ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ

ΜΗΧΑΝΙΚΩΝ ΚΑΙ

ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΓΡΑΜΜΙΚΗ ΚΑΙ ΣΥΝΔΥΑΣΤΙΚΗ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗ

ΕΡΓΑΣΙΑ 2η

ΝΑΤΑΛΙΑ ΡΟΥΣΚΑ - ΑΜ 1092581

ΥΠΕΥΘΥΝΗ ΚΑΘΗΓΗΤΡΙΑ

ΣΟΦΙΑ ΔΑΣΚΑΛΑΚΗ

ΡΙΟ

31 ΜΑΙΟΥ 2025

Άσκηση 1

(α) Το LP problem εισάγοντας μεταβλητές χαλάρωσης x_6, x_7, x_8

$\max\ 3x_1 + 11x_2 + 9x_3 - x_4 - 29x_5$
$x_2 + x_3 + x_4 - 2x_5 + x_6 = 4$
$x_1 - x_2 + x_3 + 2x_4 + x_5 - x_7 = 0$
$x_1 + x_2 + x_3 - 3x_5 + x_8 = 1$
$x_1 \epsilon \mathbb{R}, x_2, x_3, x_4, x_5 \geq 0$

Χρησιμοποιώντας το module `pulp` που χρησιμοποιεί τον `CBC MILP solver`, λύνω το πρόβλημα LP.

Κώδικας 1α

```python
import pulp

def solver():
    # A LP problem
    prob = pulp.LpProblem("exersice1", pulp.LpMaximize)

    # Variables
    x1 = pulp.LpVariable("x1", None, None)
    x2 = pulp.LpVariable("x2", 0, None)
    x3 = pulp.LpVariable("x3", 0, None)
    x4 = pulp.LpVariable("x4", 0, None)
    x5 = pulp.LpVariable("x5", 0, None)

    # Objective
    prob += 3*x1 + 11*x2 + 9*x3 - x4 - 29*x5, "obj"

    # Constraints
    prob += x2 + x3 + x4 - 2*x5 <= 4,  "c1"
    prob += x1 - x2 + x3 + 2*x4 + x5 >= 0, "c2"
    prob += x1 + x2 + x3 - 3*x5 <= 1, "c3"

    # solve the problem using the default solver
    prob.solve()

    # print the status of the solved LP
    print("Status:", pulp.LpStatus[prob.status])

    # print the value of the objective
    print("objective =", pulp.value(prob.objective))

    # print the value of the variables at the optimum
    for v in prob.variables():
        print(f'{v.name} = {v.varValue:5.2f}')

    print()
    print("Sensitivity Analysis\nConstraint\t\t\t  Shadow Price\t\tSlack")
    for name, c in prob.constraints.items():
        print(f'{name} : {c}     \t{c.pi} \t\t{c.slack}')

if __name__ == "__main__":
    solver()
```

Οπότε παίρνω το παρακάτω αποτέλεσμα.

```
Status: Optimal
objective = 28.0
x1 = -3.00
x2 =  0.50
x3 =  3.50
x4 =  0.00
x5 =  0.00

Sensitivity Analysis
Constraint                          Shadow Price        Slack
c1 : x2 + x3 + x4 - 2*x5 <= 4           6.0             -0.0
c2 : x1 - x2 + x3 + 2*x4 + x5 >= 0      -1.0            -0.0
c3 : x1 + x2 + x3 - 3*x5 <= 1           4.0             -0.0
```

Δηλαδή η βέλτιστη τιμή της αντικειμενικής συνάρτησης είναι $z^* = 28$ και οι μεταβλητές απόφασεις παίρνουν τιμές $x_1 = -3,\ x_2 = 0.5,\ x_3 = 3.5,\ x_4 = 0, x_5 = 0$. Η βασική λύση είναι $x_B = B^{-1}b = [-3, 0.5, 3.5]^T$ οπότε οι βασικές μεταβλητές είναι οι $\{x_1, x_2, x_3\}$ και οι ελεύθερες είναι οι μηδενικές $x_N = \{x_4, x_5, x_6, x_7, x_8\}$.

Ο βέλτιστος βασικός πίνακας σχηματίζεται από τις στήλες των βασικών μεταβλητών του πίνακα [A | I_3]

$$B^* = \begin{matrix} 0 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & 1 \end{matrix} \qquad N = \begin{matrix} 1 & -2 & 1 & 0 & 0 \\ 2 & 1 & 0 & -1 & 0 \\ 0 & -3 & 0 & 0 & 1 \end{matrix}$$

Οι δεσμευτικοί περιορισμοί είναι εκείνοι για τους οποίους ικανοποιείται ισότητα στην βέλτιστη κορυφή. Παρατηρούμε ότι και οι τρεις είναι δεσμευτικοί, αφού οι μεταβλητές χαλάρωσης είναι όλες μηδενικές

Η βέλτιστη κορυφή στο R^5 ορίζεται από την τομή 5 ακριβώς υπερεπίπεδων, τα τρία των δεσμευτικών περιορισμών και τα $x_4 = 0,\ x_5 = 0$. Άρα πρόκειται για μη εκφυλισμένη κορυφή.

(β) Εφαρμόζω μία διαταραχή γ στον συντελεστή c_1 της βασικής μεταβλητής x_1. Για να παραμείνει η βέλτιστη λύση στην ίδια κορυφή πρέπει οι αντικειμενικοί συντελεστές των μη βασικών μεταβλητών να παραμείνουν αρνητικοί δηλαδή $c_N^T - (c_B + \gamma e_1)^T B^{-1} N \leq 0$

$$z\,(c_B + \gamma e_1) = c_B^T B^{-1} b + \left[c_N^T - (c_B + \gamma e_1)^T B^{-1} N\right] x_N$$

$$c_B^T = [3, 11,\ 9] \quad c_N^T = [-1, -29, 0, 0, 0] \qquad B^{-1} = \begin{matrix} -1 & 0 & 1 \\ 0 & -0.5 & 0.5 \\ 1 & 0.5 & -0.5 \end{matrix}$$

$$[-1, -29, 0, 0, 0] - [3 + \gamma, 11, 9]\, B^{-1} N \leq 0 \Rightarrow$$

$$[-1, -29, 0, 0, 0] - [4 - \gamma, -25 - \gamma, -12 - \gamma, 1\ , 4 + \gamma] \leq 0 \Rightarrow\ -4 \leq \gamma \leq 4 \Rightarrow$$

$$-1 \leq\ c_1 \leq\ 7$$

Αν ο συντελεστής c_1 είναι μέσα στο διάστημα ανοχής [-1,7] τότε η βέλτιστη κορυφή δεν αλλάζει, ενώ η αντικειμενική συνάρτηση μεταβάλλεται κατά $\Delta z = \gamma e_1^T x_B = -3\delta$

