Εργασία Εξαμήνου

Γραμμική και Συνδυαστική Βελτιστοποίηση

Ιωάννης Νικολάου

4ο Έτος – 1072681

Sports Timetabling Problem for English Premier League

Περιεχόμενα

1. Εισαγωγή

1.1 Επισκόπηση

Η English Premier League [1] είναι ένα από τα πιο δημοφιλή και διάσημα πρωταθλήματα στον κόσμο, προσελκύοντας εκατομμύρια φιλάθλους και αποφέροντας δισεκατομμύρια λίρες σε έσοδα κάθε χρόνο. Με 20 ομάδες να αγωνίζονται κατά τη διάρκεια μιας σεζόν που διαρκεί από τον Αύγουστο έως το Μάιο, το πρωτάθλημα αποτελεί μια πολύπλοκη πρόκληση δημιουργίας ενός timetable που απαιτεί προσεκτική εξέταση ενός ευρέος φάσματος παραγόντων.

Η δημιουργία του πλήρους timetable των αγώνων του πρωταθλήματος δεν είναι τυχαία κλήρωση. Είναι το αποτέλεσμα μιας σχολαστικής κι επίπονης διαδικασίας που διαρκεί σχεδόν μισό χρόνο και περιλαμβάνει τον προγραμματισμό 2036 αγώνων στις τέσσερις κορυφαίες κατηγορίες της Αγγλίας. Η ενασχόληση με τη δημιουργία του timetable ξεκινάει στην αρχή της χρονιάς όταν έχουν μαθευτεί οι ημερομηνίες των αγώνων. Όλο το πρόγραμμα δημιουργείται με την εισαγωγή των διεθνών ημερομηνιών από τη FIFA, στη συνέχεια των ευρωπαϊκών διοργανώσεων συλλόγων, στη συνέχεια η Football Association προσθέτει τις ημερομηνίες για τις διοργανώσεις της και οι ημερομηνίες που απομένουν είναι αυτές που μπορούν να διεξαχθούν οι αγώνες της Premier League.

1.2 Περιορισμοί

Η μεθοδολογία που χρησιμοποιείται για την κατασκευή του timetable ονομάζεται sequencing [2] και η οποία έχει να κάνει με το σπάσιμο της σεζόν σε μια σειρά από συστατικά μέρη, τα οποία ονομάζονται sets. Αναλύουμε τη σεζόν σε πέντε sets, τα οποία αντιστρέφονται στο δεύτερο μισό της σεζόν. Από τη μεθοδολογία αυτή προκύπτουν οι «χρυσοί κανόνες» του sequencing.

Οι κανόνες αυτοί που πρέπει να ικανοποιεί το πρόγραμμα είναι οι εξής:

- Σε οποιουσδήποτε πέντε συνεχόμενους αγώνες θα πρέπει να υπάρχει κατανομή τριών εντός έδρας αγώνων, δύο εκτός έδρας αγώνων ή το αντίστροφο (Sequencing Rule).
- Όπου είναι δυνατόν ένας σύλλογος δεν θα έχει περισσότερους από δύο εντός ή εκτός έδρας αγώνες στη σειρά.
- Πρέπει να αποτραπεί οποιοσδήποτε σύλλογος να πρέπει να ξεκινήσει ή να τελειώσει τη σεζόν με δύο εντός ή δύο εκτός έδρας αγώνες.
- Αν μια ομάδα είναι στην έδρα της την ημέρα του Boxing Day (week 17) θα βρίσκεται εκτός έδρας την ημέρα της πρωτοχρονιάς (week 18), ή αντίστροφα.

Υπάρχουν και μερικοί άλλοι κανόνες στους οποίους βασίζεται το timetable:

- Είναι σημαντικό να διατηρείται μια σειρά αγώνων εντός-εκτός έδρας το Σάββατο καθ’ όλη τη διάρκεια της σεζόν, όπου αυτό είναι δυνατόν.
- Η σειρά των αγώνων στον πρώτο γύρο του πρωταθλήματος δεν είναι αναγκαστικό να είναι η ίδια και στο δεύτερο γύρο του πρωταθλήματος.
- Σύλλογοι από την ίδια περιοχή δεν πρέπει να παίζουν στην έδρα τους την ίδια ημέρα επειδή θα προκύψει κυκλοφοριακό πρόβλημα.
- Εξετάζεται εάν υπάρχουν σύλλογοι από την ίδια περιοχή που ταξιδεύουν με τα ίδια τρένα την ίδια μέρα ώστε να αποφευχθούν «σημεία συμφόρησης» στο σιδηροδρομικό και οδικό δίκτυο.

Τέλος μερικοί ακόμα παράγοντες που πρέπει να λαμβάνονται υπόψιν είναι:

- Η διαθεσιμότητα των σταδίων, π.χ. όταν πρέπει να χρησιμοποιηθεί από κάποια άλλη εκδήλωση τη συγκεκριμένη μέρα.
- Τυχόν περιορισμοί τηλεοπτικής κάλυψης ώστε να υπάρχει ισορροπία μεταξύ των ανταγωνιζόμενων τηλεοπτικών καναλιών και μεγιστοποίηση των εσόδων του πρωταθλήματος.
- Η αποφυγή συγκρούσεων με άλλες μεγάλες εκδηλώσεις ώστε να αποφεύγεται κυκλοφοριακή συμφόρηση.

2. Ανασκόπηση υπαρχόντων μεθόδων

2.1 Integer Linear Programming [3]

Ο ακέραιος προγραμματισμός χρησιμοποιείται για την επίλυση προβλημάτων προγραμματισμού αθλητικών αγώνων με τη διατύπωση της εργασίας προγραμματισμού ως μαθηματικό μοντέλο με μεταβλητές, περιορισμούς και μια αντικειμενική συνάρτηση. Οι μεταβλητές αντιπροσωπεύουν για μια συγκεκριμένη εβδομάδα ποιες ομάδες παίζουν αντίπαλες καθώς κι αν παίζει μια ομάδα συνεχόμενα εντός έδρας παιχνίδια, ενώ οι περιορισμοί διασφαλίζουν την τήρηση απαιτήσεων όπως οι περιορισμοί που αναφέρθηκαν παραπάνω. Η αντικειμενική συνάρτηση καθορίζει το μέτρο που πρέπει να βελτιστοποιηθεί, όπως η ελαχιστοποίηση των συνεχόμενων εντός έδρας παιχνιδιών ή η ελαχιστοποίηση των αποστάσεων ταξιδιού. Με την επίλυση του μοντέλου χρησιμοποιώντας τεχνικές ακέραιου προγραμματισμού, μπορεί να προκύψει ένα βέλτιστο πρόγραμμα που ικανοποιεί όλους τους περιορισμούς και επιτυγχάνει τους επιθυμητούς στόχους.

2.2 Simulated annealing [4]

Η προσέγγιση αυτή είναι ένας ευρετικός αλγόριθμος βελτιστοποίησης που χρησιμοποιείται για τον χρονοπρογραμματισμό, ο οποίος περιλαμβάνει την τυχαία δημιουργία ενός χρονοπρογράμματος και στη συνέχεια τη σταδιακή τροποποίησή του για την εύρεση ενός καλύτερου. Ο αλγόριθμος βασίζεται στη φυσική διαδικασία του annealing, όπου ένα υλικό θερμαίνεται και στη συνέχεια ψύχεται αργά για να φτάσει σε μια κατάσταση χαμηλής ενέργειας. Παρομοίως, στο simulated annealing, ένα πρόγραμμα παράγεται τυχαία και στη συνέχεια ο αλγόριθμος το τροποποιεί σταδιακά, ανταλλάσσοντας παιχνίδια, προσαρμόζοντας τις ώρες των παιχνιδιών ή τους χώρους διεξαγωγής και κάνοντας άλλες αλλαγές για τη βελτίωση της συνολικής ποιότητας του προγράμματος. Ο αλγόριθμος χρησιμοποιεί μια προσέγγιση βασισμένη σε πιθανότητες για να αποφασίσει αν θα δεχτεί ή θα απορρίψει αυτές τις τροποποιήσεις, επιτρέποντάς του να εξερευνήσει διαφορετικές περιοχές του χώρου αναζήτησης και να αποφύγει να κολλήσει σε τοπικά βέλτιστα.

2.3 Genetic algorithms [5]

Οι γενετικοί αλγόριθμοι είναι αλγόριθμοι βελτιστοποίησης που χρησιμοποιούν τις αρχές της φυσικής επιλογής και της γενετικής για να βρουν την καλύτερη λύση σε ένα πρόβλημα. Στον αθλητικό προγραμματισμό, οι γενετικοί αλγόριθμοι περιλαμβάνουν τη δημιουργία πολλαπλών χρονοδιαγραμμάτων και στη συνέχεια τη χρήση ενός γενετικού αλγορίθμου για τον συνδυασμό και την εξέλιξή τους σε ένα ενιαίο βέλτιστο χρονοδιάγραμμα. Η διαδικασία περιλαμβάνει την

επιλογή των καταλληλότερων χρονοδιαγραμμάτων και τη χρήση τους για τη δημιουργία νέων χρονοδιαγραμμάτων μέσω μετάλλαξης και διασταύρωσης. Ο αλγόριθμος συνεχίζει να επαναλαμβάνεται, με κάθε γενιά προγραμμάτων να εξελίσσεται και να βελτιώνεται σε σχέση με την προηγούμενη. Οι γενετικοί αλγόριθμοι είναι γνωστοί για την ικανότητά τους να βρίσκουν βέλτιστες λύσεις σε εξαιρετικά πολύπλοκα περιβάλλοντα με πολλαπλούς περιορισμούς και στόχους. Στον αθλητικό προγραμματισμό, χρησιμοποιούνται για τη δημιουργία προγραμμάτων που είναι δίκαια, ισορροπημένα και πληρούν διάφορους περιορισμούς, όπως η ελαχιστοποίηση της απόστασης ταξιδιού, η αποφυγή συγκρούσεων προγραμματισμού και η μεγιστοποίηση των εσόδων.

