optimization

March 31, 2023

[1]:
```
import cvxpy as cp
import numpy as np
```

[2]:
```
def check_solution(x, y, A, rhs):
    for A_, rhs_ in zip(A, rhs):
        if A_(x, y) > rhs_:
            print(f"Constraint is violated with value {A_(x, y)} > {rhs_}.")
        else:
            print(f"Constraint is satisfied with value {A_(x, y)} <= {rhs_}.")
```

1 Linear Programming

[3]:
```
# Define the decision variables
x = cp.Variable(nonneg=True)  # x >= 0
y = cp.Variable(nonneg=True)  # y >= 0

# Define the objective function
objective = cp.Maximize(20*x + 25*y)

# Define the constraints
constraints = [
    2*x + 3*y <= 30,
    3*x + 1*y <= 20,
]

# Define and solve the problem
problem = cp.Problem(objective, constraints)
problem.solve()

# Print the optimal solution and optimal value
print("Optimal solution: x =", x.value, ", y =", y.value)
print("Optimal value:", problem.value)
```

```
Optimal solution: x = 4.2857141284411195 , y = 7.142857244712961
Optimal value: 264.28571368664643
```

[4]:
```
# Define the constraints
A = [
    lambda x, y: 2*x + 3*y,
    lambda x, y: 3*x + 1*y,
]

rhs = [
    30,
    20,
]

check_solution(x.value, y.value, A, rhs)
```

```
Constraint is satisfied with value 29.999999991021124 <= 30.
Constraint is satisfied with value 19.99999963003632 <= 20.
```

2 Integer Programming

[5]:
```
# Define the decision variables
x = cp.Variable(integer=True)
y = cp.Variable(integer=True)

# Define the objective function
objective = cp.Maximize(20*x + 25*y)

# Define the constraints
constraints = [
    2*x + 3*y <= 30,
    3*x + 1*y <= 20,
    x >= 0,  # x >= 0
    y >= 0,  # y >= 0
]

# Define and solve the problem
problem = cp.Problem(objective, constraints)
problem.solve()

# Print the optimal solution and optimal value
print("Optimal solution: x =", x.value, ", y =", y.value)
print("Optimal value:", problem.value)
```

```
Optimal solution: x = 3.0 , y = 8.0
Optimal value: 260.0
```

[6]:
```
# Define the constraints
A = [
    lambda x, y: 2*x + 3*y,
```

```
    lambda x, y: 3*x + 1*y,
]

rhs = [
    30,
    20,
]

check_solution(x.value, y.value, A, rhs)
```

```
Constraint is satisfied with value 30.0 <= 30.
Constraint is satisfied with value 17.0 <= 20.
```

[ ]: