# ΠΡΟΒΛΗΜΑ ΧΡΟΝΟΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΑΓΩΝΩΝ ΓΙΑ ΕΡΑΣΙΤΕΧΝΙΚΑ ΠΡΩΤΑΘΛΗΜΑΤΑ

## (Sports Scheduling problem for amateur leagues)

## Γραμμική και Συνδυαστική Βελτιστοποίηση

## Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών

## ΓΕΩΡΓΙΟΣ ΤΣΙΑΛΙΟΣ

## ΑΜ: 1072868

ΠΕΡΙΕΧΟΜΕΝΑ:

1—ΕΙΣΑΓΩΓΗ, οργάνωση εγγράφου

2—ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

2.1—ΔΙΑΦΟΡΕΣ ΕΠΑΓΓΕΛΜΑΤΙΚΩΝ – ΕΡΑΣΙΤΕΧΝΙΚΩΝ ΠΡΩΤΑΘΛΗΜΑΤΩΝ

2.2 – ΑΝΑΣΚΟΠΗΣΗ ΣΧΕΤΙΚΩΝ ΕΡΕΥΝΩΝ

2.3 — ΠΑΡΑΔΕΙΓΜΑ ΜΕΛΕΤΗΣ

3 – ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

4 –ΕΠΙΛΥΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

5 – ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΚΤΕΛΕΣΗΣ

6 – ΕΠΕΚΤΑΣΙΜΟΤΗΤΑ ΠΡΟΓΡΑΜΜΑΤΟΣ

7 -- ΒΙΒΛΙΟΓΡΑΦΙΑ

## ΕΙΣΑΓΩΓΗ

Σε αυτήν την εργασία θα μελετήσουμε το πρόβλημα του *Χρονοπρογραμματισμού Αγώνων σε Ερασιτεχνικά Πρωταθλήματα*. Στη διεθνή βιβλιογραφία το πρόβλημα αυτό είναι γνωστό ως *Sports Scheduling* ή *Tournament Timetabling*. Το πρόβλημα συνίσταται στην εύρεση του βέλτιστου προγράμματος αγώνων, δηλ. όπως αναφέρεται στο [2], στόχος είναι να καθορίσουμε το ποιες ομάδες θα παίξουν αντίπαλες και πότε.

Το παρόν έγγραφο οργανώνεται με τον ακόλουθο τρόπο. Στην *Ενότητα 2* περιγράφεται αναλυτικά το πρόβλημα και γίνεται μία περίληψη της ήδη υπάρχουσας βιβλιογραφίας. Η *Ενότητα 3* περιέχει τη μαθηματική μοντελοποίηση του προβλήματος, χρησιμοποιώντας τεχνικές γραμμικού προγραμματισμού. Στην *Ενότητα 4* παρουσιάζεται η επίλυση του προβλήματος με τη βοήθεια της γλώσσας προγραμματισμού Python, ενώ στην *Ενότητα 5* αναλύονται τα αποτελέσματα της εκτέλεσης του κώδικα. Τέλος, στην *Ενότητα 6* εξετάζουμε την επεκτασιμότητα του προγράμματος που αναπτύχθηκε, ενώ στην *Ενότητα 7* περιλαμβάνεται η βιβλιογραφία που χρησιμοποιήθηκε.

## ΠΕΡΙΓΡΑΦΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Το πρόβλημα του χρονοπρογραμματισμού αγώνων έχει μελετηθεί εκτενώς τις τελευταίες δεκαετίες και έχουν προταθεί διάφορες μέθοδοι για την επίλυση του. Ωστόσο, η συντριπτική πλειοψηφία των ερευνών που έχουν γίνει αφορούν επαγγελματικά πρωταθλήματα (πχ ποδοσφαίρου: Premier League [10], Serie A [9], μπάσκετ: NBA [11], Argentinian Liga A [12]). Τα ερασιτεχνικά πρωταθλήματα, που αφορούν το αντικείμενο αυτής της εργασίας, έχουν σημαντικές διαφορές στον τρόπο λειτουργίας τους σε σχέση με τα επαγγελματικά πρωταθλήματα. Σε αυτά, οι συμμετέχοντες δεν είναι επαγγελματίες και δεν αμείβονται. Επομένως, δεν αναφερόμαστε σε μικρές κατηγορίες δημοφιλών αθλημάτων, όπως πχ η  Γ΄ Εθνική Κατηγορία Ποδοσφαίρου, καθώς αυτές έχουν ημιεπαγγελματικό χαρακτήρα. Αντίθετα, θα ασχοληθούμε με πρωταθλήματα που διοργανώνονται από κοινότητες σε εθελοντική βάση όπως φοιτητικά τουρνουά ποδοσφαίρου, πρωταθλήματα ποδοσφαίρου σάλας [5], τοπικά πρωταθλήματα επιτραπέζιας αντισφαίρισης (πινγκ-πονγκ) [7] κα.

### 2.1 ΣΥΓΚΡΙΣΗ ΕΠΑΓΓΕΛΜΑΤΙΚΩΝ-ΕΡΑΣΙΤΕΧΝΙΚΩΝ ΠΡΩΤΑΘΛΗΜΑΤΩΝ

Προκειμένου να κατανοήσουμε καλύτερα το πρόβλημα του Χρονοπρογραμματισμού Αγώνων σε Ερασιτεχνικά Πρωταθλήματα, ας το συγκρίνουμε με αυτό στα Επαγγελματικά Πρωταθλήματα, καθότι η πλειοψηφία του κόσμου είναι περισσότερο εξοικειωμένη με αυτά. Θα σταθούμε στους περιορισμούς που πρέπει να ικανοποιούνται ώστε να θεωρείται ένα πρόγραμμα αγώνων εφικτό και στις δύο περιπτώσεις. Επίσης, θα εξετάσουμε το κριτήριο βελτιστοποίησης των δύο περιπτώσεων, δηλ. την τιμή εκείνη που προσπαθούν να μεγιστοποιήσουν ή ελαχιστοποιήσουν τα προγράμματα αγώνων. Οι κυριότερες διαφορές μεταξύ αυτών αναγράφονται στον *Πίνακα 1*.

|  |  |
| --- | --- |
| Επαγγελματικά Πρωταθλήματα | Ερασιτεχνικά Πρωταθλήματα |
| Εναλλαγή εντός έδρας – εκτός έδρας αγώνων | Breaks ασήμαντα, όλοι οι αγώνες ισοδύναμοι |
| Ομάδες από την ίδια πόλη | Ασήμαντη η έδρα των ομάδων |
| Χρήση σταδίου για άλλες εκδηλώσεις | Δεν λαμβάνουν χώρα άλλες εκδηλώσεις στο στάδιο |
| Εμπορικοί σκοποί (οπαδοί, τηλεοπτική κάλυψη) | Λίγοι οπαδοί, δεν υπάρχει τηλεοπτική κάλυψη |
| Πρόγραμμα αγώνων ολόκληρης της σεζόν | Πρόγραμμα αγώνων σε εβδομαδιαία βάση |
| Σπάνια αναβολή αγώνων | Συχνή αναβολή αγώνων |
| Ίση κατανομή αγώνων στη σεζόν | Άνιση κατανομή αγώνων στη σεζόν |
| Ελαχιστοποίηση διανυόμενης απόστασης ή breaks | Μεγιστοποίηση διαθεσιμότητας παικτών |

*Πίνακας 1: Διαφορές Επαγγελματικών-Ερασιτεχνικών πρωταθλημάτων*

Αρχικά, βλέπουμε πως στα επαγγελματικά πρωταθλήματα είναι σημαντικό οι ομάδες να παίζουν εναλλάξ εντός έδρας – εκτός έδρας αγώνες. Είναι δηλ. επιθυμητό να μην υπάρχουν διαδοχικοί αγώνες εντός έδρας ή εκτός έδρας. Ένα τέτοιο διάστημα ονομάζεται στη βιβλιογραφία ως *break* (διάλειμμα). Στα ερασιτεχνικά πρωταθλήματα, η ύπαρξη διαλειμμάτων είναι αδιάφορη, καθώς συνήθως οι αποστάσεις μεταξύ των εδρών των ομάδων είναι μικρές ή ανύπαρκτες, όταν όλοι οι αγώνες του πρωταθλήματος λαμβάνουν χώρα στο ίδιο στάδιο. Ακόμα, στα επαγγελματικά πρωταθλήματα πρέπει να ληφθεί ειδική μέριμνα για τις ομάδες που έχουν έδρα την ίδια πόλη. Οι ομάδες αυτές πρέπει να μην παίζουν ταυτόχρονα εντός έδρας, για λόγους κυκλοφοριακούς και δημόσιας ασφάλειας. Αντίθετα, όπως ήδη αναφέραμε, στα ερασιτεχνικά πρωταθλήματα η έδρα των ομάδων είναι ασήμαντη και δεν παίζει ρόλο στη δημιουργία του προγράμματος αγώνων.

Στη συνέχεια, παρατηρούμε διάφορες μεταξύ των δυο ειδών πρωταθλημάτων που οφείλονται στο μέγεθος των διοργανώσεων. Συγκεκριμένα, τα στάδια ομάδων επαγγελματικών πρωταθλημάτων συχνά χρησιμοποιούνται και για άλλες εκδηλώσεις,, όπως συναυλίες, ομιλίες κλπ. Κάτι τέτοιο σπάνια συμβαίνει στα ερασιτεχνικά πρωταθλήματα. Επίσης, το πρόγραμμα αγώνων σε επαγγελματικά πρωταθλήματα επηρεάζεται από εμπορικούς παράγοντες, όπως η μεγιστοποίηση της παρουσίας οπαδών στο γήπεδο (να μην παίζονται αγώνες σε εργάσιμες ώρες) ή η τηλεοπτική κάλυψη των αγώνων (να μην παίζονται αγώνες παράλληλα με άλλα σημαντικά τηλεοπτικά δρώμενα). Αυτοί οι παράγοντες είναι σημαντικοί, καθώς καθορίζουν το οικονομικό κέρδος των ομάδων από την έκδοση εισιτηρίων και την τηλεθέαση των αγώνων. Αντίθετα, στα ερασιτεχνικά πρωταθλήματα δεν λαμβάνονται υπόψιν τέτοιοι παράγοντες, καθώς οι αγώνες σπάνια έχουν πολλούς οπαδούς ή τηλεοπτική κάλυψη.

Άλλες διάφορες στους περιορισμούς των προγραμμάτων αγώνων σχετίζονται με τη διαθεσιμότητα των παικτών. Σε επαγγελματικά πρωταθλήματα, οι ομάδες αποτελούνται από επαγγελματίες αθλητές και οφείλουν να παίξουν στους αγώνες, όποτε κι αν κανονιστούν αυτοί. Αυτό όμως δεν ισχύει στα ερασιτεχνικά πρωταθλήματα, όπου η ενασχόληση των παικτών με το άθλημα γίνεται στα πλαίσια χόμπι. Οι παίκτες αυτοί έχουν τις δικές τους προσωπικές και επαγγελματικές υποχρεώσεις, οι οποίες δεσμεύουν αρκετό από τον χρόνο τους. Έτσι, η διαθεσιμότητα τους για να παραστούν σε αγώνες της ομάδας τους είναι περιορισμένη. Αυτό έχει τις ακόλουθες συνέπειες. Αρχικά, το πρόγραμμα αγώνων των ερασιτεχνικών πρωταθλημάτων πρέπει να βγαίνει σε εβδομαδιαία βάση, επειδή η διαθεσιμότητα των παικτών δεν είναι σταθερή και δεν είναι γνωστή από την αρχή του πρωταθλήματος. Ακόμα, αποτελεί συχνό φαινόμενο η αναβολή αγώνων, λόγω καιρού ή έκτακτων υποχρεώσεων των παικτών. Επίσης, υπάρχουν χρονικές περίοδοι εντός της σεζόν κατά τις οποίες η διαθεσιμότητα των παικτών είναι μικρότερη κι έτσι διεξάγονται λιγότεροι αγώνες, όπως για παράδειγμα περίοδοι διακοπών (Χριστούγεννα, Πάσχα), εξεταστικές περίοδοι για φοιτητικά τουρνουά κλπ. Επομένως, παρατηρείται άνιση κατανομή των αγώνων κατά τη διάρκεια της σεζόν. Στον αντίποδα, στα επαγγελματικά τουρνουά ολόκληρο το πρόγραμμα αγώνων υπολογίζεται πριν την έναρξη του πρωταθλήματος, οι αγώνες αναβάλλονται σπάνια και μόνο για πολύ σημαντικό λόγο και οι αγώνες κατανέμονται ομοιόμορφα εντός της σεζόν.

Τέλος, τα προγράμματα αγώνων σε επαγγελματικά πρωταθλήματα έχουν συνήθως ως στόχο την ελαχιστοποίηση των διαλειμμάτων (breaks) πχ Serie A [9] ή την ελαχιστοποίηση της απόστασης που διανύουν οι ομάδες για πρωταθλήματα στα οποία οι έδρες των ομάδων απέχουν πολύ μεταξύ τους πχ ΝΒΑ [11]. Στα ερασιτεχνικά πρωταθλήματα, το μέγεθος που προσπαθεί να βελτιστοποιήσει το πρόγραμμα αγώνων είναι η διαθεσιμότητα των παικτών, δηλ. να μπορούν όσο το δυνατόν περισσότεροι παίκτες κάθε ομάδας να παίξουν στους αγώνες.

### 2.2 ΑΝΑΣΚΟΠΗΣΗ ΣΧΕΤΙΚΩΝ ΕΡΕΥΝΩΝ

Μελετώντας τη βιβλιογραφία, παρατηρούμε πως το πρόβλημα του *Χρονοπρογραμματισμού Αγώνων σε Ερασιτεχνικά Πρωταθλήματα* είναι δυνατόν να επιλυθεί με διάφορες μεθόδους. Στις εργασίες [3], [4], [7] εξετάζεται το πρόβλημα προγραμματισμού αγώνων σε μη-επαγγελματικά πρωταθλήματα επιτραπέζιας αντισφαίρισης (πινγκ-πονγκ) και επιλύεται με χρήση Γενετικού Αλγορίθμου (*Genetic Algorithm*) [3], Μιμητικού Αλγορίθμου (*Memetic Algorithm*) [4] και Ακέραιου Γραμμικού Προγραμματισμού (*Integer Linear Programming*) [7]. Οι εργασίες [1], [6] αφορούν ερασιτεχνικά τουρνουά. Συγκεκριμένα, στην εργασία [1] αναφέρεται στο πρόβλημα ενός φοιτητικού τουρνουά και χρησιμοποιείται ένας υβριδικός αλγόριθμος Προγραμματισμού Περιορισμών (*Constraint Programming*) και Αναζήτησης Γειτονιάς (*Neighborhood Search*). Στην εργασία [6] τίθεται το πρόβλημα προγραμματισμού σε ένα ερασιτεχνικό τουρνουά αντισφαίρισης (τένις) στην Ιταλία, το οποίο επιλύεται κάνοντας χρήση μίας ευρετικής μεθόδου (*heuristic*) δύο βημάτων. Τέλος, στο έγγραφο [5] μελετάται το πρόβλημα προγραμματισμού σε ένα μη-επαγγελματικό πρωτάθλημα ποδοσφαίρου εσωτερικού χώρου (σάλας) στο Βέλγιο. Το πρόβλημα λύνεται χρησιμοποιώντας Ακέραιο Προγραμματισμό και μια ευρετική μέθοδο βασισμένη στην αναζήτηση ταμπού (*tabu search*).

### 2.3 ΠΑΡΑΔΕΙΓΜΑ ΜΕΛΕΤΗΣ

Από την προηγούμενη υποενότητα, γίνεται φανερό πως κάθε περίπτωση ερασιτεχνικού πρωταθλήματος είναι διαφορετική. Οι παράμετροι από τις οποίες εξαρτάται το πρόγραμμα αγώνων αλλάζουν ελαφρώς με βάση τον τύπο του αθλήματος και τις πολιτικές που έχουν επιλέξει οι διοργανωτές του για τον τρόπο λειτουργίας του. Επομένως, κάθε λύση που προτείνεται στη βιβλιογραφία αφορά μια συγκεκριμένη περίπτωση ερασιτεχνικού πρωταθλήματος. Στην παρούσα εργασία, θα επιλύσουμε το πρόβλημα ενός φοιτητικού πρωταθλήματος ποδοσφαίρου 5x5, διότι υπάρχει μέσα από προσωπική εμπειρία, εξοικείωση με το συγκεκριμένο πρόβλημα και γνώση των παραμέτρων του. Οι κανόνες που επιλέχθηκαν για τη λειτουργία αυτού του πρωταθλήματος περιγράφονται παρακάτω.

Το πρωτάθλημα αποτελείται από έναν όμιλο των 8 ομάδων. Για την ολοκλήρωση του πρωταθλήματος απαιτείται κάθε ομάδα να παίξει με όλες τις υπόλοιπες 1 φορά (*single round robin tournament*). Υπάρχει 1 γήπεδο, το οποίο παρέχεται από τους διοργανωτές του πρωταθλήματος και όλοι οι αγώνες διεξάγονται σε αυτό. Επομένως, δεν μπορούν να διεξαχθούν 2 αγώνες ταυτόχρονα (*asynchronous tournament*). Οι διοργανωτές του πρωταθλήματος ορίζουν ότι αγώνες μπορούν να διεξαχθούν μόνο τις καθημερινές μέρες και έως 1 αγώνας την ημέρα. Επομένως, κάθε εβδομάδα μπορούν να διεξαχθούν έως 5 αγώνες. Οι ομάδες αποτελούνται από 6 παίκτες (5 + 1 αλλαγή). Μια ομάδα μπορεί να παίξει όταν είναι διαθέσιμοι τουλάχιστον 5 παίκτες της.

Οι παίκτες όλων των ομάδων δηλώνουν κάθε Σάββατο ποιες μέρες της επόμενης εβδομάδας προτιμούν να παίξουν. Δίνουν δηλ. σε κάθε μέρα μία από τις παρακάτω τιμές, οι οποίες συμβολίζουν τα εξής:

0 → δεν είμαι διαθέσιμος

4 → είμαι διαθέσιμος, αλλά δεν το προτιμώ

7 → είμαι διαθέσιμος και το προτιμώ αρκετά

10 → είμαι διαθέσιμος και το προτιμώ πολύ

Έπειτα, υπολογίζεται το πρόγραμμα αγώνων της εβδομάδας αυτής.

Εξαιτίας της περιορισμένης διαθεσιμότητας των παικτών, υπάρχει περίπτωση σε κάποια εβδομάδα να μην μπορεί να διεξαχθεί ένας αγώνας. Τότε, θα διεξαχθούν λιγότεροι αγώνες εκείνη την εβδομάδα. Πλέον, οι εμπλεκόμενες ομάδες έχουν παίξει λιγότερους αγώνες από τις υπόλοιπες και ΄χρωστάνε΄ τον συγκεκριμένο αγώνα.

Προσοχή πρέπει να δοθεί στο πόσες φορές μπορεί να παίξει μια ομάδα εντός μιας εβδομάδας. Αυτό είναι μεταβλητό μέγεθος και εξαρτάται από το εάν η ομάδα χρωστάει αγώνες ή όχι. Συγκεκριμένα, αν μια ομάδα δεν χρωστάει αγώνες, μπορεί να παίξει από 0 έως 1 αγώνα την εβδομάδα. Ωστόσο, αν μία ομάδα χρωστάει αγώνες, μπορεί να παίξει από 0 έως 2 αγώνες την εβδομάδα.

Οι διοργανωτές του πρωταθλήματος πληρώνονται από τις ομάδες για κάθε αγώνα που αυτές παίζουν. Επομένως, επιθυμούν να χαθούν συνολικά όσο το δυνατόν λιγότεροι αγώνες. Για αυτό, η διοργάνωση φροντίζει για την αναπλήρωση των χρωστούμενων αγώνων με 2 τρόπους:

* Ύπαρξη περισσότερων διαθέσιμων χρονοθυρίδων (*time slots*) από τις απαιτούμενες: 5 διαθέσιμα time slots ανά εβδομάδα > 4 αγώνες ανά εβδομάδα (αγωνιστική). Το επιπλέον αυτό time slot δίνει την ευκαιρία στις ομάδες να παίξουν τους αγώνες που χρωστάνε (*catch-up slot* [8]).
* Ύπαρξη περισσότερων αγωνιστικών πρωταθλήματος (εβδομάδων) από τις απαιτούμενες: Εφόσον η κάθε ομάδα πρέπει να παίξει με 7 ομάδες, το πρωτάθλημα υπό κανονικές συνθήκες, θα ολοκληρωθεί σε 7 αγωνιστικές. Όμως εξαιτίας των χρωστούμενων αγώνων, μπορεί να χρειαστούν παραπάνω αγωνιστικές. Οι διοργανωτές ορίζουν την ύπαρξη μίας παραπάνω αγωνιστικής (8η εβδομάδα), προκειμένου να διεξαχθούν σε αυτήν οι χρωστούμενοι αγώνες. Όσοι αγώνες δεν διεξαχθούν και σε αυτήν την εβδομάδα, θεωρούνται πλέον χαμένοι και για τις δύο ομάδες. Δεν υπάρχει δυνατότητα ορισμού παραπάνω αγωνιστικών, εξαιτίας χρονικών περιορισμών που επιβάλουν τη λήξη του πρωταθλήματος (έναρξη επόμενου πρωταθλήματος, απομάκρυνση των φοιτητών από την πόλη φοίτησης τους στο τέλος του εξαμήνου κλπ).

## ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Σε αυτήν την ενότητα θα μεταβούμε από την παραπάνω λεκτική περιγραφή του προβλήματος στη μαθηματική μοντελοποίηση του, χρησιμοποιώντας Ακέραιο Γραμμικό Προγραμματισμό (*Integer Linear Programming*).

Αρχικά, χρειάζεται να ορίσουμε κάποια σύνολα, τα οποία περιέχουν τα απαραίτητα δεδομένα για τον υπολογισμό του προγράμματος αγώνων. Τα σταθερά σύνολα αντιπροσωπεύουν δεδομένα που παραμένουν διαρκώς τα ίδια, ενώ τα μεταβλητά σύνολα αντιπροσωπεύουν δεδομένα που ανανεώνονται σε εβδομαδιαία βάση.

**Σύνολα**

* **Σταθερά**
  + T: σύνολο με όλες τις ομάδες (Teams)
  + D: σύνολο με τις διαθέσιμες ημέρες (Days)
* **Μεταβλητά**
* M: σύνολο με τους αγώνες που έχουν ήδη διεξαχθεί (Matches Played)
* Ε: σύνολο με το πλήθος αγώνων που χρωστάει η κάθε ομάδα (Extra)
* P: σύνολο με τις προτιμήσεις των παικτών για τις διαθέσιμες ημέρες (Preferences)

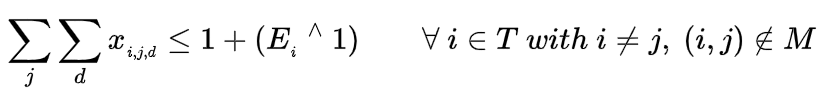
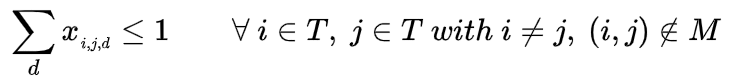
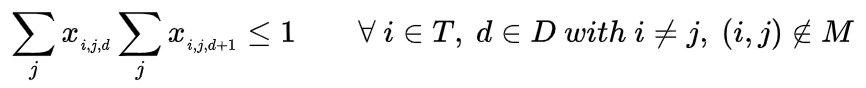
Στη συνέχεια θα ορίσουμε τις μεταβλητές απόφασης, δηλ. τις μεταβλητές που καλούμαστε να υπολογίσουμε την τιμή τους. Οι μεταβλητές αυτές καθορίζουν το πρόγραμμα αγώνων.

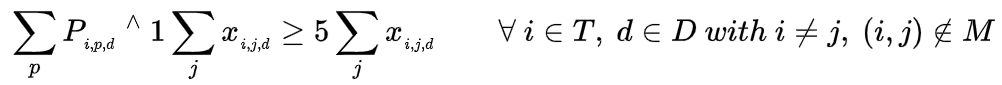
**Μεταβλητές απόφασης**

xi,j,d = , i ϵ T, j ϵ T, d ϵ D

Έπειτα, θα εκφράσουμε τους περιορισμούς του προβλήματος (*hard constraints*). Όταν ένα πρόγραμμα αγώνων ικανοποιεί όλους τους περιορισμούς του προβλήματος, τότε θεωρείται εφικτή λύση του (*feasible solution*).

**Περιορισμοί**

1. Μία ομάδα μπορεί να παίξει:
2. Μία ομάδα μπορεί να παίξει έως 1 φορά με μία άλλη ομάδα.
3. Μία ομάδα (που χρωστάει αγώνες) δεν μπορεί να παίξει σε 2 διαδοχικές μέρες.
4. Οι αγώνες εντός έδρας – εκτός έδρας είναι οι ίδιοι.
5. Εικόνα που περιέχει κείμενο, γραμματοσειρά, λευκό, καλλιγραφία

   Περιγραφή που δημιουργήθηκε αυτόματαΜπορεί να διεξαχθεί έως 1 αγώνας την ημέρα.
6. Μία ομάδα μπορεί να παίξει όταν είναι διαθέσιμοι τουλάχιστον 5 παίκτες της.

Τέλος, θα καταστρώσουμε την αντικειμενική συνάρτηση του προβλήματος μας. Θα δημιουργήσουμε δηλ. μία μετρική απόδοσης, για να μπορούμε να συγκρίνουμε τα εφικτά προγράμματα αγώνων και να επιλέγουμε το καλύτερο από αυτά. Ωστόσο, ο σχηματισμός της αντικειμενικής συνάρτησης παρουσιάζει κάποιες προκλήσεις. Συγκεκριμένα, στο πρόβλημα μας υπάρχουν δύο πλευρές με διαφορετικούς στόχους. Από τη μία μεριά, οι διοργανωτές του πρωταθλήματος επιδιώκουν τη μεγιστοποίηση των παιχνιδιών που θα διεξαχθούν, καθώς αυτό αυξάνει το κέρδος τους. Επομένως, για αυτούς είναι σημαντικό να μην χαθούν παιχνίδια. Ταυτόχρονα, οι παίκτες των ομάδων ενδιαφέρονται κυρίως για το πότε θα διεξαχθούν τα παιχνίδια, αναφορικά με τη διαθεσιμότητα τους. Επιθυμούν η ομάδα τους να αγωνίζεται όταν είναι σύσσωμη, ώστε να παίξουν με ολόκληρη την παρέα τους – άλλωστε είναι πιο κουραστικό για τους παίκτες να παίξουν σε αγώνα χωρίς αλλαγή (5 άτομα αντί για 6). Ακόμα, έτσι αποφεύγουν να χάσουν οι ίδιοι τους κάποιον αγώνα της ομάδας τους. Μάλιστα, επιθυμούν να παίζουν κατά το δυνατόν τις ημέρες που προτιμούν περισσότερο.

Εκτός από αυτά, θα πρέπει να αντιμετωπίσουμε κι ένα θέμα δικαιοσύνης που εμφανίζεται συχνά σε τέτοια πρωταθλήματα. Συγκεκριμένα, καθώς οι ομάδες χρωστάνε αγώνες κατά τη διάρκεια της σεζόν, υπάρχει διαφορά μεταξύ του πλήθους των αγώνων που έχει παίξει η κάθε ομάδα (*Games Played Difference – GPD).* Έτσι, όπως αναφέρεται στο [8], η βαθμολογία του ομίλου γίνεται ανακριβής και δεν αντικατοπτρίζει πραγματικά τη δυναμικότητα των ομάδων. Πιο σημαντικά όμως, το φαινόμενο αυτό δίνει πλεονέκτημα στις ομάδες που έχουν παίξει λιγότερους αγώνες, καθώς γνωρίζουν πότε μπορούν να αρκεστούν στην ισοπαλία, ενώ μπορεί να οδηγήσει και σε στήσιμο αγώνων (*match fixing*). Για τους λόγους αυτούς, κρίνεται σημαντική η ελαχιστοποίηση του GPD στην αντικειμενική συνάρτηση.

Λαμβάνοντας αυτά υπόψιν, κατασκευάσαμε την παρακάτω αντικειμενική συνάρτηση. Πρόκειται για συνάρτηση μεγιστοποίησης και είναι δίκλαδη, καθώς μεταβάλλεται ελαφρώς ο τύπος της όταν υπάρχουν ομάδες που χρωστάνε αγώνες.

**Αντικειμενική συνάρτηση**

* Εικόνα που περιέχει κείμενο, γραμματοσειρά, λευκό, γραφικός χαρακτήρας

  Περιγραφή που δημιουργήθηκε αυτόματαΌταν δεν υπάρχουν ομάδες που χρωστάνε αγώνες:
* Εικόνα που περιέχει κείμενο, γραμματοσειρά, λευκό, γραμμή

  Περιγραφή που δημιουργήθηκε αυτόματαΌταν υπάρχουν ομάδες που χρωστάνε αγώνες:

Ας αναλύσουμε πρώτα την αντικειμενική συνάρτηση στην περίπτωση που δεν υπάρχουν ομάδες που χρωστάνε αγώνες. Η συνάρτηση αποτελείται από 3 όρους, όπου ο κάθε ένας αναπαριστά έναν διαφορετικό στόχο. Ζητούμενο είναι η μεγιστοποίηση κάθε όρου. Οι συντελεστές μπροστά από αυτούς τους όρους χρησιμεύουν στην κανονικοποίηση της τιμής της αντικειμενικής συνάρτησης στο 100. Αυτό είναι χρήσιμο, προκειμένου να διακρίνουμε με ευκολία πόσο καλή είναι η βέλτιστη λύση του προγράμματος, δηλ. πόσο κοντά είναι στο θεωρητικά ιδανικό σενάριο, όπου ικανοποιούνται πλήρως όλοι οι στόχοι. Έτσι, ο κάθε συντελεστής είναι ένα κλάσμα, στο οποίο ο παρονομαστής αποτελεί τη μέγιστη τιμή που μπορεί να πάρει ο συγκεκριμένος στόχος (ο υπόλοιπος όρος). Διαιρώντας την τιμή που παίρνει ο στόχος διά τον παρονομαστή αυτόν, παίρνουμε το ποσοστό % στο οποίο ικανοποιείται ο συγκεκριμένος στόχος. Τέλος, το ποσοστό αυτό πολλαπλασιάζεται με τον αριθμητή του κλάσματος, ο οποίος αντιπροσωπεύει τη βαρύτητα που έχει ο στόχος αυτός για την αντικειμενική συνάρτηση. Πιο συγκεκριμένα:

* Ο πρώτος όρος αντιστοιχεί στο πλήθος των αγώνων αυτής της εβδομάδας. Αφού δεν υπάρχουν ομάδες που χρωστάνε αγώνες, το μέγιστο πλήθος αγώνων που μπορούν να διεξαχθούν σε μία εβδομάδα είναι 4. Όμως, το άθροισμα του όρου αυτού θα μετρήσει κάθε παιχνίδι δύο φορές πχ ομάδα A vs ομάδα Β αλλά και ομάδα Β vs ομάδα Α. Στην πραγματικότητα πρόκειται για το ίδιο, ένα παιχνίδι. Για αυτό απαιτείται διαίρεση με το 2, κι έτσι το 4 γίνεται 8 στον παρονομαστή του συντελεστή. Τέλος, ο αριθμητής 50 μας δείχνει το πόσο σημαντικός είναι ο συγκεκριμένος στόχος. Δηλ. αν το πρόγραμμα αγώνων καταφέρει να ικανοποιήσει αυτόν τον στόχο (να μεγιστοποιήσει το πλήθος αγώνων) θα λάβει 50 πόντους. Αν καταφέρει να τον ικανοποιήσει κατά 75% (δηλ. να διεξαχθούν 3 αγώνες αντί για 4), θα λάβει 0.75 ∙ 50 = 37.5 πόντους κοκ.
* Ο δεύτερος όρος αντιστοιχεί στο πλήθος των παικτών που θα αγωνιστούν (δηλ. είναι διαθέσιμοι τις ημέρες που αγωνίζεται η ομάδα τους). Για να υπολογίσουμε αυτό το μέγεθος, χρησιμοποιούμε το λογικό ΄και΄ (AND). Συγκεκριμένα, η έκφραση Pi,p,d ^ 1 μετατρέπει σε δυαδική την προτίμηση των παικτών. Πιο αναλυτικά, δίνει 1 όταν ο παίκτης p της ομάδας i είναι διαθέσιμος την ημέρα d (δηλ. προτίμηση Pi,p,d = 4 ή 7 ή 10) και 0 όταν ο παίκτης δεν είναι διαθέσιμος (δηλ. προτίμηση Pi,p,d = 0). Αφού το μέγιστο πλήθος αγώνων είναι 4 και σε κάθε αγώνα συμμετέχουν 12 παίκτες (6+6), τότε το μέγιστο πλήθος παικτών που μπορεί να αγωνιστεί σε μία εβδομάδα είναι 4 ∙ 12 = 48. Έτσι προκύπτει ο παρονομαστής του συντελεστή. Τέλος, επιλέγεται βαρύτητα ίση με 30 για τον συγκεκριμένο στόχο.
* Ο τρίτος όρος αντιστοιχεί στην προτίμηση των παικτών. Στόχος δηλ. είναι να αγωνίζονται οι παίκτες τις ημέρες που προτιμούν περισσότερο. Ο όρος αυτός μοιάζει πολύ με τον προηγούμενο, με τη διαφορά ότι τώρα η προτίμηση των παικτών δεν μετατρέπεται σε δυαδική. Η μέγιστη προτίμηση των παικτών προκύπτει αν αγωνιστούν και οι 48 παίκτες σε ημέρα που προτιμούν πολύ (προτίμηση Pi,p,d = 10). Επομένως, η μέγιστη προτίμηση των παικτών είναι ίση με 480. Έτσι προκύπτει ο παρονομαστής του συντελεστή. Τέλος, επιλέγεται βαρύτητα ίση με 20 για τον συγκεκριμένο στόχο.

Αξίζει να παρατηρήσουμε ότι από τους 100 διαθέσιμους πόντους της αντικειμενικής συνάρτησης, οι 50 δίνονται για την ικανοποίηση του στόχου των διοργανωτών και οι υπόλοιποι 50 για την ικανοποίηση των στόχων των παικτών. Έτσι, επιτυγχάνουμε τη δικαιοσύνη μεταξύ των δύο αυτών πλευρών.

Η αντικειμενική συνάρτηση παρουσιάζει μικρές διαφορές, στην περίπτωση που υπάρχουν ομάδες που χρωστάνε αγώνες. Οι 3 πρώτοι όροι παραμένουν σχεδόν ίδιοι, καθώς αλλάζουν μόνο οι συντελεστές τους. Πλέον το μέγιστο πλήθος αγώνων είναι 5, το μέγιστο πλήθος παικτών που θα αγωνιστούν είναι 5 ∙ 12 = 60 και η μέγιστη προτίμηση των παικτών είναι 60 ∙ 10 = 600. Τώρα όμως, χρειάζεται να προσθέσουμε έναν επιπλέον όρο για την ελάττωση του GPD, όπως είχαμε αναφέρει προηγουμένως. Στον όρο αυτό δίνεται βαρύτητα ίση με 10. Οι 10 αυτοί πόντοι που δόθηκαν στον νέο όρο, αφαιρέθηκαν από αυτόν της προτίμησής των παικτών, καθώς αυτός κρίθηκε ο λιγότερο σημαντικός. Ο νέος όρος εκφράζει το πλήθος των αγώνων που παίζουν οι ομάδες που χρωστάνε αγώνες. Προσπαθούμε δηλ. να μεγιστοποιήσουμε τις φορές που παίζουν αυτές οι ομάδες. Με αυτόν τον τρόπο, οι συγκεκριμένες ομάδες αποκτούν υψηλότερη προτεραιότητα από τις άλλες. Αυτό είναι σημαντικό, καθώς θέλουμε να αποφύγουμε το ενδεχόμενο μία ομάδα να χρωστάει πολλαπλούς αγώνες – τότε θα ήταν δύσκολο να επανέλθει και να ολοκληρώσει όλους τους αγώνες της. Για αυτό, στην περίπτωση που δεν μπορεί να πραγματοποιηθεί κάποιος αγώνας, το πρόγραμμα θα προσπαθήσει στον αγώνα αυτόν να μη συμμετέχουν ομάδες που ήδη χρωστάνε παιχνίδια. Ο παρονομαστής του συντελεστή του νέου όρου εκφράζει τη μέγιστη τιμή του στόχου. Για τον υπολογισμό της, χρησιμοποιείται η συνάρτηση μικρότερης τιμής (*min*). Συγκεκριμένα, έστω ότι 2 ομάδες χρωστάνε αγώνες. Τότε, μπορούν να αγωνιστούν από 2 φορές η κάθε μία, άρα συνολικά 2 ∙ 2 = 4 φορές. Όμως, αν χρωστάνε 6 ομάδες αγώνες, τότε αυτές δεν μπορούν να αγωνιστούν 12 φορές, αφού το μέγιστο πλήθος αγώνων σε μία εβδομάδα είναι 5. Επομένως, η μέγιστη τιμή του στόχου ισούται είτε με το διπλάσιο του πλήθους ομάδων που χρωστάνε αγώνες, όταν αυτό είναι ≤ 10, είτε με 10, όταν αυτό είναι > 10.

## ΕΠΙΛΥΣΗ ΤΟΥ ΠΡΟΒΛΗΜΑΤΟΣ

Στην παρούσα ενότητα, θα επιλύσουμε το πρόβλημα χρονοπρογραμματισμού αγώνων, μεταφράζοντας τη μαθηματική μοντελοποίηση του προβλήματος σε κώδικα, με χρήση της γλώσσας προγραμματισμού Python και της βιβλιοθήκης pulp. Ο κώδικας αυτός βρίσκεται στο αρχείο *problem-8.py*. Παρακάτω θα αναλύσουμε τα κυριότερα κομμάτια του για τη λειτουργία του προγράμματος.

Αρχικά, ας ορίσουμε τα σύνολα δεδομένων που μας χρειάζονται.

**Σύνολα**

# SETS

T = ['ΒΙΓΙΑΡΕΜΑΛ','ΓΙΟΥΒΕΤΣΙ','ΘΛΙΒΕΡΠΟΥΛ','ΜΗN ΨHNECE','ΜΠΑΡΤΣΕΛΙΩΜΑ','ΜΠΥΡΑΚΛΗΣ',

    'ΡΕAΛ MANTPI','ΧΑΒΑΛΕΝΘΙΑ']

D = ['Monday','Tuesday','Wednesday','Thursday','Friday']

# matches that have already been played

M = []

# create teams' need for extra matches and initialize to 0

E = pulp.LpVariable.dicts("E", [(i) for i in T], cat= pulp.LpInteger)

for i in T:

    E[(i)] = 0

# create players' availability

P = pulp.LpVariable.dicts("P", [(i, p, d) for i in T for p in range(1,7) for d in D], cat= pulp.LpInteger)

Πρώτα ορίζουμε τα σταθερά σύνολα των ομάδων του πρωταθλήματος και των διαθέσιμων ημερών. Στη συνέχεια ορίζουμε τα μεταβλητά σύνολα M, E, P, όπως αυτά περιεγράφηκαν στην *Ενότητα 3*. Παρατηρούμε ότι αρχικά, το σύνολο των αγώνων που έχουν ήδη διεξαχθεί είναι κενό ενώ καμία ομάδα δεν χρωστάει αγώνες. Αξιοσημείωτο είναι επίσης πως τα σύνολα E, P ορίζονται με τη βοήθεια της βιβλιοθήκης *pulp* ως λεξικά (*dictionaries*), δηλώνοντας ιεραρχικά τα κλειδιά τους, ενώ οι τιμές τους ορίζονται ως ακέραιοι αριθμοί.

Έπειτα, χρειαζόμαστε δεδομένα για τη διαθεσιμότητα των παικτών κάθε εβδομάδα. Καθότι τέτοια δεδομένα είναι δύσκολο να βρεθούν, τα παράγουμε τυχαία με τον παρακάτω κώδικα.

for w in range(1, 9):

    for i in T:

        for p in range(1,7):

            for d in D:

                P[(i, p, d)] = random.choice([0, 4, 7, 10])

Έτσι, επιλέγεται κάθε φορά τυχαία μία από τις 4 δυνατές τιμές για την προτίμηση των παικτών. Να σημειωθεί η χρήση του *for loop*, η οποία δηλώνει την κάθε αγωνιστική του πρωταθλήματος. Αυτό σημαίνει πως οι προτιμήσεις των παικτών ανανεώνονται σε εβδομαδιαία βάση.

Για την καλύτερη απεικόνιση αυτών των δεδομένων, τα τυπώνουμε σε ξεχωριστό αρχείο με όνομα *availability-8.txt* με τη βοήθεια του παρακάτω κώδικα.

with open('availability-8.txt', 'a') as f:

        f.write(f"--------------- WEEK {w} ---------------\n\n")

        for i in T:

            f.write(f"{i}")

            for p in range(1,7):

                f.write(f"\n\nPlayer {p}")

                f.write(' \n')

                for d in range(len(D)):

                    if d != len(D) - 1:

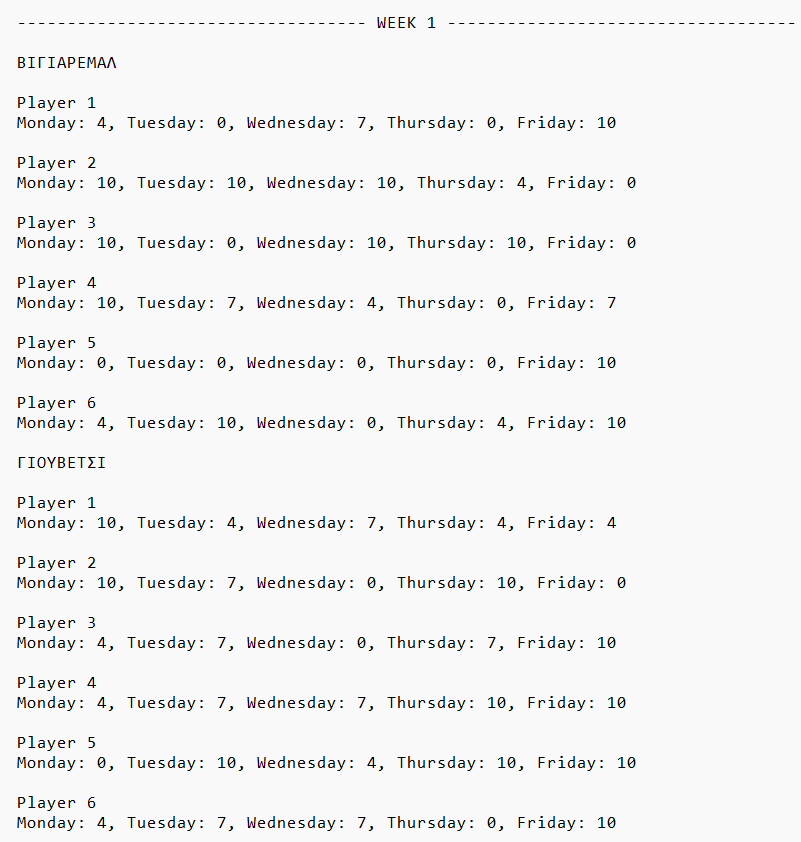
                        f.write(f"{D[d]}: {P[(i, p, D[d])]}, ")

                    else:

                        f.write(f"{D[d]}: {P[(i, p, D[d])]}")

            f.write('\n\n')

Ένα απόσπασμα από το αρχείο *availability-8.txt* φαίνεται στην *Εικόνα 1*.



Εικόνα 1: Απόσπασμα από το αρχείο availability-8.txt

Βλέπουμε δηλ. πως για κάθε εβδομάδα του πρωταθλήματος, τυπώνεται η διαθεσιμότητα όλων των παικτών όλων των ομάδων για κάθε ημέρα της εβδομάδας.

Συνεχίζοντας στον κώδικα, όλες οι παρακάτω εντολές του προγράμματος βρίσκονται εντός του προηγούμενου *for loop*. Επομένως, ο υπόλοιπος κώδικας επιλύει το πρόβλημα χρονοπρογραμματισμού αγώνων μίας εβδομάδας. Επαναλαμβάνοντας αυτή τη διαδικασία 8 φορές (όσες κι οι εβδομάδες του πρωταθλήματος) και διατηρώντας μία μνήμη των αγώνων που έχουν ήδη διεξαχθεί (σύνολο Μ), επιλύουμε το συνολικό πρόβλημα.

Έπειτα, δημιουργούμε μία λίστα με τις ομάδες που χρωστάνε αγώνες και υπολογίζουμε το πλήθος τους.

# The teams that need extra matches

teamsDouble = [i for i in T if E[(i)]]

# Number of teams that need extra matches

numberTeamsDouble = len(teamsDouble)

Τα δεδομένα αυτά θα μας χρειαστούν παρακάτω.

Στη συνέχεια, ορίζουμε το πρόβλημα ως πρόβλημα μεγιστοποίησης.

# Problem Setup

timetable = pulp.LpProblem("Weekly timetable", pulp.LpMaximize)

Ύστερα, ορίζουμε τις μεταβλητές απόφασης ως δυαδικές μεταβλητές.

**Μεταβλητές απόφασης**

# Decision Variables

x = pulp.LpVariable.dicts("x", [(i, j, d) for i in T for j in T if i != j and [i,j] not in M for d in D], cat= pulp.LpBinary)

Οι συνθήκες του *if* ελέγχουν ότι μία ομάδα δεν θα παίξει με τον εαυτό της και ότι δεν θα παίξουν δύο ομάδες που έχουν ήδη αγωνιστεί μεταξύ τους.

Πλέον, μπορούμε να ορίσουμε την αντικειμενική συνάρτηση του προβλήματος.

**Αντικειμενική συνάρτηση**

# Objective Function

if (w == 8):

    maxMatches = 0

    for i in T:

        if (E[(i)]<3):

            maxMatches += E[(i)]

        else:

            maxMatches += 2

    timetable += \

                (50 / min(10, maxMatches)) \* pulp.lpSum(x[(i, j, d)] for i in T for j in T if i != j and [i,j] not in M for d in D) \

            +   (30 / min(60, maxMatches \* 6)) \* pulp.lpSum(x[(i, j, d)] for i in T for j in T if i != j and [i,j] not in M for d in D  for p in range(1,7) if P[(i, p, d)] and 1) \

            +   (20 / min(600, maxMatches \* 60)) \* pulp.lpSum(x[(i, j, d)] \* P[(i, p, d)] for i in T for j in T if i != j and [i,j] not in M for d in D for p in range(1,7))

elif (numberTeamsDouble >=2):

    timetable += \

                (50 / 10) \* pulp.lpSum(x[(i, j, d)] for i in T for j in T if i != j and [i,j] not in M for d in D) \

            +   (30 / 60) \* pulp.lpSum(x[(i, j, d)] for i in T for j in T if i != j and [i,j] not in M for d in D  for p in range(1,7) if P[(i, p, d)] and 1) \

            +   (10 / 600) \* pulp.lpSum(x[(i, j, d)] \* P[(i, p, d)] for i in T for j in T if i != j and [i,j] not in M for d in D for p in range(1,7)) \

            +   (10 / min(10, 2 \* numberTeamsDouble)) \* pulp.lpSum(x[(i, j, d)] \* (E[(i)] and 1) for i in T for j in T if i != j and [i,j] not in M for d in D)

else:

    timetable += \

                (50 / 8) \* pulp.lpSum(x[(i, j, d)] for i in T for j in T if i != j and [i,j] not in M for d in D) \

            +   (30 / 48) \* pulp.lpSum(x[(i, j, d)] for i in T for j in T if i != j and [i,j] not in M for d in D  for p in range(1,7) if P[(i, p, d)] and 1) \

            +   (20 / 480) \* pulp.lpSum(x[(i, j, d)] \* P[(i, p, d)] for i in T for j in T if i != j and [i,j] not in M for d in D for p in range(1,7))

Στον παραπάνω κώδικα, διακρίνουμε τις δύο περιπτώσεις που αναλύσαμε προηγουμένως, δηλ. όταν δεν υπάρχουν ομάδες που χρωστάνε αγώνες (κώδικας στο *else*) και όταν υπάρχουν ομάδες που χρωστάνε αγώνες (κώδικας στο *elif*).

Ωστόσο, υπάρχει και μία 3η, ειδική περίπτωση: αυτή της τελευταίας αγωνιστικής (8η εβδομάδα) – κώδικας στο *if*. Τότε, η αντικειμενική συνάρτηση μεταβάλλεται ελαφρώς, επειδή το μέγιστο πλήθος αγώνων δεν είναι 4 ή 5, άλλα όσο και οι χρωστούμενοι αγώνες. Αυτό το μέγεθος εκφράζει η μεταβλητή *maxMatches*. Έτσι, οι συντελεστές των όρων της αντικειμενικής συνάρτησης προσαρμόζονται κατάλληλα.

Στη συνέχεια, ορίζουμε τους περιορισμούς του προβλήματος, όπως αυτοί εκφράστηκαν στην *Ενότητα 3*.

**Περιορισμοί**

# Constraints

# 1) A team can play:

#    a) up to once per week if they are not behind in matches played

#    b) up to twice per week if they are behind in matches played

for i in T:

    timetable += pulp.lpSum(x[(i,j,d)] for j in T if i != j and [i,j] not in M for d in D) <= 1 + (E[(i)] and 1)

# 2) A team can only play once vs another team per week

for i in T:

    for j in T:

        if i != j and [i,j] not in M:

                timetable += pulp.lpSum(x[(i,j,d)] for d in D) <= 1

# 3) A team (that is behind in matches played) can not play two consecutive days

for i in T:

    for k in range(len(D)-1):

        timetable += (pulp.lpSum(x[(i,j,D[k])] for j in T if i != j and [i,j] not in M) + pulp.lpSum(x[(i,j,D[k+1])] for j in T if i != j and [i,j] not in M) <= 1)

# 4) Home and away matches are the same

for i in T:

    for j in T:

        if i != j and [i,j] not in M:

            for d in D:

                timetable += x[(i,j,d)] == x[(j,i,d)]

# 5) Max 1 match per day can be played

for d in D:

    timetable += pulp.lpSum(x[(i,j,d)] for i in T for j in T if i != j and [i,j] not in M) <= 2

# 6) A team can play when at least 5 of its players are available

for i in T:

    for d in D:

        timetable += pulp.lpSum(x[(i,j,d)] for j in T if i != j and [i,j] not in M) \* pulp.lpSum(P[(i, p, d)] and 1 for p in range(1,7)) >= 5 \* pulp.lpSum(x[(i,j,d)] for j in T if i != j and [i,j] not in M)

Το μόνο που απομένει πλέον είναι η επίλυση του προβλήματος και η καταμέτρηση του χρόνου που απαιτήθηκε με τη βοήθεια της βιβλιοθήκης *timeit*.

# Solution

tStart = timeit.default\_timer()

timetable.solve(pulp.PULP\_CBC\_CMD(msg=False))

tEnd = timeit.default\_timer()

print(f"\n--------------- WEEK {w} ---------------")

print(f"\nProblem solved in: {(tEnd-tStart):5.3f} seconds")

Ο υπόλοιπος κώδικας αφορά την εκτύπωση του προγράμματος αγώνων της εβδομάδας καθώς και ορισμένων χρήσιμων στατιστικών, με τρόπο ευανάγνωστο και φιλικό προς τον χρήστη του προγράμματος. Επομένως, οι παρακάτω εντολές του προγράμματος δεν σχετίζονται με γραμμικό προγραμματισμό. Ωστόσο, κάποια σημαντικά κομμάτια κώδικα για τη σωστή λειτουργία του προγράμματος και τις επόμενες εβδομάδες είναι τα ακόλουθα.

weeklyMatches = []

for match in x:

    if x[match].varValue == 1:

        weeklyMatches.append(match)

# Add the weekly matches to the list of matches played

for match in weeklyMatches:

    M.append([match[0], match[1]])

Στο συγκεκριμένο κομμάτι κώδικα, αρχικά αποθηκεύονται οι αγώνες της εβδομάδας στη λίστα *weeklyMatches* και στη συνέχεια προστίθενται στο σύνολο Μ των αγώνων που έχουν ήδη διεξαχθεί.

# Update the teams' need for extra matches

for i in T:

    teamWeeklyMatches = sum(1 for match in weeklyMatches if match[0] == i)

    if teamWeeklyMatches == 0:

        E[(i)] += 1

    elif teamWeeklyMatches == 2:

        E[(i)] -= 1

Σε αυτό το τμήμα κώδικα, ανανεώνεται το σύνολο Ε με το πλήθος αγώνων που χρωστάει η κάθε ομάδα. Συγκεκριμένα, αν η ομάδα δεν αγωνίστηκε αυτήν την εβδομάδα, τότε οι αγώνες που χρωστάει θα αυξηθούν κατά 1. Αντίθετα, αν η ομάδα αγωνίστηκε 2 φορές αυτήν την εβδομάδα, τότε οι αγώνες που χρωστάει θα μειωθούν κατά 1. Διαφορετικά (δηλ. αν η ομάδα αγωνίστηκε 1 φορά αυτήν την εβδομάδα), οι αγώνες που χρωστάει θα παραμείνουν σταθεροί.

## ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΠΟ ΤΗΝ ΕΚΤΕΛΕΣΗ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Εκτελώντας τον κώδικα, προκύπτουν τα εξής προγράμματα αγώνων για τις 8 εβδομάδες του πρωταθλήματος:

Εικόνα που περιέχει κείμενο, κατάλογος, έγγραφο, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει κείμενο, κατάλογος, στιγμιότυπο οθόνης, έγγραφο

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, κατάλογος, έγγραφο, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει κείμενο, κατάλογος, έγγραφο, στιγμιότυπο οθόνης

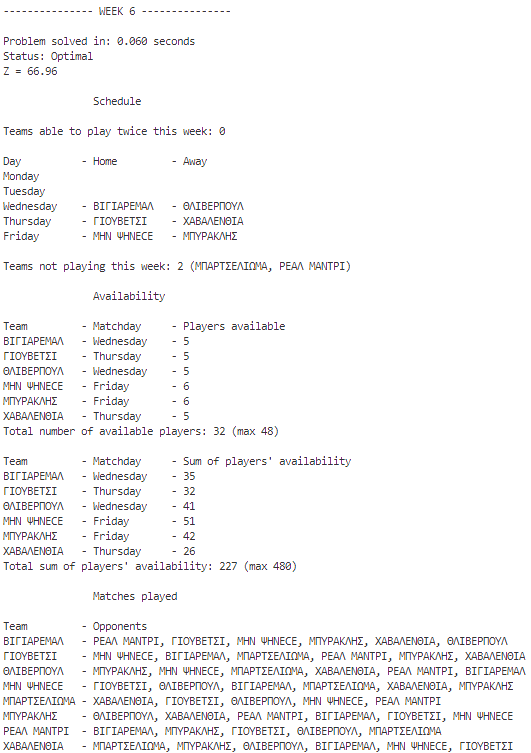
Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, έγγραφο, κατάλογος

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει κείμενο, έγγραφο, στιγμιότυπο οθόνης, κατάλογος

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, έγγραφο, γραμματοσειρά

Περιγραφή που δημιουργήθηκε αυτόματα

Ας αναλύσουμε τα παραπάνω αποτελέσματα, ώστε να επιβεβαιώσουμε την ορθή λειτουργία του προγράμματος.

Την 1η εβδομάδα, το πρόβλημα χρονοπρογραμματισμού αγώνων επιλύθηκε σε μόλις *0.06s*. Η λύση που δίνεται είναι η βέλτιστη (*optimal*) και δίνει στην αντικειμενική συνάρτηση την τιμή *89.42*. Υπενθυμίζεται πως η μέγιστη τιμή της αντικειμενικής συνάρτησης είναι 100. Με άλλα λόγια, το 100 εκφράζει την ιδανική λύση στην οποία όλοι οι στόχοι της αντικειμενικής συνάρτησης ικανοποιούνται στο έπακρο. Στην πράξη όμως, το θεωρητικό αυτό σενάριο είναι πολλές φορές αδύνατον να πραγματοποιηθεί. Για παράδειγμα, στην περίπτωση μας, το 89.42 αποτελεί την καλύτερη εφικτή τιμή. Επομένως, καταλαβαίνουμε πως η λύση που δίνεται από το πρόγραμμα μας προσεγγίζει την ιδανική.

Στη συνέχεια, τυπώνεται το πρόγραμμα αγώνων της εβδομάδας (*Schedule*). Αρχικά, βλέπουμε πως δεν υπάρχουν ομάδες που μπορούν να αγωνιστούν 2 φορές αυτήν την εβδομάδα, κάτι που είναι λογικό, αφού δεν υπάρχουν ομάδες που χρωστάνε αγώνες. Ύστερα, φαίνονται οι αγώνες που θα διεξαχθούν σε κάθε μέρα της εβδομάδας. Τέλος, ενημερωνόμαστε ότι δεν υπάρχουν ομάδες που να μην αγωνίζονται αυτήν την εβδομάδα.

Έπειτα, τυπώνεται η διαθεσιμότητα των παικτών για αυτήν την εβδομάδα, η οποία αποτελείται από 2 παρόμοια τμήματα. Στο 1ο τμήμα, τυπώνεται για κάθε ομάδα, την ημέρα που αγωνίζεται, το πόσοι παίκτες της είναι διαθέσιμοι (*available*), δηλ. το πλήθος των παικτών της που θα αγωνιστούν. Τέλος, τυπώνεται το συνολικό πλήθος παικτών που θα αγωνιστούν αυτήν την εβδομάδα. Υπενθυμίζουμε ότι το θεωρητικό μέγιστο του πλήθους αυτού είναι 48, ενώ βλέπουμε ότι στη βέλτιστη λύση παίρνει την τιμή 43. Στο 2ο τμήμα, τυπώνεται για κάθε ομάδα, την ημέρα που αγωνίζεται, το άθροισμα της προτίμησης των παικτών της (*preference*). Τέλος, τυπώνεται το συνολικό πλήθος της προτίμησης των παικτών που θα αγωνιστούν αυτήν την εβδομάδα. Υπενθυμίζουμε ότι το θεωρητικό μέγιστο του πλήθους αυτού είναι 480, ενώ βλέπουμε ότι στη βέλτιστη λύση παίρνει την τιμή 301. PREFERENCE ΕΙΚΟΝΕΣ