3. Μοντελοποίηση προβλήματος

3.1 Σύνολα

T: σύνολο με όλες τις ομάδες

Liverpool: σύνολο με όλες τις ομάδες από το Liverpool

Manchester: σύνολο με όλες τις ομάδες από το Manchester

W: σύνολο με τις εβδομάδες του πρωταθλήματος

W2: σύνολο με τις εβδομάδες για τον έλεγχο των συνεχόμενων εντός έδρας αγώνων

W3: σύνολο με τις εβδομάδες για την ικανοποίηση του πρώτου περιορισμού της 1.2

WFH: σύνολο με τις εβδομάδες των αγώνων του πρώτου γύρου

WSH: σύνολο με τις εβδομάδες των αγώνων του δεύτερου γύρου

3.2 Μεταβλητές

Ακολουθείται η προσέγγιση όπου υπάρχει μια οικογένεια μεταβλητών x που ορίζουν ποιες ομάδες παίζουν αντίπαλες την εβδομάδα w, όπου το i αναπαριστά την ομάδα που παίζει εντός έδρας και το j την ομάδα που παίζει εκτός έδρας, αν βρεθεί μια εφικτή λύση. Για τον σκοπό της ελαχιστοποίησης των συνεχόμενων εντός έδρας αγώνων, εισήχθη η μεταβλητή y που φανερώνει σε ποια εβδομάδα w ακολουθούν δύο συνεχόμενοι εντός έδρας αγώνες για την ομάδα i.

$$x_{i,j,w} \begin{cases} 1 \ \alpha\nu \ \eta \ o\mu\acute{\alpha}\delta\alpha \ i \ \pi\alpha\acute{\iota}\zeta\epsilon\iota \ \omega\varsigma \ \epsilon\nu\tau\acute{o}\varsigma \ \acute{\epsilon}\delta\rho\alpha\varsigma \ \epsilon\nu\alpha\nu\tau\acute{\iota}\omega\nu \ \tau\eta\varsigma \ j \ o\mu\acute{\alpha}\delta\alpha\varsigma \ \tau\eta\nu \ \epsilon\beta\delta o\mu\acute{\alpha}\delta\alpha \ w \\ 0 \ \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}$$

$$y_{i,w} \begin{cases} 1 \ \alpha\nu \ \eta \ o\mu\acute{\alpha}\delta\alpha \ i \ \pi\alpha\acute{\iota}\zeta\epsilon\iota \ \delta\acute{\upsilon}o \ \sigma\upsilon\nu\epsilon\chi\acute{o}\mu\epsilon\nu\alpha \ \epsilon\nu\tau\acute{o}\varsigma \ \acute{\epsilon}\delta\rho\alpha\varsigma \ \pi\alpha\iota\chi\nu\acute{\iota}\delta\iota\alpha \ \tau\eta\nu \ \epsilon\beta\delta o\mu\acute{\alpha}\delta\alpha \ w \ \kappa\alpha\iota \ w+1 \\ 0 \ \alpha\lambda\lambda\iota\acute{\omega}\varsigma \end{cases}$$

3.3 Περιορισμοί

Περιορισμοί εφικτού προγράμματος

- Μία ομάδα παίζει ακριβώς έναν αγώνα κάθε εβδομάδα

$$\sum_{a\in T, i\neq a} x_{i,a,w} + \sum_{h\in T, i\neq h} x_{h,i,w} = 1, \forall i \in T, \forall w \in W$$

Δηλαδή για κάθε εβδομάδα w και για κάθε ομάδα i πρέπει το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδας με κάθε ομάδα a την εβδομάδα w συν το άθροισμα όλων των αγώνων της i ως εκτός έδρας ομάδας με κάθε ομάδα h την εβδομάδα w να ισούται με ένα, κι άρα να παίξει σίγουρα έναν αγώνα εκείνη την εβδομάδα.

- Μια ομάδα παίζει ακριβώς έναν αγώνα στον πρώτο γύρο με κάθε άλλη ομάδα

$$\sum_{w\in WFH} x_{i,j,w} + \sum_{w\in WFH} x_{j,i,w} = 1, \forall j \in T, \forall i \in T$$

Δηλαδή για κάθε ομάδα i και για κάθε ομάδα j πρέπει το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδα ενάντια στην ομάδα j ως εκτός έδρας για κάθε εβδομάδα στον πρώτο γύρο του πρωταθλήματος συν το άθροισμα όλων των αγώνων της j ως εντός έδρας ομάδα ενάντια στην ομάδα i ως εκτός έδρας ομάδα για κάθε εβδομάδα στον πρώτο γύρο του πρωταθλήματος να ισούται με ένα, κι άρα κάθε ομάδα i να παίξει σίγουρα έναν αγώνα στον πρώτο γύρο του πρωταθλήματος με όλες τις υπόλοιπες ομάδες είτε εντός είτε εκτός έδρας.

- Μια ομάδα παίζει ακριβώς έναν αγώνα στο δεύτερο γύρο με κάθε άλλη ομάδα

$$\sum_{w\in WSH} x_{i,j,w} + \sum_{w\in WSH} x_{j,i,w} = 1, \forall j \in T, \forall i \in T$$

Δηλαδή για κάθε ομάδα i και για κάθε ομάδα j πρέπει το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδα ενάντια στην ομάδα j ως εκτός έδρας για κάθε εβδομάδα στο δεύτερο γύρο του πρωταθλήματος συν το άθροισμα όλων των αγώνων της j ως εντός έδρας ομάδα ενάντια στην ομάδα i ως εκτός έδρας ομάδα για κάθε εβδομάδα στο δεύτερο γύρο του πρωταθλήματος να ισούται με ένα, κι άρα κάθε ομάδα i να παίξει σίγουρα έναν αγώνα στο δεύτερο γύρο του πρωταθλήματος με όλες τις υπόλοιπες ομάδες είτε εντός είτε εκτός έδρας.

Προσοχή από τους δύο τελευταίες περιορισμούς δεν είναι εγγυημένο ότι μια ομάδα παρόλο που θα παίξει με όλες τις ομάδες και στον πρώτο και στο δεύτερο γύρω ακριβώς μια φορά, ότι δεν θα παίξει την ίδια εβδομάδα δύο αγώνες. Γι' αυτό το λόγο είναι απαραίτητος ο πρώτος περιορισμός!

Π.χ. Για ένα παράδειγμα με 6 ομάδες, αν δεν είχαμε τον πρώτο περιορισμό τότε όπως φαίνεται στην παρακάτω εικόνα δεν θα λάμβανε χώρα ο ίδιος αριθμός αγώνων σε κάθε εβδομάδα.

```
Week: 1  Liverpool vs. NottinghamForest
Week: 1  ManchesterCity vs. Everton
Week: 1  NewcastleUnited vs. ManchesterUnited
Week: 1  NottinghamForest vs. NewcastleUnited

Week: 2  Everton vs. ManchesterUnited
Week: 2  ManchesterUnited vs. NottinghamForest

Week: 3  Liverpool vs. ManchesterUnited
Week: 3  ManchesterCity vs. NewcastleUnited
Week: 3  NewcastleUnited vs. Liverpool
Week: 3  NottinghamForest vs. Everton

Week: 4  Everton vs. Liverpool
Week: 4  ManchesterUnited vs. ManchesterCity

Week: 5  Liverpool vs. ManchesterCity
Week: 5  ManchesterCity vs. NottinghamForest
Week: 5  NewcastleUnited vs. Everton
```

- Αν μια ομάδα i έπαιξε εντός έδρας εναντίων της ομάδας j στον πρώτο γύρο τότε πρέπει να παίξει εκτός έδρας εναντίων της ομάδας j στο δεύτερο γύρο

$$\sum_{w \in WFH} x_{i,j,w} = \sum_{w \in WSH} x_{j,i,w}, \forall j \in T, \forall i \in T$$

Δηλαδή για κάθε ομάδα i και για κάθε ομάδα j πρέπει το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδα ενάντια στην ομάδα j ως εκτός έδρας για κάθε εβδομάδα στον πρώτο γύρο του πρωταθλήματος να ισούται με το άθροισμα όλων των αγώνων της j ως εντός έδρας ομάδα ενάντια στην ομάδα i ως εκτός έδρας ομάδα για κάθε εβδομάδα στο δεύτερο γύρο του πρωταθλήματος, κι άρα για κάθε ομάδα i εφόσον έπαιξε εντός έδρας στον πρώτο γύρο να παίξει σίγουρα εκτός έδρας στο δεύτερο γύρο.

- Αν μια ομάδα i έπαιξε εκτός έδρας εναντίων της ομάδας j στον πρώτο γύρο τότε πρέπει να παίξει εντός έδρας εναντίων της ομάδας j στο δεύτερο γύρο

$$\sum_{w \in WFH} x_{j,i,w} = \sum_{w \in WSH} x_{i,j,w}, \forall j \in T, \forall i \in T$$

Δηλαδή για κάθε ομάδα i και για κάθε ομάδα j πρέπει το άθροισμα όλων των αγώνων της j ως εντός έδρας ομάδα ενάντια στην ομάδα i ως εκτός έδρας για κάθε εβδομάδα στον πρώτο γύρο του πρωταθλήματος να ισούται με το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδα ενάντια στην ομάδα j ως εκτός έδρας ομάδα για κάθε εβδομάδα στο δεύτερο γύρο του πρωταθλήματος, κι άρα για κάθε ομάδα i εφόσον έπαιξε εκτός έδρας στον πρώτο γύρο να παίξει σίγουρα εντός έδρας στο δεύτερο γύρο.

Προσοχή αν δεν υπήρχαν οι δύο τελευταίοι περιορισμοί τότε δύο ομάδες παρόλο που είναι εγγυημένο ότι θα παίξουν δύο φορές μεταξύ τους, μία στον πρώτο και μία στο δεύτερο γύρο δεν είναι εγγυημένο ότι θα παίξει η ομάδα j ως εντός έδρας εναντίων στην ομάδα i στο δεύτερο γύρο του πρωταθλήματος εφόσον έχει παίξει η ομάδα i ως εντός έδρας στον με την ομάδα j ως εκτός έδρας στον πρώτο γύρο του πρωταθλήματος. Γι' αυτό το λόγο είναι απαραίτητοι οι δύο τελευταίοι περιορισμοί.

Π.χ. Για ένα παράδειγμα με 6 ομάδες, αν είχαν αφαιρεθεί οι δύο τελευταίοι περιορισμοί τότε όπως φαίνεται στην παρακάτω εικόνα η ομάδα Manchester United θα έπαιζε δύο φορές ως εντός έδρας εναντίων της ομάδας Liverpool και στους δύο γύρους του πρωταθλήματος.

```
Week: 4  Everton vs. ManchesterCity
Week: 4  ManchesterUnited vs. Liverpool
Week: 4  NewcastleUnited vs. NottinghamForest

Week: 5  Liverpool vs. NewcastleUnited
Week: 5  ManchesterCity vs. ManchesterUnited
Week: 5  NottinghamForest vs. Everton

Week: 6  Everton vs. NottinghamForest
Week: 6  ManchesterUnited vs. Liverpool
Week: 6  NewcastleUnited vs. ManchesterCity
```

Επιπλέον περιορισμοί προγράμματος

- Τα πρώτα δύο παιχνίδια για κάθε ομάδα δεν πρέπει να είναι συνεχόμενα εντός ή εκτός έδρας

$$\sum_{j \in T, i \neq j} x_{i,j,w_1} + \sum_{j \in T, i \neq j} x_{i,j,w_2} = 1, \forall i \in T$$

Δηλαδή για κάθε ομάδα i το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα 1 (πρώτο παιχνίδι) συν το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα 2 (δεύτερο παιχνίδι) να ισούται με ένα, κι άρα κάθε ομάδα να παίξει σίγουρα έναν εντός έδρας κι έναν εκτός έδρας αγώνα στο διάστημα των 2 πρώτων εβδομάδων (στα 2 πρώτα παιχνίδια).

- Τα τελευταία δύο παιχνίδια για κάθε ομάδα δεν πρέπει να είναι συνεχόμενα εντός ή εκτός έδρας

$$\sum_{j \in T, i \neq j} x_{i,j,w_{-1}} + \sum_{j \in T, i \neq j} x_{i,j,w_{-2}} = 1, \forall i \in T$$

Δηλαδή για κάθε ομάδα i πρέπει το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδας με κάθε ομάδα j την τελευταία εβδομάδα (τελευταίο παιχνίδι) συν το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδας με κάθε ομάδα j την προτελευταία εβδομάδα (προτελευταίο παιχνίδι) να ισούται με ένα, κι άρα κάθε ομάδα να παίξει

σίγουρα έναν εντός έδρας κι έναν εκτός έδρας αγώνα στο διάστημα των 2 τελευταίων εβδομάδων (στα 2 τελευταία παιχνίδια).

- Αν μια ομάδα παίζει εντός κατά τη διάρκεια του boxing day (εβδομάδα 17) θα παίξει εκτός έδρας κατά τη διάρκεια της πρωτοχρονιάς (εβδομάδα 18), ή αντίστροφα.

$$\sum_{j \in T, i \neq j} x_{i,j,w_{17}} + \sum_{j \in T, i \neq j} x_{i,j,w_{18}} = 1, \forall i \in T$$

Δηλαδή για κάθε ομάδα i πρέπει το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα 17 (δέκατο-έβδομο παιχνίδι) συν το άθροισμα όλων των αγώνων της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα 18 (δέκατο-όγδοο παιχνίδι) να ισούται με ένα, κι άρα κάθε ομάδα να παίξει σίγουρα έναν εντός έδρας κι έναν εκτός έδρας αγώνα στο διάστημα των 2 αυτών εβδομάδων.

