Proceedings of the 2023 International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/11/20230414

Application of linear programming in two-player zero-sum games

Xinyu Wang

School of Arts and Sciences, Boston University, Boston, MA, US

wxy1113@bu.edu

Abstract. Linear programming is a useful optimization tool that has been utilized since the 1940s. It has been studied for a long time, and it can be applied to many different areas and fields. One of these applications is the two-player zero-sum game. In this game, each player has a set of possible strategies that is finite. While one player tries to maximize the net payoff, the other player tries to minimize this payoff. Since the problem involves optimizations, linear programming can be involved. This paper analyzes a famous and classic example of the game – the Colonel Blotto Game. This game is a military planning problem. Indeed, Linear programming was first developed to solve complex military and planning problems during wartime. This paper utilizes a payoff matrix and some mathematical models to build the mathematical problem of the Colonel Blotto Game. Then the results can be obtained by converting the mathematical problem into a Python Linprog problem. The paper finds the best strategy for Colonel Blotto to maximize the net payoff and how eliminating certain possible strategies will or will not affect the results. Moreover, the paper provides an understanding of the combination of the two-player zero-sum game and linear programming.

Keywords: linear optimization, linear constraints, two-player zero-sum games, the Colonel Blotto Game.

1. Introduction

Linear programming, or linear optimization, is the procedure of maximizing or minimizing objective functions that are subject to constraints. Both objective functions and constraints have to be linear. While some constraints are equalities, others can be inequalities. Linear programming has a short but rich history. It was first developed in the 1940s to solve planning problems in wartime operations that are too complicated. It was discovered three times independently, and each time differently because of the circumstances. The three discoverers were Leonid Vitalyevich Kantorovich, Tjalling Charles Koopmans, and George Bernard Dantzig. Then, in 1947, Dantzig invented the simplex method. In the same year, John von Neumann established the concept of duality [1]. When it comes to problems of optimization, linear programming remains the most effective and widely applied method.

The simplex method is frequently used when solving linear programming problems. It involves the use of slack variables and tableaus. By adding slack variables, inequality constraints are transformed into equality constraints. Moreover, Simplex tableaus are utilized when performing row operations on the linear programming model and checking optimality. The simple method has always been an extremely efficient computational tool [2]. This method is significant because it is helpful and valuable

DOI: 10.54254/2753-8818/11/20230414

in many linear programming applications. That is, many problems in the real world can be formulated as mathematical linear programming problems and solved by simplex methods [3]. The linear programming process became more intuitive with the introduction of the simplex method and the tableaus.

The dual problem of a given linear program can be derived by turning variables into constraints, turning constraints into variables, and inversing the objective function (maximum becomes minimum and vice versa). Linear programming duality is especially important when computing or solving economic problems. Additionally, suggested solutions to linear programming models are straightforward to check for feasibility but not so obvious to check for optimality. On the other hand, dual solutions provide quick checks for optimality when yielding objective values that are equal to the ones from the original model [4]. The gap in approaches of continuous and discrete methods is also reinforced by the development of primal-dual strategies. This helps with solving different kinds of nonconvex optimization problems. One of the properties of duality, however, is that the solution of the dual problem provides a relatively lower bound on the minimum value compared with that of the primal problem [5]. Still, dual programs are more helpful and coherent in many situations.

Linear programming is a useful tool that can be applied to many different fields. For example, linear programming can be utilized when finding the optimal use of raw materials. The decision makings usually based on the use of limited resources, bringing the application of linear program models as powerful tools. Linear programming and the simplex algorithm can be utilized to allocate resources and raw materials to competing variables for the purpose of maximizing profit [6]. Moreover, linear programming is an efficient tool when computing solutions to transportation problems. In these cases, total cost is minimized subject to different cost functions. The method can also be applied to more general problems, such as partial optimal transportation and barycentre problems [7]. Linear programming is also a quantitative method that helps farmers plan and make decisions. Some of the decisions involve the use of water supply, labor, and fertilizers. The reason that linear programming can be a particularly useful tool in the agriculture sector is because of its efficiency since farmers want to minimize trails and errors [8]. Besides these examples, there are many other problems and fields that utilize linear programming.

The two-player zero-sum game, the basic model in game theory, is an example of linear programming that is given emphasis in this paper. In such a game, both players have finite sets of possible strategies or partitions. Each pair of opposing strategies results in a payoff. While one player tries to maximize the gaining, the other player tries to minimize the loss. However, whatever one player wins, the other loses, making it a zero-sum game. Indeed, this game theory is also utilized in real life. One common example is using game theory techniques to help military commanders estimate different situations. In fact, decision-making in militaries was highly similar to solving zero-sum games [9]. The famous Colonel Blotto Game is analyzed in this paper. Game theory is highly based on mathematics. The zero-sum game involves choosing optimal strategies under certain constraints. Therefore, linear programming is helpful since it is an optimization tool [10]. Linear programming problems and the zero-sum game are claimed to be equivalent since any linear programming problem can be converted into a zero-sum problem and vice-versa. However, the conversion of a linear programming problem from a zero-sum game is simple and direct, while the conversion of a zero-sum game from a linear programming problem can be more complicated [11].