Εφαρμόζω μία διαταραχή γ στον συντελεστή c_4 της βασικής μεταβλητής x_4. Για να παραμείνει η βέλτιστη λύση στην ίδια κορυφή πρέπει οι αντικειμενικοί συντελεστές των μη βασικών μεταβλητών να παραμείνουν αρνητικοί δηλαδή $(c_N + \gamma e_1)^T - c_B^T B^{-1} N \leq 0$

$[-1+\gamma, -29, 0, 0, 0] - [4, -25, -12, 1, 4] \leq 0 \Rightarrow \gamma \leq 5 \Rightarrow c_4 \leq 4$

Αν ο συντελεστής c_1 είναι μέσα στο διάστημα ανοχής [-inf, 4] τότε η βέλτιστη κορυφή δεν αλλάζει, ενώ η αντικειμενική συνάρτηση επίσης δεν αλλάζει αφού $\Delta z = \gamma e_1^T x_N,\ x_N = 0$

(γ) Εφαρμόζω μία διαταραχή γ στον δεσμευτικό περιορισμό 1

$x_2 + x_3 + x_4 - 2x_5 \leq 4 + \gamma$

Η βέλτιστη λύση μπορεί να αλλάξει, όμως μας ενδιαφέρει ο ρυθμός μεταβολής της αντικείμενικής συνάρτησης καθώς μεταβάλλεται το b_1 (σκιώδες κόστος περιορισμού)

Το διάστημα ανοχής για τον περιορισμό 1 ορίζεται από το κριτήριο εφικτότητας δηλ.

$B^{-1}(b + \gamma e_1) \geq 0 \Rightarrow [-3, 0.5, 3.5]^T + \gamma\,[-1, 0, 1]^T \geq 0 \Rightarrow -3.5 \leq \gamma \leq -3$

Για αυτό το διάστημα τιμών του γ, η z μεταβάλλεται σταθερά με ρυθμό όσο η βέλτιστη τιμή της αντίστοιχης δυικής μεταβλητής $y_1 = 6$

Εφαρμόζω μία διαταραχή γ στον χαλαρό περιορισμό προσήμου $x_2 \geq \gamma$

Το σκιώδες κόστος του περιορισμού αυτού είναι μηδέν, επειδή η μεταβλητή χαλάρωσης του είναι μη μηδενική και μέσω συμπληρωματικής χαλαρότητας η αντίστοιχη δυική μεταβλητή είναι 0. Άρα για το διάστημα ανοχής $(B^{-1}b) \geq \gamma e_2$ => $(-3, 0.5, 3.5) \geq (0, \gamma, 0)$

$\gamma \leq 0.5$ η βέλτιστη κορυφή δεν αλλάζει και η αντικειμενική συνάρτηση μένει σταθερή.

(δ) Εφαρμόζοντας τους κανόνες μετασχηματισμού πρωτεύοντος σε δυικό πρόβλημα που φαίνονται παρακάτω προκύπτει το δυικό πρόβλημα

maximization	minimization
Πίνακας συντελεστών: $\mathbf{A}$	Πίνακας συντελεστών: $\mathbf{A}^T$
Διάνυσμα δεξιάς πλευράς: $\mathbf{b}$	Αντικειμενικοί συντελεστές: $\mathbf{b}$
Αντικειμενικοί συντελεστές: $\mathbf{c}$	Διάνυσμα δεξιάς πλευράς: $\mathbf{c}$
Ο j περιορισμός είναι με " $=$ "	Η y_j μεταβλητή είναι ελεύθερη $(y_j \in \mathbb{R})$
Ο j περιορισμός είναι με " $\leq$ "	Η y_j μεταβλητή είναι μη-αρνητική $(y_j \geq 0)$
Ο j περιορισμός είναι με " $\geq$ "	Η y_j μεταβλητή είναι μη-θετική $(y_i \leq 0)$
Η x_i μεταβλητή είναι ελεύθερη	Ο i περιορισμός είναι με " $=$ "
Η x_i μεταβλητή είναι μη-αρνητική	Ο i περιορισμός είναι με " $\geq$ "
Η x_i μεταβλητή είναι μη-θετική	Ο i περιορισμός είναι με " $\leq$ "

$\min\{4y_1 + y_3\}$
$y_2 + y_3 = 3$
$y_1 - y_2 + y_3 \geq 11$
$y_1 + y_2 + y_3 \geq 9$
$y_1 + 2y_2 \geq -1$
$-2y_1 + y_2 - 3y_3 \geq -29$
$y_1, y_3 \geq 0\,, y_2 \leq 0$

Εισάγοντας μεταβλητές χαλάρωσης έχουμε:

$\min\{4y_1 + y_3\}$
$y_2 + y_3 + y_4 = 3$
$y_1 - y_2 + y_3 + y_5 = 11$
$y_1 + y_2 + y_3 + y_6 = 9$
$y_1 + 2y_2 + y_7 = -1$
$-2y_1 + y_2 - 3y_3 + y_8 = -29$
$y_1, y_3 \geq 0\,, y_2 \leq 0$

