MIP Problem

Mixed Integer Programming

Problem: Maximize x+10y subject to the following constraints:

```
x + 7y <= 17.5
x <= 3.5
x >= 0
y >= 0
x, y integers
```

Constraints are linear, it's just a linear optimization problem in which the solutions are required to be integers.

Integer: not fractinoal number, can be positive, negative and zero

Basic steps for solving a MIP problme

1. Import the linear solver wrapper
2. Declare the MIP solver
3. Define the variables
4. Define the constraints
5. Define the objective
6. Call the MIP solver
7. Display the solution

In order to increase computational speed, the CP-SAT solver works over the integers.

```
from ortools.sat.python import cp_model

# Decalre hte model

model = cp_model.CpModel()


# Create the variables

x = model.NewIntVar(0, 50, 'x')

y = model.NewIntVar(0,50,'y')

# Define the constraints
```

```
model.Add(x+7*7 <= 17.5)

model.Add(x <= 3.5)

model.Add(x >= 0)

model.Add(y >= 0)


# Define the objective function

model.Maximize(x+10*y)


# Call the solver

solver = cp_model.CpSolver()

status = solver.Solve(model)


# Display the solution

if status == cp_model.OPTIMAL or status == cp_model.FEASIBLE:

print(f'Maximum of objective function: {solver.ObjectiveValue()}\n')

print(f'x = {solver.Value(x)}')

print(f'y = {solver.Value(y)}')

else:

print('No solution found.')
```

TypeError: Unrecognized linear expression: -17.5

Here are the complete programs

```
from ortools.sat.python import cp_model

```

```
# Decalre hte model

model = cp_model.CpModel()



# Create the variables

x = model.NewIntVar(0, 50, 'x')

y = model.NewIntVar(0,50,'y')




# Define the constraints

# has non-integer coefficients, you must first multiply the entire constraint by a sufficiently large integer

# to convert the coefficients to integers. In this case, you can multiply by 2, which results in the new constraint

model.Add(2*x+14*y <= 35)

model.Add(2*x <= 7)

model.Add(x >= 0)

model.Add(y >= 0)



# Define the objective function

model.Maximize(x+10*y)



# Call the solver

solver = cp_model.CpSolver()

status = solver.Solve(model)
```

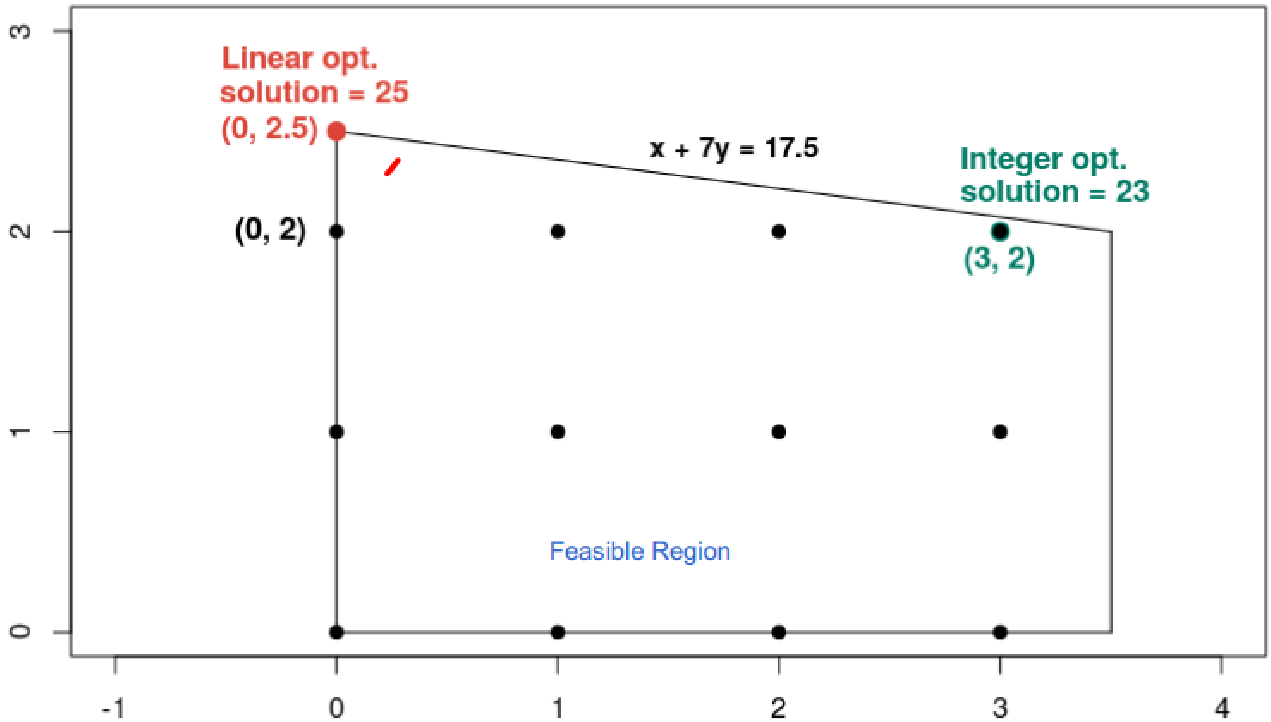
```
# Display the solution

if status == cp_model.OPTIMAL or status == cp_model.FEASIBLE:

print(f'Maximum of objective function: {solver.ObjectiveValue()}\n')

print(f'x = {solver.Value(x)}')

print(f'y = {solver.Value(y)}')

else:

print('No solution found.')
```

The output:

```
Maximum of objective function: 23.0



x = 3


y = 2
```

The optimal value of objective function is 23.
Which occurs at the point x = 3, y =

Comparing Linear and Integer Optimization



The integer solution is not close to the linear solution.

In general, the solutions to a linear optimization problem and the corresponding integer optimization problems can be far apart.

The two types of problems require different methods for their solution.