****

**MINISTERUL EDUCAȚIEI ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Informatică și Ingineria Sistemelor**

**Baîmb Eusebiu**

**Raport**

**pentru lucrarea de laborator Nr. 1**

***la cursul de “Metode Matematice și Optimizări”***

**A verificat,** asistent universitar,

Oleg Buldumac

**A elaborat**, Barîmb Eusebiu

**Chișinău – 2022**

**CUPRINS**

[INTRODUCERE 3](#_Toc120207212)

[1. DESCRIEREA PROBLEMILOR 3](#_Toc120207213)

[2. PROBLEMA REZOLVATĂ ÎN EXCELL 3](#_Toc120207214)

[3. PROBLEMA REZILVATĂ ÎN PYTHON 5](#_Toc120207215)

[4. REZULTATELE ÎN URMA RULĂRI PROGRAMULUI 6](#_Toc120207216)

[CONCLUZIE 8](#_Toc120207217)

INTRODUCERE

În optimizarea matematică, o problemă de optimizare liniară necesită minimizarea unei funcții liniare pe un poliedru convex . Funcția care este minimizată, precum și constrângerile sunt descrise de funcții liniare, de unde și numele dat acestor probleme. Optimizare liniara (LO) este disciplina care studiază problemele. Este denumit și numele de programare liniară, termen introdus de George Dantzig în jurul anului 1947, dar acest nume tinde să fie abandonat din cauza unei posibile confuzii cu noțiunea de programare pe computer.

* 1. DESCRIEREA PROBLEMILOR

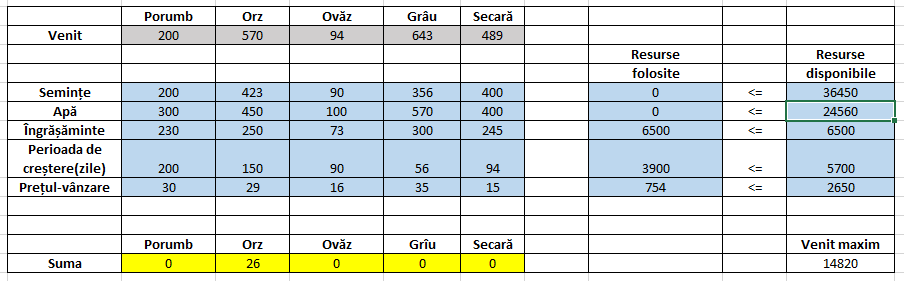
Prima problemă este o problemă de producție, are ca scop maximizarea venitului ce poate surveni în urma cultivarea plantelor de porumb, orz, ovăz, grâu și secară. Cu alte cuvinte se ia în calcul mulțimea activităților necesare pentru producerea acestor cereale resurselor necare, de asemenea și mulțimea resurselor disponibile.

A doua problemă este o problemă de minimizare costurilor pentru producerea unor anumite bucate, iar pentru a minimiza costurile pentru producere trebuie să cunoaștem caloriile pe care ele le au și deja în baza lor să facem calculele necesare pentru optimizările dorite.

* 1. PROBLEMA REZOLVATĂ ÎN EXCELL

**Problema de maximizare**

În figura 1, putem observa datele de intrare pentru problema noastră, unde avem x, constrângerile și deja datele pe baza căror se vor face calculele.



**Figura 1.** Modelul în excell

Iar în figura 2, putem observat calculele pe care au fost realizate cu ajutorul la solver și soluțiile optimile pe care a reușit să le calculeze.

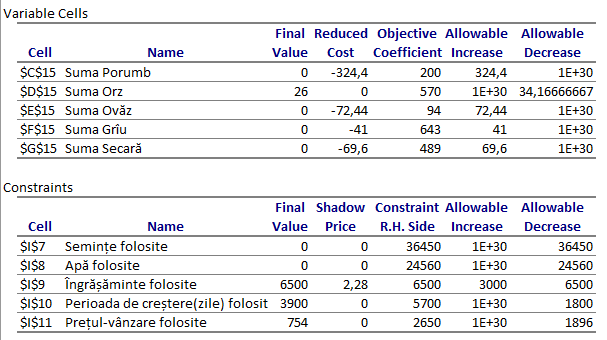
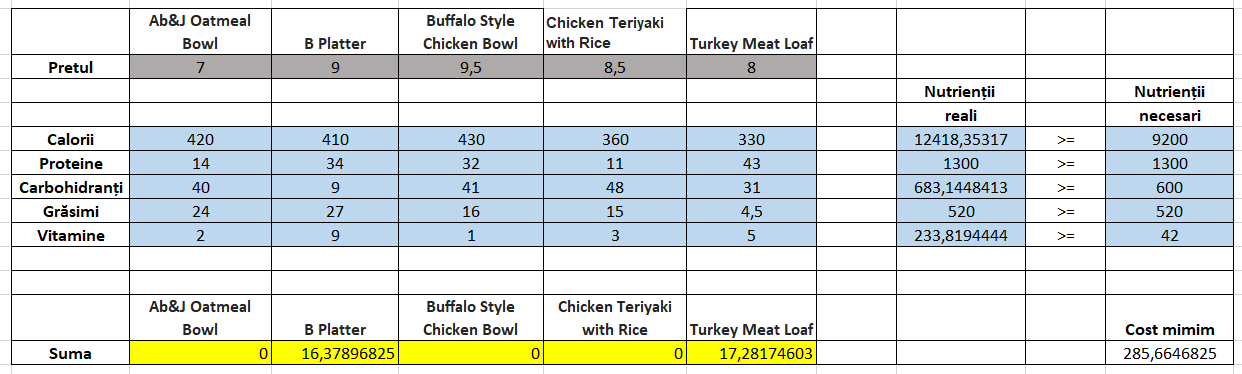


Figura 2. Output-ul la solver

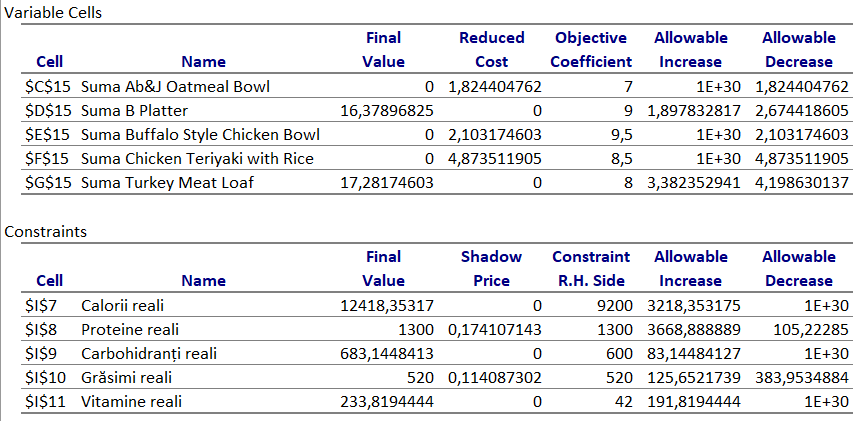
**Problema de minimizarea**

În figura 3, sunt prezentate datele pentru minimizarea a costurilor a bucatelor care se prepară pentru a fi comercializate.

****

**Figura 3.** Date din excell

Iar în figura 4, putem observa date de ieșire a modelul care a fost calculat cu ajutorul la solver.



**Figura 4.** Calculele finale

* 1. PROBLEMA REZILVATĂ ÎN PYTHON

**Listingul programul pentru problema unu.**

# import the libraries

from ortools.linear\_solver import pywraplp

import numpy as np

# Initilize the Solver

# We are using the Mixed integer Programming

# Variable can take only integer value

solver = pywraplp.Solver.CreateSolver('SCIP')

infinity = solver.infinity()

Porumb = solver.IntVar(0, solver.infinity(), 'Porumb')

Orz = solver.IntVar(0, solver.infinity(), 'Orz')

Ovaz = solver.IntVar(0, solver.infinity(), 'Ovaz')

Grau = solver.IntVar(0, solver.infinity(), 'Grau')

Secara = solver.IntVar(0, solver.infinity(), 'Secara')

# create our constraints

solver.Add(Porumb \* 200 + Orz \* 423 + Ovaz \* 90 + Grau \* 356 + Secara \* 400 <= 36450, "Semințe")

solver.Add(Porumb \* 300 + Orz \* 450 + Ovaz \* 100 + Grau \* 570 + Secara \* 400 <= 24560, "Apă")

solver.Add(Porumb \* 230 + Orz \* 250 + Ovaz \* 73 + Grau \* 300 + Secara \* 245 <= 6500, "Îngrășăminte")

solver.Add(Porumb \* 200 + Orz \* 150 + Ovaz \* 90 + Grau \* 56 + Secara \* 94 <= 5700, "Perioada de creștere")

solver.Add(Porumb \* 30 + Orz \* 29 + Ovaz \* 16 + Grau \* 16 + Secara \* 15 <= 2650, "Prețul-Vânzare")

# the income of the plants

solver.Maximize(Porumb \* 200 + Orz \* 570 + Ovaz \* 94 + Grau \* 643 + Secara \* 489)

# Solving the Equation

status = solver.Solve()

#print(status)

# display the final data

if status == pywraplp.Solver.OPTIMAL or status==pywraplp.Solver.FEASIBLE:

print('Net Profit Amount:',solver.Objective().Value())

print('Production Plan are below : ')

print('Porumb: ',int(Porumb.solution\_value()),'Unit')

print('Orz: ',int(Orz.solution\_value()),'Unit')

print('Ovaz: ',int(Ovaz.solution\_value()),'Unit')

print('Grau: ',int(Grau.solution\_value()),'Unit')

print('Secara: ',int(Secara.solution\_value()),'Unit')

**Listingul programul pentru problema doi.**

# import the libraries

from ortools.linear\_solver import pywraplp

import numpy as np

# Initilize the Solver

# We are using the Mixed integer Programming

# Variable can take only integer value

solver = pywraplp.Solver.CreateSolver('SCIP')

infinity = solver.infinity()

oatmeal = solver.IntVar(0, solver.infinity(), 'Ab&J Oatmeal Bowl')

platter = solver.IntVar(0, solver.infinity(), 'B Platter')

buffalo = solver.IntVar(0, solver.infinity(), 'Buffalo Style Chicken Bowl')

teriyaki = solver.IntVar(0, solver.infinity(), 'Chicken Teriyaki with Rice')

meat\_loaf = solver.IntVar(0, solver.infinity(), 'Turkey Meat Loaf')

# create our constraints

solver.Add(oatmeal \* 420 + platter \* 410 + buffalo \* 430 + teriyaki \* 360 + meat\_loaf \* 330 >= 9200, "Calorii")

solver.Add(oatmeal \* 14 + platter \* 34 + buffalo \* 32 + teriyaki \* 11 + meat\_loaf \* 43 >= 1300, "Proteine")

solver.Add(oatmeal \* 40 + platter \* 9 + buffalo \* 41 + teriyaki \* 48 + meat\_loaf \* 31 >= 600, "Carbohidranți")

solver.Add(oatmeal \* 24 + platter \* 27 + buffalo \* 16 + teriyaki \* 15 + meat\_loaf \* 4.5 >= 520, "Grăsimi")

solver.Add(oatmeal \* 2 + platter \* 9 + buffalo \* 1 + teriyaki \* 3 + meat\_loaf \* 5 >= 42, "Vitamine")

# the price of the food

solver.Minimize(oatmeal \* 7 + platter \* 9 + buffalo \* 9.5 + teriyaki \* 8.5 + meat\_loaf \* 8)

# Solving the Equation

status = solver.Solve()

#print(status)

# display the final data

if status == pywraplp.Solver.OPTIMAL or status == pywraplp.Solver.FEASIBLE:

print('Net Profit Amount:', solver.Objective().Value())

print('Plan are below: ')

print('Ab&J Oatmeal Bowl:', int(oatmeal.solution\_value()), 'Unit')

print('B Platter:', int(platter.solution\_value()), 'Unit')

print('Buffalo Style Chicken Bowl:', int(buffalo.solution\_value()), 'Unit')

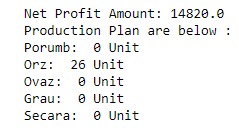
print('Chicken Teriyaki with Rice:', int(teriyaki.solution\_value()), 'Unit')

print('Turkey Meat Loaf:', int(meat\_loaf.solution\_value()), 'Unit')

* 1. REZULTATELE ÎN URMA RULĂRI PROGRAMULUI

**Rezultatul pentru programul unu.**

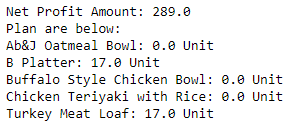
În figura 5, este prezentat output-ul programului pentru prima problemă, iar dacă să facem o comparație cu rezultatele din excell putem observa că datele sunt asemănătoare.

****

**Figura 5.** Rezultatele maximizării

**Rezultatul pentru programul unu.**

În figura 6, este prezentat output-ul programului pentru a doua problemă, iar dacă să facem o comparație cu rezultatele din excell putem observa că majoritate datelor sunt asemănătoare, nu sunt fix în fix din motiv că datele care au fost rezolvate în code sunt de timp integer.



**Figura 6.**  Rezultatele minimizări

CONCLUZIE

În urma efectuări acestui laborator am creat modele de maximizare și minimizare pentru soluționarea problemelor propuse. De asemenea am făcut cunoștință mai profund despre funcționalul excellului și cum anume funcționează solver. Tot odată destul de important consider faptul că am rezolvat probleme și în limbajul de programare python.