In conclusion, linear programming is a mathematical model utilized in many different fields to solve optimization problems. In this paper, the two-player zero-sum game is particularly focused and analyzed.

2. Methods

2.1. Game description

The Colonel Blotto Game can be summarized as the following: Colonel Blotto is getting ready for a battle with an opponent. Each of them commands five troops, and there are three mountain passes. The rule is that whoever assigns more troops to a mountain can occupy the mountain pass. If the numbers of

DOI: 10.54254/2753-8818/11/20230414

troops are the same, a draw occurs. In the end, the player that occupies more passes wins. A draw will be made when Colonel Blotto and the opponent are holding equal numbers of mountain passes.

Since the Colonel Blotto Game is an example of a finite two-player zero-sum game, each Colonel Blotto and the opponent has a set of possible strategies that is finite. For instance, the strategy (0, 1, 4) indicates that one pass is not going to be attacked at all, one pass is going to be attacked by 1 troop, and one pass is going to be attacked by 4 troops. Moreover, each of the 3! = 6 possible assignments of groups can occur in equal chances. Therefore, the strategies that Colonel Blotto and the opponent would pick determine the net payoff and the winning probabilities for both of them, and both Colonel Blotto and the opponents would try to maximize or minimize these probabilities in their direction to win the battle.

In this paper, the net payoff is defined as the winning probability of Colonel Blotto, subtracting the winning probability of the opponent. In other words, whatever Colonel Blotto wins, the opponent loses. All the payoffs can be put together to create a payoff matrix shown in Table 1. The rows correspond to the different strategies of Colonel Blotto, while the columns correspond to the different strategies of the opponent.

Table 1. Payoff matrix for Colonel Blotto Game.

	(0,0,5)	(0,1,4)	(0,2,3)	(1,1,3)	(1,2,2)
(0,0,5)	0	-1/3	-1/3	-1	-1
(0,1,4)	1/3	0	0	-1/3	-2/3
(0,2,3)	1/3	0	0	0	1/3
(1,1,3)	1	1/3	0	0	-1/3
(1,2,2)	1	2/3	-1/3	1/3	0

2.2. Mathematical models

When analyzing the payoff matrix in Table 1, Colonel Blotto tries to find the minimum value in each row and tries to choose the row in which the minimum value is the largest possible. On the other hand, the opponent tries to find the maximum value in each column and tries to choose the column in which the maximum value is the smallest possible. If Colonel Blotto plays strategy x and the opponent plays strategy y, then the net payoff to Colonel Blotto is $X^T M_y$.

For every strategy x of Colonel Blotto, the opponent will select the best response $\beta(x) = minyX^T M_y$. For every strategy y of the opponent. Colonel Blotto will select the best response $\alpha(x) = maxxX^T M_y$.

When Colonel Blotto selects a strategy, Colonel Blotto assumes that the opponent will select the best response. Therefore, Colonel Blotto seeks x^* so that $\beta(x^*) = maxx\, \beta(x) = maxx\; miny\; X^T M_y$. This is Colonel Blotto's worst-case optimal.

When the opponent selects a strategy, the opponent assumes that Colonel Blotto will select the best response, so the opponent seeks y^* so that $\alpha(y^*) = miny\; \alpha(y) = miny\; maxx\; X^T M_y$. This is the opponent's worst-case optimal.

3. Results and discussion

3.1. Mathematical linear programming problem

Almost every two-player zero-sum game can be transformed into a mathematical linear programming problem, which makes the process of solving the game more obvious and intuitive. Utilizing the payoff matrix and the models from part two, this paper obtains the mathematical linear programming problem that follows:

Maximize x_0, subject to:

DOI: 10.54254/2753-8818/11/20230414

$$\begin{cases} -x_0 + 0x_1 + \frac{1}{3}x_2 + \frac{1}{3}x_3 + x_4 + x_5 \geq 0 \\ -x_0 - \frac{1}{3}x_1 + 0x_2 + 0x_3 + \frac{1}{3}x_4 + \frac{2}{3}x_5 \geq 0 \\ -x_0 - \frac{1}{3}x_1 + 0x_2 + 0x_3 + 0x_4 - \frac{1}{3}x_5 \geq 0 \\ -x_0 - x_1 - \frac{1}{3}x_2 + 0x_3 + 0x_4 + \frac{1}{3}x_5 \geq 0 \\ -x_0 - x_1 - \frac{2}{3}x_2 + \frac{1}{3}x_3 - \frac{1}{3}x_4 + 0x_5 \geq 0 \\ 0x_0 + x_1 + x_2 + x_3 + x_4 + x_5 = 1 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \end{cases} \tag{1}$$

The function x_0 that is being maximized is the objection function, which represents the total payoff of Colonel Blotto in the example. Equations 1-5 in Constraints (1) are obtained by converting the payoff matrix into the mathematical linear inequality constraints. Equation 6 says that Colonel Blotto only gets to pick one strategy. Equation 7 shows that all variables cannot be negative.

3.2. Calculations

Python is a programming language that can be a useful tool to solve mathematical linear programming problems. The Python Linprog function call in Python is used to optimize an objective function under certain constraints. The Python Linprog call for the Colonel Blotto game can be summarized in Table 2.

Table 2. Algorithm of the Python Linprog function call.

Algorithm Python Linprog Function Call
Step 1. Create a 1-D list containing the coefficients of the linear objective function that is being minimized. Since the objective function needs to be maximized, negating the objective function converts it into a minimization problem.
Step 2. Create the inequality constraint matrix by putting all the coefficients of the linear inequality constraints into a matrix. Since the codes have to follow the model $Ax > b$, the linear inequality constraints should be negated, and all the numbers in the matrix change signs.
Step 3. Create the vector containing the inequality constraints, and each element represents an upper bound on the corresponding value of A.
Step 4. Create the equality constraint matrix by putting all the coefficients of the linear equality constraints into a matrix.
Step 5. Create the vector containing the equality constraints, and each element of A_eq must be equal to the corresponding element of b_eq.
Step 6. Define the maximum and minimum values of the decision variables.
Step 7. Put all the information together and use the simplex method to solve the linear programming problem.

After running the Python Linprog function codes, Python will provide the solutions to the mathematical linear programming problem. Obtaining a 0.0 for fun (objective functions) means that if Colonel Blotto successfully picks his optimal strategy, the net payoff would be zero. The array of x gives us the strategy that makes Colonel Blotto maximize his payoff. One thing to notice is that both strategy three and four gives the number 0.5. This means that Colonel Blotto's worst-case optimal strategy is not unique. The solution given by Python Linprog, x = (0,0,0.5,0.5,0), has the same value for the objective function as the solution x = (0,0,1,0,0). The objective function is the expected payoff. In the case x = (0,0,0.5,0.5,0), Colonel Blotto is playing position 3 fifty percent of the time and position 4 fifty percent of the time.

3.3. Alternating Colonel Blotto's strategies

In the previous section, the results obtained by running the Python Linprog function show that Colonel Blotto will play his third strategy fifty percent of the time and his fourth strategy fifty percent of the time. There is one way to prove that Colonel Blotto would choose between these two strategies. If any strategy besides strategy 3 and strategy 4 is eliminated, the results should still be (0,0,0.5,0.5,0). This paper will consider the case of removing the fifth strategy of Colonel Blotto. Table 3 is the new payoff matrix.

Table 3. Payoff matrix for Colonel Blotto Game after removing Colonel Blotto's fifth strategy.

	(0,0,5)	(0,1,4)	(0,2,3)	(1,1,3)	(1,2,2)
(0,0,5)	0	-1/3	-1/3	-1	-1
(0,1,4)	1/3	0	0	-1/3	-2/3
(0,2,3)	1/3	0	0	0	1/3
(1,1,3)	1	1/3	0	0	-1/3

Similarly, the new mathematical representation of the problem is as follows:

Maximize x_0, subject to:

$$\begin{cases} -x_0 + 0x_1 + \frac{1}{3}x_2 + \frac{1}{3}x_3 + x_4 \geq 0 \\ -x_0 - \frac{1}{3}x_1 + 0x_2 + 0x_3 + \frac{1}{3}x_4 \geq 0 \\ -x_0 - \frac{1}{3}x_1 + 0x_2 + 0x_3 + 0x_4 \geq 0 \\ -x_0 - x_1 - \frac{1}{3}x_2 + 0x_3 + 0x_4 \geq 0 \\ -x_0 - x_1 - \frac{2}{3}x_2 + \frac{1}{3}x_3 - \frac{1}{3}x_4 \geq 0 \\ 0x_0 + x_1 + x_2 + x_3 + x_4 = 1 \\ x_1, x_2, x_3, x_4 \geq 0 \end{cases} \quad (2)$$

Utilizing Algorithm 1, which was provided in Table 2, the mathematical linear programming problem can be converted into a Python Linprog function call in Python. After running the program, it is not surprising that the same results were obtained. The program obtained a 0.0 for the objective function, which represents zero net payoffs. The array of x is still (0,0,0,0.5,0.5), meaning that removing one strategy that Colonel Blotto will never pick is not going to change the net payoff and his strategy selections.

