ΔΕΥΤΕΡΗ ΕΡΓΑΣΙΑ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής

Μάθημα:

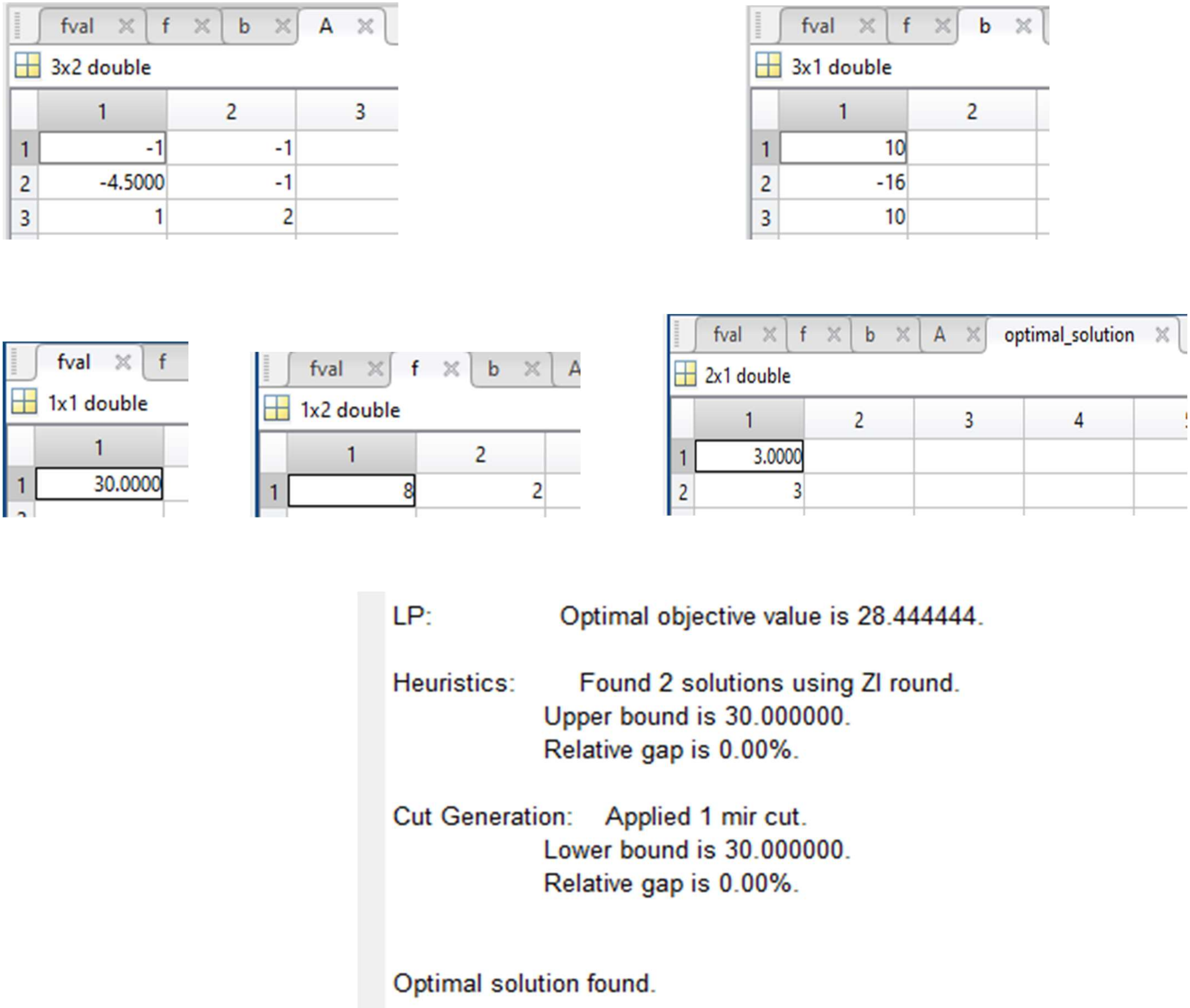
«Ειδικά Θέματα Επιχειρησιακής Έρευνας»

ΑΝΑΣΤΑΣΙΑ ΙΩΑΝΝΑ ΜΕΞΑ

Π18101

Άσκηση 1:

Τρέχοντας το αρχείο Matlab που μας δώσατε, έλαβα τα παρακάτω αποτελέσματα που φαίνονται στα screenshots.



Εικόνα 1: Αποτελέσματα Matlab

Επομένως, το μοντέλο που δημιουργείται έχει την εξής μορφή:

$$\min z = 8x_1 + 2x_2$$

$$\upsilon.\pi.$$

$$x_1 + x_2 \geq -10$$

$$4.5x_1 + x_2 \geq 16$$

$$x_1 + x_2 \leq 10$$

$$x_1, x_2 \geq 0$$

$$x_1, x_2 \in \mathbb{N}$$

Στο αρχείο Matlab *«exercise1.m»* έχω γράψει κώδικα που λύνει το παραπάνω πρόβλημα ακέραιου προγραμματισμού χρησιμοποιώντας τον LP solve, για να επαληθεύσω τα αποτελέσματα που μου επέστρεψε το *«Random_Generator_of_Integer_Programs.m».* Για να επιλύσω το παραπάνω μοντέλο με την μέθοδο Branch and Bound, χρησιμοποίησα το Excel και τον solver του. Το αρχείο Excel είναι *«exercise1.xlsx».*

Παρακάτω αναλύεται η διαδικασία επίλυσης με την Branch and Bound.

Αρχικά, πρέπει να μετατρέψω το πρόβλημα ακεραίου προγραμματισμού σε γραμμικό πρόβλημα. Άρα το μοντέλο μετατρέπεται σε:

$$\min z = 8x_1 + 2x_2$$

$$\upsilon.\pi.$$

$$x_1 + x_2 \geq -10$$ Υπο-πρόβλημα 1

$$4.5x_1 + x_2 \geq 16$$

$$x_1 + x_2 \leq 10$$

$$x_1, x_2 \geq 0$$

Η λύση του υπο-προβλήματος 1 είναι $z^* = 28.444$, $x_1^* = 3.555$, $x_2^* = 0$. Η λύση δεν είναι ακέραια, άρα συνεχίζουμε. Θα πρέπει να εισάγουμε τους περιορισμούς $x_1 \geq 4$ και $x_1 \leq 3$, επομένως δημιουργούμε ακόμα δύο υπο-προβλήματα τα 2 και 3.

Το μοντέλο του υπο-προβλήματος 2 είναι:

$$\min z = 8x_1 + 2x_2$$

$$\upsilon.\pi.$$

$$x_1 + x_2 \geq -10$$ Υπο-πρόβλημα 2

$$4.5x_1 + x_2 \geq 16$$

$$x_1 + x_2 \leq 10$$

$$x_1 \geq 4$$

$$x_1, x_2 \geq 0$$

Η λύση του υπο-προβλήματος 2 είναι $z^* = 32$, $x_1^* = 4$, $x_2^* = 0$. Η λύση είναι ακέραια, άρα σταματάμε. Πρέπει να λύσουμε το υπο-πρόβλημα 3.

Το μοντέλο του υπο-προβλήματος 3 είναι:

$$\min z = 8x_1 + 2x_2$$

$$\upsilon.\pi.$$

$$x_1 + x_2 \geq -10$$ Υπο-πρόβλημα 3

$$4.5x_1 + x_2 \geq 16$$

$$x_1 + x_2 \leq 10$$

$$x_1 \leq 3$$

$$x_1, x_2 \geq 0$$

Η λύση του υπο-προβλήματος 3 είναι $z^* = 29$, $x_1^* = 3$, $x_2^* = 2.5$. Η λύση δεν είναι ακέραια, άρα συνεχίζουμε. Θα πρέπει να εισάγουμε τους περιορισμούς $x_2 \geq 3$ και $x_2 \leq 2$, επομένως δημιουργούμε ακόμα δύο υπο-προβλήματα τα 4 και 5.

Το μοντέλο του υπο-προβλήματος 4 είναι:

$$\min z = 8x_1 + 2x_2$$

$$\upsilon.\pi.$$

$$x_1 + x_2 \geq -10 \qquad \text{Υπο-πρόβλημα 4}$$

$$4.5x_1 + x_2 \geq 16$$

$$x_1 + x_2 \leq 10$$

$$x_1 \leq 3$$

$$x_2 \geq 3$$

$$x_1, x_2 \geq 0$$

Η λύση του υπο-προβλήματος 4 είναι $z^* = 29.111$, $x_1^* = 2.888$, $x_2^* = 3$. Η λύση δεν είναι ακέραια, άρα συνεχίζουμε. Θα πρέπει να εισάγουμε τους περιορισμούς $x_1 \geq 3$ και $x_1 \leq 2$, επομένως δημιουργούμε ακόμα δύο υπο-προβλήματα τα 6 και 7.

Το μοντέλο του υπο-προβλήματος 6 είναι:

$$\min z = 8x_1 + 2x_2$$

$$\upsilon.\pi.$$

$$x_1 + x_2 \geq -10 \qquad \text{Υπο-πρόβλημα 6}$$

$$4.5x_1 + x_2 \geq 16$$

$$x_1 + x_2 \leq 10$$

$$x_1 = 3$$

$$x_2 \geq 3$$

$$x_1, x_2 \geq 0$$

Οι δύο περιορισμοί $x_1 \leq 3$ και $x_1 \geq 3$ γίνονται $x_1 = 3$.

Η λύση του υπο-προβλήματος 6 είναι $z^* = 30$, $x_1^* = 3$, $x_2^* = 3$. Η λύση είναι ακέραια, άρα σταματάμε. Πρέπει να λύσουμε το υπο-πρόβλημα 7.

Σημείωση: Το Matlab επέστρεψε αυτή την λύση του υπο-προβλήματος 6!

Το μοντέλο του υπο-προβλήματος 7 είναι:

$$\min z = 8x_1 + 2x_2$$

$$\upsilon.\pi.$$

$$x_1 + x_2 \geq -10 \qquad \text{Υπο-πρόβλημα 7}$$

$$4.5x_1 + x_2 \geq 16$$

$$x_1 + x_2 \leq 10$$

$$x_1 \leq 2$$

$$x_2 \geq 2$$

$$x_1, x_2 \geq 0$$

Ο περιορισμός $x_1 \leq 3$ είναι πλεονασματικός πλέον αφού εισάγουμε τον $x_1 \leq 2$, άρα μπορούμε να τον αφαιρέσουμε.

Η λύση του υπο-προβλήματος 7 είναι $z^* = 30,\ x_1^* = 2,\ x_2^* = 7$. Η λύση είναι ακέραια, άρα σταματάμε. Πρέπει να λύσουμε το υπο-πρόβλημα 5.

Σημείωση: Είναι μια εναλλακτική λύση την οποία επιστρέφει ο solver του Excel. Ωστόσο το Matlab επιστρέφει την λύση του υπο-προβλήματος 6, άρα κρατάω ως λύση αυτήν!

Το μοντέλο του υπο-προβλήματος 5 είναι:

$$\min z = 8x_1 + 2x_2$$

$$\upsilon.\pi.$$

$$x_1 + x_2 \geq -10 \qquad \text{Υπο-πρόβλημα 5}$$

$$4.5x_1 + x_2 \geq 16$$

$$x_1 + x_2 \leq 10$$

$$x_1 \leq 3$$

$$x_2 \leq 2$$

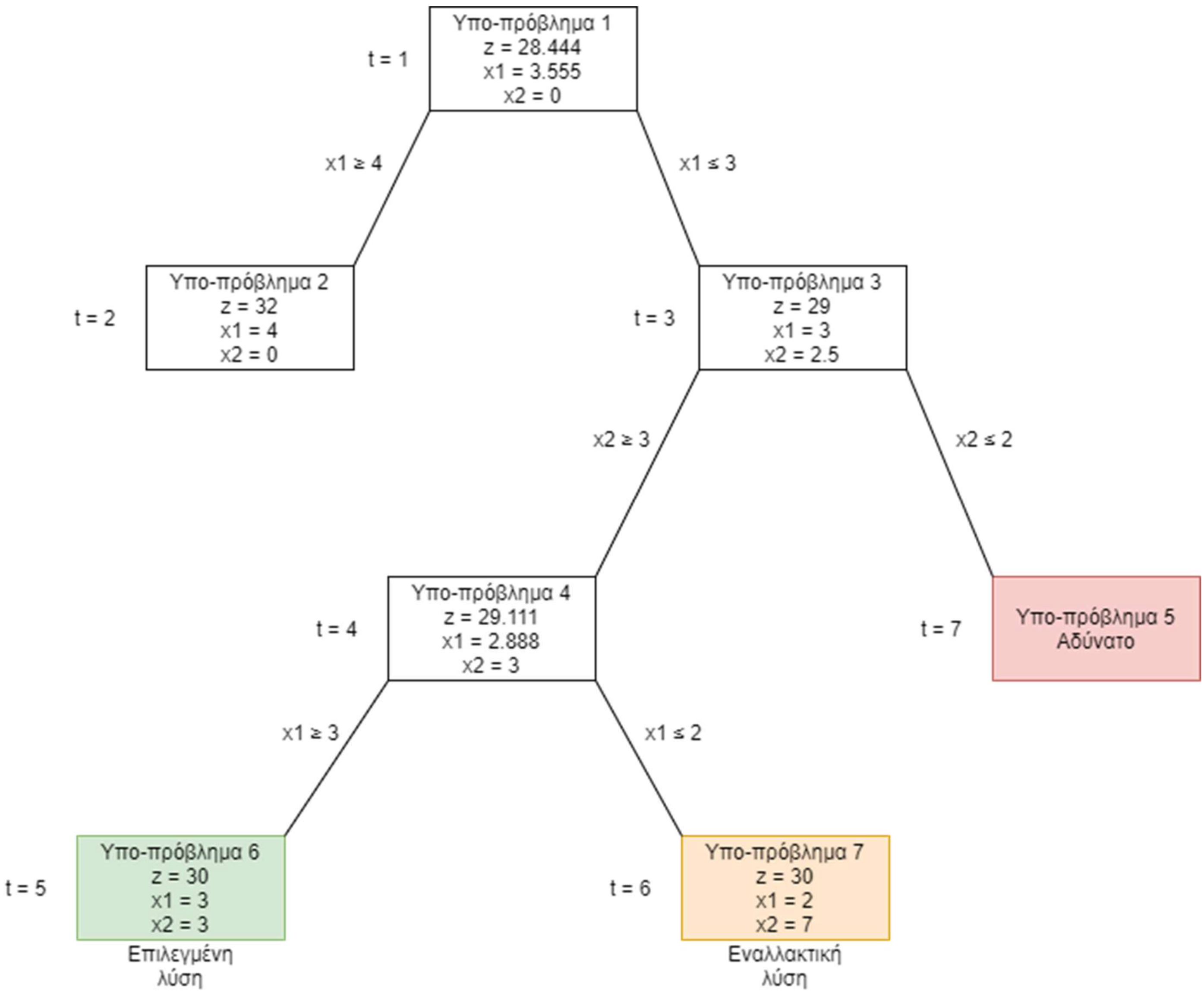
$$x_1, x_2 \geq 0$$

Το υπο-πρόβλημα 5 είναι αδύνατο.

Λύσαμε όλα τα υπο-προβλήματα, επομένως η βέλτιστη λύση είναι αυτή του υπο-προβλήματος 6, δηλαδή $z^* = 30,\ x_1^* = 3,\ x_2^* = 3$.

Πρέπει να σημειωθεί ότι μια εναλλακτική βέλτιστη λύση επιστρέφει και το υπο-πρόβλημα 7, δηλαδή $z^* = 30,\ x_1^* = 2,\ x_2^* = 7$. Ωστόσο, για λόγους που αναφέρονται παραπάνω επιλέγουμε αυτήν του υπο-προβλήματος 6.

Στην παρακάτω εικόνα αναπαρίσταται το δένδρο αναζήτησης της βέλτιστης λύσης.



Εικόνα 2: Branch and Bound tree

Άσκηση 2:

Τρέχοντας το αρχείο Matlab που μας δώσατε, έλαβα τα παρακάτω αποτελέσματα που φαίνονται στα screenshots.

optimal_solution | A

12x12 double

	1	2	3	4	5	6	7	8	9	10	11	12
1	0	-1	0	0	0	-1	0	-1	-1	-1	0	-1
2	-1	0	-1	0	0	-1	0	0	0	0	0	-1
3	0	0	0	-1	0	0	-1	0	-1	-1	0	0
4	0	0	0	0	0	-1	0	0	0	0	-1	0
5	0	0	0	-1	0	0	-1	0	0	0	-1	0
6	-1	-1	0	0	0	0	0	0	-1	0	0	0
7	0	-1	-1	-1	0	0	0	0	-1	0	-1	0
8	-1	0	-1	0	0	0	0	0	-1	0	-1	0
9	0	-1	0	0	0	0	0	0	0	0	-1	0
10	0	0	-1	-1	0	0	0	-1	-1	0	0	0
11	-1	0	0	0	0	-1	0	0	0	0	0	-1
12	-1	-1	0	-1	-1	0	0	0	0	-1	-1	0

optimal_solution

12x1 double

	1	2
1	0	
2	0	
3	0	
4	0	
5	0	
6	0	
7	0	
8	0	
9	1	
10	0	
11	1	
12	1	

Command Window

LP: Optimal objective value is 3.000000.

Optimal solution found.

Εικόνα 3: Αποτελέσματα Matlab

Επομένως, το μοντέλο που δημιουργείται έχει την εξής μορφή:

$$\min z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} + x_{12}$$

$\upsilon.\pi.$

$$x_2 + x_6 + x_8 + x_9 + x_{10} + x_{12} \geq 1$$

$$x_1 + x_3 + x_6 + x_{12} \geq 1$$

$$x_4 + x_7 + x_9 + x_{10} \geq 1$$

$$x_6 + x_{11} \geq 1$$

$$x_4 + x_7 + x_{11} \geq 1$$

$$x_1 + x_2 + x_9 \geq 1$$

$$x_2 + x_3 + x_4 + x_9 + x_{11} \geq 1$$

$$x_1 + x_3 + x_9 + x_{11} \geq 1$$

$$x_2 + x_{11} \geq 1$$

$$x_3 + x_4 + x_8 + x_9 \geq 1$$

$$x_1 + x_6 + x_{12} \geq 1$$

$$x_1 + x_2 + x_4 + x_5 + x_{10} + x_{11} \geq 1$$

$$x_j \in \{0,1\}\ j = 1, \dots ,12$$

Στο αρχείο Matlab *«exercise2.m»* έχω γράψει κώδικα που λύνει το παραπάνω πρόβλημα δυαδικού ακέραιου προγραμματισμού χρησιμοποιώντας την intlinprog, για να επαληθεύσω τα αποτελέσματα που μου επέστρεψε το *«Random_Generator_of_Adjacency_Matrix.m»*.

Η βέλτιστη λύση είναι $z^* = 3,\ x_9^* = 1,\ x_{11}^* = 1,\ x_{12}^* = 1$ και όλα τα άλλα 0. Άρα, θα χτιστούν 3 νοσοκομεία στις περιοχές 9, 11 και 12.