Κώδικας 1δ

```python
import pulp

def solver():
    # A LP problem
    prob = pulp.LpProblem("exersice1d", pulp.LpMinimize)

    # Variables
    y1 = pulp.LpVariable("y1", 0, None)
    y2 = pulp.LpVariable("y2", None, 0)
    y3 = pulp.LpVariable("y3", 0, None)

    # Objective
    prob += 4*y1 + y3 , "obj"

    # Constraints
    prob += y2 + y3 == 3,  "c1"
    prob += y1 - y2 + y3 >= 11, "c2"
    prob += y1 + y2 + y3 >= 9, "c3"
    prob += y1 + 2*y2 >= -1, "c4"
    prob += -2*y1 + y2 -3*y3 >= -29, "c5"

    # solve the problem using the default solver
    prob.solve()

    # print the status of the solved LP
    print("Status:", pulp.LpStatus[prob.status])
    # print the value of the objective
    print("objective =", pulp.value(prob.objective))

    # print the value of the variables at the optimum
    for v in prob.variables():
        print(f'{v.name} = {v.varValue:5.2f}')

    print("\nSensitivity Analysis")
    print("{:<30} {:<15} {:<15}".format("Constraint", "Shadow Price", "Slack"))
    for name, c in prob.constraints.items():
        print("{:<30} {:<15} {:<15}".format(
            f"{name} : {c}",
            str(c.pi),
            str(c.slack)
        ))

if __name__ == "__main__":
    solver()
```

Παρατηρούμε την λύση του δυικού, που έχει ίδια βέλτιστη τιμή αντικειμενικής συνάρτησης $z^* = 28$ με το πρωτεύον.

```
Status: Optimal
objective = 28.0
y1 =  6.00
y2 = -1.00
y3 =  4.00

Sensitivity Analysis
Constraint                    Shadow Price    Slack
c1 : y2 + y3 = 3              -3.0            -0.0
c2 : y1 - y2 + y3 >= 11       0.5             -0.0
c3 : y1 + y2 + y3 >= 9        3.5             -0.0
c4 : y1 + 2*y2 >= -1          0.0             -5.0
c5 : -2*y1 + y2 - 3*y3 >= -29 0.0             -4.0
```

Συμπληρωματικά ζεύγη μεταβλητών στη βέλτιστη λύση

$x_1^* = -3$	$y_4^* = 0$
$x_2^* = 0.5$	$y_5^* = 0$
$x_3^* = 3.5$	$y_6^* = 0$
$x_4^* = 0$	$y_7^* = -5$
$x_5^* = 0$	$y_8^* = -4$
$x_6^* = 0$	$y_1^* = 6$
$x_7^* = 0$	$y_2^* = -1$
$x_8^* = 0$	$y_3^* = 4$

Άρα επαληθεύται το θεώρημα συμπληρωματικής χαλαρότητας, δηλαδή $y^{*T}x_s^* = 0$ και $y_s^{*T}x^* = 0$

Άσκηση 2

(α) Εφαρμόζοντας τους κανόνες μετασχηματισμού πρωτεύοντος σε δυικό πρόβλημα προκύπτει το δυικό πρόβλημα

$min\, z = 6y_2 + 3y_3$
$2y_1 - y_2 + 3y_3 \geq 1$
$3y_1 + y_2 + y_3 \leq 1$
$y_1 + 2y_2 + 4y_3 = 0$
$y_1 + y_2 + 2y_3 \leq 0$
$y_2 \epsilon \mathbb{R},\ y_1 \leq 0, y_3 \geq 0$

Χρησιμοποιώντας το module `pulp` που χρησιμοποιεί τον `CBC MILP solver`, λύνω το πρόβλημα LP.

```
Status: Optimal
objective = -1.80
y1 =  0.00
y2 = -0.40
y3 =  0.20

Sensitivity Analysis
Constraint                    Shadow Price    Slack
c1 : 2*y1 - y2 + 3*y3 >= 1    -1.8            -0.0
c2 : 3*y1 + y2 + y3 <= 1      -0.0            1.2
c3 : y1 + 2*y2 + 4*y3 = 0     2.1             -0.0
c4 : y1 + y2 + 2*y3 <= 0      -0.0            -0.0
```

Η βέλτιστη βασική λύση αποτελείται από 4 μεταβλητές, τις μη μηδενικές $y_2,\ y_3,\ s_2$ και μία μηδενική από τις y_1, s_1, s_3, s_4. Άρα πρόκειται για εκφυλισμένη βασική λύση.

Οι Π1, Π3, Π4 είναι δεσμευτικοί περιορισμοί και ο περιορισμός προσήμου για την y_1 είναι δεσμευτικός. Η βέλτιστη κορυφή στο R^3 σχηματίζεται από την τομή 4 υπερεπιπέδων, τα τρία των δεσμευτικών περιορισμών και το $y_1 = 0$, άρα πρόκειται για εκφυλισμένη κορυφή.

(β) Με βάση τις συνθήκες συμπληρωματικής χαλαρότητας

1. Αν ο περιορισμός i του δυϊκού (πρωτεύοντος) ικανοποιείται ως γνήσια ανισότητα (μη δεσμευτικός περιορισμός), τότε η i μεταβλητή του πρωτεύοντος (δυϊκού) είναι υποχρεωτικά μηδέν,
2. Αν η j μεταβλητή του δυϊκού (πρωτεύοντος) είναι θετική τότε ο j περιορισμός του πρωτεύοντος (δυϊκού) ικανοποιείται υποχρεωτικά ως ισότητα (δεσμευτικός περιορισμός).

Ο Π2 του δυικού είναι μη δεσμευτικός, οπότε η $x_2 = 0 \xrightarrow{z = x_2 + x_1 = -1.8} x_1 = -1.8$

Η $y_3 > 0$, οπότε ο Π3 του πρωτεύοντος είναι δεσμευτικός στη βέλτιστη λύση

$$\left.\begin{array}{l} \Pi 3\colon 3x_1 + x_2 + 4x_3 + 2x_4 = 3 \Rightarrow 4x_3 + 2x_4 = 8.4 \\ \Pi 2\colon -x_1 + x_2 + 2x_3 + x_4 = 6 \Rightarrow 2x_3 + x_4 = 4.2 \end{array}\right\}$$ γραμμικώς εξαρτημένα επίπεδα

$$\Pi 1\colon x_3 + x_4 \leq 3.6$$

Οπότε $x_3 \geq 0.6$ *και* $x_4 \leq 3$ *τ. ω να ισχύουν οι* $\Pi 2, \Pi 1, x_2 = 0, x_1 = -1.8$, δηλαδή το πρωτεύον έχει πολλαπλές βέλτιστες λύσεις

Βέλτιστη λύση πρωτεύοντος		**Βέλτιστη λύση δυϊκού**
Πολλαπλές λύσεις	⇒	Εκφυλισμένη λύση
Μοναδική μη-εκφυλισμένη	⇒	Μοναδική μη-εκφυλισμένη
Πολλαπλές μη-εκφυλισμένες	⇒	Μοναδική εκφυλισμένη
Μοναδική εκφυλισμένη	⇒	Πολλαπλές λύσεις

Κώδικας 2α

```python
import pulp

def solver():
    # A LP problem
    prob = pulp.LpProblem("ex2", pulp.LpMaximize)

    # Variables
    y1 = pulp.LpVariable("y1", None, 0)
    y2 = pulp.LpVariable("y2", None, None)
    y3 = pulp.LpVariable("y3", 0, None)

    # Objective
    prob += 6*y2 + 3*y3 , "obj"

    # Constraints
    prob += 2*y1 - y2 +3*y3 >= 1,  "c1"
    prob += 3*y1 + y2 + y3 <= 1, "c2"
    prob += y1 + 2*y2 + 4*y3 == 0, "c3"
    prob += y1 + y2 + 2*y3 <= 0, "c4"

    # solve the problem using the default solver
    prob.solve()

    # print the status of the solved LP
    print("Status:", pulp.LpStatus[prob.status])
    # print the value of the objective
    print(f"objective = {pulp.value(prob.objective):5.2f}")

    # print the value of the variables at the optimum
    for v in prob.variables():
        print(f'{v.name} = {v.varValue:5.2f}')

    print("\nSensitivity Analysis")
    print("{:<30} {:<15} {:<15}".format("Constraint", "Shadow Price", "Slack"))
    for name, c in prob.constraints.items():
        print("{:<30} {:<15} {:<15}".format(
            f"{name} : {c}",
            str(c.pi),
            str(c.slack)
        ))

if __name__ == "__main__":
    solver()
```

Άσκηση 3

$\max z = 6x_1 + x_2 - x_3 - x_4$
$x_1 + 2x_2 + x_3 + x_4 \leq 5$
$3x_1 + x_2 - x_3 \leq 8$
$x_2 + x_3 + x_4 = 1$
$x_1, x_2 \epsilon \mathbb{R},\ x_3, x_4 \geq 0$

Εξετάζω τη λύση $x = (3, -1, 0, 2)$ με βάση το παρακάτω θεώρημα από το βιβλίο των Sierksma and Zwols για τη συνθήκη βελτιστότητας.

Theorem 4.2.5. (*Optimality condition for nonstandard LO-models*)
The vector $\mathbf{x}$ is an optimal solution of the nonstandard LO-model (GM) if

(i) $\mathbf{x}$ is feasible, and

(ii) there exists a vector $\mathbf{y}$ that is feasible for (DGM) with $\mathbf{c}^T\mathbf{x} = \mathbf{b}^T\mathbf{y}$.

Η τιμή της αντικειμενικής συνάρτησης για $x = (3, -1, 0, 2)$ είναι z = 15 και ελέγχω αν ισχύουν οι περιορισμοί

$\Pi 1 : 3 - 2 + 0 + 2 = 3 \leq 5$	μη δεσμευτικός $x_5 = 2$
$\Pi 2 : 3 * 3 - 1 - 0 = 8$	δεσμευτικός $x_6 = 0$
$\Pi 3 : -1 + 0 + 2 = 1$	δεσμευτικός $x_7 = 0$

Άρα ικανοποιούνται όλοι οι περιορισμοί, οπότε το $x = (3, -1, 0, 2)$ πρόκειται για εφικτή λύση.

Το δυικό πρόβλημα φαίνεται παρακάτω

$\min\ z = 5y_1 + 8y_2 + y_3$
$\Pi 1: y_1 + 3y_2 = 6$
$\Pi 2: 2y_1 + y_2 + y_3 = 1$
$\Pi 3: y_1 - y_2 + y_3 \geq -1$
$\Pi 4: y_1 + y_3 \geq -1$
$y_3 \epsilon \mathbb{R}, \quad y_1, y_2 \geq 0$

Ο Π1 του πρωτεύοντος είναι μη δεσμευτικός, άρα από συμπληρωματική χαλαρότητα ισχύει $y_1 = 0$.

Από τον Π1 του δυικού προκύπτει ότι $y_2 = 2$ και έπειτα από τον Π2 του δυικού προκύπτει ότι $y_3 = -1$

Όμως ο Π3 του δυικού για τη λύση $y = (0, 2, -1)$ δεν ικανοποιείται, οπότε δεν είναι εφικτή.

Τελικά αφού δεν βρέθηκε εφικτή λύση y στο δυικό μέσω συμπληρωματικής χαλαρότητας για την οποία να ισχύει b^Ty =15 σημαίνει ότι η λύση $x = (3, -1, 0, 2)$ δεν είναι βέλτιστη στο πρωτεύον.

Άσκηση 4

(α) Μοντελοποιήση του προβλήματος ακέραιου γραμμικού προγραμματισμού.

Μέρες εργασίας Βάρδια	Αριθμός σερβιτόρων (μεταβλητή απόφασης)
Κυριακή - Πέμπτη	x_1
Δευτέρα - Παρασκευή	x_2
Τρίτη - Σάββατο	x_3
Τετάρτη - Κυριακή	x_4
Πέμπτη - Δευτέρα	x_5
Παρασκευή - Τρίτη	x_6
Σάββατο - Τετάρτη	x_7

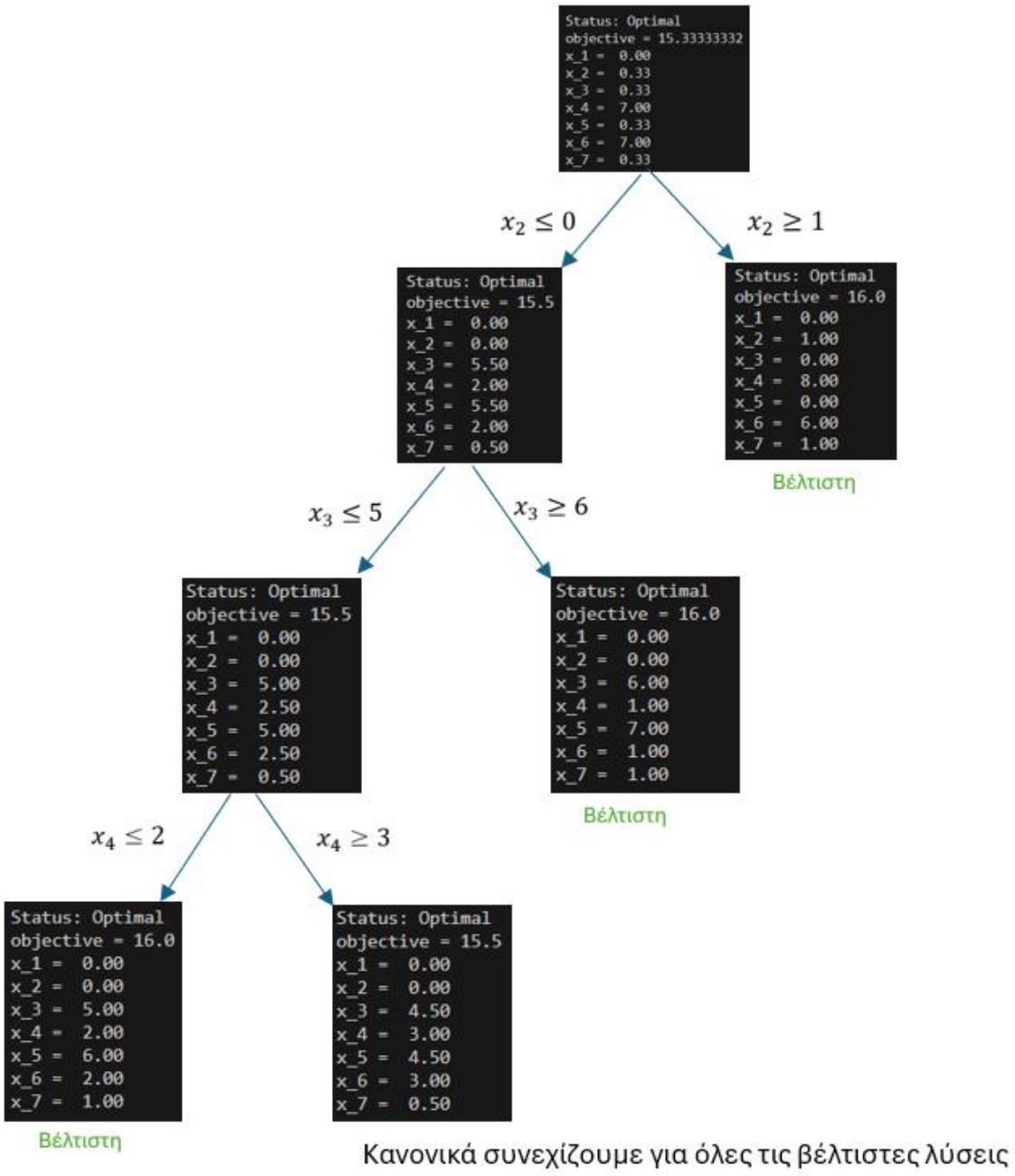
Περιορισμοί

Κυριακή	$x_1 + x_4 + x_5 + x_6 + x_7 \geq 10$
Δευτέρα	$x_1 + x_2 + x_5 + x_6 + x_7 \geq 8$
Τρίτη	$x_1 + x_2 + x_3 + x_6 + x_7 \geq 8$
Τετάρτη	$x_1 + x_2 + x_3 + x_4 + x_7 \geq 8$
Πέμπτη	$x_1 + x_2 + x_3 + x_4 + x_5 \geq 8$
Παρασκευή	$x_2 + x_3 + x_4 + x_5 + x_6 \geq 15$
Σάββατο	$x_3 + x_4 + x_5 + x_6 + x_7 \geq 15$
	$x_i \geq 0 \ \forall i$
	x_i *ακέραιος* $\forall i$

Πρόβλημα ελαχιστοποίησης του πλήθους των σερβιτόρων άρα η αντικειμενική συνάρτηση

$\min\{ x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 \}$

(β) Αρχικά χαλαρώνουμε τον περιορισμό για ακέραιες λύσεις και λύνουμε το πρόβλημα γ.π χρησιμοποιώντας το module `pulp` που χρησιμοποιεί τον `CBC MILP solver`.



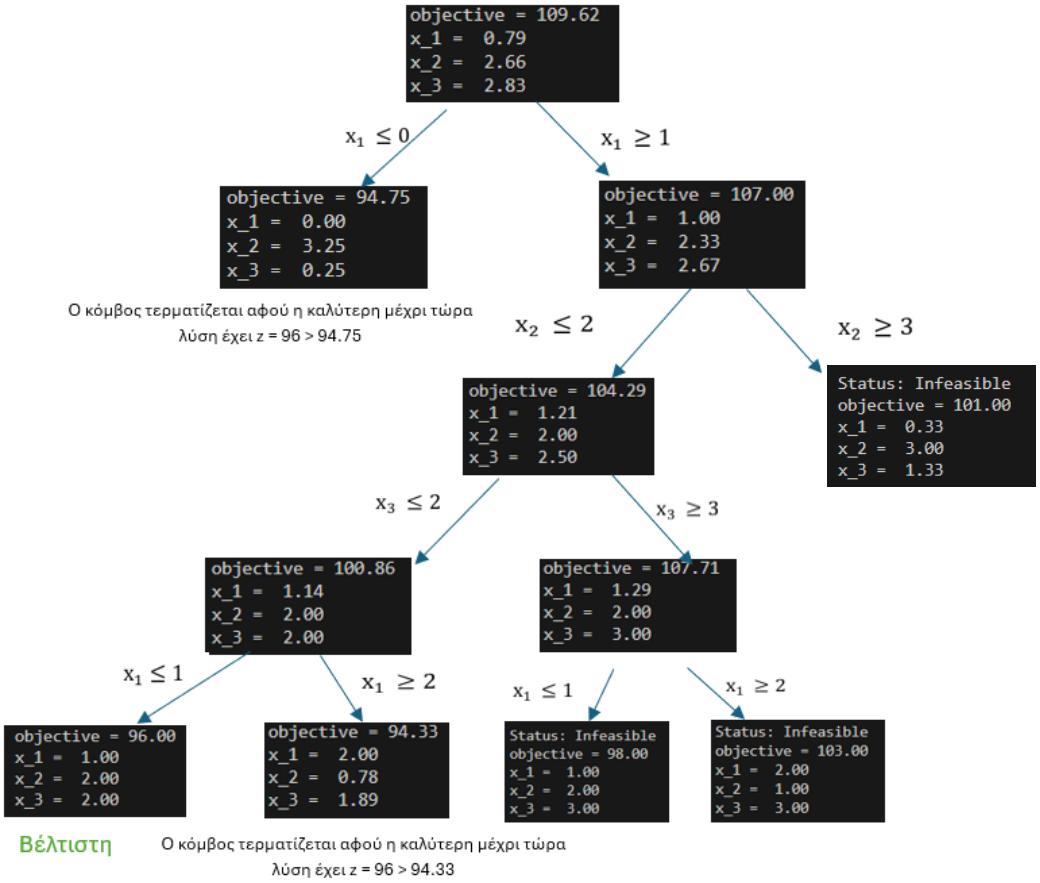
Με τη μέθοδο Branch and Bound βρέθηκαν 3 βέλτιστες λύσεις με z =16, οπότε σταματάμε τη διακλάδωση

