables decision-makers to examine several solutions and decide with knowledge by their preferences and priorities [4]. Recent developments in multi-objective programming include incorporating machine learning methodologies, creating hybrid algorithms, and using multi-objective programming to solve challenging real-world issues like resource management and healthcare optimization.

This paper will start by introducing the background and leading solutions of the three linear programming models and then analyze the background, purpose, and results of three classical linear programming examples. A case study will be used to compare the three different linear programming and research objectives. It will conclude with research adjustments and future perspectives.

2. Prerequisite knowledge

Develop a mathematical model for linear programming:

1. list the constraints and objective function
2. Draw the feasible domain represented by the constraints.
3. Find the optimal solution and the optimal value of the objective function in the feasible domain.

2.1. Standard linear programming and solution method

The standard form consists of three parts: 1. a function to be maximized. 2. problem constraints, 3. Non-negative variables [5]. generally in the form of:

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 \le b_1 \\ a_{21}x_1 + a_{22}x_2 \le b_2 \\ a_{31}x_1 + a_{32}x_2 \le b_3 \end{gathered} \tag{1}$$

Other problems, such as minimization problems, problems with different forms of constraints, and problems with negative variables, can be rewritten as standard types of their equivalent problems.

Therefore the standard form looks like [6]: Maximize $c_1x_1 + c_2x_2 + \cdots c_nx_n$, subject to

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + \cdots a_{1n}x_n \le b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots a_{1n}x_n \le b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots a_{mn}x_n \le b_m \\ x_1, x_2, \cdots, x_n \ge 0 \end{gathered} \tag{2}$$

It can also rewrite as matrix form:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}, \quad (3)$$
$$b = (b_1, b_2, \cdots, b_n)^T,$$
$$c = (c_1, c_2, \cdots, c_n)^T,$$
$$x = (x_1, x_2, \cdots, x_n)^T.$$

The simplex approach is typically used to solve problems using classic linear programming [7]. George Dantzig's Simplex Form Algorithm was first conceived as a method for gradually descending to a vertex on a convex polyhedron [8]. As long as there is an optimal solution for this linear programming, the optimal solution will be found after a finite number of steps of iteration. The basic idea of the simplex method is to start from a vertex of the feasible set of linear programming and seek the next vertex in the direction that makes the value of the objective function decrease. The three concerns that follow must be resolved to use an iterative approach to discover the best linear programming solution: Three factors can cause the value of the objective function to decrease more than others: 1. The optimal solution discrimination criterion, which is the iteration termination criterion. 2. The base-switching operation, which is the method of iterating from one base-feasible solution to another base-feasible solution, and 3. The selection of the base-entry columns, which is the selection of appropriate columns to carry out the base-change operation, can result in a significant decrease in the value of the objective function [9].

The simplex method's general solution steps can be summed up as follows [10]:

(1) Find the initial basic feasible solution to the linear programming problem and express the system of constraint equations as a canonical system of equations.

(2) The problem has no solution if the basic feasible solution is not possible, i.e., the constraints are incompatible.

(3) If the basic feasible solution is present, start with it and use the optimality condition and feasibility condition to introduce non-basic variables to replace a particular basic variable to find a different basic feasible solution with a higher objective function value.

(4) Repeat steps 3 and 2 until the relevant test number satisfies the criterion for optimality (at this point, the objective function value cannot be increased). This is the best way to solve the issue.

(5) The iteration is stopped if the objective function value of the issue is discovered to be unbounded during the iteration procedure.

The amount of constraints has a major impact on how many iterations of the simplex approach are needed to solve a linear programming problem. Today, the simplex method is used on computers to handle issues involving general linear programming.

2.2. 0-1 integer programming and solution method

In 0-1 integer programming, the only possible values for the decision variables are 0 and 1. When making the choice, 1 equals "yes" and 0 equals "no." In most cases, the branch and bound method of hidden enumeration is used to solve 0-1 integer programming.

The main concept behind the branch and bound method is to compute a target lower term (for minimum value problems) or a target upper term (for maximum value problems) for the set of solutions within each subset, which is referred to as delimitation. This process is repeated until the entire space of feasible solutions has been divided into smaller and smaller subsets, called branches. Pruning is the process of stopping further branching after each subset whose boundaries go beyond the given objective value of the feasible set, allowing for the disregard of numerous subsets [11].

The branch-and-bound method's primary steps are as follows (using minimization as an example):

1. If minimization is the goal of the problem, set as the value of the best solution.

2. In accordance with the Branching rule, a node is chosen from the unfathomed nodes (local solutions) and split into numerous new nodes in the node's subsequent stratum.

