Geometry of Simplex Lab

Objectives

- Understand the geometry of a linear program's feasible region.
- Use isoprofit lines and planes to solve 2D and 3D LPs graphically.
- Identify the most limiting constraint in an iteration of simplex both algebra geometrically.
- Identify the geometric features corresponding to dictionaries.
- Describe the geometrical decision made at each iteration of simplex.

Review

Recall, linear programs (LPs) have three main components: decision variables, objective function. The goal of linear programming is to find a **feasible solutio** satisfying every constraint) with the highest objective value. The set of feasible **feasible region**. In lecture, we learned about isoprofit lines. For every objectiv define an isoprofit line. Isoprofit lines have the property that two solutions on t same objective value and all isoprofit lines are parallel.

In the first part of the lab, we will use a Python package called GILP to solve li graphically. We introduce the package now.

GILP

If you are running this file in a Google Colab Notebook, uncomment the follow Otherwise, you can ignore it.

In [51]:
```
#pip install gilp
```

This lab uses default LPs built in to GILP. We import them below.

In [1]:
```
from gilp import examples as ex
```

We access the LP examples using `ex.NAME` where `NAME` is the name of th example, consider:

$$\begin{array}{rl} \max & 5x_1 + 3x_2 \\ \text{s.t.} & 2x_1 + 1x_2 \leq 20 \\ & 1x_1 + 1x_2 \leq 16 \\ & 1x_1 + 0x_2 \leq 7 \\ & x_1, x_2 \geq 0 \end{array}$$

This example LP is called `ALL_INTEGER_2D_LP` . We assign this LP to the v

In [2]:
```
lp = ex.ALL_INTEGER_2D_LP
```

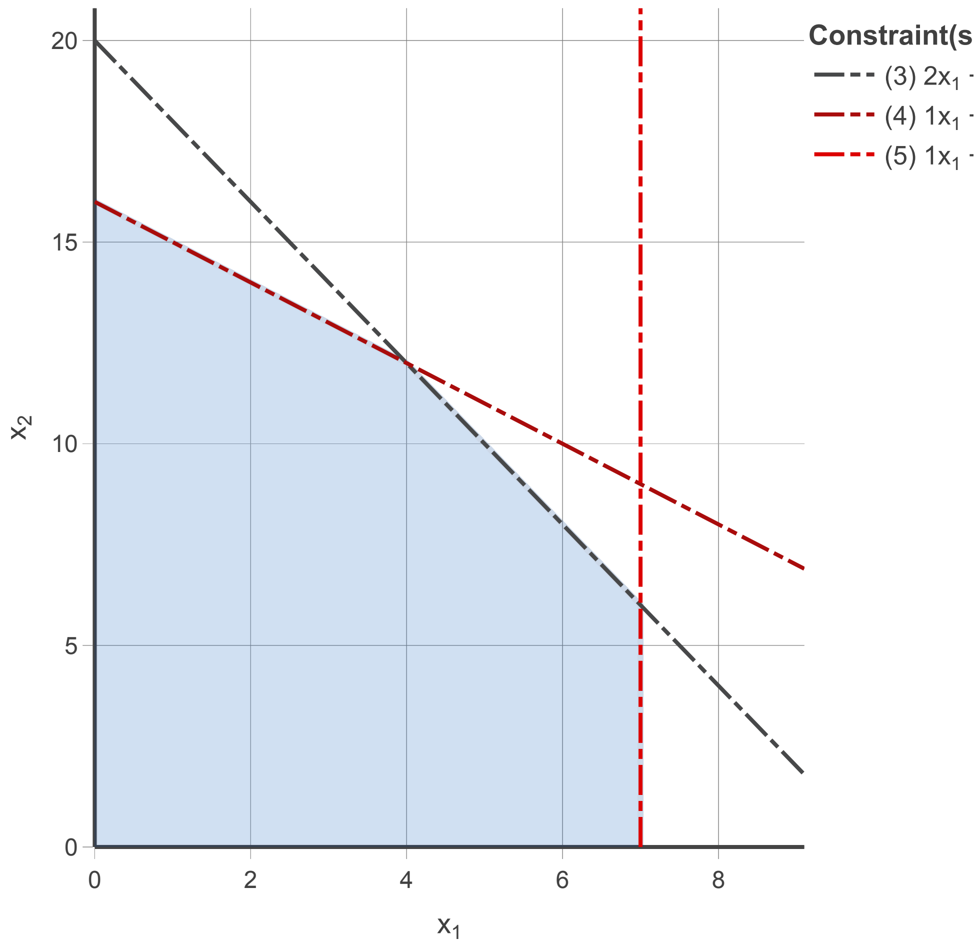
We can visualize this LP using a function called `lp_visual()` . First, we mu

In [3]:
```
from gilp.visualize import lp_visual
```

The function `lp_visual()` takes an LP and returns a visualization. We then function to display the visualiazation.

In [4]:
```
lp_visual(lp).show()
```

Geometric Interpretation of LPs



On the left, you can see a coordinate plane where the x-axis corresponds to t the y-axis corresponds to the value of x_2. The region shaded blue is the feasil perimeter of the feasible region, you can see points where two edges come to hover over these **corner points** to see information about them. Only some of hover box will be relevant for Part I. The first two values of **BFS** represent the respectively and **Obj** is the objective value. For example, the upper left corner $x_1 = 0$ and $x_2 = 16$ with objective value 48. The dashed lines represent the click on the constraints in the legend to mute and un-mute them. Note this do just changes visibility. Lastly, the objective slider allows you to see the isoprofi objective values.

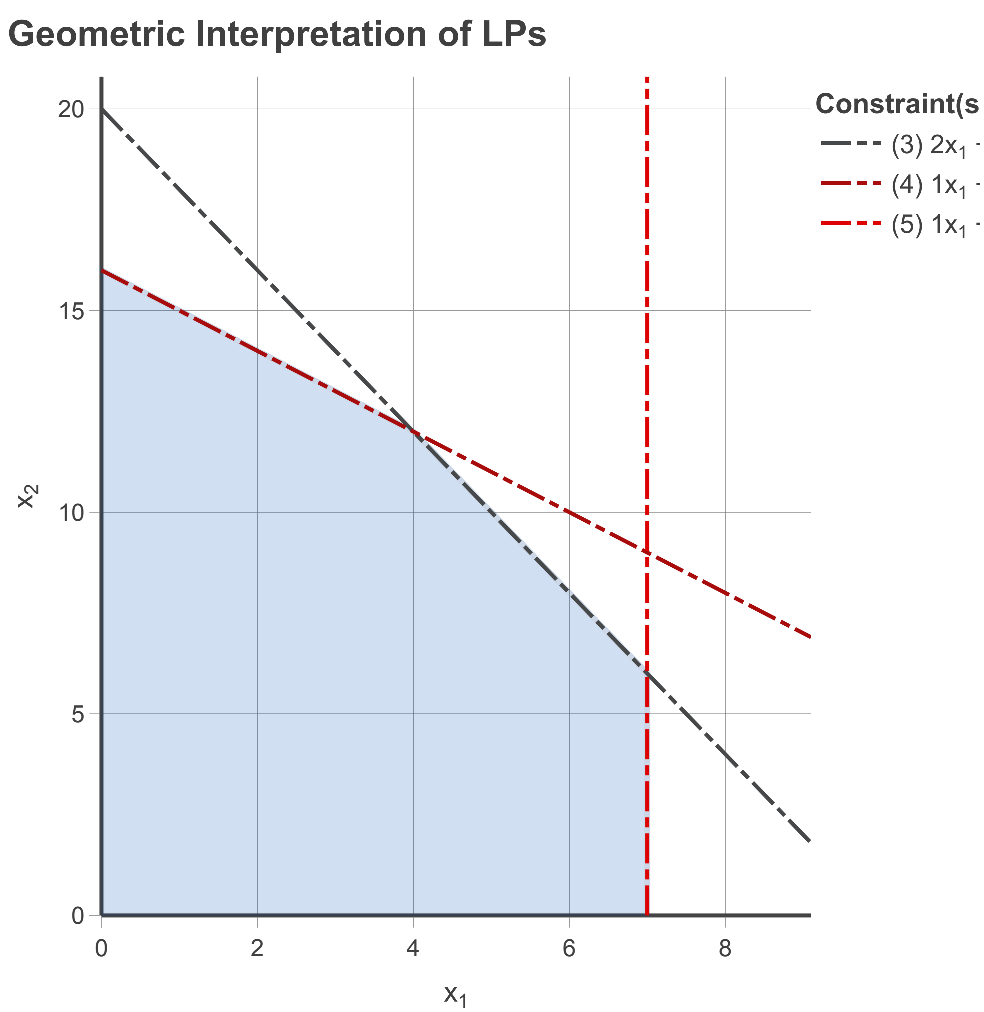
Part I: Solving Linear Programs Graphic

Let's use GILP to solve the following LP graphically:

$$\begin{aligned} \max \quad & 5(4) + 3(12) \\ \text{s.t.} \quad & 2(4) + 1(12) \leq 20 \\ & 1(4) + 1(12) \leq 16 \\ & 1(4) + 0(12) \leq 7 \\ & 1(4), (12) \geq 0 \end{aligned}$$

Recall, this LP is called `ALL_INTEGER_2D_LP`.

In [5]:
```
lp = ex.ALL_INTEGER_2D_LP # get LP example
lp_visual(lp).show() # visualize it
```



Q1: How can you use isoprofit lines to solve LPs graphically?

A: The optimal objective value would be the isoprofit line that is furthest from t
still in the domain.

Q2: Use the objective slider to solve this LP graphically. Give an optimal soluti
value. Argue why it is optimal. (Hint: The objective slider shows the isoprofit lir
objective value.)

A: Our objective value is 56 and the optimal solution is where x1 = 4 and x2 =
because the line is on the boundary of the domain.

