

# Operational Research

ΑΤΟΜΙΚΗ ΕΡΓΑΣΙΑ

Κωνσταντίνος Μυλωνάς | 10027 | 03.06.2023

Word Count: 984

## Πίνακας περιεχομένων

Εισαγωγή.....	3
Τεχνικές Προδιαγραφές .....	3
Υλοποίηση.....	4
Προγραμματισμός Κατασκευής Εξαρτημάτων .....	4
Προγραμματισμός παραγωγής αυτοκινήτων.....	4
Παράρτημα.....	6
(A) Αποτελέσματα - Λύσεις.....	6
Πρώτο πρόβλημα.....	6
Δεύτερο πρόβλημα .....	7

## Εισαγωγή

Στην συγκεκριμένη εργασία ζητήθηκε η υλοποίηση ενός προγράμματος σε python με τη χρήση gurobi ή σε ampl, το οποίο να βελτιστοποιεί δύο προβλήματα.

Το πρώτο πρόβλημα αφορά στη σειρά διεργασιών σε μια γραμμή παραγωγής, παρόμοιο με το γνωστό job shop scheduling problem. Πρόκειται για ένα πρόβλημα στο οποίο πρέπει να υπολογιστεί η βέλτιστη λύση, η οποία ελαχιστοποιεί τον χρόνο που απαιτείται για όλα τα εξαρτήματα να περάσουν και από τα 3 μηχανήματα (χύτευσης, λείανσης και σύνδεσης) υπό κάποιους περιορισμούς. Το κάθε εξάρτημα πρέπει να περάσει με συγκεκριμένη σειρά από τα τρία μηχανήματα, ωστόσο μας ζητήθηκε να υπολογίσουμε τη βέλτιστη σειρά που εισέρχονται τα εξαρτήματα στο κάθε μηχάνημα.

Το δεύτερο πρόβλημα ήταν πιο απλό καθώς ζητήθηκε η υλοποίηση ενός προγράμματος που να συγκρίνει και να μεγιστοποιεί τα κέρδη για την εταιρεία, και να ληφθεί μία απόφαση σχετικά με το αν πρέπει η εταιρεία να συγχωνεύσει κάποιες γραμμές παραγωγής ή όχι. Στη περίπτωση αυτή οι περιορισμοί αφορούσαν στη ζήτηση για κάθε μοντέλο αυτοκινήτου, τη δυνατότητα παραγωγής από τη πλευρά της Αυτοκίνησης και άλλους παράγοντες όπως το ποσοστό των δυσαρεστημένων πελατών που θα συμβιβαζόταν με κάποιο άλλο μοντέλο εκτός της πρώτης επιλογής.

## Τεχνικές Προδιαγραφές

Για την υλοποίηση του πρώτου μέρους της εργασίας χρησιμοποιήθηκε Python ενώ ως περιβάλλον εργασίας το PyCharm. Απαραίτητη ήταν επίσης η ORTools βιβλιοθήκη για τη δημιουργία και τη βελτιστοποίηση του μοντέλου.

Στο δεύτερο σκέλος 'του προβλήματος χρησιμοποιήθηκε η AMPL σε AMPL IDE. Πρόκειται για ένα λογισμικό βελτιστοποίησης που διαθέτει και περιβάλλον επεξεργασίας κώδικα, επομένως δεν απαιτείται η ενσωμάτωση βιβλιοθηκών. Η λύση που προτείνεται κάνει χρήση του αλγορίθμου CPLEX, όπως φαίνεται στο παράρτημα (Α).

Η προδιαγραφή της συσκευής που χρησιμοποιήθηκε για την εκτέλεση του κώδικα και τη λήψη μετρήσεων είναι οι εξής:

- Processor: 12th Gen Intel(R) Core(TM) i7-1260P @ 2.10 GHz
- Ram memory: 32 GB
- System type: 64-bit operating system, x64-based processor

---

<sup>1</sup> Σημείωση: Χρησιμοποιήθηκε διαφορετικό λογισμικό στα 2 προβλήματα για λόγους εξοικείωσης με δύο τελείως διαφορετικά, αλλά εξίσου χρήσιμα και διαδεδομένα, περιβάλλοντα για μοντελοποίηση και βελτιστοποίηση προβλημάτων επιχειρησιακής έρευνας. Τελικά, παρόλο που η γλώσσα python είναι γνωστή και από προηγούμενη εμπειρία, στη πράξη το λογισμικό της AMPL φάνηκε ιδιαίτερα εύχρηστο και βολικό για την επίλυση του συγκεκριμένου προβλήματος.

## Υλοποίηση

### ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΚΑΤΑΣΚΕΥΗΣ ΕΞΑΡΤΗΜΑΤΩΝ

Αρχικά εισάγουμε τα δεδομένα που μας δίνονται με τη μορφή πίνακα. Κάθε στοιχείο του πίνακα αποτελεί ένα σετ αριθμών που φανερώνουν το μηχάνημα και τον αντίστοιχο χρόνο για το συγκεκριμένο εξάρτημα.

Στη συνέχεια ορίζουμε τον αριθμό των μηχανών και υπολογίζουμε δυναμικά το άθροισμα όλων των χρόνων. Δημιουργούμε το μοντέλο προς βελτιστοποίηση και 2 tuples για την αποθήκευση και τροποποίηση πληροφοριών. Εισάγουμε τα απαραίτητα δεδομένα στα δύο αυτά tuples και συνεχίζουμε με τους περιορισμούς.

Πρώτος περιορισμός είναι να μην υπάρχει υπερκάλυψη των μηχανών από 2 διαφορετικά εξαρτήματα, και στη συνέχεια είναι οι προτεραιότητες για ένα εξάρτημα, δηλαδή πως δεν μπορεί να περάσει από το στάδιο 2 (τη λείανση) πριν περάσει από το στάδιο 1 (τη χύτευση) και αντίστοιχα για το στάδιο 3.

Ορίζουμε και τη μεταβλητή που επιθυμούμε να ελαχιστοποιήσουμε, δηλαδή στη συγκεκριμένη περίπτωση τον συνολικό χρόνο, και τελικά ζητάμε από το μοντέλο μας τη βέλτιστη λύση.

Τέλος ακολουθεί ένας κώδικας για την εκτύπωση των αποτελεσμάτων και της βέλτιστης λύσεις, που στη περίπτωση μας προκύπτει να είναι 35.0 ώρες. Η σειρά με την οποία εισάγονται τα εξαρτήματα στο κάθε μηχάνημα φαίνεται στο παράρτημα.

### ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΠΑΡΑΓΩΓΗΣ ΑΥΤΟΚΙΝΗΤΩΝ

Στο πρόβλημα της Αυτοκίνησης, δημιουργούμε δύο χωριστά αρχεία με extension .dat για το αρχείο των δεδομένων και .mod για το αρχείο του μοντέλου.

Στο αρχείο .dat εισάγουμε τα δεδομένα που μας δίνονται με τη μορφή πινάκων και αφορούν στο κόστος των αυτοκινήτων, το κέρδος από κάθε μοντέλο, τη ζήτηση, τη παραγωγική δυνατότητα της Αυτοκίνησης για κάθε μοντέλο και τέλος, τα ποσοστά των πελατών που είναι διατεθειμένα να αλλάξουν το μοντέλο της παραγγελίας τους, δεδομένου πως το μοντέλο που επιθυμούσαν εξ αρχής δεν είναι διαθέσιμο.

Στο αρχείο του μοντέλου θέτουμε αρχικά τις μεταβλητές για τις γραμμές παραγωγής και τους τύπους των αυτοκινήτων και στη συνέχεια εισάγουμε σε αυτό τις 5 μεταβλητές που θέσαμε νωρίτερα στο αρχείο .dat.

Δημιουργούμε τις μεταβλητές απόφασης για το μοντέλο μας, όπως τα συνολικά έσοδα (πρέπει να είναι μη αρνητικά), τα συνολικά έξοδα (πρέπει να είναι μη αρνητικά), τον αριθμό αυτοκινήτων από κάθε μοντέλο που πωλείται (πρέπει να είναι μη αρνητικός και ακέραιος), τον αριθμό των δυσανεσθημένων πελατών που το μοντέλο της επιλογής τους δεν είναι διαθέσιμο (πρέπει να είναι μη αρνητικός και ακέραιος), και τέλος μια binary μεταβλητή, η οποία μας φανερώνει ποια γραμμή λειτουργεί και ποια όχι, καθώς είναι προϋπόθεση ότι δεν λειτουργούν ταυτόχρονα οι 3 παλιές γραμμές με τις 2 δυνητικά καινούριες.