3. Determine each freshly branched-out node's Lower bound.

4. Each node is subject to the fathom condition; if its lower bound value exceeds or is equal to the Z value, the node is no longer taken into consideration. This node contains a feasible solution that has the lowest lower bound value. If this condition is true, the feasible solution is compared to the Z value, and if the former is smaller, the Z value is modified to reflect the feasible solution's value [11].

5. If there are any unfathomed nodes, move on to step 2. If there are no more unfathomed nodes, the process ends and the best solution is found.

The branch and bound method has the advantage that finding the best solution can be done more quickly and with fewer subproblems to check. However, a large number of sub-node boundaries and accompanying consumption matrices must be stored. It uses up a lot of memory [11].

2.3. Mixed integer programming and solution method

Partial decision variables with integer values are referred to as "mixed integer programming" problems. This type of problem can be solved using the branch and bound method and the simplex method mentioned above, as well as the memory cut in the plane, cut method.

The primary procedures in memory cut are:

1. relaxation of the overall variables
2. Use the simplex approach to solve the linear programming problem resulting from the relaxation of the integer programming problem.
3. Cut the solution space while ensuring that the feasible solution is not cut off and that the solution from the preceding linear programming is not present in the solution space after cutting.
4. After cutting, keep using linear relaxation and keep going through the stages above until you reach the definitive answer.

It is important to keep in mind that repeating the aforementioned processes will produce many distinct ideal solutions. If the intersection of the viable half-spaces of the constraint space is the empty set, then the problem lacks a workable solution [12].

2.4. Multi-objective programming and solution method

Multi-objective programming involves solving several objectives simultaneously. Since these objectives mostly interact with each other, when one objective is optimized, the other ones to be optimized may worsen. In contrast to the other planning mentioned above, multi-objective planning cannot find an optimal solution; the multi-objective planning generation goal is to find the best state within a certain range.

The weighting sum method, the priority method, and the ideal point method are the three basic approaches to solving multi-objective planning problems.

2.4.1. The weighting sum method. Since there are different preferences for different objectives, it is possible to add weights to each objective function in multi-objective planning, thus transforming multi-objective planning into single-objective planning. This method is suitable for real-world environments where expert scoring is available.This is expressed as [13]:

$$\min Z = \sum_{i=1}^{n} w_i f_i, \tag{4}$$

$$\text{subject to: } \begin{cases} lb \le x \le ub \\ Aeq * x = beq \\ Ax \le b \\ \sum_{i=1}^{n} w_i = 1 \end{cases} \tag{5}$$

2.4.2. The priority method. The main idea of the priority method is to categorize the importance of the objectives into different levels, first seek the optimal value of the high-priority objective function, then seek the low-priority objective function while making sure that the high-priority objective receives at least the best quality.

Take an example of double-goal programming:

DOI: 10.54254/2753-8818/25/20240979

$$\min Z_1 = f_1(x)$$
$$\min Z_2 = f_2(x)$$
$$\text{subject to} \begin{cases} lb \leq x \leq ub \\ Aeq * x = beq \\ Ax \leq b \end{cases} \tag{6}$$

Assuming that the priority of $f_1(x)$ is higher than the priority of $f_2(x)$, then first find the optimal solution of $f_1(x)$ the solution Z_1. Subsequently solve for the optimal value of $f_2(x)$ while ensuring that $f_1(x)$ is optimal:

$$\min Z_2 = f_2(x)$$
$$\text{subject to} \begin{cases} lb \leq x \leq ub \\ Aeq * x = beq \\ Ax \leq b \\ f_1(x) = Z_1 \end{cases} \tag{7}$$

By analogy, all multi-objective planning models with more than 2 objectives can be solved according to the above ideas.

2.4.3. The ideal point method. By solving n single objective problems: $\min Z_i(x)$, Suppose that the single optimal solution to each single-objective problem is Z_i,then $Z^* = (Z_1, Z_2, \ldots, Z_n)$ can be called a single ideal point in the value domain, which corresponds to the single optimal solution of each single-objective problem, and if the multi-objective problem can reach this point, then the optimal solution can be obtained [14].

However, since it is difficult to reach the optimal solution among multi-objective optimization, we can only find the closest Z to Z^* as an approximate solution within some range of feasible points. The most direct way is to minimize the distance of each planning problem from its corresponding optimal solution, the equation looks like:

$$\min Z = \sqrt{[f_1(x) - Z_1]^2 + [f_2(x) - Z_2]^2 + \cdots [f_n(x) - Z_n]^2}$$
$$\text{subject to} \begin{cases} lb \leq x \leq ub \\ Aeq * x = beq \\ Ax \leq b \end{cases} \tag{8}$$

3. Application analysis

3.1. 0-1 integer programming application: a university timetabling problem

The university timetable problem is a typical instance of 0-1 integer programming being used to establish a workable schedule for classes, professors, and students while satisfying a variety of constraints and objectives [15].