- Ομάδες από την ίδια πόλη δεν μπορούν να παίξουν και οι δύο εντός έδρας την ίδια εβδομάδα

$$\sum_{h \in Liverpool} \sum_{a \in T, a \neq h} x_{h,a,w} = 1, \forall w \in W$$

Δηλαδή για κάθε εβδομάδα w πρέπει το άθροισμα όλων των πιθανών αγώνων για τις 2 ομάδες που βρίσκονται στην πόλη του Liverpool ως εντός έδρας εναντίων όλων των υπόλοιπων ομάδων ως εκτός έδρας να ισούται με 1, κι άρα για κάθε εβδομάδα η μία ομάδα από το Liverpool να παίζει εντός έδρας κι άλλη εκτός έδρας.

$$\sum_{h \in Manchester} \sum_{a \in T, a \neq h} x_{h,a,w} = 1, \forall w \in W$$

Δηλαδή για κάθε εβδομάδα w πρέπει το άθροισμα όλων των πιθανών αγώνων για τις 2 ομάδες που βρίσκονται στην πόλη του Manchester ως εντός έδρας εναντίων όλων των υπόλοιπων ομάδων ως εκτός έδρας να ισούται με 1, κι άρα για κάθε εβδομάδα η μία ομάδα από το Manchester να παίζει εντός έδρας κι άλλη εκτός έδρας.

- Σε οποιουσδήποτε πέντε συνεχόμενους αγώνες θα πρέπει να υπάρχει κατανομή τριών εντός έδρας αγώνων, δύο εκτός έδρας αγώνων ή το αντίστροφο (Sequencing Rule).

$$\sum_{j \in T, i \neq j} \left(x_{i,j,w} + x_{i,j,w+1} + x_{i,j,w+2} + x_{i,j,w+3} + x_{i,j,w+4} \right) \leq 3, \forall w \in W3, \forall i \in T$$

Δηλαδή για κάθε ομάδα i και για κάθε εβδομάδα w από το σύνολο W3 πρέπει όλοι οι πιθανοί αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα w συν όλους τους πιθανούς αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα w + 1 συν όλους τους πιθανούς αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την

εβδομάδα w + 2 συν όλους τους πιθανούς αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα w + 3 συν όλους τους πιθανούς αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα w + 4 να είναι μικρότεροι ίσοι του τρία, κι άρα κάθε ομάδα σε οποιουσδήποτε πέντε αγώνες να παίζει το πολύ τρεις αγώνες στην έδρα της.

$$\sum_{j\in T, i\neq j} \left(x_{i,j,w} + x_{i,j,w+1} + x_{i,j,w+2} + x_{i,j,w+3} + x_{i,j,w+4}\right) \geq 2, \forall w \in W3, \forall i \in T$$

Δηλαδή για κάθε ομάδα i και για κάθε εβδομάδα w από το σύνολο W3 πρέπει όλοι οι πιθανοί αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα w συν όλους τους πιθανούς αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα w + 1 συν όλους τους πιθανούς αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα w + 2 συν όλους τους πιθανούς αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα w + 3 συν όλους τους πιθανούς αγώνες της i ως εντός έδρας ομάδας με κάθε ομάδα j την εβδομάδα w + 4 να είναι μεγαλύτεροι ίσοι του δύο, κι άρα κάθε ομάδα σε οποιουσδήποτε πέντε αγώνες να παίζει το λιγότερο δύο αγώνες στην έδρα της.

Από τα δύο προηγούμενα αθροίσματα μπορεί να ικανοποιηθεί ο περιορισμός για μία ομάδα σε οποιουσδήποτε πέντε αγώνες να υπάρχει κατανομή τριών εντός έδρας αγώνων, δύο εκτός έδρας αγώνων ή το αντίστροφο, αφού κάθε ομάδα θα παίζει το ελάχιστο δύο και το μέγιστο τρεις αγώνες εντός έδρας σε οποιουσδήποτε πέντε αγώνες.

- Όπου είναι δυνατόν ένας σύλλογος δεν θα έχει περισσότερους από δύο εντός ή εκτός έδρας αγώνες στη σειρά.

$$\left(\sum_{j\in T, i\neq j} x_{i,j,w} + x_{i,j,w+1}\right) - y_{i,w} \leq 1, \forall w \in W2, \forall i \in T$$

Δηλαδή για κάθε ομάδα i και για κάθε εβδομάδα w πρέπει το άθροισμα όλων των πιθανών αγώνων της i ως εντός έδρας εναντίων όλων των υπόλοιπων ομάδων ως εκτός έδρας την εβδομάδα w συν όλους τους πιθανούς αγώνες της i ως εντός έδρας εναντίων όλων των υπόλοιπων ομάδων ως εκτός έδρας την εβδομάδα w + 1 μείον τη μεταβλητή $y_{w,i}$ να είναι μικρότερο ίσο με 1, κι άρα σε οποιουσδήποτε δύο συνεχόμενους αγώνες να αποτραπεί να είναι συνεχόμενοι εντός έδρας αγώνες.

