DAMA 50

Written Assignment VI

Submitted by:

Panagiotis Paltsokas

ID: std163861

[DAMA 50] Written Assignment 6

Panagiotis Paltsokas - std163861

Problem 7

Solve the following problem using `sagemath` :
A pension fund of \$42 million is available for investment across treasury notes, bonds, and stocks. Administration guidelines mandate a minimum investment of \$3 million in each category, with at least half of the total funds allocated to treasury notes and bonds. Additionally, the investment in bonds must not exceed twice the investment in treasury notes. The annual yields are 7% for treasury notes, 9% for bonds, and 10% for stocks. Find how the allocation of funds should be optimized to maximize returns using the following steps:
(a) Formulate this problem as a linear programming problem and write the resulting equations (in a `sagemath` markdown cell).
(b) Solve the problem using the appropriate class of `sagemath` functions.

In [1]:
```
restore()
%display latex
```

(a)

Let x be the investment on treasury notes, y the investment on bonds and z the investment on stocks

$$\begin{array}{ll} \text{maximize} & 0.07x + 0.09y + 0.1z \\ \text{such that} & x + y + z \leq 42 \\ & x + y \geq 21 \\ & 2x - y \geq 0 \\ & x, y, z \geq 3 \end{array}$$

#1 Create a linear programming model using MixedIntegerLinearProgram() with parameter maximization = True, since we want to maximize the returns. \ #2 Define a new variable dictionary v where variables are real and nonnegative.\ #3 Define the variables used in the linear program, extracted from the dictionary v. Each variable represents a decision variable in the model whose optimal value is sought under the given constraints.\ #4 Set the objective function for the linear program.\ #5 Add the constraints as defined from the problem.

In [2]:
```
p=MixedIntegerLinearProgram(maximization=True) #1
v = p.new_variable(real=True, nonnegative=True) #2
x,y,z = v['x'], v['y'], v['z'] #3
p.set_objective(0.07*x+0.09*y+0.1*z) #4
p.add_constraint(x+y+z<=42) #5
p.add_constraint(x+y>=21)
p.add_constraint(2*x-y>=0)
p.add_constraint(x>=3)
p.add_constraint(y>=3)
p.add_constraint(z>=3)
```

(b)

Execute the optimization and solve the problem.

```
In [3]: p.solve()
```

```
Out[3]: 3.85
```

Retrieve the optimal values

```
In [4]: opt_x, opt_y, opt_z = p.get_values(x), p.get_values(y), p.get_values(z)
```

Print the results

```
In [5]: print("Optimal investment in treasury notes (in million $):", opt_x)
        print("Optimal investment in bonds (in million $):", opt_y)
        print("Optimal investment in stocks (in million $):", opt_z)
```

```
Optimal investment in treasury notes (in million $): 7.0
Optimal investment in bonds (in million $): 14.0
Optimal investment in stocks (in million $): 21.0
```

[DAMA 50] Written Assignment 6

Panagiotis Paltsokas - std163861

Problem 8

Given the function $f(x_1, x_2) = e^{x_1}(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$ and working with **sagemath**:
(a) Find analytically and classify (e.g. minimum,maximum,saddle point) the stationary points (at finite x_1, x_2).
(b) Plot the above function using a contour plot (with appropriate parameter bounds) and depict the position of the stationary points. Use axes labels, contour labels and an appropriate number of contours.
(c) Working numerically, find the global minimum (at finite x_1, x_2) starting from the point $x_0 = (-1, 2)$ and utilizing two different algorithms (one using the derivatives and one not using them). Comment on the results compared to the results of (a).

In [1]:
```
restore()
%display latex
```

In [2]:
```
var('x1','x2')
f=e^x1*(4*x1^2+2*x2^2+4*x1*x2+2*x2+1)
show(f)
```

$$\left(4\,x_1^2 + 4\,x_1 x_2 + 2\,x_2^2 + 2\,x_2 + 1\right)e^{x_1}$$

(a)

Calculate the partial derivatives $\frac{\partial f}{\partial x_1}$ and $\frac{\partial f}{\partial x_2}$

In [3]:
```
fx1=diff(f,x1)
fx2=diff(f,x2)
show(fx1)
show(fx2)
```

$$\left(4\,x_1^2 + 4\,x_1 x_2 + 2\,x_2^2 + 2\,x_2 + 1\right)e^{x_1} + 4\,(2\,x_1 + x_2)e^{x_1}$$

$$2\,(2\,x_1 + 2\,x_2 + 1)e^{x_1}$$

Solve the equations $\frac{\partial f}{\partial x_1} = 0$ and $\frac{\partial f}{\partial x_2} = 0$ to find the abscissa of the stationary points

In [4]:
```
stationary_points = solve([fx1==0,fx2==0],x1,x2);stationary_points
```