Επομένως, ζητάμε από το μοντέλο αρχικά να μεγιστοποιήσει το κέρδος, δηλαδή τη διαφορά inc-exp, ορίζοντας ως συνολικό έσοδο τη κέρδος από κάθε αυτοκίνητο που πωλείται επί τον αριθμό των αυτοκινήτων από κάθε μοντέλο και ως συνολικό έξοδο ορίζουμε το κόστος για κάθε γραμμή παραγωγής, προφανώς ελέγχοντας αν λειτουργεί ή όχι, επομένως έχουμε πολλαπλασιάσει το κόστος της κάθε γραμμής με τη binary μεταβλητή που μας δείχνει αν η γραμμή βρίσκεται σε λειτουργία.

Στους περιορισμούς εισάγουμε αρχικά ότι το άθροισμα της binary μεταβλητής για τη γραμμή Basic και για τη γραμμή NewBasic, δεν μπορεί να είναι μεγαλύτερο του 1, καθώς πρέπει να μη λειτουργούν ταυτόχρονα αυτές οι 2 γραμμές. Αντίστοιχη είναι η λογική για τις γραμμές ShowOff και NewShowOff. Επίσης, η παραγωγή θα πρέπει να είναι προφανώς μικρότερη ή ίση από το production capacity των γραμμών που λειτουργούν, και τέλος συνυπολογίζουμε τον αριθμό των αυτοκινήτων της κάθε κατηγορίας που θα προκύψει από τα conversion rates που μας δίνονται.

Τελικά, όπως παρατηρούμε από τα αποτελέσματα στο δεύτερο μέρος του παραρτήματος (Α), η Αυτοκίνηση μεγιστοποιεί τα κέρδη της στη τελική τιμή των 4040 (σε 1000€). Αυτή, όπως φαίνεται στη συνέχεια, προκύπτει από τις 2 καινούργιες γραμμές παραγωγής, δηλαδή την NewBasic και την NewShowOff. Επομένως συμφέρει στην Αυτοκίνηση να αναβαθμίσει/αντικαταστήσει τις 3 παλιές γραμμές παραγωγής με τις 2 καινούργιες. Μάλιστα, στο αντίστοιχο παράρτημα φαίνεται και ο αριθμός από κάθε μοντέλο που απαιτείται να παραχθεί σε κάθε μία γραμμή που λειτουργεί.

## Παράρτημα

### (Α) ΑΠΟΤΕΛΕΣΜΑΤΑ - ΛΥΣΕΙΣ

#### Πρώτο πρόβλημα

```
Solution:
Optimal Schedule Length: 35.0
Machine 0: job_0_task_0   job_2_task_0   job_3_task_0   job_5_task_0   job_4_task_0   job_1_task_0
           [0,3]         [3,6]         [6,11]        [11,18]        [18,23]        [23,29]
Machine 1: job_0_task_1   job_2_task_1   job_3_task_1   job_5_task_1   job_4_task_1   job_1_task_1
           [3,8]         [8,10]        [11,15]       [18,23]       [23,27]       [29,33]
Machine 2: job_0_task_2   job_2_task_2   job_3_task_2   job_5_task_2   job_4_task_2   job_1_task_2
           [8,13]        [13,17]       [17,23]       [23,29]       [29,32]       [33,35]

Statistics
- conflicts: 5
- branches : 39
- wall time: 0.019606 s

Process finished with exit code 0
```

## Δεύτερο πρόβλημα

```
C:\Users\conmy\Documents\i × + v
ampl: model model10027.mod
ampl: data data10027.dat
ampl: option solver cplex;
ampl: solve;
CPLEX 22.1.1.0: optimal integer solution; objective 4040
13 MIP simplex iterations
0 branch-and-bound nodes
ampl: display n
ampl? ;
n :=
BASIC      BASIC      0
BASIC      SHOWOFF    0
BASIC      LUXURIOUS   0
BASIC      NEWBASIC    1400
BASIC      NEWSHOWOFF  0
SHOWOFF    BASIC      0
SHOWOFF    SHOWOFF    0
SHOWOFF    LUXURIOUS   0
SHOWOFF    NEWBASIC    100
SHOWOFF    NEWSHOWOFF  1000
LUXURIOUS  BASIC      0
LUXURIOUS  SHOWOFF    0
LUXURIOUS  LUXURIOUS   0
LUXURIOUS  NEWBASIC    0
LUXURIOUS  NEWSHOWOFF  800
;

ampl: |
```