Материал доступен по ссылке [https://proglib.io/p/lineynoe-programmirovanie-praktika-resheniya-zadach-optimizacii-na-python-2020-11-26 (https://proglib.io/p/lineynoe-programmirovanie-praktika-resheniya-zadach-optimizacii-na-python-2020-11-26)](https://proglib.io/p/lineynoe-programmirovanie-praktika-resheniya-zadach-optimizacii-na-python-2020-11-26)

In [1]:
```
from scipy.optimize import linprog
```

In [2]:
```
obj = [-1, -2]
#      ─┬  ─┬
#       │   └┤ Коэффициент для y
#       └────┤ Коэффициент для x

lhs_ineq = [[ 2,  1],  # левая сторона красного неравенства
            [-4,  5],  # левая сторона синего неравенства
            [ 1, -2]]  # левая сторона желтого неравенства

rhs_ineq = [20,  # правая сторона красного неравенства
            10,  # правая сторона синего неравенства
             2]  # правая сторона желтого неравенства

lhs_eq = [[-1, 5]]  # левая сторона зеленого равенства
rhs_eq = [15]       # правая сторона зеленого равенства
```

In [3]:
```
bnd = [(0, float("inf")),  # Границы x
       (0, float("inf"))]  # Границы y
```

In [4]:
```
opt = linprog(c=obj, A_ub=lhs_ineq, b_ub=rhs_ineq,
              A_eq=lhs_eq, b_eq=rhs_eq, bounds=bnd,
              method="revised simplex")

opt
```

```
C:\Users\pc\AppData\Local\Temp\ipykernel_4000\1795202355.py:1: DeprecationWar
ning: `method='revised simplex'` is deprecated and will be removed in SciPy
1.11.0. Please use one of the HiGHS solvers (e.g. `method='highs'`) in new co
de.
  opt = linprog(c=obj, A_ub=lhs_ineq, b_ub=rhs_ineq,
```

Out[4]:
```
     con: array([0.])
     fun: -16.818181818181817
 message: 'Optimization terminated successfully.'
     nit: 3
   slack: array([ 0.        , 18.18181818,  3.36363636])
  status: 0
 success: True
       x: array([7.72727273, 4.54545455])
```

In [5]:
```
opt = linprog(c=obj, A_ub=lhs_ineq, b_ub=rhs_ineq, bounds=bnd,
              method="revised simplex")

opt
```

```
C:\Users\pc\AppData\Local\Temp\ipykernel_4000\3668097702.py:1: DeprecationWar
ning: `method='revised simplex'` is deprecated and will be removed in SciPy
1.11.0. Please use one of the HiGHS solvers (e.g. `method='highs'`) in new co
de.
  opt = linprog(c=obj, A_ub=lhs_ineq, b_ub=rhs_ineq, bounds=bnd,
```

Out[5]:
```
     con: array([], dtype=float64)
     fun: -20.714285714285715
 message: 'Optimization terminated successfully.'
     nit: 2
   slack: array([0.        , 0.        , 9.85714286])
  status: 0
 success: True
       x: array([6.42857143, 7.14285714])
```

In [6]:
```
obj = [-20, -12, -40, -25]

lhs_ineq = [[1, 1, 1, 1],  # Рабочая сила
            [3, 2, 1, 0],  # Материал A
            [0, 1, 2, 3]]  # Материал B

rhs_ineq = [ 50,  # Рабочая сила
            100,  # Материал A
             90]  # Материал B

opt = linprog(c=obj, A_ub=lhs_ineq, b_ub=rhs_ineq,
              method="revised simplex")

opt
```

```
C:\Users\pc\AppData\Local\Temp\ipykernel_4000\1541892138.py:11: DeprecationWa
rning: `method='revised simplex'` is deprecated and will be removed in SciPy
1.11.0. Please use one of the HiGHS solvers (e.g. `method='highs'`) in new co
de.
  opt = linprog(c=obj, A_ub=lhs_ineq, b_ub=rhs_ineq,
```

Out[6]:
```
     con: array([], dtype=float64)
     fun: -1900.0
 message: 'Optimization terminated successfully.'
     nit: 2
   slack: array([ 0., 40.,  0.])
  status: 0
 success: True
       x: array([ 5.,  0., 45.,  0.])
```

In [11]:
```
!pip install -U pulp
```

```
Collecting pulp
  Using cached PuLP-2.9.0-py3-none-any.whl (17.7 MB)
Installing collected packages: pulp
Successfully installed pulp-2.9.0

WARNING: Ignoring invalid distribution -cipy (c:\users\pc\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cipy (c:\users\pc\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cipy (c:\users\pc\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cipy (c:\users\pc\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cipy (c:\users\pc\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cipy (c:\users\pc\anaconda3\lib\site-packages)
WARNING: Ignoring invalid distribution -cipy (c:\users\pc\anaconda3\lib\site-packages)
```

In [12]:
```
from pulp import LpMaximize, LpProblem, LpStatus, lpSum, LpVariable
```

In [14]:
```
model = LpProblem(name="small-problem", sense=LpMaximize)
```

In [15]:
```
x = LpVariable(name="x", lowBound=0)
y = LpVariable(name="y", lowBound=0)
```

In [16]:
```
expression = 2 * x + 4 * y
print(type(expression))
constraint = 2 * x + 4 * y >= 8
print(type(constraint))
```

```
<class 'pulp.pulp.LpAffineExpression'>
<class 'pulp.pulp.LpConstraint'>
```

In [17]:
```
model += (2 * x + y <= 20, "red_constraint")
model += (4 * x - 5 * y >= -10, "blue_constraint")
model += (-x + 2 * y >= -2, "yellow_constraint")
model += (-x + 5 * y == 15, "green_constraint")
```

In [18]:
```
obj_func = x + 2 * y
model += obj_func
```

In [19]:
```
model
```

Out[19]:
```
small-problem:
MAXIMIZE
1*x + 2*y + 0
SUBJECT TO
red_constraint: 2 x + y <= 20

blue_constraint: 4 x - 5 y >= -10

yellow_constraint: - x + 2 y >= -2

green_constraint: - x + 5 y = 15

VARIABLES
x Continuous
y Continuous
```

In [20]:
```
status = model.solve()
```

In [21]:
```
print(f"status: {model.status}, {LpStatus[model.status]}")

print(f"objective: {model.objective.value()}")

for var in model.variables():
    print(f"{var.name}: {var.value()}")

for name, constraint in model.constraints.items():
    print(f"{name}: {constraint.value()}")
```

```
status: 1, Optimal
objective: 16.8181817
x: 7.7272727
y: 4.5454545
red_constraint: -9.99999993922529e-08
blue_constraint: 18.181818300000003
yellow_constraint: 3.3636362999999996
green_constraint: -2.0000000233721948e-07
```

In [22]:
```
# Создаем модель
model = LpProblem(name="small-problem", sense=LpMaximize)

# Инициализируем переменные решения: x - целое число, y меняется непрерывно
x = LpVariable(name="x", lowBound=0, cat="Integer")
y = LpVariable(name="y", lowBound=0)

# Добавляем ограничения
model += (2 * x + y <= 20, "red_constraint")
model += (4 * x - 5 * y >= -10, "blue_constraint")
model += (-x + 2 * y >= -2, "yellow_constraint")
model += (-x + 5 * y == 15, "green_constraint")

# Добавляем целевую функцию
# Вариант добавления через lpSum
model += lpSum([x, 2 * y])

# Решаем задачу оптимизации
status = model.solve()
```

In [23]:
```
print(f"status: {model.status}, {LpStatus[model.status]}")

print(f"objective: {model.objective.value()}")

for var in model.variables():
    print(f"{var.name}: {var.value()}")

for name, constraint in model.constraints.items():
    print(f"{name}: {constraint.value()}")
```

```
status: 1, Optimal
objective: 15.8
x: 7.0
y: 4.4
red_constraint: -1.5999999999999996
blue_constraint: 16.0
yellow_constraint: 3.8000000000000007
green_constraint: 0.0
```

In [25]:
```
# Определяем модель
model = LpProblem(name="resource-allocation", sense=LpMaximize)

# Описываем переменные
x = {i: LpVariable(name=f"x{i}", lowBound=0) for i in range(1, 5)}

# Добавляем ограничения
model += (lpSum(x.values()) <= 50, "manpower")
model += (3 * x[1] + 2 * x[2] + x[3] <= 100, "material_a")
model += (x[2] + 2 * x[3] + 3 * x[4] <= 90, "material_b")

# Описываем цель
model += 20 * x[1] + 12 * x[2] + 40 * x[3] + 25 * x[4]

# Решаем задачу оптимизации
status = model.solve()

# Выводим результаты решения
print(f"status: {model.status}, {LpStatus[model.status]}")
print(f"objective: {model.objective.value()}")

for var in x.values():
    print(f"{var.name}: {var.value()}")

for name, constraint in model.constraints.items():
    print(f"{name}: {constraint.value()}")
```

```
status: 1, Optimal
objective: 1900.0
x1: 5.0
x2: 0.0
x3: 45.0
x4: 0.0
manpower: 0.0
material_a: -40.0
material_b: 0.0
```

In [26]:
```
model = LpProblem(name="resource-allocation", sense=LpMaximize)

x = {i: LpVariable(name=f"x{i}", lowBound=0) for i in range(1, 5)}
y = {i: LpVariable(name=f"y{i}", cat="Binary") for i in (1, 3)}

model += (lpSum(x.values()) <= 50, "manpower")
model += (3 * x[1] + 2 * x[2] + x[3] <= 100, "material_a")
model += (x[2] + 2 * x[3] + 3 * x[4] <= 90, "material_b")

M = 100
model += (x[1] <= y[1] * M, "x1_constraint")
model += (x[3] <= y[3] * M, "x3_constraint")
model += (y[1] + y[3] <= 1, "y_constraint")

model += 20 * x[1] + 12 * x[2] + 40 * x[3] + 25 * x[4]

status = model.solve()

print(f"status: {model.status}, {LpStatus[model.status]}")
print(f"objective: {model.objective.value()}")

for var in model.variables():
    print(f"{var.name}: {var.value()}")

for name, constraint in model.constraints.items():
    print(f"{name}: {constraint.value()}")
```

```
status: 1, Optimal
objective: 1800.0
x1: 0.0
x2: 0.0
x3: 45.0
x4: 0.0
y1: 0.0
y3: 1.0
manpower: -5.0
material_a: -55.0
material_b: 0.0
x1_constraint: 0.0
x3_constraint: -55.0
y_constraint: 0.0
```

In [ ]: