2η Εργασία μαθήματος «Ειδικά θέματα επιχειρησιακής έρευνας»

Αντώνιος Ελευθέριος Καρναβάς - Π18063

* **Άσκηση 1**

Min z= 8\*x1+6\*x2

Υ.π.

-2x1-2x2<=9

-9/2x1-1x2<=-28

1x1+2x2<=10

x1,x2>=0

x1,x2 integer

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Χαλάρωση του παραπάνω ακεραίου προβλήματος σε γραμμικό πρόβλημα.

1

Min z= 8\*x1+6\*x2

Υ.π.

-2x1-2x2<=9

-9/2x1-1x2<=-28

1x1+2x2<=10

x1,x2>=0

x1,x2 ∈ℕ

Z= 56

X1=7

X2=0

3

Min z= 8\*x1+6\*x2

Υ.π.

-2x1-2x2<=9

-9/2x1-1x2<=-28

1x1+2x2<=10

X1>=7

x1,x2>=0

x1,x2 ∈ℕ

Z= 54

X1=6

X2=1

2

X1>=7

Z= 49.778

X1=6.222

X2=0

X1<=6

Min z= 8\*x1+6\*x2

Υ.π.

-2x1-2x2<=9

-9/2x1-1x2<=-28

1x1+2x2<=10

X1<=6

x1,x2>=0

x1,x2 ∈ℕ

Παρατηρήσεις

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Από το παραπάνω αποτέλεσμα της αρχικής προσέγγισης της λύσης από την intlinprog που λύθηκε από το έτοιμο script που μας δόθηκε παρατηρώ ότι το αποτέλεσμα είναι το ίδιο. Δηλαδή το upper bound όπως βρήκα και εγώ με την branch and bound είναι 56 ενώ το lower bound είναι ίσο με 54.

* **Άσκηση 2**

To Adjacency Matrix που παράχθηκε τυχαία από τον κώδικα του MATLAB:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Π1 | Π2 | Π3 | Π4 | Π5 | Π6 | Π7 | Π8 | Π9 | Π10 | Π11 | Π12 |
| Π1 | **0** | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **1** | **0** | **1** | **0** |
| Π2 | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **0** |
| Π3 | **1** | **0** | **0** | **0** | **1** | **0** | **0** | **1** | **1** | **1** | **0** | **0** |
| Π4 | **0** | **0** | **0** | **0** | **1** | **1** | **1** | **0** | **0** | **0** | **1** | **1** |
| Π5 | **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **1** | **0** |
| Π6 | **0** | **0** | **1** | **0** | **0** | **0** | **0** | **0** | **1** | **1** | **0** | **0** |
| Π7 | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |
| Π8 | **0** | **0** | **0** | **1** | **1** | **0** | **0** | **0** | **0** | **1** | **0** | **0** |
| Π9 | **0** | **1** | **0** | **0** | **1** | **1** | **1** | **0** | **0** | **0** | **0** | **0** |
| Π10 | **0** | **0** | **0** | **0** | **1** | **0** | **1** | **0** | **0** | **0** | **0** | **0** |
| Π11 | **0** | **1** | **0** | **1** | **0** | **0** | **1** | **1** | **0** | **0** | **0** | **0** |
| Π12 | **1** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** | **0** | **1** | **0** |

Μαθηματικό μοντέλο προβλήματος:

Min z=x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12

s.t.

x4+x5+x7+x9+x11>=1

x4+x5+x11>=1

x1+x5+x8+x9+x10>=1

x5+x6+x7+x11+x12>=1

x7+x9+x11>=1

x3+x9+x10>=1

x12>=1

x4+x5+x10>=1

x2+x5+x6+x7>=1

x5+x7>=1

x2+x4+x7+x8>=1

x1+x2+x5+x8+x11>=1

Τα αποτελέσματα του κώδικα ήταν τα εξής και ειδικότερα η βέλτιστη λύση αφορούσε τις περιοχές 5,7 και 9.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

* **Άσκηση 3**

Μοντελοποίηση του προβλήματος του περιοδεύοντος πωλητή

Αντικειμενική συνάρτηση: **Min Σij=cij\*yij** ,όπου yij=1 ανν η πόλη j έρχεται μετά την πόλη i αλλιώς yij=0 και cij είναι η απόσταση μεταξύ της πόλης i και j.

Περιορισμοί: Εφόσον οι μεταβλητές y12 και y21 είναι διαφορετικές αφού η μια σημαίνει μετάβαση από την πόλη 1 στη πόλη 2 ενώ η άλλη σημαίνει μετάβαση από την πόλη 2 στη πόλη 1 χρειαζόμαστε 2 περιορισμούς έτσι ώστε να εξασφαλίσουμε ότι κάθε διαδρομή έγινε μια φορά μόνο. Κατ’ επέκταση αυτό σημαίνει και κάθε πόλη επισκέφθηκε μόνο μια φορά και οι διαδρομές που την συμπεριλαμβάνουν είναι μια κατά την άφιξη και μια κατά την αναχώρηση. Άρα έχουμε:

**Σi:(i,j)yij=1** για κάθε πόλη j από 1 έως n πόλεις

**Σj:(i,j)yji=1** για κάθε πόλη i από 1 έως n πόλεις

Όπου **yji=0 ή 1** για κάθε (i,j)

Τέλος για να αποφύγουμε κύκλους στο γράφημα διαδρομών θα πρέπει να ισχύει ο περιορισμός ότι το άθροισμα του συνόλου των διαδρομών yij θα πρέπει να είναι μικρότερο από το σύνολο των κόμβων (πόλεων). Άρα θα πρέπει να ισχύει: **ΣiΣjyji<=S-1** όπου S το σύνολο των πόλεων.

* **Άσκηση 4**

Μοντελοποίηση του προβλήματος του Sudoku

Επειδή στο sudoku δεν υπάρχουν πολλές λύσεις εκ των οποίων κάποιες ή κάποια είναι βέλτιστη, δεν υπάρχει αντικειμενική συνάρτηση. Παρόλα αυτά υπάρχουν κανόνες επίλυσης του sudoku οι οποίοι εκφράζονται υπό την μορφή περιορισμών. Οι περιορισμοί αυτοί είναι (όπως και οι κανόνες):

Αρχικά θα υπάρχουν 9\*9\*9=729 μεταβλητές x(i,j,k)=1 ή 0 όπου i=γραμμή, j=στήλη, k=ακέραιος που αυτές θα αντιστοιχούν στους παρακάτω περιορισμούς. Τιμή 1 θα λαμβάνει αν στη γραμμή i και στήλη j θα έχει τον αριθμό k αλλιώς θα έχει την τιμή 0.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Για κάθε γραμμή θα πρέπει να έχει μόνο έναν μοναδικό αριθμό k, για j,k=1 έως 9.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Για κάθε στήλη θα πρέπει να έχει μόνο έναν μοναδικό αριθμό k, για i,k=1 έως 9.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Πρέπει όλες οι θέσεις του πίνακα i=1 έως 9 και j=1 έως 9 να έχουν έναν ακέραιο k συμπληρωμένο.

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Ο παραπάνω περιορισμός να ισχύει για κάθε 3χ3 υπό-πίνακα δηλαδή κάθε ένας τους να περιέχει διαφορετικές, μοναδικές, μη επαναλαμβανόμενες τιμές του k. Επειδή ο παραπάνω περιορισμός αφορά ένα τέτοιο μόνο υπό-πίνακα ενώ υπάρχουν 9 τέτοιοι ο περιορισμός γίνεται:



* **Άσκηση 5**

Μοντελοποίηση του προβλήματος του Company

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

Αντικειμενική συνάρτηση: max z=4ΠΑ+3ΠΒ+2ΠΓ+0B1+0B2+0B3

Περιορισμοί:

3,5\*ΠΑ+1,5\*ΠΒ+2,5\*ΠΓ<=6500

32\*ΠΑ+34\*ΠΒ+36\*ΠΓ<=65000

ΠΑ<=ΜΒ1 , ΠΑ>=1000-Μ(1-Β1)

ΠΒ<=ΜΒ2 , ΠΒ>=1000-Μ(1-Β2)

ΠΓ<=ΜΒ3 , ΠΓ>=1000-Μ(1-Β3)

* **Άσκηση 6**

Πίνακας κέρδους έργων συναρτήσει των προγραμματιστών, παραγμένος τυχαία από το MATLAB:

|  |  |  |  |
| --- | --- | --- | --- |
| Προγραμματιστές | Έργο Α | Έργο Β | Έργο Γ |
| 0 | 0 | 0 | 0 |
| 1 | 30 | 20 | 20 |
| 2 | 60 | 70 | 50 |
| 3 | 80 | 80 | 90 |
| 4 | 100 | 110 | 100 |
| 5 | 140 | 120 | 140 |

Μεταβλητές xn: αριθμός προγραμματιστών που θα ανατεθούν στο έργο n.

Ξεκινάμε αντίστροφα δηλαδή από το έργο Γ. Για n=1, η x1 δείχνει πόσους προγραμματιστές θα ανατεθούν στο έργο Γ.

S= Κατάσταση, δηλαδή διαθέσιμους προγραμματιστές προς ανάθεση στο κάθε βήμα.

X1= Προγραμματιστές στο έργο Γ.

F1(s)= Βέλτιστη τιμή εκάστοτε βήματος.

X1\*= Απαιτούμενος αριθμός προγραμματιστών ώστε να πάρουμε τη βέλτιστη τιμή για κάθε κατάσταση si δηλαδή το fi(s).

Βήμα 1:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| s | X1=0 | X1=1 | X1=2 | X1=3 | X1=4 | X1=5 | F1(s) | X1\* |
| 0 | 0 | - | - | - | - | - | 0 | 0 |
| 1 | - | 20 | - | - | - | - | 20 | 1 |
| 2 |  |  | 50 |  |  |  | 50 | 2 |
| 3 |  |  |  | 90 |  |  | 90 | 3 |
| 4 |  |  |  |  | 100 |  | 100 | 4 |
| 5 |  |  |  |  |  | 140 | 140 | 5 |

Βήμα 2:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| s | X2=0 | X2=1 | X2=2 | X2=3 | X2=4 | X2=5 | F2(s) | X2\* |
| 0 | 0+0 | - | - | - | - | - | 0 | 0 |
| 1 | 0+20 | 20+0 | - | - | - | - | 20 | 0,1 |
| 2 | 0+50 | 20+20 | 70+0 | - | - | - | 70 | 2 |
| 3 | 0+90 | 20+50 | 70+20 | 80+0 | - | - | 90 | 1,2 |
| 4 | 0+100 | 20+90 | 70+50 | 80+20 | 110+0 | - | 120 | 2 |
| 5 | 0+140 | 20+100 | 70+90 | 80+50 | 110+20 | 120+0 | 160 | 2 |

Επειδή έχουμε 3 έργα μόνο, και έχουμε κάνει ήδη για τα έργα Γ και Β πίνακες το βήμα n=3 είναι το τελευταίο και φορά το έργο Α.

Επομένως, στη συγκεκριμένη περίπτωση, δεν έχει νόημα να εξετάσουμε στο βήμα 3 την περίπτωση s={0,1,2,3} προγραμματιστές καθώς αυτό θα σήμαινε ότι ενώ θα έχουμε 5 προγραμματιστές να διαθέσουμε στα 3 έργα, εμείς δεν θα τα διαθέσουμε όλα.

Συνεπώς, αρκεί για n=3 να εξετάσουμε μόνο την περίπτωση s=4,5.

Βήμα 3:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| s | X3=0 | X3=1 | X3=2 | X3=3 | X3=4 | X3=5 | F3(s) | X3\* |
| 4 | 0+120 | 30+90 | 60+70 | 80+20 | 100+0 | - | 130 | 2 |
| 5 | 0+160 | 30+120 | 60+90 | 80+70 | 100+20 | 140+0 | 160 | 0 |

|  |  |  |
| --- | --- | --- |
| s | F2(s) | X2\* |
| 0 | 0 | 0 |
| 1 | 20 | 0,1 |
| 2 | 70 | 2 |
| 3 | 90 | 1,2 |
| 4 | 120 | 2 |
| 5 | 160 | 2 |

|  |  |  |
| --- | --- | --- |
| s | F3(s) | X3\* |
| 4 | 130 | 2 |
| 5 | 160 | 0 |

|  |  |  |
| --- | --- | --- |
| s | F1(s) | X1\* |
| 0 | 0 | 0 |
| 1 | 20 | 1 |
| 2 | 50 | 2 |
| 3 | 90 | 3 |
| 4 | 100 | 4 |
| 5 | 140 | 5 |

Παρατηρώ πως η βέλτιστη λύση

X1\*=3

X2\*=2

X3\*=0

Άρα για να έχουμε βέλτιστη αποτελεσματικότητα θα δώσουμε 0 προγραμματιστές στο έργο Α, 2 προγραμματιστές στο έργο Β και 3 προγραμματιστές στο έργο Γ.

* **Άσκηση 7**

Άθροισμα αριθμού μητρώου=18

Επομένως α=18/30=0,6 ή 3/5.

Στάδιο n=3:

|  |  |  |
| --- | --- | --- |
| S3 | F3\*(S3) | X3\* |
| 0 | 0 | - |
| 1 | 0 | - |
| 2 | 0,6 | 2 |
| 3 | 0,6 | >=1 |
| >=4 | 1 | 0(ή S3-4) |

Στάδιο n=2: (f2(s2,x2)=0,4F3\*(s2-x2)+0,6 F3\*(s2+x2))

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| S2 | X2 | 0 | 1 | 2 | 3 | F2\*(S2) | X2\* |
| 0 | | 0 | - | - | - | 0 | - |
| 1 | | 0 | 0,36 | - | - | 0,36 | 1 |
| 2 | | 0,6 | 0,36 | 0,6 | - | 0,6 | 0 ή 2 |
| 3 | | 0,6 | 0,84 | 0,6 | 0,6 | 0,84 | 1 |
| 4 | | 1 |  |  |  | 1 | 0(ή S2-4) |

Στάδιο n=3: (f1(s1,x1)=0,4F2\*(s1-x1)+0,6 F2\*(s1+x1))

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S1 | X2 | 0 | 1 | 2 | F1\*(S1) | X1\* |
| 2 | | 0,6 | 0,648 | 0,6 | 0,648 | 1 |

Επιτυχία X3\*=0

Επιτυχία X2\*=1

Αποτυχία X3\*=2

X1\*=1

Αποτυχία X2\*=1

Επιτυχία X3\*=2

Αποτυχία Αποτυχία