Κώδικας 4

```python
import pulp
def solver():
    # A LP problem
    prob = pulp.LpProblem("dovetail", pulp.LpMinimize)

    #Variables
    x =  pulp.LpVariable.dicts('x', range(1,8),lowBound=0, upBound = None, cat=pulp.LpContinuous)

    # Objective
    prob += x[1]+x[2]+x[3]+x[4]+x[5]+x[6]+x[7], 'obj'

    prob += x[1] + x[4] + x[5] + x[6] + x[7] >= 10 #sunday
    prob += x[1] + x[2] + x[5] + x[6] + x[7] >= 8 #monday
    prob += x[1] + x[2] + x[3] + x[6] + x[7] >= 8 #tuesday
    prob += x[1] + x[2] + x[3] + x[4] + x[7] >= 8 #wednesday
    prob += x[1] + x[2] + x[3] + x[4] + x[5] >= 8 #thursday
    prob += x[2] + x[3] + x[4] + x[5] + x[6] >= 15 #friday
    prob += x[3] + x[4] + x[5] + x[6] + x[7] >= 15 #saturday
    ##1o branching
    prob += x[2] <=0
    # prob += x[2] >=1

    ##2o branching
    prob += x[3]<= 5
    #prob += x[3]>= 6

    ##3o branching
    #prob += x[4]<= 2
    prob += x[4]>= 3
    # solve the problem using the default solver
    prob.solve()

    # print the status of the solved LP
    print("Status:", pulp.LpStatus[prob.status])

    # print the value of the objective
    print("objective =", pulp.value(prob.objective))

    # print the value of the variables at the optimum
    for v in prob.variables():
        print(f'{v.name} = {v.varValue:5.2f}')


if __name__=='__main__':
    solver()
```

Άσκηση 5

Αρχικά χαλαρώνουμε τον περιορισμό για ακέραιες λύσεις και λύνουμε το πρόβλημα γ.π. Επιλέγω τη x_1 για διόρθωση, οπότε προκύπτει το πρώτο branching και δημιουργούνται δύο νέα προβλήματα γ.π. Η επιλόγη του επόμενου κόμβου για διακλάδωση γίνεται με Jumptracking, δηλ. επιλέγεται αυτός με τη μεγαλύτερη τιμή αντικειμενικής συνάρτησης.


objective = 109.62
x_1 = 0.79
x_2 = 2.66
x_3 = 2.83
x1 ≤ 0
x1 ≥ 1
objective = 94.75
x_1 = 0.00
x_2 = 3.25
x_3 = 0.25
Ο κόμβος τερματίζεται αφού η καλύτερη μέχρι τώρα λύση έχει z = 96 > 94.75
objective = 107.00
x_1 = 1.00
x_2 = 2.33
x_3 = 2.67
x2 ≤ 2
x2 ≥ 3
objective = 104.29
x_1 = 1.21
x_2 = 2.00
x_3 = 2.50
Status: Infeasible
objective = 101.00
x_1 = 0.33
x_2 = 3.00
x_3 = 1.33
x3 ≤ 2
x3 ≥ 3
objective = 100.86
x_1 = 1.14
x_2 = 2.00
x_3 = 2.00
objective = 107.71
x_1 = 1.29
x_2 = 2.00
x_3 = 3.00
x1 ≤ 1
x1 ≥ 2
x1 ≤ 1
x1 ≥ 2
objective = 96.00
x_1 = 1.00
x_2 = 2.00
x_3 = 2.00
objective = 94.33
x_1 = 2.00
x_2 = 0.78
x_3 = 1.89
Status: Infeasible
objective = 98.00
x_1 = 1.00
x_2 = 2.00
x_3 = 3.00
Status: Infeasible
objective = 103.00
x_1 = 2.00
x_2 = 1.00
x_3 = 3.00
Βέλτιστη
Ο κόμβος τερματίζεται αφού η καλύτερη μέχρι τώρα λύση έχει z = 96 > 94.33

