Linear Programming

Ονοματεπώνυμο : Νικόλας Φιλιππάτος
AM: 1072754
Εργασία: 1η
Ημερομηνία: March 25, 2024

Table Of Contents

- Table Of Contents
- Εκφωνήσεις
  - Exercise 1
  - Exercise 2
  - Exercise 3
  - Exercise 4
  - Exercise 5
  - Exercise 6
- Solutions
  - Solution Exercise 01
    - 1a
    - 1b
    - 1γ
  - Python Scripts
    - *plt_line.py*
    - *linearGui.py*
    - *exercise_01.py *
  - Solution Exercise 2
    - 2a
      - ==Μενει να απαντησω αυτο
    - 2b
  - Solution Exercise 3
    - Προεργασια
  - Solution Exercise 4
    - 4a
    - 4b
  - Solution Exercise 5
    - 5a
  - Solution Exercise 6

Εκφωνήσεις

Exercise 1

Δίνεται το παρακάτω πρόβλημα γραμμικού προγραμματισμού:

$$\min\ Z = 2x_1 - x_2$$

όταν

$$\begin{array}{lll} (\Pi 1) & x_1 + x_2 & \geq 10 \\ (\Pi 2) & -10x_1 + x_2 & \leq 10 \\ (\Pi 3) & -4x_1 + x_2 & \leq 20 \\ (\Pi 4) & x_1 + 4x_2 & \geq 20 \\ & x_1, x_2 & \geq 0 \end{array}$$

1. α) Να παραστήσετε γραφικά την εφικτή περιοχή του προβλήματος καθώς και όλες τις κορυφές της. Περιγράψτε τη μορφή της εφικτής περιοχής. Με γραφικό τρόπο βρείτε τη βέλτιστη κορυφή του προβλήματος, εάν υπάρχει.
2. (β) Ομοίως με γραφικό τρόπο βρείτε τη βέλτιστη λύση, αν υπάρχει, όταν η αντικειμενική συνάρτηση γίνει:

   $$min\ Z = 11x_1 - x_2$$

   και η εφικτή περιοχή παραμένει ίδια με το ερώτημα (α)
3. (γ) Αν η εφικτή περιοχή είναι όπως περιγράφεται στο ερώτημα (α) και η αντικειμενική συνάρτηση δίνεται ως:

   $$max\ Z = c_1x_1 - x_2$$

   ποιά θα πρέπει να είναι η τιμή του c1 ώστε η βέλτιστη λύση να βρίσκεται στην τομή των ευθειών που ορίζονται από τους περιορισμούς Π1 και Π4;

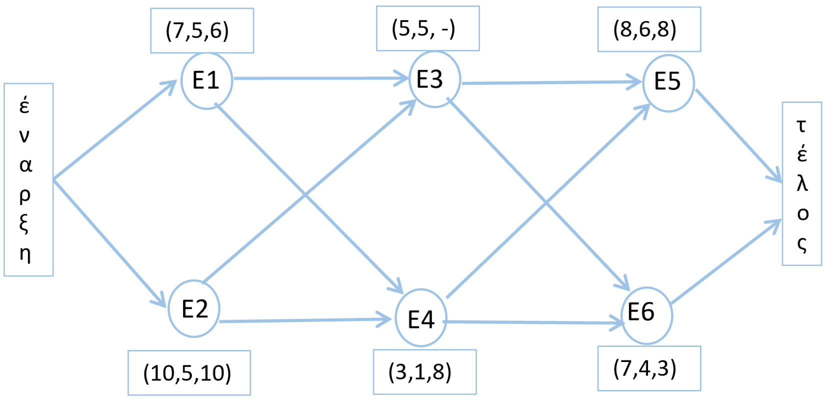
Exercise 2

΄Ενας φρουτοχυμός που παράγει η εταιρεία FRESH περιέχει το πολύ 3 μέρη χυμό πορτοκαλιού και τουλάχιστον 1 μέρος χυμό μήλου, σύμφωνα με τα αναγραφόμενα στη συσκευασία του. Η εβδομαδιαία δυναμικότητα της εταιρείας για το συγκεκριμένο προϊόν δεν ξεπερνάει τα 500 lit ενώ σύμφωνα με τα δεδομένα των πωλήσεων είναι σίγουρο ότι η αγορά μπορεί να απορροφήσει τουλάχιστον 400 lit κάθε εβδομάδα. Για την επόμενη εβδομάδα η εταιρεία μπορεί να διαθέσει μέχρι 250 lit χυμό μήλου και το τμήμα παραγωγής στοχεύει να παράξει φρουτοχυμό που θα περιέχει τουλάχιστον 40% χυμό πορτοκαλιού. Το κόστος για 1 lit χυμό πορτοκαλιού είναι 2 χ.μ. (χρηματικές μονάδες) και για 1 lit χυμό μήλου είναι 1 χ.μ. ενώ η εταιρεία πουλάει τον φρουτοχυμό (ανεξαρτήτου αναλογίας των δύο φρούτων) προς 5 χ.μ.

1. a) Βοηθήστε το τμήμα παραγωγής της εταιρείας FRESH να αποφασίσει για τις ποσότητες χυμού που θα πρέπει να παράγει την επόμενη εβδομάδα όταν αντικειμενικός στόχος είναι η μεγιστοποίηση του κέρδους. Μοντελοποιήστε το παραπάνω σενάριο ως πρόβλημα γραμμικού προγραμματισμού και λύστε το γραφικά. Περιγράψτε αναλυτικά τις μεταβλητές απόφασης, την αντικειμενική συνάρτηση και τους περιορισμούς. Περιγράψτε τη βέλτιστη λύση αναφορικά με τους περιορισμούς, δηλ. ποιους ικανοποιεί οριακά (δεσμευτικοί περιορισμοί) και ποιους χαλαρά (μη δεσμευτικοί περιορισμοί).
2. b) Αν αντίθετα το τμήμα παραγωγής στοχεύσει να παράξει φρουτοχυμό που θα περιέχει τουλάχιστον 50%/60% χυμό πορτοκαλιού, τι επιπτώσεις θα έχει αυτή η απόφασή τους στη βέλτιστη λύση του προβλήματος;

Exercise 3

Θεωρήστε ένα έργο το οποίο για να ολοκληρωθεί απαιτεί την διεκπεραίωση 6 επί μέρους εργασίες (Ε1 - Ε6). Οι εργασίες είναι εξαρτημένες μεταξύ τους και οι εξαρτήσεις δίνονται με το παρακάτω σχήμα:



Σύμφωνα με το σχήμα οι εργασίες Ε1 και Ε2 μπορούν να εκτελεστούν παράλληλα, όμως για να εκτελεστεί η εργασία Ε3 θα πρέπει να έχουν ολοκληρωθεί οι εργασίες Ε1 και Ε2.
Ομοίως και για τις υπόλοιπες εργασίες. Το έργο ολοκληρώνεται όταν εκτελεστούν (παράλληλα) οι εργασίες Ε5 και Ε6. Για κάθε εργασία δίνονται ο κανονικός χρόνος διεκπεραίωσης της εργασίας (σε εβδομάδες), το απόλυτο ελάχιστο για τον χρόνο αυτό, και το κόστος που θα προκύψει αν προσπαθήσουμε να μειώσουμε τον κανονικό χρόνο κατά μία εβδομάδα.