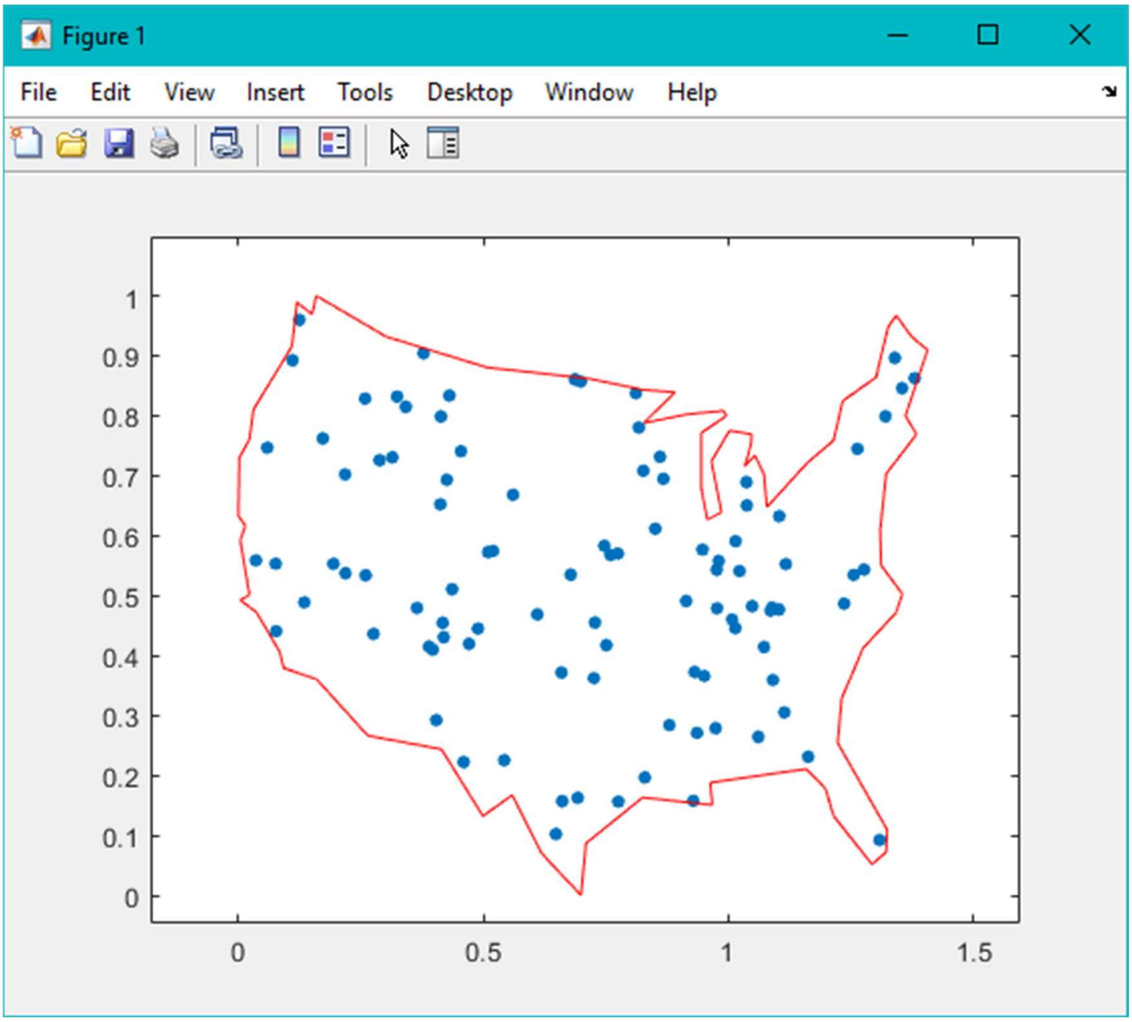
Άσκηση 3:

Αρχικά έτρεξα το script που μας δώσατε, ορίζοντας την μεταβλητή nStops = 97, διότι το άθροισμα των ψηφίων του αριθμού μητρώου μου είναι 11. Άρα 7 * 11 + 20 = 97 (+20 γιατί όπως λέτε και στην εκφώνηση αν βγει μικρότερο του 100 να προσθέσουμε ακόμα 20).

```
AM_SUM=11;
nStops = 7*AM_SUM + 20;
```

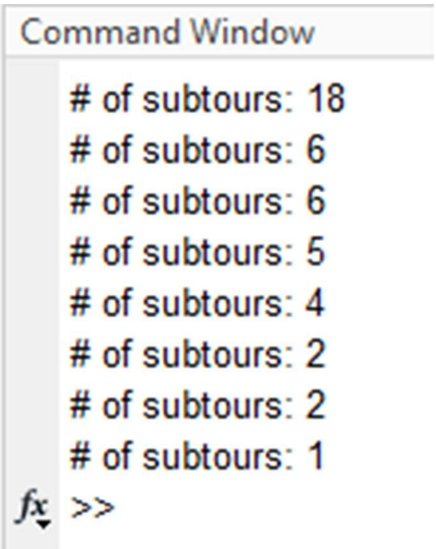
Εικόνα 4: Ορισμός μεταβλητής nStops

Το αποτέλεσμα του script είναι ο χάρτης με τις 97 πόλεις, ακολουθεί αντίστοιχο screenshot.