Κώδικας 5

```python
import pulp
def solver():
    # A LP problem
    prob = pulp.LpProblem("ex5", pulp.LpMaximize)

    #Variables
    x =  pulp.LpVariable.dicts('x', range(1,4),lowBound=0, upBound = None, cat=pulp.LpContinuous)

    # Objective
    prob += 34*x[1]+29*x[2]+2*x[3], 'obj'

    prob += 7*x[1] + 5*x[2] - x[3] <= 16
    prob += -x[1] + 3*x[2] + x[3] <= 10
    prob += -x[2] + 2*x[3] <= 3

    ##1o branching
    #prob += x[1]<=0
    #prob += x[1]>=1

    ##2o branching
    #prob += x[2]<=2
    #prob += x[2]>=3

    ##3o branching
    #prob += x[3]<=2
    # prob += x[3]>=3

    ##4o branching
    #prob += x[1]<=1
    # prob += x[1]>=2

    ##5o branching
    #prob += x[3]<=1
    #prob += x[3]>=2

    ##6o branching
    #prob += x[1]<=1
    #prob += x[1]>=2

    ##7o branching
    #prob += x[2]<=3
    #prob += x[2]>=4
```

```python
    ##8o branching
    #prob += x[2]<=0
    #prob += x[2]>=1

    # solve the problem using the default solver
    prob.solve()

    # print the status of the solved LP
    print("Status:", pulp.LpStatus[prob.status])

    # print the value of the objective
    print(f"objective = {pulp.value(prob.objective):5.2f}")

    # print the value of the variables at the optimum
    for v in prob.variables():
        print(f'{v.name} = {v.varValue:5.2f}')


if __name__=='__main__':
    solver()
```

Άσκηση 6

(α)

Δέμα	Όγκος	Κέρδος
1	2	10
2	3	14
3	4	31
4	6	48
5	8	60

Μοντελοποίηση: Έστω η δυαδική μεταβλητή $x_i, i = 1, \dots, 5$ που παίρνει τιμή 1 όταν το δέμα παραδίδεται και 0 στην αντίθετη περίπτωση.

Πρόβλημα μεγιστοποίησης της αντικειμενικής συνάρτησης

$z = 10x_1 + 14x_2 + 31x_3 + 48x_4 + 60x_5$

Περιορισμοί

$2x_1 + 3x_2 + 4x_3 + 6x_4 + 8x_5 \leq 11, \qquad x_i \in \{0,1\}\ \forall i$

Πρόκειται για πρόβλημα τύπου Knapsack

Διατάσσω τα δέματα με βάση το κέρδος τους ανά μονάδα όγκου, και ορίζω νέες μεταβλητές σύμφωνα με τη διάταξη. Χαλαρώνω τον περιορισμό για δυικές μεταβλητές.

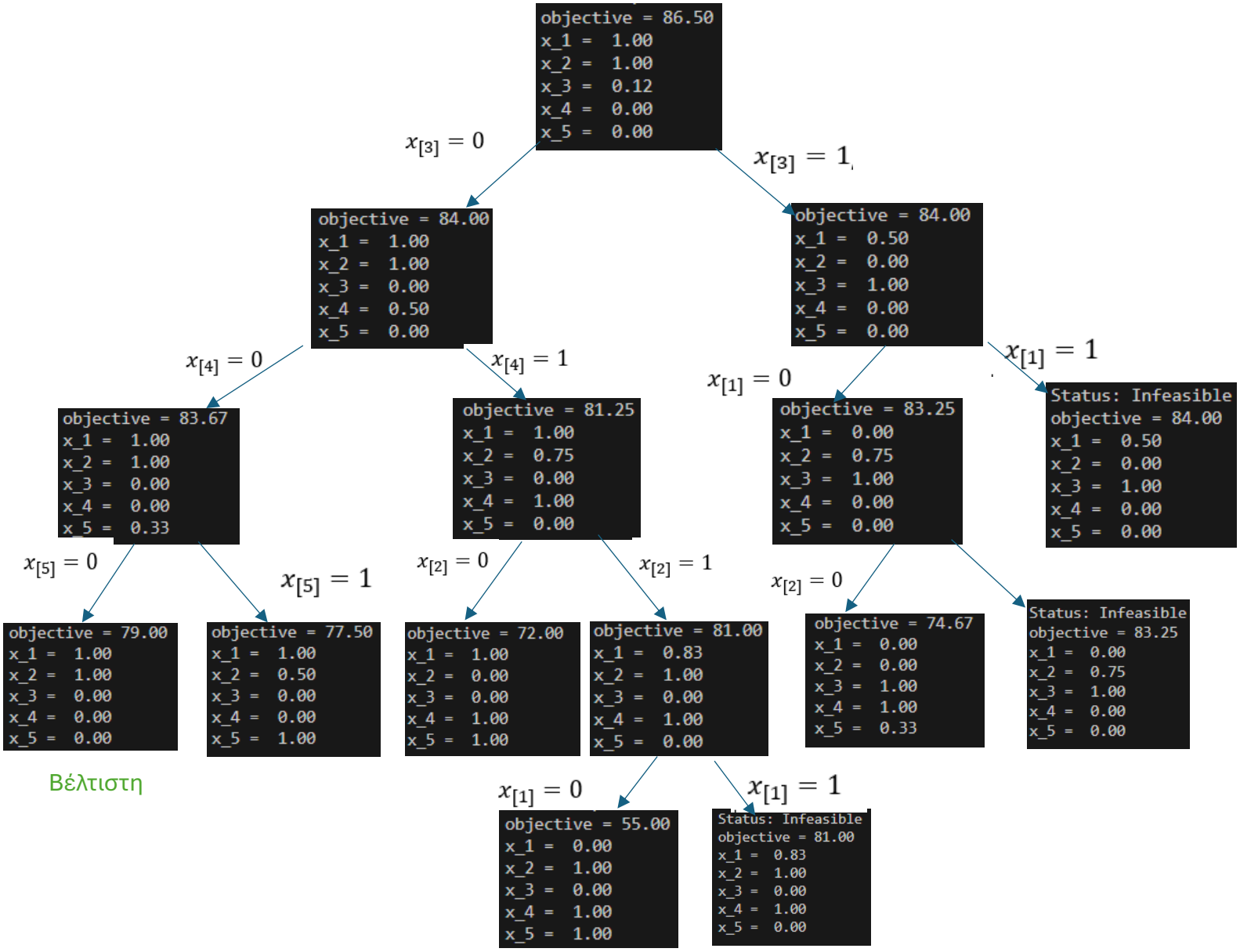
Δέμα	Κέρδος/Όγκος	Σειρά
1	5	$x_{[4]}$
2	14/3	$x_{[5]}$
3	7.75	$x_{[2]}$
4	8	$x_{[1]}$
5	7.5	$x_{[3]}$

Χαλαρωμένο πρόβλημα

$z = 10x_{[4]} + 14x_{[5]} + 31x_{[2]} + 48x_{[1]} + 60x_{[3]}$

$2x_{[4]} + 3x_{[5]} + 4x_{[2]} + 6x_{[1]} + 8x_{[3]} \leq 11, \qquad x_i \in [0,1]\ \forall i$

Προφανής λύση $x_{[1]} = 1,\ x_{[2]} = 1,\ x_{[3]} = \frac{1}{8}\ ,\ x_{[4]} = x_{[5]} = 0\ , z = 86.5$



Οι υπόλοιποι κόμβοι τερματίζονται γιατί η καλύτερη μέχρι τώρα λύση έχει $z = 79 > z_u$ του κάθε κόμβου

Άρα η βέλτιστη λύση είναι να παραδοθούν τα δέματα 3, 4.