Out[4]:

$$\left[\left[x_1 = \left(\frac{1}{2}\right), x_2 = (-1)\right], \left[x_1 = \left(-\frac{3}{2}\right), x_2 = 1\right]\right]$$

Calculate the stationary points' coordinates

In [5]:
```
point1 = (stationary_points[0][0].rhs(),stationary_points[0][1].rhs(),f.subs(stationary_points[0][0],stationary_points[0][1]))
point2 = (stationary_points[1][0].rhs(),stationary_points[1][1].rhs(),round(n(f.subs(stationary_points[1][0],stationary_points[1][1])),2))
show(point1,point2)
```

$$\left(\frac{1}{2}, -1, 0\right) \left(-\frac{3}{2}, 1, 1.79\right)$$

Store the points into a list

In [6]:
```
points = [point1, point2];points
```

Out[6]:

$$\left[\left(\frac{1}{2}, -1, 0\right), \left(-\frac{3}{2}, 1, 1.79\right)\right]$$

#1 Compute the Hessian matrix \ #2 Iterate over each stationary point in the list and \ #3 Substitute the stationary point into the Hessian matrix \ #4 Compute the eigenvalues of the Hessian matrix at that specific stationary point \ #5 Check the sign of the eigenvalues to determine the type of the stationary point. \ If all eigenvalues are positive that indicates a local minimum\ If all eigenvalues are negative that indicates a local maximum\ If eigenvalues have mixed signs, that indicates a saddle point

In [7]:
```
H=f.hessian();H #1
for s_point in stationary_points: #2
    hess_eval = H.subs(s_point) #3
    eigenvalues = hess_eval.eigenvalues() #4
    if all(e > 0 for e in eigenvalues): #5
        print(f"Point {s_point[0].rhs(),s_point[1].rhs(),f(s_point[0].rhs(),s_point[1].rhs())} is a local minimum.")
    elif all(e < 0 for e in eigenvalues):
        print(f"Point {s_point[0].rhs(),s_point[1].rhs(),f(s_point[0].rhs(),s_point[1].rhs())} is a local maximum.")
    else:
        print(f"Point {s_point[0].rhs(),s_point[1].rhs(),round(n(f(s_point[0].rhs(),s_point[1].rhs())),3)} is a saddle point.")
```

```
Point (1/2, -1, 0) is a local minimum.
Point (-3/2, 1, 1.785) is a saddle point.
```

(b)

Create the contour plot adding a red point for the saddle point and a green one for the local minimum, and combine the plot with the points.

In [8]:
```
cplot = contour_plot(f, (x1, -6, 2), (x2, -3, 3), fill=False, labels=True, contours=[0.5,1,1.5,2,2.5,3,3.5], \
                     axes_labels=['$x_1$', '$x_2$'], plot_points=100)\
+ point([-3/2, 1], size=40, color='red', zorder=3)\
+ point([1/2,-1], size=40, color='green', zorder=3)
```

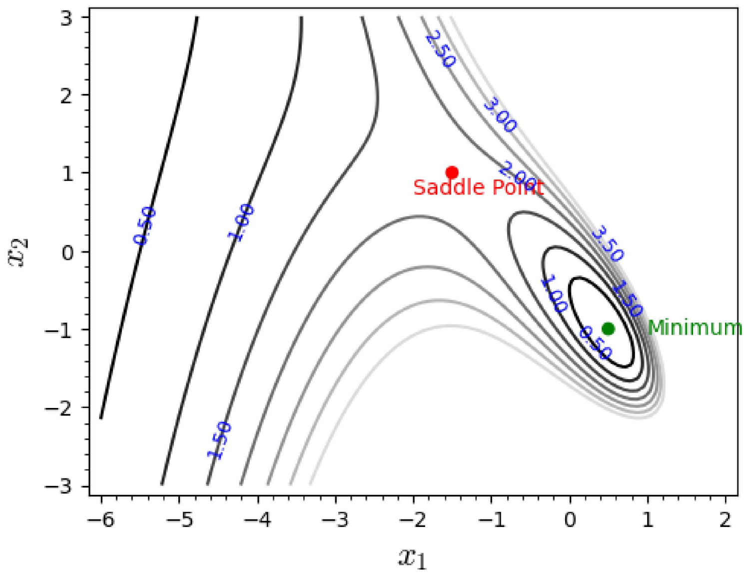
Add text labels to the plot to improve its clarity and make it easier to interpret.