3.1.1. Consideration of factors. 1. Course scheduling: the curriculum includes both core and elective courses that need to be assigned to specific slots and rooms to ensure that courses with conflicting schedules are not assigned to the same slot. 2. Professor Availability: Courses are scheduled according to the professors' specialisations and preferences [15]. 3. Time Allocation: Minimise overlap between courses and free time between each session [15]. 4. Classroom consistency: all class periods for a course should be held in the same classroom. The Usage of 0-1 Integer Programming in the Problem of Developing a University Timetable.

Decision Variables: 0-1 variables are used to indicate whether a particular class or course is assigned to a particular time and room. If a class is assigned to a specific time slot and room, then the variable has a value of 1; otherwise, it has a value of 0.

DOI: 10.54254/2753-8818/25/20240979

Objective function: The goal is to minimise conflicts, room usage or teacher workload [15]. This can be expressed as an objective function to be minimised or maximised.

Constraints: the constraints of course scheduling, classroom consistency, professor availability, and time allocation must be met.

3.1.2. Result. Scheduling using 0-1 integer programming optimizes the use of resources, such as classrooms, teachers, and time, while reducing expenses. Additionally, scheduling issues between students and teachers are decreased while satisfaction among both groups is raised [16].

3.2. Mixed Integer Programming Application: Production Line Optimisation

Mixed integer programming is frequently used in operations and manufacturing management for production line optimization. Its goal is to boost revenue while boosting productivity and cutting expenses [16].

3.2.1. Consideration of factors. To maximize productivity, cost, and resource utilization in a production line, a number of factors must be taken into account. These elements consist of: 1. Production planning: To achieve a balanced production line rate and a balanced distribution of work, determine the sequence and timing of production operations for each product and build a production plan appropriately. 2. Resource allocation: Effectively distribute scarce resources, including as labor, materials, and machines, across various production activities while taking into account issues like the space needed for equipment, the simplicity of moving semi-finished goods, and the competing uses of tools. 3. Manage inventory levels to strike a balance between inventory expenses and the risk of stock-outs and production lags.

3.2.2. The usage of mixed integer programming in the problem of Production Line Optimization. Decision variables: Define decision variables, some of which may be 0 or 1, to indicate choices such as whether to produce a particular product in a particular time period or whether to allocate resources to a particular task. Continuous variables may indicate the number of products to be produced as well as the sequence of production processes.

Objective function: Develop an objective function for minimization or maximization. This function can be a measure of production cost, production profit, production time, etc.

Constraints: These include consideration of factors as well as capacity constraints, resource availability, demand requirements, and maintenance schedules. These constraints ensure that the production plan meets the actual constraints and requirements.

3.2.3. Result. By optimizing the production line, a mixed integer program can effectively improve the efficiency of the production line, the balance rate of the production line, and the production line process, thereby increasing the production of products. At the same time, it can improve the profit and increase the order quantity to increase the competitiveness in the market.

3.3. Multi-objective programming application: water resource allocation

In order to establish thorough evaluation statistics of water resource allocation options and to choose the most reasonable water resource allocation option, multi-objective programming is a significant and useful application approach in the field of water resource allocation [17].

3.3.1. Consideration of factors. 1. Water resource development: maximizing environmental sustainability through the protection of water habitats, particularly riverine wetlands, and seasonal influences on water resources [17]. 2. water demand: satisfy a range of demands, including environmental, industrial, agricultural, and domestic ones. Measures to lessen the effects of seasonal flooding and droughts on infrastructure, agriculture, and communities [17]. 3. Existing projects: Projects relating to water supply, water conservation, wastewater reuse, and the possibility of water supply in reservoirs and plants.

3.3.2. The usage of multi-objective programming in the problem of water resource allocation. Decision variables: water resources' added value, global water supply, per-person water supply, water supply assurance, water shortage, and ecological satisfaction [17].

Objective function: Economic benefits and total investment; water supply; water Supply guarantee and scarcity; environmental impact of public water.

Constraints: Including consideration of factors and local development and living standards.

Target weight: Six indicators are given the target weights: value added of water resources, total water supply, water supply per person, water supply that is assured, water shortage, and ecological satisfaction [17].

3.3.3. Result. Improved decision-making that takes into account many goals at once improves sustainability, boosts resistance to droughts and floods, and reduces costs by better allocating ecological and environmental needs.

3.4. Comparative analysis

This section compares and contrasts 0-1 IP, MIP, and MOP in terms of application objects, research scenarios, method characteristics, benefits, and drawbacks.

Table 1. Comparison of 0-1 integer programming, mixed integer programming, objective multi-objective programming.