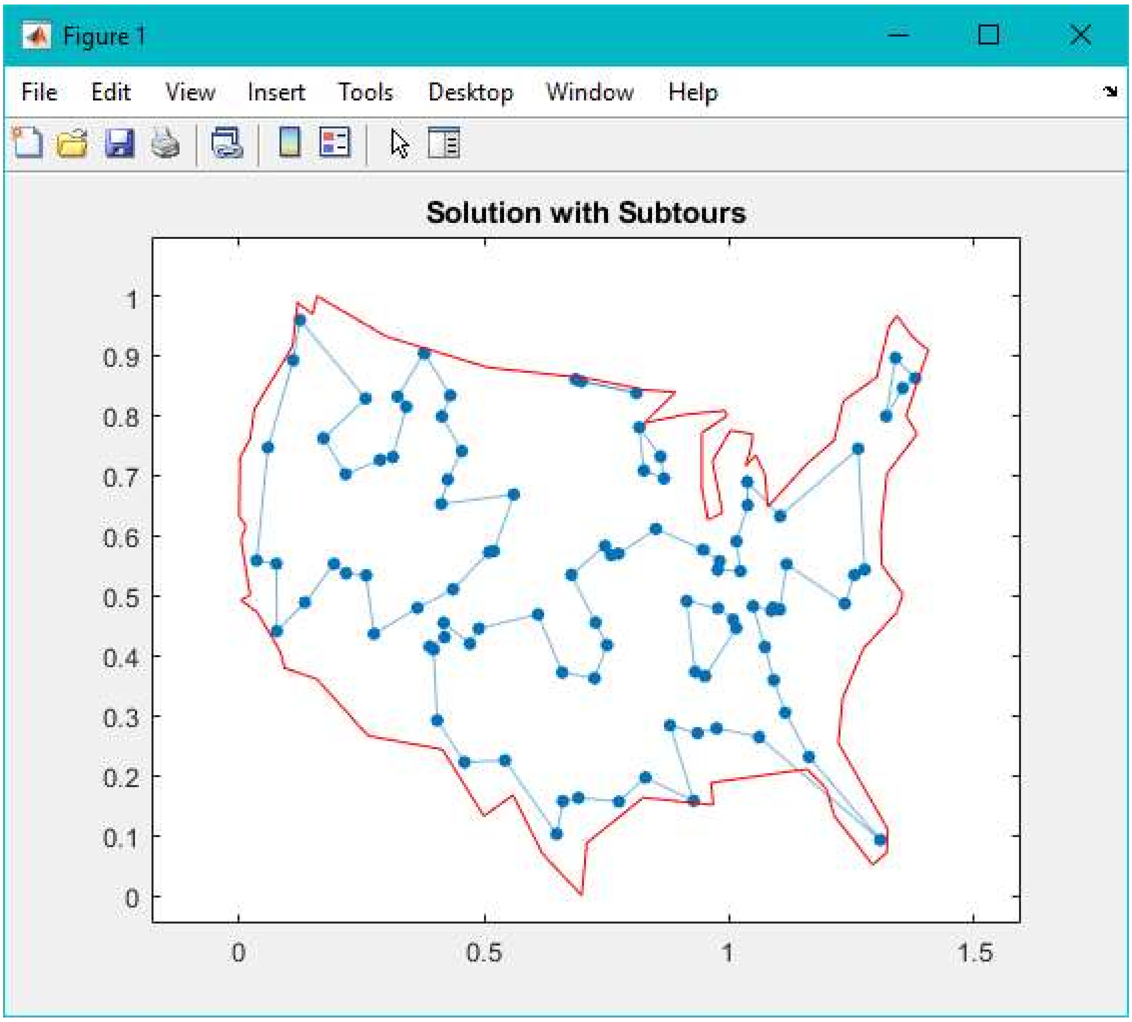
Εικόνα 5: Ο χάρτης με τις 97 πόλεις

Έπειτα, ακολουθώντας τις οδηγίες που υπάρχουν στον σύνδεσμο που μας δίνετε έλυσα το πρόβλημα βρίσκοντας των αριθμό των υπο-διαδρομών σε κάθε επανάληψη αλλά και την βέλτιστη διαδρομή. Συνολικά χρειάστηκαν 8 επαναλήψεις. Το πλήθος των υπο-διαδρομών φαίνεται στο παρακάτω screenshot.



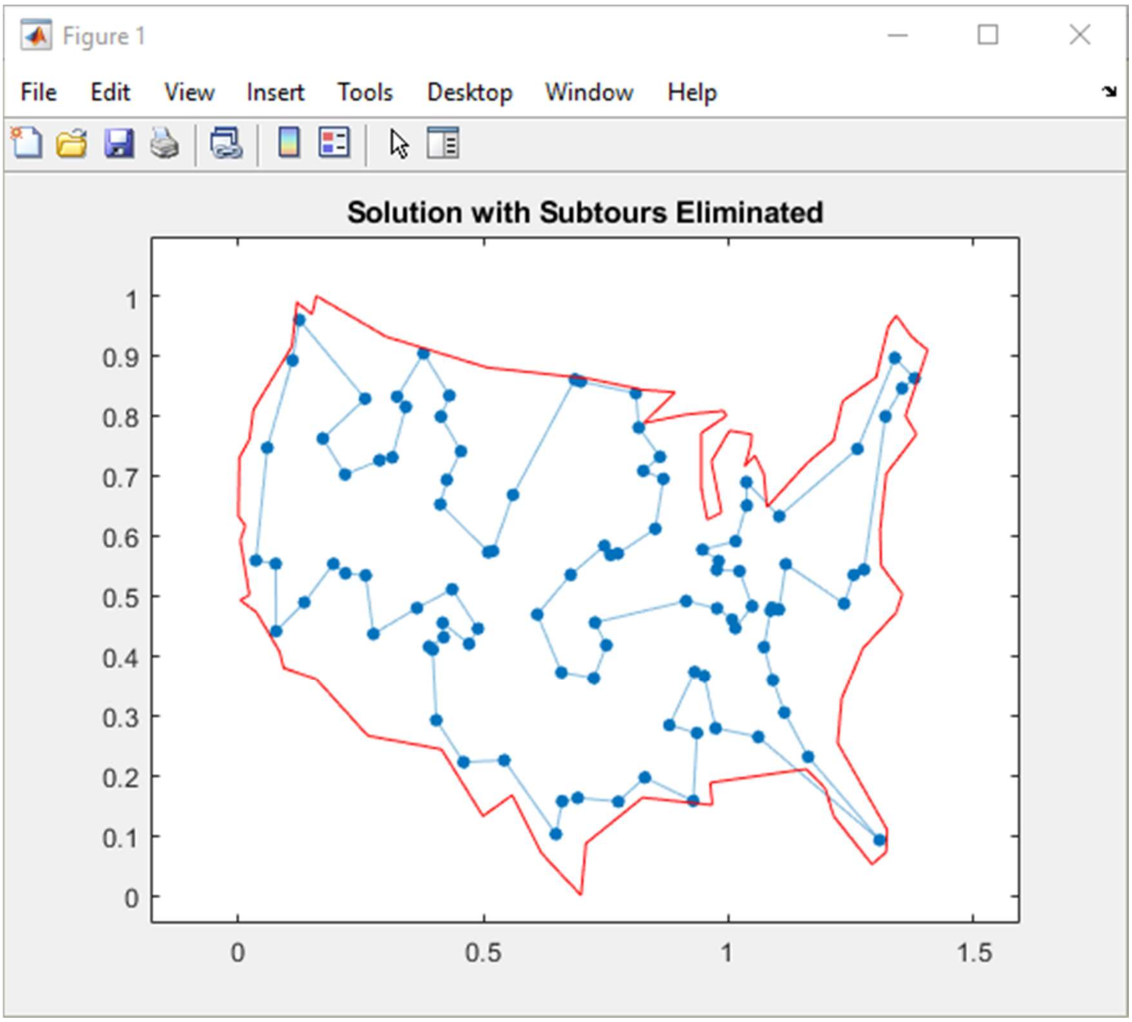
Εικόνα 6: πλήθος υπο-διαδρομών σε κάθε επανάληψη

Επίσης, έβγαλα ένα screenshot όπου φαίνονται στον χάρτη οι 6 υπο-διαδρομές που είχε εντοπίσει ο αλγόριθμος καθώς έλυνε το πρόβλημα.



Εικόνα 7: Ο χάρτης με 6 υπο-διαδρομές

Εν τέλη, η βέλτιστη διαδρομή που εντόπισε και επέστρεψε ως αποτέλεσμα ο αλγόριθμος, φαίνεται στην παρακάτω εικόνα.



Εικόνα 8: Ο χάρτης με την βέλτιστη διαδρομή

Άσκηση 4:

Ακολουθώντας τις οδηγίες τις εκφώνησης, κατάφερα να σχεδιάσω το πρόβλημα του sudoku χρησιμοποιώντας την συνάρτηση drawSudoku που μας δίνετε. Έλεγξα για τυχόν διπλότυπες τιμές, ωστόσο δεν υπήρχαν οπότε συνέχισα στην δημιουργία της συνάρτησης sudokuEngine, ακολουθώντας τις οδηγίες που μας παρέχετε στα δύο links. Η συνάρτηση βρίσκεται στο αρχείο Matlab «sudokuEngine.m».