Δεν χρειάζεται να γραφεί κανόνας για να αποτρέψει συνεχόμενους εκτός έδρας αγώνες αφού όταν υπάρχουν συνεχόμενοι εντός έδρας αγώνες θα υπάρξουν αναγκαστικά και συνεχόμενοι εκτός έδρας αγώνες. Άρα ο παραπάνω περιορισμός χρησιμοποιείται για να αποτρέψει και τις δύο περιπτώσεις.

3.4 Αντικειμενική συνάρτηση

Η αντικειμενική συνάρτηση στοχεύει να ελαχιστοποιήσει τον αριθμό των 2 συνεχόμενων παιχνιδιών εντός κι εκτός έδρας.

$$\min \sum_{i \in T} \sum_{w \in W2} y_{i,w}$$

4. Υλοποίηση με κώδικα

Για την υλοποίηση της παραπάνω μοντελοποίησης του προβλήματος με κώδικα χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python [6] με τη βοήθεια της βιβλιοθήκης pulp [7].

```
import pulp
```

4.1 Σύνολα

Σε αντιστοιχία με την παράγραφο 3.1 τα σύνολα που περιεγράφηκαν αναπαρίστανται στην Python με την μορφή λιστών.

```
T = [
    'Arsenal',
    'AstonVilla',
    'Bournemouth',
    'Brentford',
    'Brighton',
    'Chelsea',
    'CrystalPalace',
    'Everton',
    'Fulham',
    'LeedsUnited',
    'LeicesterCity',
    'Liverpool',
    'ManchesterCity',
    'ManchesterUnited',
    'NewcastleUnited',
    'NottinghamForest',
    'Southampton',
    'TottenhamHotspur',
    'WestHamUnited',
    'WolverhamptonWanderers'
    ]

Liverpool = ['Liverpool', 'Everton']
Manchester = ['ManchesterCity', 'ManchesterUnited']

W = range(1, 2*len(T) - 1)
W2 = range(1, 2*len(T) - 2)
W3 = range(1, 2*len(T) - 5)
WFH = range(1, len(T))
WSH = range(len(T), 2*len(T) - 1)
```

Έχουμε συνολικά 20 ομάδες άρα ο αριθμός των αγώνων που θα διεξαχθούν είναι 38.

4.2 Μεταβλητές

Σε αντιστοιχία με την παράγραφο 3.2 οι μεταβλητές που περιεγράφηκαν αναπαρίστανται στην Python με έναν ειδικό τύπο με τη βοήθεια της βιβλιοθήκης pulp ως λεξικά δηλώνοντας την ιεραρχία τους, δηλαδή τα κλειδιά του λεξικού και τη σειρά που ακολουθεί η ιεραρχία. Με το όρισμα cat ορίζεται η κατηγορία στην οποία βρίσκονται η μεταβλητές και στην συγκεκριμένη περίπτωση είναι Binary, δηλαδή 0 ή 1.

```
x = pulp.LpVariable.dicts('x', (T, T, W), cat=pulp.LpBinary)
y = pulp.LpVariable.dicts('y', (T, W2), cat=pulp.LpBinary)
```

Από τις δηλώσεις των μεταβλητών φαίνεται για τη μεταβλητή x ότι στην ιεραρχία πρώτα υπάρχει ως κλειδί η ομάδα, έπειτα ως δεύτερο κλειδί πάλι η ομάδα και ως τελευταίο κλειδί η εβδομάδα. Για τη μεταβλητή y πρώτα υπάρχει στην ιεραρχία ως κλειδί η ομάδα κι έπειτα η εβδομάδα.

```
# Team cannot play itself
for w in W:
    for h in T:
        for a in T:
            if h == a:
                del x[h][a][w]
```

Το κομμάτι αυτό του κώδικα προστέθηκε ώστε να μην υπάρχουν μεταβλητές x όπου μια ομάδα παίζει με τον εαυτό της σε οποιαδήποτε εβδομάδα.

```
schedule = pulp.LpProblem("Schedule", pulp.LpMinimize)
```

Τελευταίο μέρος αυτής της παραγράφου είναι να ορίσουμε σε μια μεταβλητή το πρόβλημα που πρέπει να επιλυθεί δίνοντας του ένα όνομα “Schedule” και τι είδος είναι (minimization problem).