Η εταιρεία που έχει αναλάβει το έργο ενδιαφέρεται να μειώσει τον συνολικό χρόνο διεκπεραίωσης του (αν είναι εφικτό) σε 19 εβδομάδες, επομένως η διάρκεια μίας ή περισσοτέρων εργασιών θα πρέπει να μειωθεί σε σχέση με την κανονική τους διάρκεια. Προφανώς ο στόχος αυτός θα πρέπει να επιτευχθεί με το μικρότερο δυνατό κόστος.

Μοντελοποιήστε το συγκεκριμένο σενάριο προγραμματισμού εργασιών ως πρόβλημα γραμμικού προγραμματισμού. Ορίστε κατάλληλες μεταβλητές απόφασης και διαμορφώστε τους περιορισμούς όπως περιγράφονται στο σχήμα. Περιγράψτε και μοντελοποιήστε τον αντικειμενικό στόχο της εταιρείας. Δώστε την αντικειμενική συνάρτηση του προβλήματος γραμμικού προγραμματισμού. (Σημ. Η άσκηση δεν ζητάει τη λύση του προβλήματος, μόνο τη μοντελοποίησή του.)

Exercise 4

- (Π1) Η τομή X δύο κυρτών συνόλων X1 και X2 είναι κυρτό σύνολο. Ισχύει το ίδιο για την ένωση των κυρτών συνόλων;
- (Π2) Το σύνολο $M = \{(x, y) \in R^2_{++} | xy \geq k,\ k \in\ R\}$ είναι κυρτό σύνολο. (Υπόδειξη. Μπορείτε να χρησιμοποιήσετε ως γνωστό το Λήμμα ότι για θετικούς αριθμούς a και b ισχύει πάντα: $\frac{a}{b} + \frac{b}{a} \geq 2$

Exercise 5

Ασκηση 5. Θεωρήστε το πρόβλημα γραμμικού προγραμματισμού:

$$\max\ Z = -2x_1 + x_2 - 4x_3 + 3x_4$$

όταν

$$\begin{aligned} x_1 + x_2 + 3x_3 + 2x_4 &\leq 4 \\ x_1 - x_3 + x_4 &\leq 2 \\ 2x_1 + x_2 &\leq 3 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$

(α) Θεωρήστε το πολύτοπο των εφικτών λύσεων του παραπάνω προβλήματος γραμμικού προγραμματισμού. Βρείτε όλες τις κορυφές που δημιουργούνται από τις τομές των υπερεπιπέδων του και ξεχωρίστε ποιες από αυτές είναι κορυφές του πολύτοπου των εφικτών λύσεων. Εντοπίστε, αν υπάρχουν, τις εκφυλισμένες κορυφές.
(β) Προσθέστε μεταβλητές χαλάρωσης στο σύστημα ανισώσεων και βρείτε όλες τις βασικές (εφικτές και μη-εφικτές) λύσεις για το μη ομογενές σύστημα εξισώσεων που δημιουργείται. Εντοπίστε (αν υπάρχουν) τις εκφυλισμένες βασικές λύσεις.
(γ) Αντιστοιχίστε τις βασικές λύσεις που βρήκατε στο (β) ερώτημα με τις κορυφές του ερωτήματος (α) και τέλος υποδείξτε τη βέλτιστη λύση και βέλτιστη κορυφή του προβλήματος.

Exercise 6

Θεωρήστε το πρόβλημα γραμμικού προγραμματισμού

$$max\ Z = 5x_1 + 4x_2 - x_3 + 3x_4$$

όταν

$$\begin{gathered} 3x_1 + 2x_2 - 3x_3 + x_4 \leq 24 \\ 3x_1 + 3x_2 + x_3 + 3x_4 \leq 36 \\ x_1, x_2, x_3, x_4 \geq 0 \end{gathered}$$

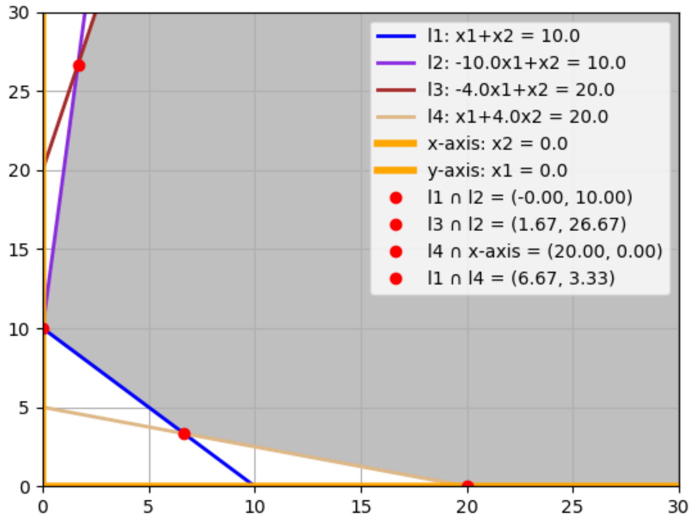
1. (α) Εφαρμόστε τον αλγόριθμο Simplex για να βρείτε τη βέλτιστη λύση του, αν υπάρχει. Σε κάθε επανάληψη του αλγορίθμου περιγράψτε συνοπτικά τα βήματα που ακολουθείτε και τις αποφάσεις που παίρνετε μέχρι το επόμενο βήμα.
2. (β) Εφαρμόστε όλες τις εναλλακτικές επιλογές που μπορεί να έχετε σε κάθε βήμα επιλογής της εισερχόμενης ή εξερχόμενης μεταβλητής στις επαναλήψεις του αλγορίθμου και δημιουργήστε έναν γράφο με τα βήματα (κορυφές) του αλγορίθμου μέχρι τη βέλτιστη λύση.

Solutions

Solution Exercise 01

1a

> 1. α) Να παραστήσετε γραφικά την εφικτή περιοχή του προβλήματος καθώς και όλες τις κορυφές της. Περιγράψτε τη μορφή της εφικτής περιοχής. Με γραφικό τρόπο βρείτε τη βέλτιστη κορυφή του προβλήματος, εάν υπάρχει.



Θα σχεδιάσουμε τις ευθειες

$$
\begin{gathered}
y = -x + 10 \\
y = 10x + 20 \\
y = 4x + 20 \\
y = -\frac{1}{4}x + 5
\end{gathered}
$$

Με βάση τους περιορισμους Π1-Π4 και $x_1, x_2 \geq 0$ η εφικτή περιοχή περιγράφεται από την γραμμοσκιασμένη περιοχή του σχήματος.

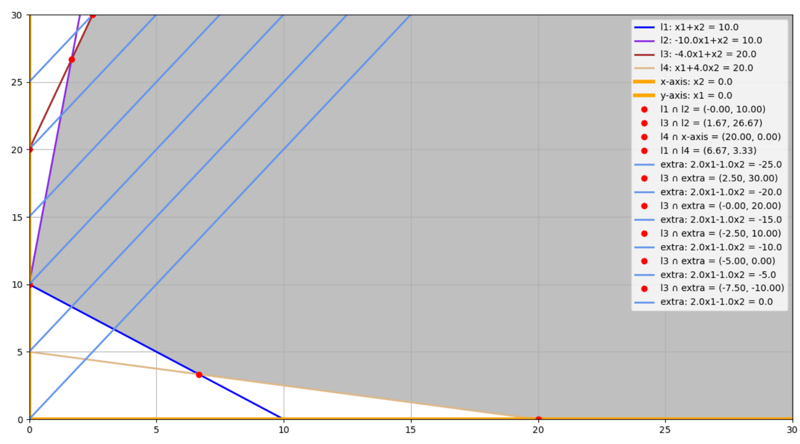
Κορυφές της εφικτής περιοχής :