3.4. Alternating the opponent's strategies

In the previous sections, it is shown that Colonel Blotto will always pick a strategy between his third and fourth strategies. Moreover, if a strategy besides the third one and the fourth one is removed, the net payoff and the strategy array for Colonel Blotto will not change. In this section, this paper will analyze how changing the strategies of the opponent will affect Colonel Blotto's payoff and selection of strategies. The new payoff matrix obtained by eliminating the opponent's fifth strategy is shown in Table 4.

Table 4. Payoff matrix for Colonel Blotto Game after removing the opponent's fifth strategy.

	(0,0,5)	(0,1,4)	(0,2,3)	(1,1,3)
(0,0,5)	0	-1/3	-1/3	-1
(0,1,4)	1/3	0	0	-1/3
(0,2,3)	1/3	0	0	0
(1,1,3)	1	1/3	0	0
(1,2,2)	1	2/3	-1/3	1/3

The new payoff matrix can also be converted into a mathematical linear programming problem:

Maximize x_0, subject to:

$$\begin{cases} -x_0 + 0x_1 + \frac{1}{3}x_2 + \frac{1}{3}x_3 + x_4 + x_5 \geq 0 \\ -x_0 - \frac{1}{3}x_1 + 0x_2 + 0x_3 + \frac{1}{3}x_4 + \frac{2}{3}x_5 \geq 0 \\ -x_0 - \frac{1}{3}x_1 + 0x_2 + 0x_3 + 0x_4 - \frac{1}{3}x_5 \geq 0 \\ -x_0 - x_1 - \frac{1}{3}x_2 + 0x_3 + 0x_4 + \frac{1}{3}x_5 \geq 0 \\ 0x_0 + x_1 + x_2 + x_3 + x_4 + x_5 = 1 \\ x_1, x_2, x_3, x_4, x_5 \geq 0 \end{cases} \quad (3)$$

Utilizing Algorithm 1, which was provided in Table 2, the mathematical linear programming problem can be converted into a Python Linprog function call in Python. This time, there is a slight change in the results. The net payoff stays at 0.0. However, the array of x changes from (0,0,0,0.5,0.5,0) to (0,0,0,0,1,0). This represents that removing the fifth strategy of the opponent would make Colonel Blotto always pick strategy 4.

4. Conclusion

The objective of this paper is to perform the calculations of solving the Colonel Blotto Game using Python Linprog optimization to show the relationship between linear optimization and two-player zero-sum games. These games can be solved with mathematical linear programming problems. Therefore, the solutions to the Colonel Blotto Game can be obtained by converting and plugging the problem into a Python Linprog function. After running the codes, the results show that there is indeed an optimal strategy for Colonel Blotto to maximize his payoff, which turns out to be zero. By alternating the mathematical problem, it is shown that removing Colonel Blotto's fifth strategy would not change the results, while removing the opponent's fifth strategy would help Colonel Blotto be more certain when choosing strategies. In conclusion, programming languages like Python can be useful tools when trying to solve two-player zero-sum game problems, which is one of the applications of linear programming in real life.

References

[1] Dorfman R 1984 The discovery of linear programming. Annals of the History of Computing 6(3) 283-295.

[2] Shamir R 1987 The efficiency of the simplex method: a survey. Management science 33(3) 301-334.

[3] Nash J C 2000 The (Dantzig) simplex method for linear programming. Computing in Science & Engineering 2(1) 29-31.

[4] Williams, H. Paul. "Duality in mathematics and linear and integer programming." Journal of Optimization Theory and Applications 90 (1996): 257-278.

[5] Komodakis N, Jean-Christophe P 2015 Playing with duality: An overview of recent primal? dual approaches for solving large-scale optimization problems. IEEE Signal Processing Magazine 32(6) 31-54.

[6] Akpan N P, Iwok I A 2016 Application of linear programming for optimal use of raw materials in bakery. International journal of mathematics and statistics invention 4(8) 51-57.

[7] Oberman A M, Ruan Y L 2015 An efficient linear programming method for optimal transportation. Cornell University Numerical Analysis. arXiv preprint arXiv:1509.03668.

[8] Alotaibi A, Farrukh N 2021 A Review of Applications of Linear Programming to Optimize Agricultural Solutions. International Journal of Information Engineering & Electronic Business 13(2).

[9] Cantwell G L 2003 Can two person zero sum game theory improve military decision-making course of action selection. Army Command and General Staff Coll Fort Leavenworth Ks School of Advanced Military Studies.

DOI: 10.54254/2753-8818/11/20230414

[10] Aplak H, Mehmet K, Erkan K 2014 A two person zero sum game oriented to integration of objectives. Journal of Military Studies 5(2) 65-85.
[11] Adler I 2013 The equivalence of linear programs and zero-sum games. International Journal of Game Theory 42(1) 165.