Η μοντελοποίηση του προβλήματος που χρησιμοποιεί η συνάρτηση sudokuEngine αναλύεται παρακάτω.

1.Κάθε ακέραιη τιμή k ∈ [1, 9] πρέπει να εμφανίζεται ακριβώς μια φορά σε κάθε γραμμή
2.Κάθε ακέραιη τιμή k ∈ [1, 9] πρέπει να εμφανίζεται ακριβώς μια φορά σε κάθε στήλη
3.Κάθε ακέραιη τιμή k ∈ [1, 9] πρέπει να εμφανίζεται ακριβώς μια φορά σε κάθε υπο-πίνακα

Επίσης, x = 1 αν το στοιχείο (i,j) περιέχει τον αριθμό k αλλιώς 0, δηλαδή: $x_{ijk} = \begin{cases} 1 \\ 0 \end{cases}$

Μαθηματικά αυτοί οι περιορισμοί μπορούν να γραφτούν ως

$$\sum_{i=1}^{9} x_{ijk} = 1 \; για \; j, k = 1 \; μέχρι \; 9 \;\; (μόνο \; ένα \; στοιχείο \; k \; σε \; κάθε \; στήλη)$$

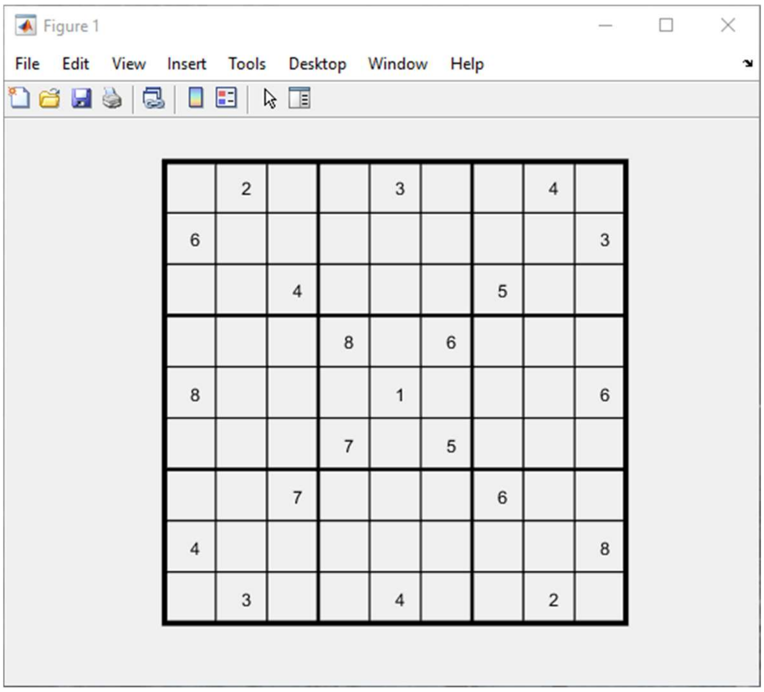
$$\sum_{j=1}^{9} x_{ijk} = 1 \; \gamma\iota\alpha \; i, k = 1 \; \mu\acute{\varepsilon}\chi\rho\iota \; 9 \;\; (\mu\acute{o}\nu o \; \acute{\varepsilon}\nu\alpha \; \sigma\tau o\iota\chi\varepsilon\acute{\iota}o \; k \; \sigma\varepsilon \; \kappa\acute{\alpha}\theta\varepsilon \; \gamma\rho\alpha\mu\mu\acute{\eta})$$

$$\sum_{i=1}^{3} \; \sum_{j=1}^{3} x_{ijk} = 1 \; \gamma\iota\alpha \; k = 1 \; \mu\acute{\varepsilon}\chi\rho\iota \; 9 \;\; (\mu\acute{o}\nu o \; \acute{\varepsilon}\nu\alpha \; \sigma\tau o\iota\chi\varepsilon\acute{\iota}o \; k \; \sigma\varepsilon \; \kappa\acute{\alpha}\theta\varepsilon \; \upsilon\pi o - \pi\acute{\iota}\nu\alpha\kappa\alpha \; 3x3)$$

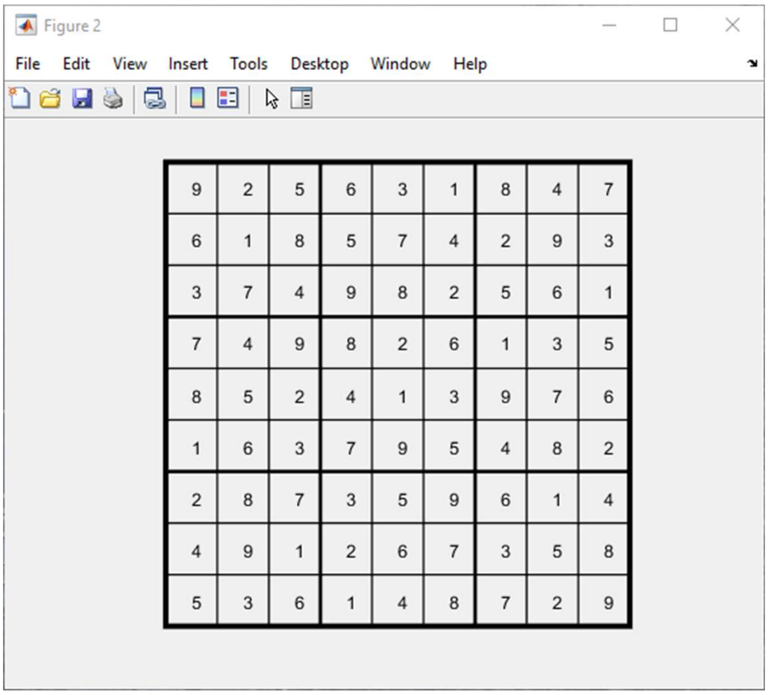
$$\sum_{k=1}^{9} x_{ijk} = 1 \; \gamma\iota\alpha \; i, j = 1 \; \mu\acute{\varepsilon}\chi\rho\iota \; 9 \;\; (\acute{o}\lambda\varepsilon\varsigma \; o\iota \; \theta\acute{\varepsilon}\sigma\varepsilon\iota\varsigma \; \tau o\upsilon \; \pi\acute{\iota}\nu\alpha\kappa\alpha \; \pi\rho\acute{\varepsilon}\pi\varepsilon\iota \; \nu\alpha \; \sigma\upsilon\mu\pi\lambda\eta\rho\omega\theta o\acute{\upsilon}\nu)$$

$x_{ijk} = 1$ *για όλες τις θέσεις του πίνακα που γνωρίζουμε την τιμή τους.*

Το αποτέλεσμα που επέστρεψε η drawSudoku και η sudokuEngine φαίνονται στα παρακάτω screenshots.