Κώδικας 6α

```python
import pulp
def solver():
    # A LP problem
    prob = pulp.LpProblem("ex6", pulp.LpMaximize)

    #Variables
    x =  pulp.LpVariable.dicts('x', range(1,6),lowBound=0, upBound = 1, cat=pulp.LpContinuous)

    # Objective
    prob += 10*x[4]+14*x[5]+31*x[2]+48*x[1]+60*x[3], 'obj'

    prob += 2*x[4] + 3*x[5] + 4*x[2] + 6*x[1] + 8*x[3] <= 11

    #1o branching
    prob += x[3]==0

    #2o branching
    prob += x[4]==1

    #3o branching
    prob += x[1]==1

    #4o branching
    #prob += x[5]==1

    #5o branching
    #prob += x[2]==1

    #6o branching
    prob += x[2] ==1

    # solve the problem using the default solver
    prob.solve()

    # print the status of the solved LP
    print("Status:", pulp.LpStatus[prob.status])

    # print the value of the objective
    print(f"objective = {pulp.value(prob.objective):5.2f}")

    # print the value of the variables at the optimum
    for v in prob.variables():
        print(f'{v.name} = {v.varValue:5.2f}')

if __name__=='__main__':
    solver()
```

(β) Το χαλαρωμένο πρόβλημα μαζί με το δυικό του φαίνονται παρακάτω:

$\max z = 10x_{[4]} + 14x_{[5]} + 31x_{[2]} + 48x_{[1]} + 60x_{[3]}$
$2x_{[4]} + 3x_{[5]} + 4x_{[2]} + 6x_{[1]} + 8x_{[3]} \leq 11$
$x_{[1]} \leq 1$
$x_{[2]} \leq 1$
$x_{[3]} \leq 1$
$x_{[4]} \leq 1$
$x_{[5]} \leq 1$
$x_i \geq 0 \forall i$

$\min z = 11y_1 + y_2 + y_3 + y_4 + y_5 + y_6$
$6y_1 + y_2 \geq 48$
$4y_1 + y_3 \geq 31$
$8y_1 + y_4 \geq 60$
$2y_1 + y_5 \geq 10$
$3y_1 + y_6 \geq 14$
$y_i \geq 0 \forall i$

Στη βέλτιστη λύση του πρωτεύοντος ισχύει

$$x_{[1]} = 1,\ x_{[2]} = 1,\ x_{[3]} = \frac{1}{8},\ x_{[4]} = \ x_{[5]} = 0\ , z = 86.5$$

Άρα με βάση τις συνθήκες συμπληρωματικής χαλαρότητας

1. Αν ο περιορισμός i του δυϊκού (πρωτεύοντος) ικανοποιείται ως γνήσια ανισότητα (μη δεσμευτικός περιορισμός), τότε η i μεταβλητή του πρωτεύοντος (δυϊκού) είναι υποχρεωτικά μηδέν,
2. Αν η j μεταβλητή του δυϊκού (πρωτεύοντος) είναι θετική τότε ο j περιορισμός του πρωτεύοντος (δυϊκού) ικανοποιείται υποχρεωτικά ως ισότητα (δεσμευτικός περιορισμός).

Οι Π1, Π2, Π3 του δυικού είναι δεσμευτικοί και $y_4 = \ y_5 = y_6 = 0$,

Οπότε βρίσκουμε επίσης $y_1 = 7.5, y_2 \ = 3, y_3 \ = 1$

Η λύση αυτή είναι εφικτή και δίνει z =86.5, οπότε οι δύο λύσεις είναι βέλτιστες στα προβλήματα.

(γ) Η βέλτιστη ακέραια λύση του δυικού είναι

```
Status: Optimal
objective = 88.00
y_1 =  8.00
y_2 =  0.00
y_3 =  0.00
y_4 =  0.00
y_5 =  0.00
y_6 =  0.00
```

Παρατηρώ ότι στο πρωτεύον ακέραιο πρόβλημα η βέλτιστη λύση έχει z = 79, ενώ στο δυικό ακέραιο έχει z = 88

Επομένως δεν ισχύουν οι συνθήκες συμπληρωματικής χαλαρότητας για το ακέραιο πρωτεύον και το δυικό του.

Κώδικας 6γ

```python
import pulp
def solver():
    # A LP problem
    prob = pulp.LpProblem("ex6c", pulp.LpMinimize)

    #Variables
    y =  pulp.LpVariable.dicts('y', range(1,7),lowBound=0, upBound = None, cat=pulp.LpInteger)

    # Objective
    prob += 11*y[1]+y[2]+y[3]+y[4]+y[5]+y[6], 'obj'

    prob += 6*y[1] + y[2] >=48
    prob += 4*y[1] + y[3] >=31
    prob += 8*y[1] + y[4] >=60
    prob += 2*y[1] + y[5] >=10
    prob += 3*y[1] + y[6] >=14


    # solve the problem using the default solver
    prob.solve()

    # print the status of the solved LP
    print("Status:", pulp.LpStatus[prob.status])

    # print the value of the objective
    print(f"objective = {pulp.value(prob.objective):5.2f}")

    # print the value of the variables at the optimum
    for v in prob.variables():
        print(f'{v.name} = {v.varValue:5.2f}')

if __name__=='__main__':
    solver()
```