In [9]:
```
legend_text_red = text('Saddle Point', (-2, 0.8), color='red', horizontal_alignment='left', vertical_alignment='center')
legend_text_green = text('Minimum', (1, -1), color='green', horizontal_alignment='left', vertical_alignment='center')
```

Combine the plot with the text annotations and display

In [10]:
```
cplot+legend_text_red+legend_text_green
```

Out[10]:



(c)

Initial guess for the minimum

In [11]:
```
x0=[-1,2]
```

Perform minimization using the Non-derivative based Simplex algorithm

In [12]:
```
min_simplex = minimize(f(x1,x2),x0,verbose=True, algorithm='simplex');min_simplex
```

```
Optimization terminated successfully.
         Current function value: 0.000000
         Iterations: 60
         Function evaluations: 112
```

Out[12]: $(0.49999911066660385, -0.9999631034560157)$

In [13]:
```
round(min_simplex[0],3),round(min_simplex[1],3)
```

Out[13]: $(0.5, -1.0)$

Perform minimization using the derivative based BFGS algorithm (Sagemath default)

In [14]:
```
min_bfgs = minimize(f(x1,x2),x0, verbose=True, algorithm='bfgs');min_bfgs
```

```
Optimization terminated successfully.
         Current function value: 0.000008
         Iterations: 14
         Function evaluations: 18
         Gradient evaluations: 18
```

Out[14]: $(-18.54624861145268, 8.751212526386928)$

In [15]:
```
round(min_bfgs[0],3),round(min_bfgs[1],3)
```

Out[15]: $(-18.546, 8.751)$

The Simplex method, even though significally slower (60 iterations), shows high accuracy by converging very close to the analytical minimum. In contrast, the much faster BFGS method (14 iterations) falls way off in this instance, which might suggest issues with either the initial conditions, the nature of the function, or the specific implementation of the algorithm. Derivative-based methods like BFGS are considered efficient due to their use of gradient information to quickly navigate the function's topology. However, if gradients are poorly estimated or if the function has discontinuities or sharp curvature, these methods run the risk of converging to incorrect values. The Simplex method, not requiring derivative information, avoids complications from inaccurate gradient computations and can be more robust across a diverse range of functions.

[DAMA 50] Written Assignment 6

Panagiotis Paltsokas - std163861

Problem 9

Using by hand calculation, minimize $f(x, y) = \frac{1}{2}x^2 + 2y^2$ subject to the constraint $x + 3y = b$, where b is a real number, following the steps below
(a) What is the Lagrangian $L(x, y, \lambda)$ of the problem?
(b) What are the equations that determine the minimum ?
(c) Solve the equations in (b) to find the minimum (x^*, y^*) and the Lagrange multiplier λ.
(d) Find the value of the function f at the minimum.

(a)

Let g be the function $g(x) = x + 3y - b$

The Lagrangian $L(x, y, \lambda)$ of the problem is

$$L(x,y,\lambda) = f(x,y) + \lambda \cdot g(x) = \frac{1}{2}x^2 + 2y^2 + \lambda \cdot (x + 3y - b)$$

(b)

First, we compute the gradient.

$$\frac{dL}{dx} = x + \lambda \qquad \frac{dL}{dy} = 4y + 3\lambda \qquad \frac{dL}{d\lambda} = x + 3y - b$$

To find the equations that determine the minimum we have to set the gradient to zero, which yields

$$\frac{dL}{dx} = 0 \quad \Rightarrow \quad x - \lambda = 0$$
$$\frac{dL}{dy} = 0 \quad \Rightarrow \quad 4y + 3\lambda = 0$$
$$\frac{dL}{d\lambda} = 0 \quad \Rightarrow \quad x + 3y - b = 0$$

(c)

Solving those equations, we get

$$x + \lambda = 0 \quad \Rightarrow \quad x = -\lambda$$

$$4y = -3\lambda \quad \Rightarrow \quad y = \frac{-3\lambda}{4}$$

$$x + 3y - b = 0 \quad \Rightarrow \quad -\lambda - \frac{9\lambda}{4} - b = 0 \quad \Rightarrow \quad -\frac{13\lambda}{4} = b \quad \Rightarrow \quad \lambda = -\frac{4b}{13}$$

Therefore, the minimum is $\left(x^*, y^*\right)$ where $x^* = \frac{4b}{13}$ and $y^* = \frac{-3}{4}\cdot\left(\frac{-4b}{13}\right) = \frac{3b}{13}$.

(d) The value of the function f at the minimum is

$$f\left(x^*, y^*\right) = f\left(\frac{4b}{13}, \frac{3b}{13}\right) = \frac{1}{2}\cdot\left(\frac{4b}{13}\right)^2 + 2\cdot\left(\frac{3b}{13}\right)^2 = \frac{1}{2}\cdot\frac{16b^2}{169} + 2\cdot\frac{9b^2}{169} = \frac{8b^2}{169} + \frac{18b^2}{169} = \frac{26b^2}{169} = \frac{2b^2}{13}$$

[DAMA 50] Written Assignment 6

Panagiotis Paltsokas - std163861

Problem 10

Using by hand calculation, show that the LogSumExp function in $\mathbb{R}^2$

$$LSE(x_1, x_2) = \ln(e^{x_1} + e^{x_2})$$

is convex using the following steps:
(a) Calculate the Hessian matrix H. (b) Calculate the eigenvalues of H. (c) Determine whether the function is convex or not.