- Σημειο Τομής του Π2 και Π3 [1.67, 26.67]
- Σημείο Τομής του Π1 και Π2 [0, 10]
- Σημείο Τομής του Π1 και Π4 [6,67, 3.33]
- Σημείο Τομής του Π4 και του άξονα x [20, 0]

Παρατηρουμε η εφικτή περιοχή
Ειναι bounded απο τα αριστερα λογω των περιορισμων Π3,Π2,Π1,Π4, αλλά απο τα δεξία οσο αυξάνονται τα x βλέπουμε ότι εχουμε μονο ενα κάτω όριο (ευθεια x2 =0 και ο Π4)

Για να βρουμε την βελτιστη κορυφη, σχηματιζουμε τις ευθειες $2x_1 - x_2 = c$, όπου c=[0,-5,-10,-15,-20,-25] και κοιταζουμε εαν συμπιπτει με καποια κορυφη

Βλέπουμε ότι μειώνοντας το c η αντικειμενική συνάρτηση παραμενει μεσα στην εφικτή περιοχη χωρις να βρισκει καποιο ανω οριο, οποτε δνε υπαρχει η βελτιστη κορυφη .

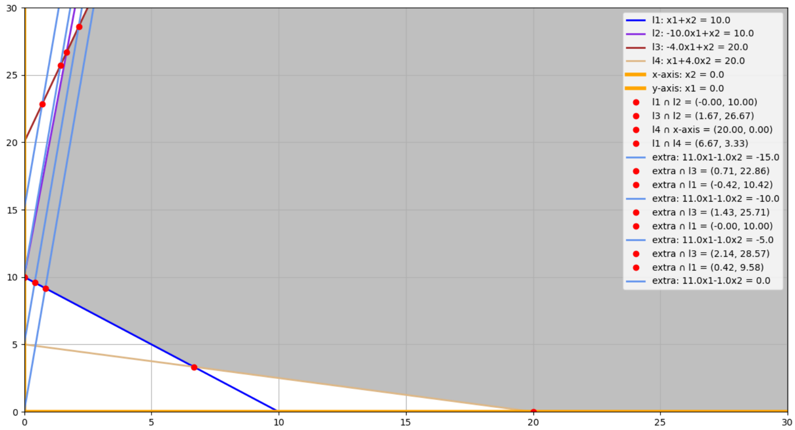


Το προβλημα εχει απεριοριστη λυση.

1b

> 2. (β) Ομοίως με γραφικό τρόπο βρείτε τη βέλτιστη λύση, αν υπάρχει, όταν η αντικειμενική συνάρτηση γίνει:
>
> $$min\ Z = 11x_1 - x_2$$
>
> και η εφικτή περιοχή παραμένει ίδια με το ερώτημα (α)

Για να λυσουμε γραφικα το προβλημα γραφουμε την αντικειμενικη συναρτηση ίση με c :
$2x_1 - x_2 = c$, όπου c=[0,-5,-10,-15,-20,-25]
Μειωνουμε την τιμη του c και βλεπουμε να πηγαινει προς τα αριστερα .



Παρατηρουμε οτι μετα το c=-10 η αντικειμενικη συναρτηση βγαινει εκτος της εφικτης περιοχης. Επομενως η βελτιστη κορυφη είναι η (0,10) Τομη του περιορισμου Π1 και Π2 και η μικροτερη τιμη που μπορει να παρει η αντικειμενικη συναρτηση ειναι -10

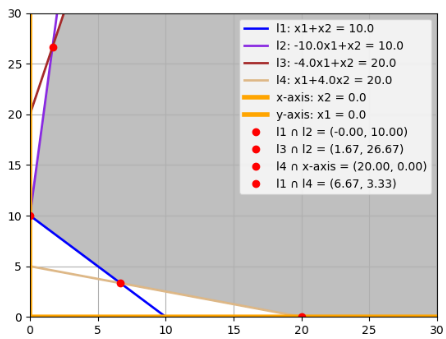
1γ

> 3. (γ) Αν η εφικτή περιοχή είναι όπως περιγράφεται στο ερώτημα (α) και η αντικειμενική συνάρτηση δίνεται ως:
>
> $$max\ Z = c_1x_1 - x_2$$
>
> ποιά θα πρέπει να είναι η τιμή του c1 ώστε η βέλτιστη λύση να βρίσκεται στην τομή των ευθειών που ορίζονται από τους περιορισμούς Π1 και Π4;

Η κορυφη των περιορισμων Π1, Π2

$$y = -x + 10$$
$$y = -\frac{1}{4}x + 5$$

- Σημείο Τομής του Π1 και Π4 [6,67, 3.33]



> $$\min\ Z = 2x_1 - x_2$$
> όταν
> $$\begin{array}{lll} (\Pi1) & x_1 + x_2 & \geq 10 \\ (\Pi2) & -10x_1 + x_2 & \leq 10 \\ (\Pi3) & -4x_1 + x_2 & \leq 20 \\ (\Pi4) & x_1 + 4x_2 & \geq 20 \\ & x_1, x_2 & \geq 0 \end{array}$$

Η αντικειμενικη συναρτηση του ερωτηματος γ ειναι $min\ Z = c_1x_1 - x_2$. Αν λυσουμε ως προς x_2: $$
x{2} = c{1}x_{1}-Z

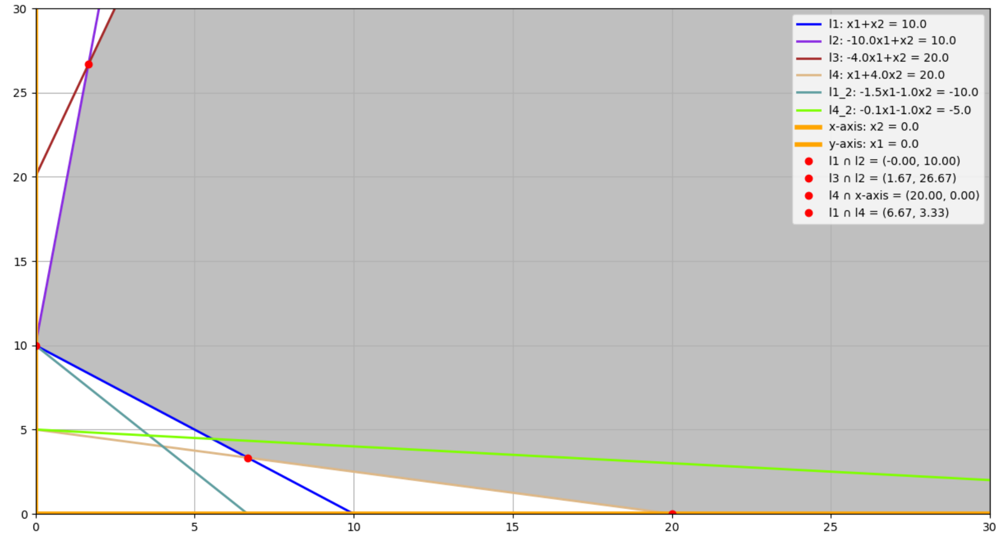
$Πρεπειναβρουμετοευροςτιμωντουc_1ετσιωστεηβελτιστηλυσηναειναιητομητωνπεριορισμωνΠ1καιΠ2: -ΣημείοΤομήςτουΠ1καιΠ4[6,67,3.33]Μπορουμεναγραψ$

\begin{matrix}
-x*{1}*-x{2}\leq -10 \

- \frac{1}{4} - x_{2} \leq -5
  \end{matrix}
  $$

Χωρις να επηρεασει το σχημα.

