Linear Programming

Name: Sanghyeon Kim

Student ID: 20181605

Date: 11/13/2020

The linear program on problem 7.5 has three constraints on its variables, either an upper bound or a lower bound for each. The goal is to maximize the function $\max_{x \in R^2, \xi \in R} \boldsymbol{p}^\top \boldsymbol{x} + \xi$, with constraints on $\boldsymbol{x}, \xi$. The problem is expressed as a linear program $\boldsymbol{c}^\top \mathbf{u}$, where $\boldsymbol{c}^\top = [p_0\ p_1\ 1], \boldsymbol{u}^\top = [x_0\ x_1\ \xi]$. The vector $\boldsymbol{c}$ determines whether the function decreases or increases with respect to $\boldsymbol{u}$. For each element of $\boldsymbol{c}$ and the constraint on its corresponding element in $\boldsymbol{u}$, the program determines if the function increases without bound. If the function increases unbounded, the unique solution cannot be found, and the program terminates. If the function increases with bound, the values of the constraints on $\boldsymbol{u}$ and $\boldsymbol{c}$ is used to calculate the maximum of the function. In the case of problem 7.5, the solution is precisely on the boundary set by the constraint. Therefore, the unique solution to the linear program is at the bound of the constraints.

```
# Name: lp.py
# Author: Sanghyeon Kim
# Date: 11/13/20

# Check if the solution is unbounded and return 0 if unbounded or return 1 if
unique/bounded
def check_solutionBound(c, constraints):
    # Check unbounding condition for each element of c and the corresponding c
onstraint for its variable
    for i in range(len(c)):
        if(constraints[i][0] == 'l'):
            state = -c[i]
        else:
            state = c[i]
        if(state < 0):
            return 0

    return 1

# Get the maximum of c^T*u and the solution vector u
def findSolution(c, constraints):
    total = 0.0
    solution = list()
    for i in range(len(c)):
```

```
        total = total + c[i] * constraints[i][1]
        solution.append(constraints[i][1])

    return solution, total

def main():
    c = list(map(float, input("Type the first two elements(c_0, c_1) of c: ").
split()))
    c.append(1.0) # The third element(c_2) of c is 1
    constraints = list()

    print("Set upper or lower bounds for the variables(u_0 = x_0, u_1 = x_1, u
_2 = xi):")
    for i in range(3):
        # Set constraints for each variable as upper or lower bound
        while True:
            # Force user to type the correct bound type(u/l)
            bound_type = input("-
upper or lower bound for u_{}? <Type u for upper or l for lower>: ".format(i))
            if(bound_type == 'u' or bound_type == 'l'):
                bound = float(input("-
Type the bound value for the u_{}: ".format(i)))
                constraints.append([bound_type, bound]) # Save bound type(u/l)
 and the value of the constraint for each variable
                break

    # Print the summary of the linear program
    print("\nThe problem is to maximize c^T*u, where")
    print("<c^T> = ", c)
    print("with the constraints:")
    for i in range(len(constraints)):
        print("u_{}: {}".format(i, constraints[i]))

    state = check_solutionBound(c, constraints)

    if(state == 0):
        print("\nSolution is unbounded at infinity")
    else:
        solution, result = findSolution(c, constraints)
        print("\nThe optimal solution is u = {} with {}".format(solution, resu
lt))

main()
```