(a)

First, lets calculate the Hessian matrix. We need to compute the partial derivatives of $f(x_1, x_2) = \ln(e^{x_1} + e^{x_2})$

$$\frac{df}{dx_1} = \frac{1}{e^{x_1} + e^{x_2}} \cdot e^{x_1} \Rightarrow \frac{df}{dx_1} > 0 \text{ for all } (x_1, x_2) \in \mathbb{R}^2$$

$$\frac{df}{dx_2} = \frac{1}{e^{x_1} + e^{x_2}} \cdot e^{x_2} \Rightarrow \frac{df}{dx_2} > 0 \text{ for all } (x_1, x_2) \in \mathbb{R}^2$$

$$\frac{d^2f}{dx_1^{\,2}} = \frac{e^{x_1} \cdot (e^{x_1} + e^{x_2}) - e^{x_1} \cdot e^{x_1}}{(e^{x_1} + e^{x_2})^2} = \frac{e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2} \Rightarrow \frac{d^2f}{dx_1^{\,2}} > 0 \text{ for all } (x_1, x_2) \in \mathbb{R}^2$$

$$\frac{d^2f}{dx_2^{\,2}} = \frac{e^{x_2} \cdot (e^{x_1} + e^{x_2}) - e^{x_2} \cdot e^{x_2}}{(e^{x_1} + e^{x_2})^2} = \frac{e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2} \Rightarrow \frac{d^2f}{dx_2^{\,2}} > 0 \text{ for all } (x_1, x_2) \in \mathbb{R}^2$$

$$\frac{d^2f}{dx_1dx_2} = \frac{-e^{x_1}}{(e^{x_1} + e^{x_2})^2} \cdot e^{x_2} = \frac{-e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2} \Rightarrow \frac{d^2f}{dx_1dx_2} < 0 \text{ for all } (x_1, x_2) \in \mathbb{R}^2$$

$$\frac{d^2f}{dx_2dx_1} = \frac{-e^{x_2}}{(e^{x_1} + e^{x_2})^2} \cdot e^{x_1} = \frac{-e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2} \Rightarrow \frac{d^2f}{dx_2dx_1} < 0 \text{ for all } (x_1, x_2) \in \mathbb{R}^2$$

The Hessian matrix H is:

$$H = \begin{pmatrix} \frac{d^2f}{dx_1^{\,2}} & \frac{d^2f}{dx_1dx_2} \\ \frac{d^2f}{dx_2dx_1} & \frac{d^2f}{dx_2^{\,2}} \end{pmatrix} = \begin{pmatrix} \frac{e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2} & \frac{-e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2} \\ \frac{-e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2} & \frac{e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2} \end{pmatrix} = \frac{e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2} \cdot \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \kappa \cdot A$$

where matrix $A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ and $\kappa = \frac{e^{x_1+x_2}}{(e^{x_1} + e^{x_2})^2}$ a scalar.

(b)

We can easily find the <u>eigenvalues of matrix A</u>:

$$\det(A-\lambda I)=\begin{vmatrix}1-\lambda & -1\\ -1 & 1-\lambda\end{vmatrix}=(1-\lambda)^2-(-1)(-1)=1-2\lambda+\lambda^2-1=\lambda^2-2\lambda=\lambda(\lambda-2)$$

$$\det(A-\lambda I)=0 \;\Leftrightarrow\; \lambda(\lambda-2)=0 \;\Leftrightarrow\; \lambda=0 \quad \text{or} \quad \lambda=2$$

Since $H=\kappa\cdot A$, the eigenvalues of H are the eigenvalues of A, multiplied by the scalar κ.

<u>Eigenvalues of H:</u>

$$\lambda_1=\frac{e^{x_1+x_2}}{\left(e^{x_1}+e^{x_2}\right)^2}\cdot 0 \;\Rightarrow\; \lambda_1=0$$

$$\lambda_2=\frac{e^{x_1+x_2}}{\left(e^{x_1}+e^{x_2}\right)^2}\cdot 2 \;\Rightarrow\; \lambda_2=\frac{2e^{x_1+x_2}}{\left(e^{x_1}+e^{x_2}\right)^2}>0 \quad \text{for all} \;\; (x_1,x_2)\in\mathbb{R}^2$$

which means that the Hessian matrix H is **positive semi-definite**.

Therefore, the LSE function is **convex**.