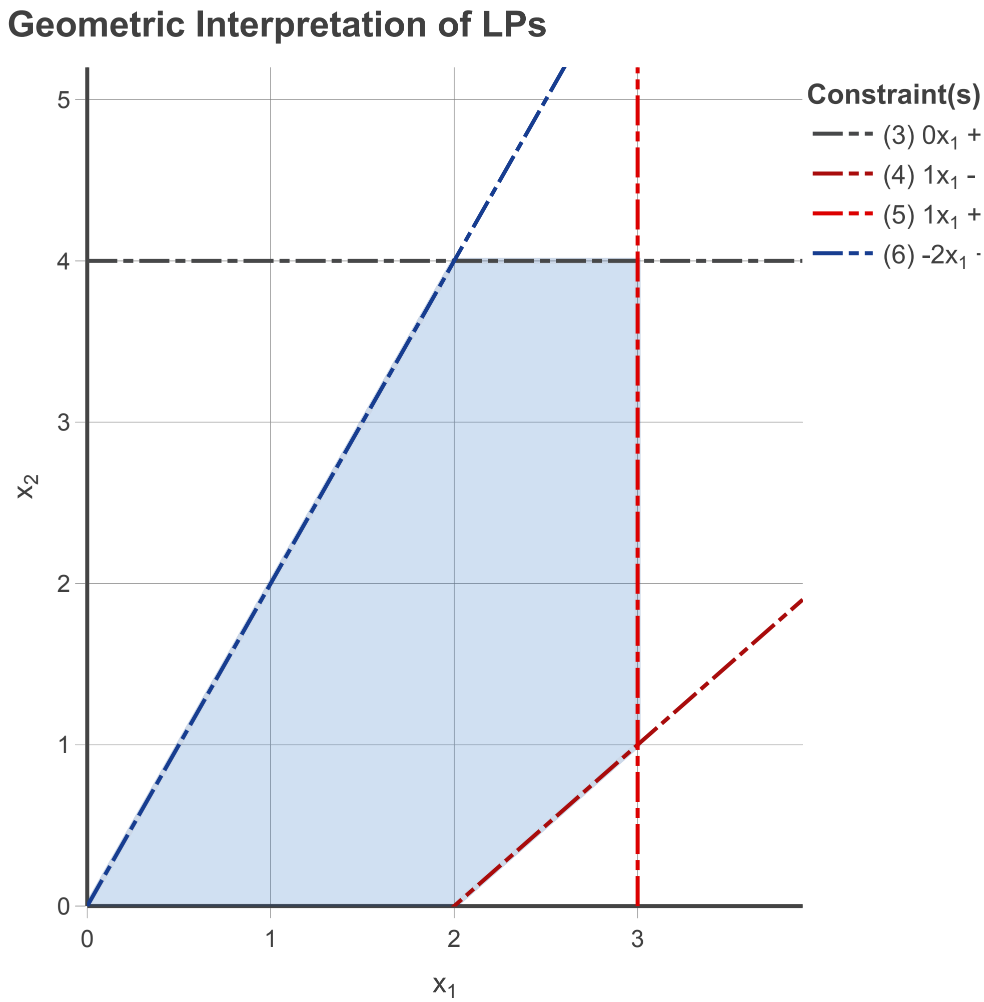
Q3: Plug your solution from **Q2** back into the LP and verify that each constrair
forget non-negativity constraints!) and the objective value is as expected. Sho

A: Each constraint is satisfied. We put the values into solving LP graphically.

Let's try another! This LP is called `DEGENERATE_FIN_2D_LP` .

$$\begin{aligned} \max \quad & 1x_1 + 2x_2 \\ \text{s.t.} \quad & 0x_1 + 1x_2 \leq 4 \\ & 1x_1 - 1x_2 \leq 2 \\ & 1x_1 + 0x_2 \leq 3 \\ & -2x_1 + 1x_2 \leq 0 \\ & x_1, x_2 \geq 0 \end{aligned}$$

In [6]:
```
lp = ex.DEGENERATE_FIN_2D_LP # get LP example
lp_visual(lp).show() # visualize it
```



Q4: Use the objective slider to solve the `DEGENERATE_FIN_2D_LP` LP grapl optimal solution and objective value. (Hint: The objective slider shows the isop some objective value.)

A: Our objective value is 11 and the optimal solution is 3 for x1 and 4 for x2.

You should now be comfortable solving linear programs with two decision vari this case, each constraint is a line representing an inequality. These inequalite: region in the coordinate plane which is our feasible region. Lastly, the isoprofit find an optimal solution, we just increase the objective value while the corresp still intersects the 2D feasible region.

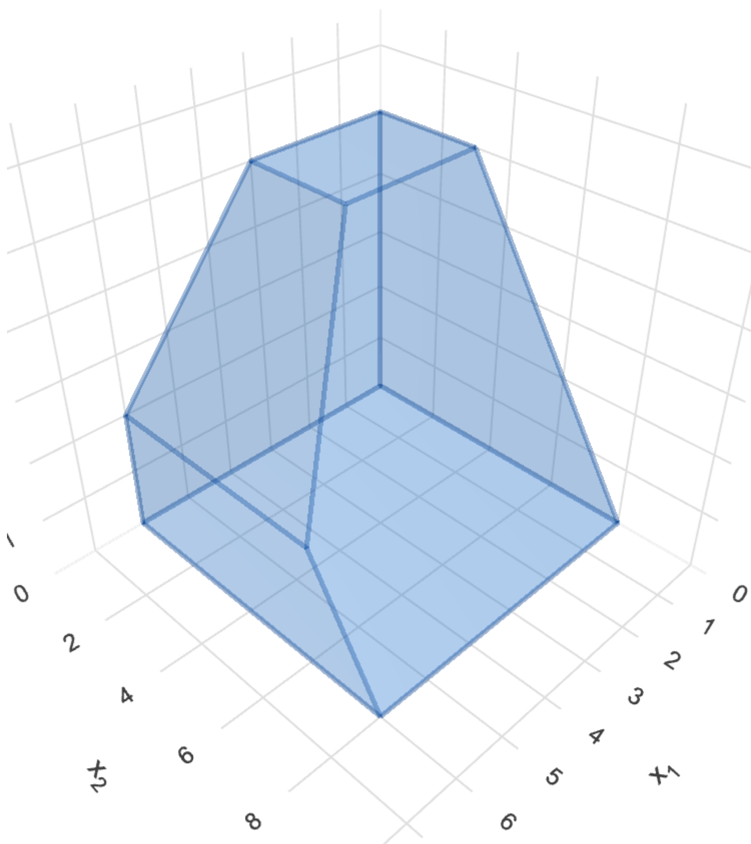
Now, we will try to wrap our head around an LP with three decision variables! can plot solutions to a 3D LP on a plot with 3 axes. Here, the x-axis correspor and the y-axis corresponds to the value of x_2 as before. Furthermore, the z-a the value of x_3. Now, constraints are *planes* representing an inequality. These define a 3D shaded region which is our feasible region. The isoprofits are isop parallel. To find an optimal solution, we just increase the objective value while isoprofit plane still intersects the 3D feasible region. Let us look at an example

This LP is called `ALL_INTEGER_3D_LP` :

$$
\begin{aligned}
\max \quad & 1x_1 + 2x_2 + 4x_3 \\
\text{s.t.} \quad & 1x_1 + 0x_2 + 0x_3 \leq 6 \\
& 1x_1 + 0x_2 + 1x_3 \leq 8 \\
& 0x_1 + 0x_2 + 1x_3 \leq 5 \\
& 0x_1 + 1x_2 + 1x_3 \leq 8 \\
& x_1, x_2 \geq 0
\end{aligned}
$$

In [7]:
```
lp = ex.ALL_INTEGER_3D_LP # get LP example
lp_visual(lp).show() # visualize it
```

Geometric Interpretation of LPs



Constraint(s)

(4) $1x_1$ + $0x_2$
(5) $1x_1$ + $0x_2$
(6) $0x_1$ + $0x_2$
(7) $0x_1$ + $1x_2$

The 3D feasible region is shown on the left. Hold and drag the mouse to exam angles. Next, click on a constraint to un-mute it. Each constraint is a gray plar mute the constraints one by one to see how they define the 3D feasible region slider to see the isoprofit planes. The isoprofit plane is light gray and the inters feasible region is shown in red. Like the 2D visualization, you can hover over c information about that point.

Q5: Use the objective slider to solve this LP graphically. Give an optimal soluti value. (Hint: The objective slider shows the isoprofit plane for some objective v the intersection with the feasible region in red.)

A: Our objective value is 29 and our optimal solution is x1 = 5 and x2 = 6.

When it comes to LPs with 4 or more decision variables, our graphical approa
find a different way to solve linear programs of this size.

Part II: The Simplex Algorithm for Solvir

Dictionary Form LP

First, let's answer some guiding questions that will help to motivate the simple

Q6: Does there exist a unique way to write any given inequality constraint? If s
constraint can only be written one way. Otherwise, give 2 ways of writing the s
constraint.

A: Yes, there is a unique way to write any given ineqaulity constraint. We can r
by the same number or divide them.

Q7: Consider the following two constraints: $2x_1 + 1x_2 \leq 20$ and $2x_1 + 1x_2$
x are nonnegative. Are these the same constraint? Why? (This question is tricl

A: Yes, they are the same constraint. When we subtract x3 on both sides, 20-x

Q8: Based on your answers to **Q6** and **Q7**, do you think there exists a unique
given LP?

A: While there are unique ways to write the constraint, there aren't any unique
LP.

You should have found that there are many ways to write some LP. This begs a
some ways of writing an LP harder or easier to solve than others? Consider th

$$\begin{array}{rl} \max & 56 - 2x_3 - 1x_4 \\ \text{s.t.} & x_1 = 4 - 1x_3 + 1x_4 \\ & x_2 = 12 + 1x_3 - 2x_4 \\ & x_5 = 3 + 1x_3 - 1x_4 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0 \end{array}$$

Q9: Just by looking at this LP, can you give an optimal solution and its objectiv
what property of the LP allows you to do this. (Hint: Look at the objective func

A: The objective value is 56 and the optimal solution is (4,12,0,0,3). And the pr
to do this would be having non-negative coefficients.

The LP above is the same as `ALL_INTEGER_2D_LP` just rewritten in a differe

form (which we found is easier to solve) was found using the simplex algorithn
simplex algorithm strategically rewrites an LP until it is in a form that is "easy"

The simplex algorithm relies on an LP being in **dictionary form**. Recall the foll
an LP in dictionary form:

- All constraints are equality constraints
- All variables are constrained to be nonnegative
- Each variable only appears on the left-hand side (LHS) or the right-hand s constraints (not both)
- Each constraint has a unique variable on the LHS
- The objective function is in terms of the variables that appear on the RHS only.
- All constants on the RHS of the constraints are nonnegative

Q10: Rewrite the example LP `ALL_INTEGER_2D_LP` in dictionary form. Sho

$$\begin{array}{rl} \max & 5x_1 + 3x_2 \\ \text{s.t.} & 2x_1 + 1x_2 \leq 20 \\ & 1x_1 + 1x_2 \leq 16 \\ & 1x_1 + 0x_2 \leq 7 \\ & x_1, x_2 \geq 0 \end{array}$$

A: x3 = 20 - 2x1 - 1x2 x4 = 16 - 1x1 - 1x2 x5 = 7 - 1x1 -0x2

x1, x2, x3, x4, x5 >= 0

Most Limiting Constraint

Once our LP is in dictionary form, we can run the simplex algorithm! In every it
simplex algorithm, we will take an LP in dictionary form and strategically rewri
dictionary form. Note: it is important to realize that rewriting the LP **does not** c
feasible region. Let us examine an iteration of simplex on a new LP.

$$\begin{array}{rl} \max & 5x_1 + 3x_2 \\ \text{s.t.} & 1x_1 + 0x_2 \leq 4 \\ & 0x_1 + 1x_2 \leq 6 \\ & 2x_1 + 1x_2 \leq 9 \\ & 3x_1 + 2x_2 \leq 15 \\ & x_1, x_2 \geq 0 \end{array}$$

Q11: Is this LP in dictionary form? If not, rewrite this LP in dictionary form.