Βλεπουμε οτι με το $c_1 = -1, c_2 - \frac{1}{4}$ η αντικειμενικη συναρτηση ταυτιζεται με τις ευθειες των περιορισμων Π1 και Π4 αντιστοιχα



Βλεπουμε οτι αμα αυξησουμε το c1 απο -0.25 σε -0.1 αποκταει περισσοτερες λυσεις και σταματαει η τομη [6.67, 3.33] να ειναι η βελτιστη κορυφη. Αντιστοιχα αμα μειωσουμε το c1 απο -1 σε -1.5 η αντικειμενικη συναρτηση βγαινει εκτος περιοχης.

Επομενως το c_1 θα ανηκει :

$$-1 < c_1 < -\frac{1}{4}$$

Δεν περιλαμβανουμε το -1 και -0.25 γιατι τοτε δνε θα ειχαμε μονο το σημειο [6.67,3.33] ως βελτιστη λυση

Python Scripts

Για την επιλυση αξιοποιηθηκαν 3 python scripts:

plt_line.py

```python
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.colors as mcolors


class line:
    """ax1 + bx2 = c"""

    def __init__(
        self,
        a: float,
        b: float,
        c: float,
        name: str,
        legend_show=True,
        type: str = "line",
    ) -> None:
        """
        Args:
            a (float): _description_
            b (float): _description_
            c (float): _description_
            name (str): _description_
            legend_show (bool, optional): _description_. Defaults to True.
        """
        self.a, self.b, self.c = map(float, (a, b, c))
        self.name = name
        self.legend_show = legend_show
        self.type = type

    def __str__(self):
        """returns the equation"""

        output = f"{str(self.a if self.a !=1 else '')+'x1' if self.a else ''}{'+' if self.a and self.b>0 else ''}{str(self.b if self.b !=1 else '')+'x2' if self.b else ''} = {self.c}"
        return output

    def __call__(self, x, reverse=False) -> float:
        if reverse:
            return self.reverse_equation(x)
        return self.equation(x)

    def intersection(self, line, plotting=True):
        """returns the intersection point of two lines, and plots it if plotting is True"""

        x_up = line.c * self.b - self.c * line.b
        x_down = line.a * self.b - self.a * line.b
        if x_down == 0:
            return None

        if self.b == 0:
            x = self.c / self.a
            y = line.equation(x)
        elif line.b == 0:
            x = line.c / line.a
            y = self.equation(x)
        else:
            x = x_up / x_down
            y = self.equation(x)

        if plotting:
            plt.plot(
                x, y, "ro", label=f"{self.name} ∩ {line.name} = ({x:.2f}, {y:.2f})"
            )
            if self.legend_show:
                plt.legend()

        return [x, y]

    def equation(self, x):
        """returns the y value of the line for a given x value"""
        if self.b != 0:
            return (-self.a * x + self.c) / self.b
        return self.c / self.a
```

```
    def reverse_equation(self, y):
        """returns the x value of the line for a given y value"""
        if self.a != 0:
            return (-self.b * y + self.c) / self.a
        return self.c / self.b

    def plot(self, x, color=None, lw=2, ms=12):
        """plots the line, and the equation as a label"""

        if self.b == 0:
            plt.axvline(
                x=self.c / self.a,
                label=f"{self.name}: {self}",
                color=self.auto_color_chooser(color),
                linewidth=lw,
                markersize=ms,
            )
        elif self.a == 0:
            plt.axhline(
                y=self.c / self.b,
                label=f"{self.name}: {self}",
                color=self.auto_color_chooser(color),
                linewidth=lw,
                markersize=ms,
            )
        else:
            plt.plot(
                x,
                self.equation(x),
                label=f"{self.name}: {self}",
                color=self.auto_color_chooser(color),
                linewidth=lw,
                markersize=ms,
            )
        # calls the plot settings function to show the legend
        if self.legend_show:
            plt.legend()

    def auto_color_chooser(self, color=None):
        """returns a color from the matplotlib color list, if color is not given, it returns the next color in the list, if color
is given, it returns the color if it is in the list, else it returns the first color in the list"""
        colors = sorted(mcolors.cnames)
        if not color and not hasattr(self, "ind"):
            self.ind = 0
        elif not color and hasattr(self, "ind"):
            self.ind = (self.ind + 1) % len(colors)
        elif color and not hasattr(self, "ind"):
            self.ind = colors.index(color) if color in colors else 0
        return colors[self.ind]
```

linearGui.py

```
import matplotlib.pyplot as plt
import numpy as np
import matplotlib.colors as mcolors
import sys
from pathlib import Path
from shapely.geometry.polygon import Polygon
from shapely.geometry import Point

sys.path.append(str(Path(__file__).parents[1]))
from Tools.plt_line import line


class LinearGUI:
    def __init__(self, *args, **kwargs) -> None:
        """
        Args:
            plt (module): the matplotlib module
            xAxis (np.array): the x axis
            limit (int): the limit of the graph
            ylim (tuple): the y axis limits
            xlim (tuple): the x axis limits
            lines (list): the lines to plot
            save_images (bool): save the images
            v (list): the feasible points
            name (str): the name of the graph
        """
```

```
        self.plt = kwargs.get("plt", plt)

        self.limit = kwargs.get("limit", 10)

        xAxis = np.linspace(-10, self.limit + 10, 10)
        self.xAxis = kwargs.get("xAxis", xAxis)

        self.ylim = kwargs.get("ylim", (0, self.limit))
        self.xlim = kwargs.get("xlim", (0, self.limit))

        self.lines = kwargs.get("lines", [])
        self.name = kwargs.get("name", "LinearGUI")
        self.figsize = kwargs.get("figsize")
        self.step = kwargs.get("step", 1)
        self.save_images = kwargs.get("save_images", False)

        self.v = []

    def show(self):
        self.plt.show()

    def plot_equations(self):
        for i, line in enumerate(self.lines):
            if line.type == "line":
                if i == 0:
                    color = "blue"
                else:
                    color = self.lines[i - 1].auto_color_chooser()
                line.plot(self.xAxis, color)

        self.add_axis_lines()

    def add_axis_lines(self):
        right_limit = line(1, 0, self.limit, "limit", type="axis")
        x = line(0, 1, 0, "x-axis", type="axis")
        y = line(1, 0, 0, "y-axis", type="axis")

        self.lines.append(right_limit)
        self.lines.append(x)
        self.lines.append(y)

        x.plot(self.xAxis, "orange", lw=4)
        y.plot(self.xAxis, "orange", lw=4)


    def find_feasible_points(self):
        pass

    def fill_feasible(self):

        self.find_feasible_points()

        # Create the x and y Coordinates for the fill area
        x = [i[0] for i in self.v]
        y = [i[1] for i in self.v]

        # Fill takes the x and y of a polygon and fills it with color
        self.plt.fill(x, y, color="gray", alpha=0.5)

    def create_figure(self, name: str = None):
        if name:
            self.name = name
        if self.figsize:
            plt.figure(self.name, figsize=self.figsize)
        else:
            plt.figure(self.name)

        self.plt.ylim(*self.ylim)
        self.plt.xlim(*self.xlim)
        self.plt.grid()
        self.plot_equations()
        self.fill_feasible()

    def check_feasible_point(self, points):
        p = Point(points)
        polygon = Polygon(self.v)
        return polygon.contains(p)

    def intersections_in_feasible(self, extra):
        for l in self.lines:
```

```
            possible = extra.intersection(l, plotting=False)
            if self.check_feasible_point(possible):
                extra.intersection(l)

    def graphical_solution(self, a, b, minlim, maxlim, legend=True):

        counter = minlim
        while counter < maxlim:
            extra = line(a, b, counter, "extra")
            extra.plot(self.xAxis, "cornflowerblue")
            extra.legend_show = False
            self.intersections_in_feasible(extra)
            counter += self.step

    def save_image(self, file_name):
        if self.save_images:
            img_folder = Path(self.parent, "img")

            image_file = Path(img_folder, file_name)
            self.plt.savefig(image_file, dpi="figure")
```