	0-1 integer programming	mixed integer programming	multi-objective programming
Application object	Used for portfolio decisions, and the decision variable is 0 or 1.	Some variables must be 0 or 1; others can take continuous values.	Tackling optimizing issues with many competing objectives requires several factors to be considered at once.
Research scenario	-Scheduling design: selecting or excluding courses or schedules for a specific time. -Backpack problem: selecting or rejecting items to maximize a particular value or to meet a weight limit. -Item selection: deciding which items to take on or exclude to optimize a set of criteria.	-Production Planning: Optimise production planning through decisions such as machine setup (discrete) and productivity continuum (continuous). -Supply Chain Optimisation: Balancing inventory levels, labor, and machine allocation(discrete) with transport costs, productivity, and profitability(continuous).	-Environmental planning: balancing objectives such as pollution reduction and minimize cost in development plans. -Portfolio optimization: maximizing profit and minimizing risk. -Product design: Balancing cost, profit, and performance in engineering design.
Methods	Branch and bound method, cutting plane algorithms.	Branch and bound, Linear programming relaxation, cutting plane algorithms.	Weighted sum method, idea point method, priority method.
Advantages	The constraints and objectives are incorporated into a unified framework, and exact optimal	Provides flexibility and breadth in real-world problems with precise solutions.	Various problem types can be handled and help decision makers understand the inherent trade-offs to

Table 1. (continued).

	solutions can be found.		make informed choices by weighting different objectives.
Disadvantages	It is limited to binary decisions which cannot accurately represent all situations in the real world. Problem solving can become lengthy for large problems.	The effectiveness of the formulation depends on the quality of the model formulation. Poorly formulated models can lead to sub-optimal or infeasible solutions. Due to the complexity of the variables, more sophisticated equipment may be required for the calculations.	Assigning weights or priorities to objectives is often subjective, and the choice of objective weights can significantly impact the results. Significant computational resources are required for large-scale or highly non-linear problems.

3.5. Discussion
This chapter uses classic examples of class scheduling, production line optimization, and water balancing to analyze the application of different linear programming models in real-life cases. This chapter briefly describes the applications, research scenarios, methodological features, advantages, and disadvantages of the three different linear programming methods to provide more relevant and effective modeling for different real-world applications. There are more than three types of linear programming and many different application areas. However, one or more linear programming methods will be more efficient and accurate than the others in arriving at the optimal solution in different real-world scenarios.

4. Conclusions
In many disciplines, including operations research, economics, engineering, and management, linear programming is crucial for tackling various optimization issues. By selecting the suitable model for the job, linear programming can be more effective. Similar to integer programming, more and more optimization techniques will be used with linear programming in the future. Technical difficulties are also included in the development of linear programming. More significant and complicated problems can be addressed faster by using faster and more efficient linear programming solvers that are being developed thanks to algorithms, hardware, and parallel computing improvements. In conclusion, linear programming is still an essential and adaptable method for optimization in various fields. Its bright prospects for the future are anticipated to fuel its ongoing significance and expansion as computer power, integration with other optimization techniques, and applications in data-driven decision-making all develop.

References
[1] Gomory R E 1958 Bulletin of the American Mathematical Society 64 275278
[2] Kantor I, Robineau J, Bütün H, Maréchal F 2020 Frontiers in Energy Research 8 49
[3] Alves M J, Clímaco J 2001 Encyclopedia of Optimization Springer Boston MA
[4] Karimi N, Feylizadeh M R, Govindan K, Bagherpour M 2022 Expert systems with applications 196 116663
[5] Matousek J and Gartner B 2007 Springer-Verlag, Berlin Heidelberg 1-6
[6] Omar Antolín C 2022 Standard Form for Linear Programs
[7] Guoqing H 2020 Optimization cbe cornell edu
[8] Dantzig G B 1987 Trchnical Report Sol 87-5
[9] Guogen X, Housui Z, Zhiyong H 2018 OptimizationBeijing University of Aeronautics and Astronautics Press 73

DOI: 10.54254/2753-8818/25/20240979

[10] Fanghuai W, Ning X 2011Haitian Publishing House 130
[11] Subham D, Branch, Bound A 2020 Baeldung on Computer Science. 23 21
[12] István M 2023 Mixed Integer Linear Programming: Formal Definition and Solution Space.
[13] Xiujuan L, Zhongke S 2004 J. of Systems Engineering and Electronics 15 2 142-146
[14] Weixiong C, Yifei R, Jian L, Lixin Y 2022 J. of Beijing Jiaotong University 46 04 23-30 2022
[15] Bakir M, Akif, and Aksop C 2018Hacettepe Journal of Mathematics and Statistics 37 1 41-55
[16] Wenqi L, Wei L 2018 Fujian Computer 34 12 46-47
[17] Cuizhu Z, Zhitao L, Qiuping Y, Yin Z, Yaming H, Jianyu L 2022 Mechanical and Electrical Engineering Technology 51 05 198-203