A: This is not in dictionary form because we say an LP is in dictionary form if a
equality constraints, and all variables are constrained to be nonnegative.

max z = 5x1 + 3x2

x3 = 4 - 1x1 - 0x2 x4 = 6 - 0x1 - 1x2 x5 = 9 - 2x1 - 1x2 x6 = 15 - 3x1 - 2x2

x1, x2, x3, x4, x5, x6 >= 0

Q12: Recall from **Q9** how you found a feasible solution (which we argued to b looking at the LP. Using this same stratagy, look at the LP above and give a fe objective value for this LP. Describe how you found this feasible solution. Is it

A: (0, 0, 4, 6, 9, 15). This is not an optimal solution because x3 through x6 is n objective value will be 0.

From **Q12** we see that every dictionary form LP has a corresponding feasible there are positive coefficents in the objective function. Hence, we can increase by increaseing the corresponding variable. In our example, both x_1 and x_2 ha coefficents in the objective function. Let us choose to increase x_1.

Q13: What do we have to be careful about when increasing x_1?

A: We have to make sure that x1 follows all of the constraints in the dictionary

Q14: After choosing a variable to increase, we must determine the most limitir look at the first constraint $x_3 = 4 - 1x_1 - 0x_2$. How much can x_1 increase? dictionary form LP require about the constant on the RHS of constraints?)

A: We can increase x1 by 4.

Q15: Like in **Q14**, determine how much each constraint limits the increase in most limiting constraint.

A:

x3 = by 4, x4 = no limit, x5 = by 4.5, x6 = by 5,

The most limiting constraint would be the first constraint.

If we increase x_1 to 4, note that x_3 will become zero. Earlier, we identified tha has a corresponding feasible solution acheived by setting variables on the RH objective function) to zero. Hence, since x_3 will become zero, we want to rew x_3 appears on the RHS. Furthermore, since x_1 is no longer zero, it should nov

Q16: Rewrite the most limiting constraint $x_3 = 4 - 1x_1 - 0x_2$ such that x_1 a and x_3 appears on the right.

A: x1 = 4 - x3

Q17: Using substitution, rewrite the LP such that x_3 appears on the RHS and LHS. (Hint: Don't forget the rule about which variables can appear in the objec

A:

max z = 20 - 5x3 + 3x2

x1 = 4 - 1x3 x4 = 6 - 1x2 x5 = 1 + 2x3 - 1x2 x6 = 3 + 3x3 - 2x2

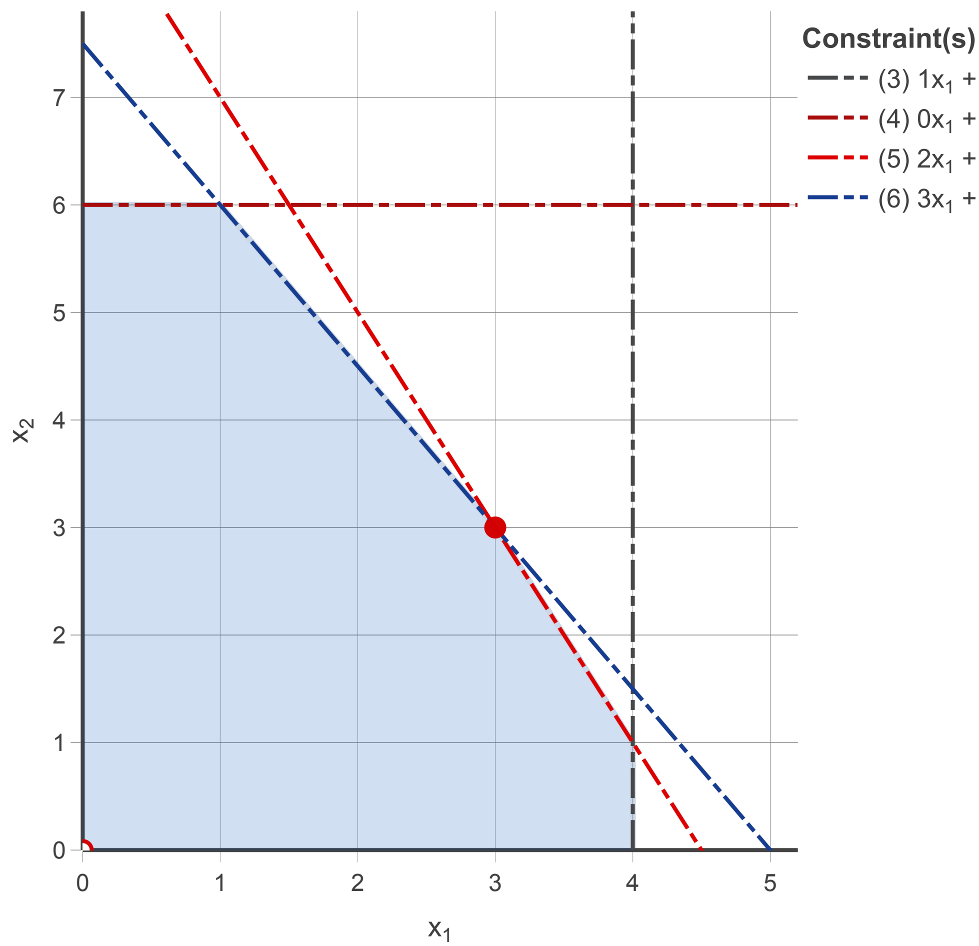
Q18: We have now completed an iteration of simplex! What is the correspondi
of the new LP?