exercise_01.py

```
import matplotlib.pyplot as plt
import sys
from pathlib import Path

sys.path.append(str(Path(__file__).parents[2]))
from Tools.plt_line import line
from Tools.linearGui import LinearGUI


class Exerc01Gui(LinearGUI):
    def __init__(self, *args, **kwargs) -> None:
        limit = 30

        kwargs["limit"] = limit
        super().__init__(*args, **kwargs)
        self.lines = []
        self.lines.append(line(1, 1, 10, "l1", type="line"))
        self.lines.append(line(-10, 1, 10, "l2", type="line"))
        self.lines.append(line(-4, 1, 20, "l3", type="line"))
        self.lines.append(line(1, 4, 20, "l4", type="line"))

    def find_feasible_points(self):

        # Intersection lines 1 and 2
        self.v.append(self.lines[0].intersection(self.lines[1]))

        # Intersection lines 2 and 3
        self.v.append(self.lines[2].intersection(self.lines[1]))

        # Upper left limit corner
        self.v.append([self.lines[2](self.limit, reverse=True), self.limit])

        # Upper right corner
        self.v.append([self.limit, self.limit])

        # Lower right corner
        self.v.append([self.limit, 0])

        # Intersection x_axis and line 4

        x_axis = [l for l in self.lines if l.name == "x-axis"][0]
        self.v.append(self.lines[3].intersection(x_axis))

        # Intersection line 4 and line 1
        self.v.append(self.lines[0].intersection(self.lines[3]))

    def save_image(self, file_name):
        if self.save_images:
            parent = Path(__file__).parent
            img_folder = Path(parent, "img")

            image_file = Path(img_folder, file_name)
            self.plt.savefig(image_file, dpi="figure")

    def feasible_region(self):
        self.create_figure()
```

```python
        self.save_image("exerc01_a.png")

    def exerc01_a(self):
        self.name = "Z: min 2x1-x2"
        self.create_figure()
        self.step = 5
        self.graphical_solution(2, -1, -25, 5)
        self.save_image("exerc01_a_1.png")

    def exerc01_b(self):
        self.name = "Z: min 11x1-x2"
        self.create_figure()
        self.step = 5
        self.graphical_solution(11, -1, -15, 5)
        self.save_image("exerc01_b.png")

    def exerc01_c(self):
        self.name = "Z: min c1x1-x2"

        top = self.lines[0].intersection(self.lines[3], plotting=False)
        self.lines.append(line(-1.5, -1, -10, "l1_2", type="line"))
        self.lines.append(line(-0.1, -1, -5, "l4_2", type="line"))

        self.create_figure()
        self.save_image("exerc01_c.png")

    def main(self):

        self.save_images = False
        self.feasible_region()
        self.exerc01_a()
        self.exerc01_b()
        self.exerc01_c()
        self.show()


if __name__ == "__main__":
    try:
        gui = Exerc01Gui(plt, name="Feasible Region", figsize=(15, 8))
        gui.main()
    except KeyboardInterrupt:
        pass
    except Exception as e:
        print(e)
```

Solution Exercise 2

> ## Exercise 2
>
> Ενας φρουτοχυμός που παράγει η εταιρεία FRESH περιέχει το πολύ 3 μέρη χυμό πορτοκαλιού και τουλάχιστον 1 μέρος χυμό μήλου, σύμφωνα με τα αναγραφόμενα στη συσκευασία του. Η εβδομαδιαία δυναμικότητα της εταιρείας για το συγκεκριμένο προϊόν δεν ξεπερνάει τα 500 lit ενώ σύμφωνα με τα δεδομένα των πωλήσεων είναι σίγουρο ότι η αγορά μπορεί να απορροφήσει τουλάχιστον 400 lit κάθε εβδομάδα. Για την επόμενη εβδομάδα η εταιρεία μπορεί να διαθέσει μέχρι 250 lit χυμό μήλου και το τμήμα παραγωγής στοχεύει να παράξει φρουτοχυμό που θα περιέχει τουλάχιστον 40% χυμό πορτοκαλιού. Το κόστος για 1 lit χυμό πορτοκαλιού είναι 2 χ.μ. (χρηματικές μονάδες) και για 1 lit χυμό μήλου είναι 1 χ.μ. ενώ η εταιρεία πουλάει τον φρουτοχυμό (ανεξαρτήτου αναλογίας των δύο φρούτων) προς 5 χ.μ.
>
> 1. a) Βοηθήστε το τμήμα παραγωγής της εταιρείας FRESH να αποφασίσει για τις ποσότητες χυμού που θα πρέπει να παράγει την επόμενη εβδομάδα όταν αντικειμενικός στόχος είναι η μεγιστοποίηση του κέρδους. Μοντελοποιήστε το παραπάνω σενάριο ως πρόβλημα γραμμικού προγραμματισμού και λύστε το γραφικά. Περιγράψτε αναλυτικά τις μεταβλητές απόφασης, την αντικειμενική συνάρτηση και τους περιορισμούς. Περιγράψτε τη βέλτιστη λύση αναφορικά με τους περιορισμούς, δηλ. ποιους ικανοποιεί οριακά (δεσμευτικοί περιορισμοί) και ποιους χαλαρά (μη δεσμευτικοί περιορισμοί).
> 2. b) Αν αντίθετα το τμήμα παραγωγής στοχεύσει να παράξει φρουτοχυμό που θα περιέχει τουλάχιστον 50%/60% χυμό πορτοκαλιού, τι επιπτώσεις θα έχει αυτή η απόφασή τους στη βέλτιστη λύση του προβλήματος;

2a

Έστω `x` η ποσότητα πορτοκαλιου στον Χυμο και `y` η ποσοτητα μηλου στον Χυμο

Το κόστος για την παραγωγη τους εκφραζεται απο την σχεση :

$$2 \cdot x + y$$

καθώς 1l χυμο πορτοκαλι κοστιζει 2 χ.μ. και 1l χυμος μηλου κοστιζει 1 χμ

Απο τον εβδομαδιαιο περιορισμο παραγωγης, βγαζουμε οτι

$$x + y \leq 500$$

Επισης βλεπουμε οτι εφοσον το υποστηριζει η αγορα, πρεπει να παραξουμε τουλαχιστον 400l αρα :

$$x + y \geq 400$$

Τα εσοδα απο τον χυμο ειναι 5 χ.μ. επι την ποσοτητα του χυμου που παραγουμε, αρα

$$5 \cdot (x + y)$$

Θελουμε να μεγιστοποιησουμε το κερδος, το οποιο ειναι η διαφορα των εσοδων απο τα εξοδα :

$$5(x + y) - (2x + y) = 3x + 4y$$

Ακομη γραφει οτι η ο φρουτοχυμος πρεπει να περιεχει το πολυ 3 μερη χυμο πορτοκαλιου και τουλαχιστον 1 μερος χυμο μηλου

