Peer Feedback - Hand-In

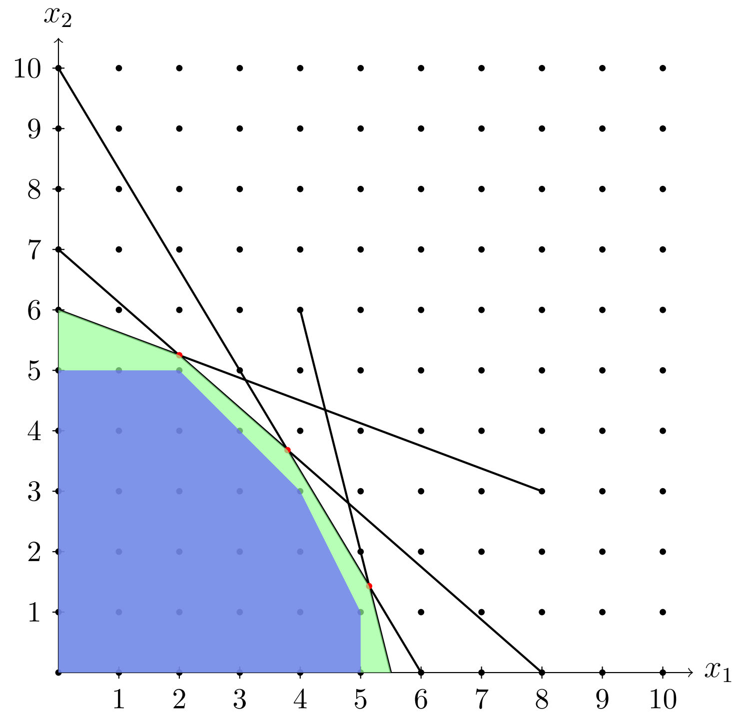
Modellering og Løsning af Optimeringsproblemer

April 28, 2025

Opgave 2.

1.

Det blå område omkrandser alle brugbare heltalpunkter for det linelre programmeringsproblem.



2.

Branch and Bound

Trin 1: Relaxation
Først løser vi ILP-problemet uden heltalskrav. Hvis løsningen allerede giver heltal på alle heltalsvariabler → færdig.

Trin 2: Hvis løsningen ikke er heltallig
Så brancher vi på en variabel, der ikke er heltallig.
Vi laver to grene:

- Én gren hvor vi tvinger variablen til at være mindre end eller lig den største hele værdi under den fundne værdi.
- Én gren hvor vi tvinger variablen til at være større end eller lig den mindste hele værdi over den fundne værdi.

Trin 3: Bound Når vi løser de nye LP-problemer i grenene:

- Hvis løsningen er infeasible (ingen løsning) → gren lukkes.
- Hvis løsningen er heltallig → vi gemmer den som en mulig kandidat.
- Hvis løsningen er bedre end de andre kendte løsninger, så fortsætter vi.
- Hvis løsningen er værre end en kendt løsning, kan grenen afskæres — vi behøver ikke undersøge videre.

Trin 4: Slut Når alle grene er undersøgt eller lukket, vælger vi den bedste heltallige løsning vi fandt.

Løsning

Trin 1: Relaxation
Problemet løses med koden:

```
import pulp as PLP
import numpy as np

# Minimeringsproblem
Model = PLP.LpProblem("EksJuni21", PLP.LpMaximize)

# Integer decision variables: x1 and x2
x = PLP.LpVariable.dicts("x", range(2), 0, None)

# Objective function)
Model += 8*x[0] + 7*x[1], "Objektfunktion"

# Constraints:
Model += 3*x[0] + 8*x[1] <= 48, "Begrænsning_1"
Model += 7*x[0] + 8*x[1] <= 56, "Begrænsning_2"
Model += 5*x[0] + 3*x[1] <= 30, "Begrænsning_3"
Model += 4*x[0] + x[1]   <= 22, "Begrænsning_4"

# Solve the optimization problem
Model.solve(PLP.PULP_CBC_CMD(msg=False))

# Print solution status
print("Status:", PLP.LpStatus[Model.status])

# Print nonzero decision variables
for v in Model.variables():
    if v.varValue > 0.01:
        print(v.name, "=", v.varValue)


# Print optimal objective function value
print("Obj. = ", PLP.value(Model.objective))
```