A: (4, 0, 0, 6, 1, 3) objective function = 20

Now that we have seen an iteration of simplex algebraically, let's use GILP to
example we have been using is called `LIMITING_CONSTRAINT_2D_LP` . To
must import a function called `simplex_visual()` .

In [8]:
```
from gilp.visualize import simplex_visual # import the fun
import numpy as np
lp = ex.LIMITING_CONSTRAINT_2D_LP # get the LP example
simplex_visual(lp, initial_solution=np.array([[0],[0]])).s
```

Geometric Interpretation of LPs



This visualization is much the same as the previous one but we now have an a
allows you to toggle through iterations of simplex. Furthermore, the correspon
every iteration of simplex is shown in the top right. If you toggle between two i
see the dictionary form for both the previous and next LP at the same time.

Q19: Starting from point (0,0), by how much can you increase x_1 before the p
feasible? Which constraint do you *hit* first? Does this match what you found al

A: We can increase x1 by 4, and we hit constraint (3) first. And yes this match
algebraically.

Q20: Which variable will be the next increasing variable and why? (Hint: Look
LP at iteration 1)

A: We have to increase x2 because in our graph that is the only remaining vari
And also looking back at our calculations, x2 has a positive coefficient.

Q21: Visually, which constraint do you think is the most limiting constraint? H
increase? Give the corresponding feasible solution and its objective value of t
form LP. (Hint: hover over the feasible points to see information about them.)

A: We can increase x2 by 1. Corresponding feasible solution = 4,1, 0, 5, 0, 1.

Q22: Move the slider to see the next iteration of simplex. Was your guess from
describe how your guess was wrong.

A: Our guess was correct.

Q23: Look at the dictionary form LP after the second iteration of simplex. Wha
variable? Identify the most limiting constraint graphically and algebraically. Sh
verify they are the same constraint. In addition, give the next feasible solution
value.

A: We are trying to maximize x3 because it has a positive coefficient. The mos
the last one. 3x1 + 2x2 <= 15, x6 = 15 - 3x1 - 2x2. We are going to substitute
original input (with x1 and x2). The next feasible solution is (3,3,1,3,0,0). The o

x6 is the most limiting constraint. x6 = 1 - x3 + 2x5 (iteration 2)

substitute values for x3 and x5.

Q24: Is the new feasible solution you found in **Q23** optimal? (Hint: Look at the

A: The solution is optimal because the coefficients in the objective function ar

Q25: In **Q21** and **Q23**, how did you determine the most limiting constraint gra

A: We travel along the constraints and along the boundaries of the constraint.

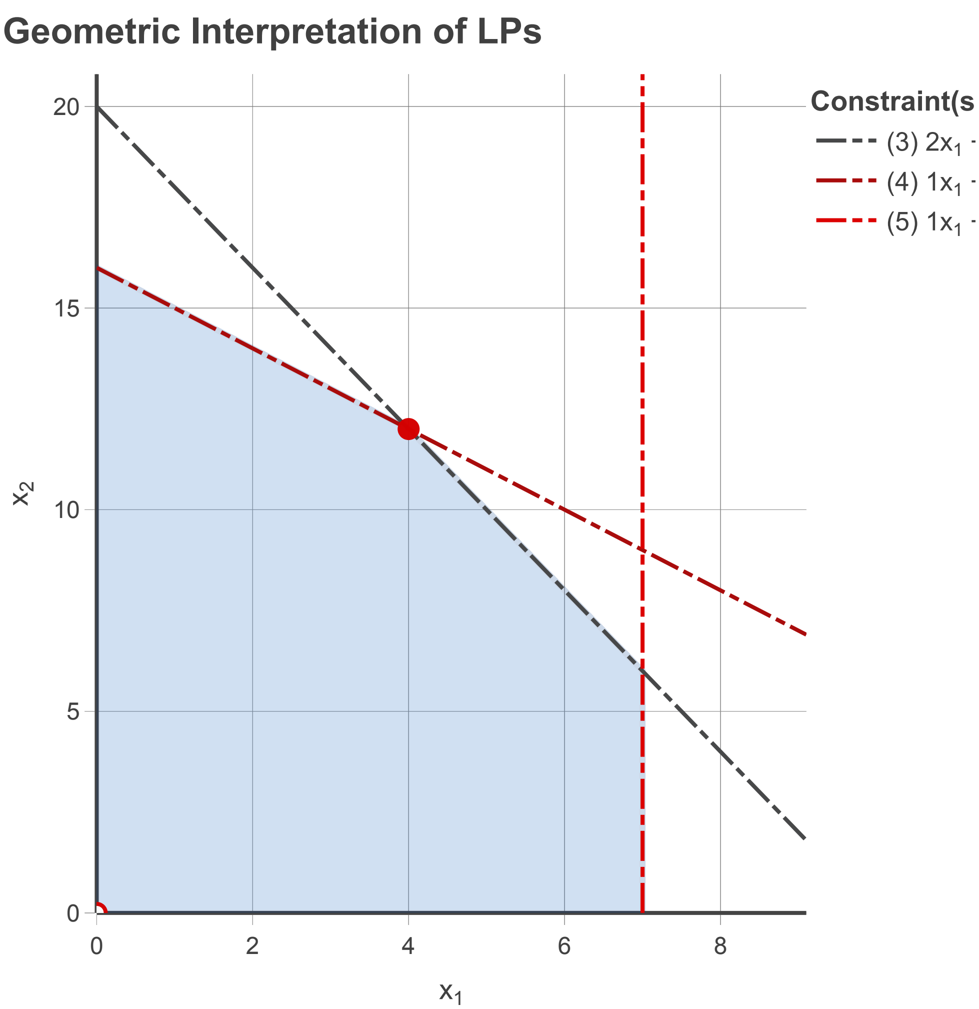
(BONUS): In 2D, we can increase a variable until we hit a 2D line representing
constraint. What would be the analagous situation in 3D?