Για να βρουμε το ποσοστο του χυμου πορτοκαλιου χρησιμοποιουμε τον τυπο :

$$\frac{x}{x + y}$$

Οποτε :

$$\frac{x}{x + y} \leq \frac{3}{4} \implies x - 3y \leq 0$$

Και για το μηλο :

$$\frac{y}{x + y} \geq \frac{1}{4} \implies x - 3y \leq 0$$

το μηλο για την επομενη εβδομαδα μπορει να ειναι μεχρι 250l

$$y \leq 250$$

Τελος μας δινει οτι εχουν στοχο το φρουτοποτο να περιεχει τουλαχιστον 40% χυμο πορτοκαλι :

$$\frac{x}{x + y} \geq 0.4 \implies 0.6x - 0.4y \geq 0 \implies 6x - 4y \geq 0 \implies 3x - 2y \geq 0$$

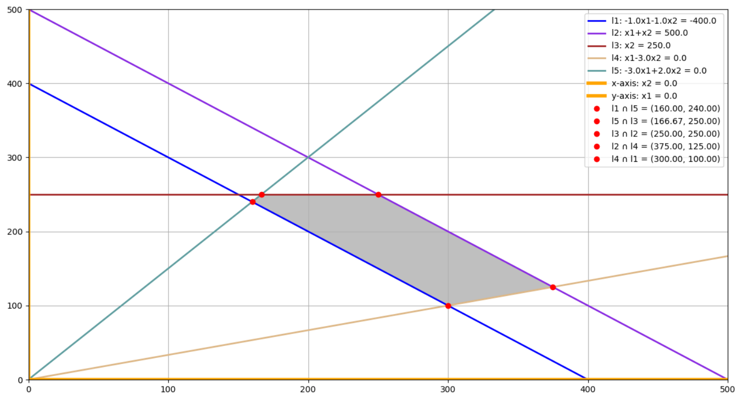
Το μοντελο γραμμικου προγραμματισμου ειναι :

$$\begin{array}{lccccll} max\ Z: & 3x & + & 4y & & \\ st & & & & & \\ & x & + & y & \geq 400 & (\Pi 1) \\ & x & + & y & \leq 500 & (\Pi 2) \\ & & & y & \leq 250 & (\Pi 3) \\ & x & & -3y & \leq 0 & (\Pi 4) \\ & 3x & & -2y & \geq 0 & (\Pi 5) \end{array}$$

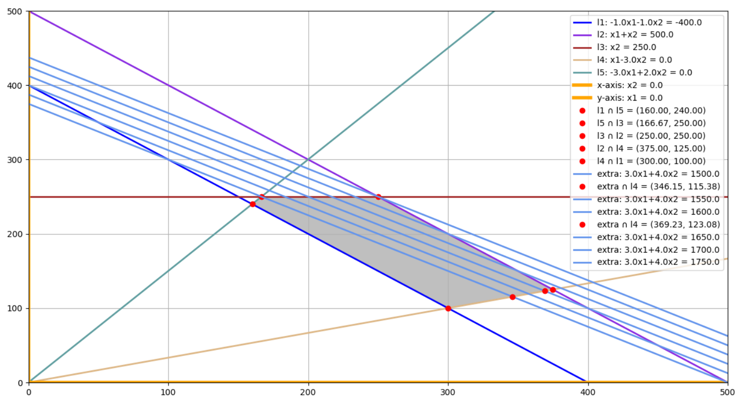
Το οποιο το μετατρεπουμε σε :

$$
\begin{array}{ll}
max\ Z: & 3x+4y \\
st & \\
& \begin{array}{lll}
-x-y & \leq 400 & (\varPi 1) \\
x+y & \leq 500 & (\varPi 2) \\
y & \leq 250 & (\varPi 3) \\
x-3y & \leq 0 & (\varPi 4) \\
-3x+2y & \leq 0 & (\varPi 5)
\end{array}
\end{array}
$$

Παιρνουμε την εφικτη περιοχη :



Ψαχνοντας να το λυσουμε γραφικα :



Βλεπουμε οτι η βελτιστη κορυφη ειναι το σημειο τομης του Π3 και Π2 [250,250]

Αυτο σημαινει οτι χρησιμοποιωντας 50% χυμο πορτοκαλι και 50% χυμο μηλο θα εχουν το μεγαλυτερο κερδος στις 1750 χ.μ.
Πληρουνται ολοι οι περιορισμοι.

==Μενει να απαντησω αυτο

Περιγράψτε αναλυτικά τις μεταβλητές απόφασης, την αντικειμενική συνάρτηση και τους περιορισμούς. Περιγράψτε τη βέλτιστη λύση αναφορικά με τους περιορισμούς, δηλ. ποιους ικανοποιεί οριακά (δεσμευτικοί περιορισμοί) και ποιους χαλαρά (μη δεσμευτικοί περιορισμοί).

==.

2b

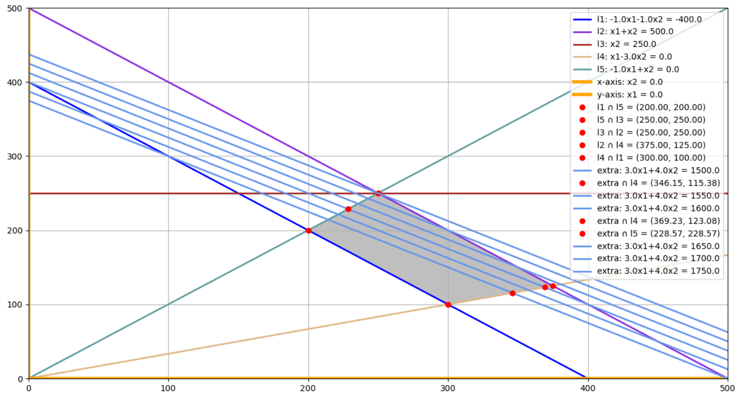
> 2. b) Αν αντίθετα το τμήμα παραγωγής στοχεύσει να παράξει φρουτοχυμό που θα περιέχει τουλάχιστον 50%/60% χυμό πορτοκαλιού, τι επιπτώσεις θα έχει αυτή η απόφασή τους στη βέλτιστη λύση του προβλήματος;

Για να εχουμε τουλαχιστον 50% χυμο πορτοκαλι αλλαζουμε τον περιορισμο Π5. Βλεπουμε οτι το 50% ειναι κατω απο το 75% του περιορισμου Π4. Οποτε εχουμε :

$$
\frac{x}{x+y} \geq 0.5 \implies \frac{x}{x+y} \geq \frac{1}{2} \implies x-y \geq 0 \implies -x+y \leq 0
$$

$$
\begin{array}{ll}
max\ Z: & 3x+4y \\
st & \\
& \begin{array}{lll}
-x-y & \leq 400 & (\varPi 1) \\
x+y & \leq 500 & (\varPi 2) \\
y & \leq 250 & (\varPi 3) \\
x-3y & \leq 0 & (\varPi 4) \\
-x+y & \leq 0 & (\varPi 5)
\end{array}
\end{array}
$$

Κανουμε τη γραφικη παρασταση :



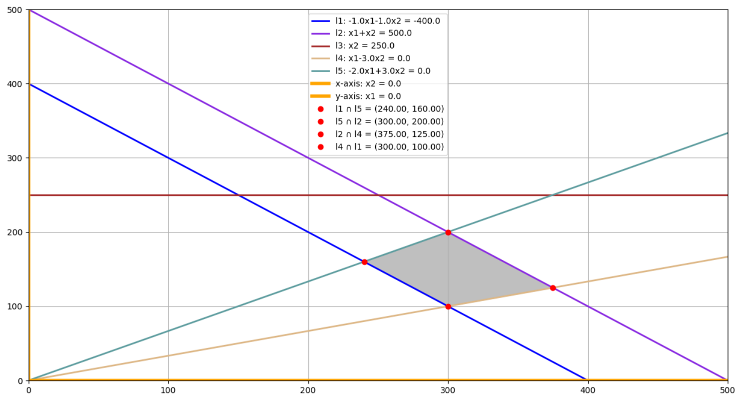
Παρατηρουμε οτι η βελτιστη κορυφη δεν εχει αλλαξει και ειναι η τομη του περιορισμου Π3 και Π2 [250,250]

Για να εχει τουλαχιστον 60% χυμο πορτοκαλι αλλαζουμε παλι τον περιορισμο Π5

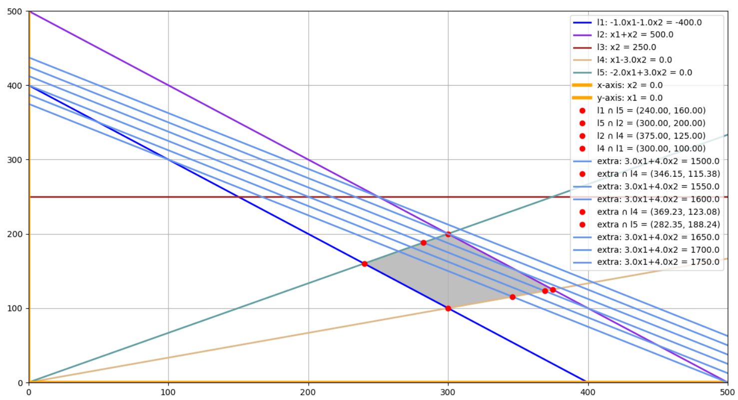
$$\frac{x}{x+y} \geq 0.6 \implies x \geq 0.6x + 0.6y \implies -2x + 3y \leq 0$$

$$\begin{array}{lllll} max\ Z: & 3x + 4y & & \\ st & & & \\ & -x - y & \leq -400 & (\Pi 1) \\ & x + y & \leq 500 & (\Pi 2) \\ & y & \leq 250 & (\Pi 3) \\ & x - 3y & \leq 0 & (\Pi 4) \\ & -2x + 3y & \leq 0 & (\Pi 5) \end{array}$$

Η εφικτή περιοχη ειναι πλεον :



Εφαρμοζουμε την αντικειμεινικη συναρτηση :

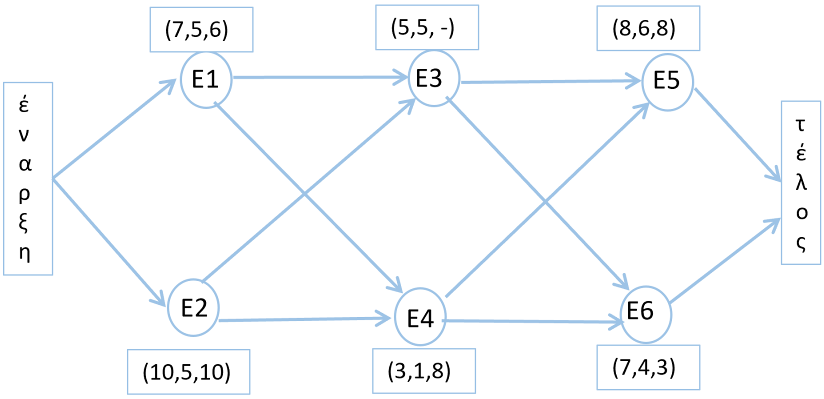


Και η βελτιση κορυφη βρισκεται στην τομη των περιορισμων Π5 και Π2 [300,200]

Solution Exercise 3

Exercise 3

Θεωρήστε ένα έργο το οποίο για να ολοκληρωθεί απαιτεί την διεκπεραίωση 6 επί μέρους εργασίες (Ε1 - Ε6). Οι εργασίες είναι εξαρτημένες μεταξύ τους και οι εξαρτήσεις δίνονται με το παρακάτω σχήμα:



Σύμφωνα με το σχήμα οι εργασίες Ε1 και Ε2 μπορούν να εκτελεστούν παράλληλα, όμως για να εκτελεστεί η εργασία Ε3 θα πρέπει να έχουν ολοκληρωθεί οι εργασίες Ε1 και Ε2.
Ομοίως και για τις υπόλοιπες εργασίες. Το έργο ολοκληρώνεται όταν εκτελεστούν (παράλληλα) οι εργασίες Ε5 και Ε6. Για κάθε εργασία δίνονται ο κανονικός χρόνος διεκπεραίωσης της εργασίας (σε εβδομάδες), το απόλυτο ελάχιστο για τον χρόνο αυτό, και το κόστος που θα προκύψει αν προσπαθήσουμε να μειώσουμε τον κανονικό χρόνο κατά μία εβδομάδα.

Η εταιρεία που έχει αναλάβει το έργο ενδιαφέρεται να μειώσει τον συνολικό χρόνο διεκπεραίωσης του (αν είναι εφικτό) σε 19 εβδομάδες, επομένως η διάρκεια μίας ή περισσοτέρων εργασιών θα πρέπει να μειωθεί σε σχέση με την κανονική τους διάρκεια. Προφανώς ο στόχος αυτός θα πρέπει να επιτευχθεί με το μικρότερο δυνατό κόστος.

Μοντελοποιήστε το συγκεκριμένο σενάριο προγραμματισμού εργασιών ως πρόβλημα γραμμικού προγραμματισμού. Ορίστε κατάλληλες μεταβλητές απόφασης και διαμορφώστε τους περιορισμούς όπως περιγράφονται στο σχήμα. Περιγράψτε και μοντελοποιήστε τον αντικειμενικό στόχο της εταιρείας. Δώστε την αντικειμενική συνάρτηση του προβλήματος γραμμικού προγραμματισμού. (Σημ. Η άσκηση δεν ζητάει τη λύση του προβλήματος, μόνο τη μοντελοποίησή του.)

Προεργασια

Ξερουμε οτι τα (Ε1,Ε2), (Ε3,Ε4), (Ε5,Ε6) μπορουν να τρεξουν ταυτοχρονα

επομενως εστω οι μεταβλητες $t_1, t_2, t_3, t_4, t_5, t_6$ των διεργασιων αυτων

Ο τελικος χρονος που θα παρουν για να ολοκληρωθουν ολες οι διεργασιες ειναι :

$max(t_1, t_2) + max(t_3, t_4) + max(t_5, t_6)$

οπου ξερουμε :

st :

$$\begin{gathered} 7 \geq t_1 \geq 5 \\ 10 \geq t_2 \geq 5 \\ t_3 = 5 \\ 3 \geq t_4 \geq 1 \\ 8 \geq t_5 \geq 6 \\ 7 \geq t_6 \geq 4 \end{gathered}$$

Εαν θεωρησουμε $c_1, c_2, c_3, c_4, c_5, c_6$ το βαρος του κοστους της καθε βελτιωσης χρονου διεργασιας :

$$\begin{gathered} c_1 = 6 \\ c_2 = 10 \\ c_3 = 0 \\ c_4 = 8 \\ c_5 = 8 \\ c_6 = 3 \end{gathered}$$

Θελουμε να εχουμε το συνολικο χρονο :

Ξερουμε οτι για να ξεκινησει η Ε3 πρεπει να εχει περασει το max(t1,t2)

βλεπουμε οτι μπορουμε να βαλουμε μαζι τους χρονους t_{12}, t_{34}, t_{56}

Οπου :
Κανονικοι χρονοι :

$$\begin{gathered} 7 \leq t_{12} \leq 10 \\ t_{34} = 5 \\ 7 \leq t_{56} \leq 8 \end{gathered}$$

Ελαχιστοι Χρονοι:

$$5 \le t_{12} \le 10$$
$$t_{34} = 5$$
$$6 \le t_{56} \le 8$$

Θεωρουμε μαζι τα ζευγαρια των εργασιων, καθως μολις τελειωσει η μεγαλυτερης διαρκειας διαδικασια μπορουμε να περασουμε στο επομενο ζευγαρι.

Το δευτερο ζευγαρι εχει σταθερο χρονο εκτελεσης, καθως το Ε3 θα παιρνει παντα παραπανω χρονο απο το Ε4 και το Ε3 δεν μπορει να βελτιωθει καθολου ο χρονος του

Θεωρουμε τις μεταβλητες s_1, s_2, s_4, s_5, s_6, τον χρονο που μπορουμε να αφαιρεσουμε απο τον κανονικο

$$0 \le s_1 \le 2$$
$$0 \le s_2 \le 5$$
$$0 \le s_4 \le 2$$
$$0 \le s_5 \le 2$$
$$0 \le s_6 \le 3$$

Και το κοστος για την καθε μια βελτιωση ειναι :

$$cost_1 = s_1 \cdot c_1$$
$$cost_2 = s_2 \cdot c_2$$
$$cost_4 = s_4 \cdot c_4$$
$$cost_5 = s_5 \cdot c_5$$
$$cost_6 = s_6 \cdot c_6$$

Μπορουμε να παραβλεψουμε οποιαδηποτε βελτιωση για το 4 γιατι παντα θα χρειαζεται 5 χρονους για να μπορεσει εκτελεσθει

$$0 \le s_1 \le 2 \qquad cost_1 = s_1 \cdot c_1$$
$$0 \le s_2 \le 5 \qquad cost_2 = s_2 \cdot c_2$$
$$0 \le s_5 \le 2 \qquad cost_5 = s_5 \cdot c_5$$
$$0 \le s_6 \le 3 \qquad cost_6 = s_6 \cdot c_6$$

Solution Exercise 4

4a

> - (Π1) Η τομή Χ δύο κυρτών συνόλων Χ1 και Χ2 είναι κυρτό σύνολο. Ισχύει το ίδιο για την ένωση των κυρτών συνόλων;

Από την θεωρία γνωρίζουμε οτι ενα συνολο Χ ειναι κυρτο εαν και μονο αν για $\forall \, x, y$ και $\lambda \ \in [0,1]$ ισχύει οτι :

$$\lambda x + (1-\lambda)y \in X$$

Θεωρουμε X_1, X_2 δυο κυρτα συνολα

Θα παρουμε 3 ζευγαρια σημειων (x,y)

Πρωτη Περιπτωση : $x, y \in X_1 \cap X_2$

Οποτε ικανοποιουνται :

$\lambda x + (1-\lambda)y \in X_1$ και $\lambda x + (1-\lambda)y \in X_2$, και γνωριζουμε οτι η τομη κυρτων συνολων ειναι κυρτο συνολο.

Δευτερη Περιπτωση : $x, y \in X_1$ αλλά $x, y \notin X_2$

==Δεν ειναι σωστο

4b

> - (Π2) Το σύνολο $M = \{(x,y) \in R^2_{++} | xy \geq k, \ k \in \ R\}$ είναι κυρτό σύνολο. (Υπόδειξη. Μπορείτε να χρησιμοποιήσετε ως γνωστό το Λήμμα ότι για θετικούς αριθμούς a και b ισχύει πάντα: $\frac{a}{b} + \frac{b}{a} \geq 2$

Εχουμε το συνολο $M = \{(x,y) \in R^2_{++} | xy \geq k, \ k \in \ R\}$

Θα παρουμε δυο σημεια $(x_1, y_1), (x_2, y_2)$ τα οποια ανηκουν στο συνολο Μ, δηλαδη $x_1 y_1 \geq k, x_2 y_2 \geq k$

Για να δειξουμε οτι το συνολο Μ ειναι κυρτο θα πρεπει και τα $\lambda x_1 + (1-\lambda)x_2 \in M, \, \lambda y_1 + (1-\lambda)y_2 \in M$

Οποτε θελουμε να ισχυει για $x = \lambda x_1 + (1-\lambda)x_2, \ y = \lambda y_1 + (1-\lambda)y_2 : \ xy \geq k$

==Καπως συνεχιζει αυτο και λυνεις με βαση το xy>k και το χτιζεις μεχρι να το αποδειξεις

Solution Exercise 5

> ## Exercise 5
>
> Ασκηση 5. Θεωρήστε το πρόβλημα γραμμικού προγραμματισμού:
>
> $$\max\ Z = -2x_1 + x_2 - 4x_3 + 3x_4$$
>
> όταν
>
> $$\begin{aligned} x_1 + x_2 + 3x_3 + 2x_4 &\leq 4 \\ x_1 - x_3 + x_4 &\leq 2 \\ 2x_1 + x_2 &\leq 3 \\ x_1, x_2, x_3, x_4 &\geq 0 \end{aligned}$$
>
> (α) Θεωρήστε το πολύτοπο των εφικτών λύσεων του παραπάνω προβλήματος γραμμικού προγραμματισμού. Βρείτε όλες τις κορυφές που δημιουργούνται από τις τομές των υπερεπιπέδων του και ξεχωρίστε ποιες από αυτές είναι κορυφές του πολύτοπου των εφικτών λύσεων. Εντοπίστε, αν υπάρχουν, τις εκφυλισμένες κορυφές.
> (β) Προσθέστε μεταβλητές χαλάρωσης στο σύστημα ανισώσεων και βρείτε όλες τις βασικές (εφικτές και μη-εφικτές) λύσεις για το μη ομογενές σύστημα εξισώσεων που δημιουργείται. Εντοπίστε (αν υπάρχουν) τις εκφυλισμένες βασικές λύσεις.
> (γ) Αντιστοιχίστε τις βασικές λύσεις που βρήκατε στο (β) ερώτημα με τις κορυφές του ερωτήματος (α) και τέλος υποδείξτε τη βέλτιστη λύση και βέλτιστη κορυφή του προβλήματος.

5a

Θελουμε να βρουμε ολες τις κορυφες που δημιουργουνται απο τις τομες των υπερπιπεδων του

Ειμαστε στο R^4, αρα μια κορυφη ειναι τομη 4 υπερεπιπεδων. Εχουμε 7 υπερεπιπεδα αρα:

$$\binom{7}{4} = \frac{7!}{4!(7-4)!} = \frac{7!}{4!3!} = \frac{7 \cdot 6 \cdot 5}{3 \cdot 2 \cdot 1} = 35$$

Θα εχουμε 35 πιθανες κορυφες

Θα παρουμε τα παρακατω συστηματα :

$$min\ Z = 12x_1 - 10x_2$$

==Μετα χρειαζεται να φτιαξουμε συνδυασμους 4 εξισωσεων, να βρισκουμε τη λυση του συστηματος και να την ελεγχουμε με ολους τους περιορισμους

Solution Exercise 6