Homework 2 (Vanderbei Exercise 1.1 : Standard Form)

In [1]:

```
import sys
!{sys.executable} -m pip install pulp
```

```
Requirement already satisfied: pulp in c:\users\muntakim\appdata\local\programs\python\p
ython38-32\lib\site-packages (2.4)
Requirement already satisfied: amply>=0.1.2 in c:\users\muntakim\appdata\local\programs
\python\python38-32\lib\site-packages (from pulp) (0.1.4)
Requirement already satisfied: pyparsing in c:\users\muntakim\appdata\local\programs\pyt
hon\python38-32\lib\site-packages (from amply>=0.1.2->pulp) (2.4.7)
Requirement already satisfied: docutils>=0.3 in c:\users\muntakim\appdata\local\programs
\python\python38-32\lib\site-packages (from amply>=0.1.2->pulp) (0.17.1)
```

In [2]:

```
import pulp
from pulp import *
```

In [3]:

```
"""
The Steel Company Band and Coil Production Problem (Standard Form)

Author: Muntakim Rahman 2020
"""
```

Out[3]:

```
'\nThe Steel Company Band and Coil Production Problem (Standard Form)\n\nAuthor: Muntaki
m Rahman 2020\n'
```

Textbook Problem

1.1 A steel company must decide how to allocate next week's time on a rolling mill, which is a machine that takes unfinished slabs of steel as input and can produce either of two semi-finished products: bands and coils. The mill's two products come off the rolling line at different rates:

Bands	200 tons/h
Coils	140 tons/h.

They also produce different profits:

Bands	\$25/ton
Coils	\$30/ton.

Based on currently booked orders, the following upper bounds are placed on the amount of each product to produce:

Bands	6,000 tons
Coils	4,000 tons.

1. INTRODUCTION

Given that there are 40 h of production time available this week, the problem is to decide how many tons of bands and how many tons of coils should be produced to yield the greatest profit. Formulate this problem as a linear programming problem. Can you solve this problem by inspection?

Steel Company Band and Coil Production Problem

$$
\begin{aligned}
\text{Maximize} \quad & 25 * bands + 30 * coils \\
\text{Subject to} \quad & 140 * bands + 200 * coils = 40 * 200 * 140 \\
& 0 \leq bands \leq 6000 \\
& 0 \leq coils \leq 4000
\end{aligned}
$$

Standard Inequality Form

$$
\begin{aligned}
\text{Maximize} \quad & -25 * bands - 30 * coils \\
\text{Subject to} \quad & -7 * bands - 10 * coils \leq -26000 \\
& 7 * bands + 10 * coils \leq 26000 \\
& bands \geq 0 \\
& coils \geq 0
\end{aligned}
$$

In [4]:

```
# Create a LP Maximization Problem.
LP_Prob_St = LpProblem('Steel_Company_Problem_St', LpMaximize)
```

In [5]:

```
# Create Prime Decision Variables.
bands__ = LpVariable(name = 'bands__', lowBound = 0)
coils__ = LpVariable(name = 'coils__', lowBound = 0)
```

In [6]:

```
# Add Objective Function to LP Problem.
LP_Prob_St += -25 * bands__ - 30 * coils__

# Now Add Constraints.
LP_Prob_St += -7 * bands__ - 10 * coils__ <= -26000
LP_Prob_St += 7 * bands__ + 10 * coils__ <= 26000
LP_Prob_St += bands__ >= 0
LP_Prob_St += coils__ >= 0
```

In [7]:

```
# Write the LP Problem to a File.
LP_Prob_St.writeLP('SteelCompanyProblemStandardForm.lp')

# Display the LP Problem.
print(LP_Prob_St)
```

```
Steel_Company_Problem_St:
MAXIMIZE
-25*bands__ + -30*coils__ + 0
SUBJECT TO
_C1: - 7 bands__ - 10 coils__ <= -26000

_C2: 7 bands__ + 10 coils__ <= 26000

_C3: bands__ >= 0

_C4: coils__ >= 0

VARIABLES
bands__ Continuous
```

```
coils__ Continuous
```

In [8]:

```
LP_Prob_St.solve()
```

Out[8]: 1

In [9]:

```
print(f'Status: {LpStatus[LP_Prob_St.status]}\n')

decision_variables__ = {}
for variable in LP_Prob_St.variables() :
    print(f'{variable.name} = {variable.varValue}')
    decision_variables__[variable.name] = variable.varValue
if (LpStatus[LP_Prob_St.status] == 'Optimal') :
    print(f'Optimal Value : z = {value(LP_Prob_St.objective)}')
else :
    print(f'No Optimal Value. Status Code : {value(LP_Prob_St.objective)}')
```

```
Status: Optimal

bands__ = 0.0
coils__ = 2600.0
Optimal Value : z = -78000.0
```

In [10]:

```
print('Prime Decision Variables : ', str(decision_variables__))
```

```
Prime Decision Variables :  {'bands__': 0.0, 'coils__': 2600.0}
```

In [11]:

```
bands = 6000 - decision_variables__['bands__']
coils = 4000 - decision_variables__['coils__']
```

In [12]:

```
print(f'Bands from Original Problem = {bands}')
print(f'Coils from Original Problem = {coils}')
```

```
Bands from Original Problem = 6000.0
Coils from Original Problem = 1400.0
```