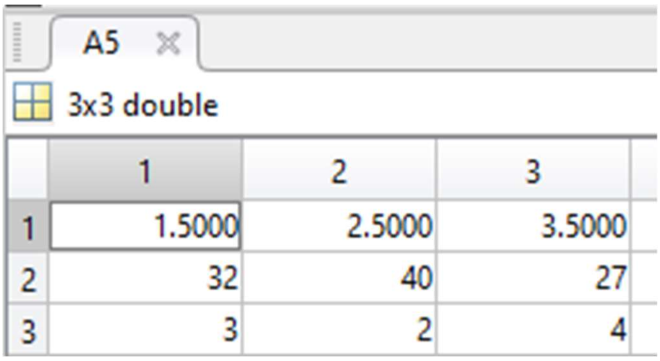
Εικόνα 9: Το πρόβλημα sudoku



Εικόνα 10: Η λύση του sudoku

Άσκηση 5:

Τρέχοντας το αρχείο Matlab που μας δώσατε, έλαβα τα παρακάτω αποτελέσματα για την δημιουργία του πίνακα που φαίνονται στο screenshot.



	1	2	3
1	1.5000	2.5000	3.5000
2	32	40	27
3	3	2	4

Εικόνα 11: Αποτελέσματα Matlab

Επομένως, ο πίνακας έχει την εξής μορφή:

	Προϊόν Α	Προϊόν Β	Προϊόν Γ
Πόρος 1	1.5	2.5	3.5
Πόρος 2	32	40	27
Κέρδος	3	2	4

Για να εξασφαλίσουμε την περίπτωση ότι αν αποφασιστεί η παραγωγή του εκάστοτε προϊόντος, τότε αυτή θα πρέπει να ξεπερνά τις 1000 μονάδες, πρέπει να εισάγουμε τρεις binary μεταβλητές στο μοντέλο για το κάθε προϊόν. Έστω ότι η παραγόμενη ποσότητα του κάθε προϊόντος συμβολίζεται με x1, x2, x3 και η κάθε binary μεταβλητή με y1, y2, y3 αντίστοιχα για τα προϊόντα Α, Β και Γ.

Λογικοί περιορισμοί:

$y = 0$ αν $x \leq 0$

$y = 1$ αν $x \geq 1000$

Πραγματικοί περιορισμοί:

$x \leq My$ (1)

$x \geq 1000 - M(1 - y)$ (2)

όπου $x \geq 0, y \in \{0, 1\}$ και M ένας πολύ μεγάλος αριθμός.

Άρα θα έχουμε δύο περιπτώσεις:

- Αν $y = 0$ τότε $x \leq 0$ και $x \geq 0$, δηλαδή $x = 0$, άρα το προϊόν δεν παράγεται. Δηλαδή, ο περιορισμός (1) ισχύει και ο περιορισμός (2) απενεργοποιείται.
- Αν $y = 1$ τότε $x \geq 1000$, άρα το προϊόν παράγεται. Δηλαδή, ο περιορισμός (2) ισχύει και ο περιορισμός (1) απενεργοποιείται.

Με αυτόν τον τρόπο λοιπόν, εξασφαλίζουμε την περίπτωση ότι για να παραχθεί ένα προϊόν, πρέπει η εταιρία να παράξει τουλάχιστον 1000 τεμάχια.

Επομένως, το μαθηματικό μοντέλο έχει την εξής μορφή:

$$\max z = 3x_1 + 2x_2 + 4x_3$$

$$\upsilon.\pi.$$

$$1.5x_1 + 2.5x_2 + 3.5x_3 \leq 6500$$

$$32x_1 + 40x_2 + 27x_3 \leq 65000$$

$$x_1 \leq My_1$$

$$x_1 \geq 1000 - M(1 - y_1)$$

$$x_2 \leq My_2$$

$$x_2 \geq 1000 - M(1 - y_2)$$

$$x_3 \leq My_3$$

$$x_3 \geq 1000 - M(1 - y_3)$$

$$x_1, x_2, x_3 \geq 0 \; \kappa\alpha\iota \; y_1, y_2, y_3 \in \{0,1\}$$

Έχω λύσει το πρόβλημα για επαλήθευση και με το Matlab (*«exercise5.m»*) χρησιμοποιώντας την intlinprog, αλλά και με τον solver του Excel (*«exercise5.xlsx»*).

Η βέλτιστη λύση είναι $z^* = 7888$, $x_1^* = 1000$, $x_2^* = 0$, $x_3^* = 1222$. Άρα, για να μεγιστοποιήσει το κέρδος της, η εταιρία πρέπει να παράξει 1000 τεμάχια από το προϊόν A, κανένα από το προϊόν B και 1222 από το προϊόν Γ.

Άσκηση 6:

Τρέχοντας το αρχείο Matlab που μας δώσατε, έλαβα τα παρακάτω αποτελέσματα για την δημιουργία του πίνακα που φαίνονται στο screenshot.

A6

6x3 double

	1	2	3
1	0	0	0
2	40	40	20
3	50	50	60
4	80	80	90
5	100	100	110
6	120	140	120

Εικόνα 12: Αποτελέσματα Matlab

Επομένως, ο πίνακας έχει την εξής μορφή:

Προγραμματιστές	Έργο Α	Έργο Β	Έργο Γ
0	0	0	0
1	40	40	20
2	50	50	60
3	80	80	90
4	100	100	110
5	120	140	120

Η διαδικασία θα ξεκινήσει «ανάποδα», δηλαδή από το έργο Γ.

Ξεκινάμε για το στάδιο n = 1 (Έργο Γ)

s	$x_1 = 0$	$x_1 = 1$	$x_1 = 2$	$x_1 = 3$	$x_1 = 4$	$x_1 = 5$	$f_1(s)$	x_1^*
0	0	-	-	-	-	-	0	0
1	-	20	-	-	-	-	20	1
2	-	-	60	-	-	-	60	2
3	-	-	-	90	-	-	90	3
4	-	-	-	-	110	-	110	4
5	-	-	-	-	-	120	120	5

Ο μειωμένος πίνακας που θα χρησιμοποιήσουμε για το στάδιο 2 είναι:

s	$f_1(s)$	x_1^*
0	0	0
1	20	1
2	60	2
3	90	3
4	110	4
5	120	5

Συνεχίζουμε για το στάδιο n = 2 (Έργο Β)

s	$x_2 = 0$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$x_2 = 4$	$x_2 = 5$	$f_2(s)$	x_2^*
0	0 + 0	-	-	-	-	-	0	0
1	0 + 20	40 + 0	-	-	-	-	40	1
2	0 + 60	40 + 20	50 + 0	-	-	-	60	0,1
3	0 + 90	40 + 60	50 + 20	80 + 0	-	-	100	1
4	0 + 110	40 + 90	50 + 60	80 + 20	100 + 0	-	130	1
5	0 + 120	40 + 110	50 + 100	80 + 60	100 + 20	140 + 0	150	1

Ο μειωμένος πίνακας που θα χρησιμοποιήσουμε για το στάδιο 2 είναι:

s	$f_2(s)$	x_2^*
0	0	0
1	40	1
2	60	0,1
3	100	1
4	130	1
5	150	1

Συνεχίζουμε για το τελευταίο στάδιο n = 3 (Έργο Α)

Στη συγκεκριμένη περίπτωση, δεν έχει νόημα να εξετάσουμε στο στάδιο 3 την περίπτωση s = {0,1,2,3,4} προγραμματιστές καθώς αυτό θα σήμαινε ότι, ενώ θα έχουμε 5 προγραμματιστές να διαθέσουμε στα τρία έργα, δεν θα τους διαθέσουμε όλους. Άρα, αρκεί να εξετάσουμε μόνο την περίπτωση s = 5.

s	$x_3 = 0$	$x_3 = 1$	$x_3 = 2$	$x_3 = 3$	$x_3 = 4$	$x_3 = 5$	$f_3(s)$	x_3^*
5	0 + 150	40 + 130	50 + 100	80 + 60	100 + 40	120 + 0	170	1

Άρα, ο μειωμένος πίνακας που προκύπτει είναι:

s	$f_3(s)$	x_3^*
5	170	1

Από τους μειωμένους πίνακες, μπορούμε να βρούμε πως θα διαμοιράσουμε εν τέλη τους προγραμματιστές.

Στο στάδιο n = 3 δίνουμε $x_3^* = 1$, άρα πάμε στο στάδιο n = 2 με s = 4 προγραμματιστές, δηλαδή $x_2^* = 1$. Επομένως, πάμε στο στάδιο n = 1 με s = 3 προγραμματιστές, που τους δίνουμε όλους, άρα $x_1^* = 3$.

Συνεπώς, έχουμε βέλτιστη λύση με βέλτιστη τιμή (συνολική κερδοφορία) 170:

$x_1^* = 3, x_2^* = 1, x_3^* = 1.$

Άσκηση 7:

Το άθροισμα των ψηφίων του ΑΜ μου είναι sum = 11, επομένως ακολουθώ τον κανόνα Β και το α = 11/25 = 0.44. Άρα, η πιθανότητα επιτυχίας είναι 0.44 και η πιθανότητα αποτυχίας είναι 1 – 0.44 = 0.56.

Η αναδρομική σχέση που πρέπει να ακολουθήσω είναι:

$$f_n\ (s_n, x_n) = 0.56 f_{n+1}^*(s_n - x_n) + 0.44 f_{n+1}^*(s_n + x_n)$$

Η διαδικασία ξεκινάει από το τελευταίο στάδιο (παρτίδα) n = 3.

Στάδιο n = 3:

s_3	$f_3^*(s_3)$	x_3^*
0	0	-
1	0	-
2	0.44	2
3	0.44	≥ 1
$\geq$ **4**	1	0 (ή $\leq s_3 - 4$)

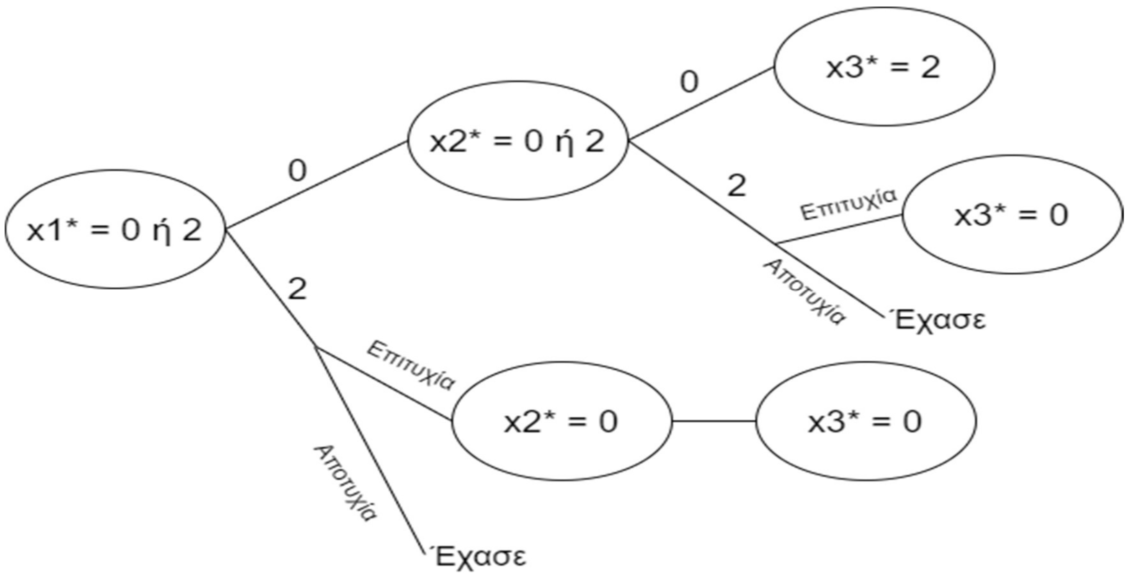
Στάδιο n = 2: $(f_2\ (s_2, x_2) = 0.56 f_3^*(s_2 - x_2) + 0.44 f_3^*(s_2 + x_2))$

s_2	$x_2 = 0$	$x_2 = 1$	$x_2 = 2$	$x_2 = 3$	$f_2^*(s_2)$	x_2^*
0	0	-	-	-	0	-
1	0	0.1936	-	-	0.1936	1
2	0.44	0.1936	0.44	-	0.44	0,2
3	0.44	0.6864	0.44	0.44	0.6864	1
$\geq$ **4**	1	-	-	-	1	0 (ή $\leq s_2 - 4$)

Στάδιο n = 1: $(f_1\ (s_1, x_1) = 0.56 f_2^*(s_1 - x_1) + 0.44 f_2^*(s_1 + x_1))$

s_1	$x_1 = 0$	$x_1 = 1$	$x_1 = 2$	$f_1^*(s_1)$	x_1^*
2	0.44	0.4104	0.44	0.44	0,2

Η βέλτιστη πολιτική για να συγκεντρώσει ο παίκτης τουλάχιστον 4 μάρκες έχει πιθανότητα 0.44:



Εικόνα 13: Πιθανά σενάρια

Άρα, μπορεί να ακολουθήσει τρεις διαφορετικές τακτικές για να κερδίσει:

$x_1^* = 0, x_2^* = 0, x_3^* = 2.$

$x_1^* = 0, x_2^* = 2, x_3^* = 0.$

$x_1^* = 2, x_2^* = 0, x_3^* = 0.$