A:

Part III: Geometrical Interpretation of th Dictionary

We have seen how the simplex algorithm transforms an LP from one dictionar Each dictionary form has a corresponding dictionary defined by the variables constraints. Furthermore, each dictionary form has a corresponding feasible s setting all non-dictionary variables to 0 and the dictionary variables to the con this section, we will explore the geometric interpretation of a dictionary.

In [9]:
```
lp = ex.ALL_INTEGER_2D_LP # get LP example
simplex_visual(lp, initial_solution=np.array([[0],[0]])).s
```



Recall, we can hover over the corner points of the feasible region. **BFS** indicat solution corresponding to that point. For example, (7,0,6,9,0) means $x_1 = 7, x_2 = 0, x_3 = 6, x_4 = 9$, and $x_5 = 0$. **B** gives the indices of the varia For example, (1,3,4) means that x_1, x_3, and x_4 are in the dictionary. Lastly, the that point is given.

Q26: Hover over the point (7,6) where $x_1 = 7$ and $x_2 = 6$. What is the feasibl point ?

A: The feasible solution at this point would be (7,6,0,3,0).

We have a notion of *slack* for an inequality constraint. Consider the constraint
solution where $x_1 = 7$ has a slack of 7 in this constraint. Consider the constra
The feasible solution with $x_1 = 7$ and $x_2 = 6$ has a slack of 0 in this constrai

Q27: What is the slack in constraint $1x_1 + 1x_2 \leq 16$ when $x_1 = 7$ and $x_2 =$

A: The slack is 3 for this constraint.

Q28: Look at the constraint $2x_1 + 1x_2 \leq 20$. After rewriting in dictionary form
$x_3 = 20 - 2x_1 - 1x_2$. What does x_3 represent?

A: x3 is the slack of the inequality constraint.

Q29: What do you notice about the feasible solution at point (7,6) and the slac

A: The slack in each constraint is the solution for each of the x variables in the

It turns out that each decision variable is really a measure of slack in some co
constraint!

Q30: If the slack between a constraint and a feasible solution is 0, what does t
relationship between the feasible solution and constraint geometrically?

A: It has reached the maximum capacity of the constraint.

Q31: For (7,6), which variables are **not** in the dictionary? For which constraints
the slack? (Hint: The **B** in the hover box gives the indices of the variables in th

A: x3 and x5 are not in the dictionary. The slack is equal to 0.

Q32: For (7,6), what are the values of the non-dictionary variables? Using wha
Q30, what does their value tell you about the feasible solution at (7,6)?

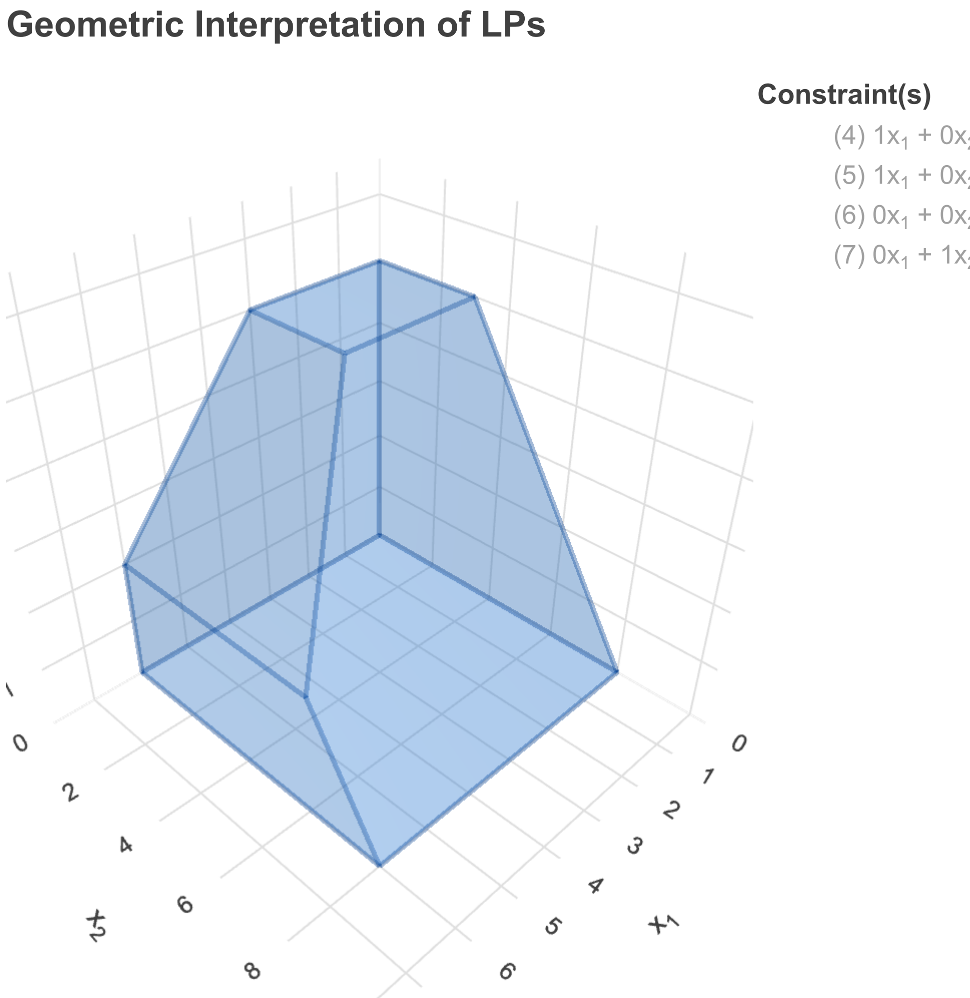
A: 0. The fact that the values of the non-dictionary variables are 0 tell us that c

Q33: Look at some other corner points with this in mind. What do you find?

A: I find that there are more feasible solutions but the one the top right has the
value, leading us to make the conclusion that that is our optimal solution.

Now, let's look at a 3 dimensional LP!

In [10]:
```
lp = ex.ALL_INTEGER_3D_LP # get LP example
lp_visual(lp).show() # visualize it
```



Q34: Hover over the point (6,6,2) where $x_1 = 6$, $x_2 = 6$, and $x_3 = 2$. Note wl
in the dictionary. Toggle the corresponding constraints on. What do you notice

A: The variables x4, x5, and x7 are not in the dictionary. We can see that this p
constraints. (on the borders). The constraints have been optimized.

Q35: Look at some other corner points and do as you did in Q34. Do you see
Combining what you learned in Q33, what can you say about the relationship
not in the dictionary at some corner point, and the corresponding constraints?

A: We can say that the variables not in the dictionary put constraints for the va
dictionary so as to optimize the constraint (intersection between the planes).

Q36: What geometric feature do feasible solutions for a dictionary correspond

A: The point of intersection between the planes that the constraints create.

Part IV: Pivot Rules

The first step in an iteration of simplex is to choose an increasing variable. Sor
multiple options since multiple variables have a positive coefficent in the objec
we will explore what this decison translates to geometrically.

In this section, we will use a special LP commonly referred to as the Klee-Mint

Furthermore, we will use an optional parameter called `rule` for the `simple`
function. This rule tells simplex which varaible to choose as an increasing vari
multiple options.

In [ ]:
```
simplex_visual(ex.KLEE_MINTY_3D_LP, rule='dantzig', initia
```

In [ ]:
```

```

Q37: Use the iteration slider to examine the path of simplex on this LP. What c

A: We notice that the path starts increases in the order of x1, x2, x4, x3, x5, x4

Above, we used a pivot rule proposed by Dantzig. In this rule, the variable with
coefficient in the objective function enters the dictionary. Go through the iterat
this.

Let us consider another pivot rule proposed by Bland, a professor here at Cor
variables with positive coefficents in the objective function, the one with the sr
Let us examine the path of simplex using this pivot rule! Again, look at the dict
every iteration.

In [ ]:
```
simplex_visual(ex.KLEE_MINTY_3D_LP,rule='manual_select', i
```

Q38: What is the difference between the path of simplex using Dantzig's rule a

A: Bland's rule goes from x1, x2, x3, x5, x4. Less iterations.

Can you do any better? By setting `rule='manual_select'` , you can choo
variable explicitly at each simplex iteration.

Q39: Can you do better than 5 iterations? How many paths can you find? (By

A: Yes, we can do better than 5 iterations. We can straight up maximize x3 and
1 iteration.

In [ ]:
```
simplex_visual(ex.KLEE_MINTY_3D_LP,rule='manual_select', i
```

Q40: What does the choice of increasing variable correspond to geometrically

A: It means that we are moving onto the next constraint's plane from the previ

Q41: Are there any paths you could visualize taking to the optimal solution tha `rule='manual_select'` prevented you from taking? If yes, give an examp is not a valid path for simplex to take. (Hint: Look at the objective value after e iteration.)

A: The path 2, 1, 3, 5, 4 would not work because it would unoptimize the cons

Part V: Creating LPs in GILP (Optional)

We can also create our own LPs! First, we must import the `LP` class.

In [ ]:
```
from gilp.simplex import LP
```

Let us create the following LP.

$$\begin{aligned} \max \quad & 3x_1 + 2x_2 \\ \text{s.t.} \quad & 2x_1 + 1x_2 \leq 6 \\ & 0x_1 + 1x_2 \leq 2 \\ & x_1, x_2 \geq 0 \end{aligned}$$

We will create this LP by specifying 3 arrays of coefficents. We define the Num and `c` and then pass them to the `LP` class to create the LP.

In [ ]:
```
A = np.array([[2,1],  # LHS constraint coefficents
              [0,1]])
b = np.array([6,2])  # RHS constraint coefficents
c = np.array([3,2])  # objective function coefficents
lp = LP(A,b,c)
```

Let's visualize it!

In [ ]:
```
lp_visual(lp).show()
```

... and solve it!

In [ ]:
```
simplex_visual(lp, initial_solution=np.array([[0],[0]])).s
```

In [ ]:
```
```

In [ ]:
```
```

In [ ]:
```
```