Og vi får en ikke heltalsløsning:

```
Status: Optimal
x_0 = 3.7894737
x_1 = 3.6842105
Obj. =  56.1052631
```

Trin 2: Hvis løsningen ikke er heltallig

Der branches på x1. Altså én gren med helltallig $x1 \leq 3$ og én gren med heltallig $x1 \geq 3$.

Resulterende Branch and Bound træ er:

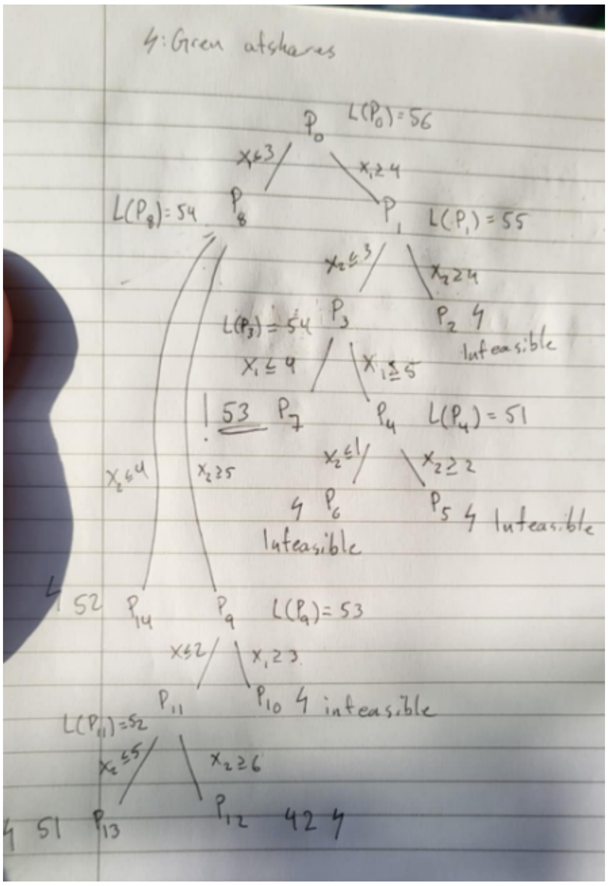


Figure 1: branch and Bound træ

P7 er løsningen med:

```
Status: Optimal
x_0 = 4.0
x_1 = 3.0
Obj. =  53.0
```

3.

```
# Minimeringsproblem
Model = PLP.LpProblem("EksJuni21", PLP.LpMaximize)

# Integer decision variables: x1 and x2
x = PLP.LpVariable.dicts("x", range(2), 0, None, PLP.LpInteger)

# Objective function)
Model += 8*x[0] + 7*x[1], "Objektfunktion"

# Constraints:
#P0 No PLP.LpInteger
Model += 3*x[0] + 8*x[1] <= 48, "Begrænsning_1"
Model += 7*x[0] + 8*x[1] <= 56, "Begrænsning_2"
Model += 5*x[0] + 3*x[1] <= 30, "Begrænsning_3"
Model += 4*x[0] + x[1]   <= 22, "Begrænsning_4"

# Solve the optimization problem
Model.solve(PLP.PULP_CBC_CMD(msg=False))

# Print solution status
print("Status:", PLP.LpStatus[Model.status])

# Print nonzero decision variables
for v in Model.variables():
    if v.varValue > 0.01:
        print(v.name, "=", v.varValue)


# Print optimal objective function value
print("Obj. = ", PLP.value(Model.objective))
```

Med samme løsning fra Branch and Bound:

```
Status: Optimal
x_0 = 4.0
x_1 = 3.0
Obj. =  53.0
```