4.3 Περιορισμοί

Οι περιορισμοί στην python θα παρατεθούν ακριβώς με τη σειρά που έχουν γραφεί και στην παράγραφο 3.3. Αντιστοιχίζοντας τα σύμβολα των μαθηματικών εξισώσεων με τις εντολές της Python, όπου υπάρχει εξωτερικό for loop αντιστοιχεί στο σύμβολο ∀ ενώ η

εντολή pulp.lpSum αντιστοιχεί στο σύμβολο $\sum$ κι οι μεταβλητές που έχει από κάτω το σύμβολο αυτό αντιστοιχούν στο for loop μέσα στην εντολή pulp.lpSum.

Περιορισμοί εφικτού προγράμματος

```
# Each team plays exactly one match each week
for w in W:

    for i in T:

        schedule += pulp.lpSum(x[i][a][w] for a in T if a != i) + \
            pulp.lpSum(x[h][i][w] for h in T if h != i) == 1
```

```
# Teams play each other twice during the season, with one home match and one away match
li = []
for i in T:
    li.append(i)
    for j in T:
        if j != i and j not in li:
            # team i plays one match in the first half against team j
            schedule += pulp.lpSum(x[i][j][w] for w in WFH) + \
                pulp.lpSum(x[j][i][w] for w in WFH) == 1
            # team i plays one match in the second half against team j
            schedule += pulp.lpSum(x[i][j][w] for w in WSH) + \
                pulp.lpSum(x[j][i][w] for w in WSH) == 1
            # if team i played at home against team j in the first half,
            # they will play away against team j in the second half, and vice versa
            schedule += pulp.lpSum(x[i][j][w] for w in WFH) == pulp.lpSum(
                x[j][i][w] for w in WSH)
            schedule += pulp.lpSum(x[j][i][w] for w in WFH) == pulp.lpSum(
                x[i][j][w] for w in WSH)
```

Εδώ χρησιμοποιείται μια λίστα ώστε αν για μία ομάδα γραφούν οι περιορισμοί να την προσθέσουμε στη λίστα αυτή κι όταν την συναντήσουμε ως αντίπαλη για άλλη ομάδα να μη γράψουμε ξανά τους περιορισμούς διότι είναι ακριβώς οι ίδιοι.

Επιπλέον περιορισμοί προγράμματος

```
# First two and last two games cannot be consecutive away or home matches
for i in T:
    schedule += pulp.lpSum(x[i][j][W[0]] for j in T if i != j) + \
        pulp.lpSum(x[i][j][W[1]] for j in T if i != j) == 1
    schedule += pulp.lpSum(x[i][j][W[-1]] for j in T if i != j) + \
        pulp.lpSum(x[i][j][W[-2]] for j in T if i != j) == 1
```

```
# If a team is home during boxing day (week 17) it will be away during
# new year's day (week 18)
for i in T:
    schedule += pulp.lpSum(x[i][j][W[16]] for j in T if i != j) + \
        pulp.lpSum(x[i][j][W[17]] for j in T if i != j) == 1
```

```
# Teams from the same city can't play both home in the same week
for w in W:
    schedule += pulp.lpSum(x[h][a][w] for a in T for h in Liverpool if h != a) == len(Liverpool) // 2

for w in W:
    schedule += pulp.lpSum(x[h][a][w] for a in T for h in Manchester if h != a) == len(Manchester) // 2
```

Εδώ το μήκος των πινάκων είναι 2 οπότε ο περιορισμός προκύπτει ίδιος όπως στην παράγραφο 3.3

```
# Sequencing rule
for i in T:
    for w in W3:
        schedule += pulp.lpSum(x[i][j][w] + x[i][j][w + 1] + x[i][j][w + 2] + x[i][j][w + 3] + x[i][j][w + 4]
                               for j in T if i != j) <= 3
        schedule += pulp.lpSum(x[i][j][w] + x[i][j][w + 1] + x[i][j][w + 2] + x[i][j][w + 3] + x[i][j][w + 4]
                               for j in T if i != j) >= 2
```

```
# Minimize the number of consecutive home and away games
for i in T:
    for w in W2:
        schedule += pulp.lpSum(x[i][j][w] + x[i][j][w + 1] for j in T if i != j) - y[i][w] <= 1
```

4.4 Αντικειμενική συνάρτηση

Η αντικειμενική συνάρτηση συνηθίζεται να προηγείται από τους περιορισμούς και γράφεται ακριβώς κάτω από τη δημιουργία της μεταβλητής για τον ορισμό του προβλήματος.

```
schedule += pulp.lpSum(y[i][w] for w in W2 for i in T)
```

Όλοι οι περιορισμοί και η αντικειμενική συνάρτηση προστίθενται στη μεταβλητή schedule κι έτσι για να βρεθεί βέλτιστη λύση πρέπει η τιμή της μεταβλητής αυτής να ελαχιστοποιηθεί. Η τιμή της μεταβλητής αυτής θα ελαχιστοποιηθεί όπως έχει αναφερθεί και στην παράγραφο 3.4 με σκοπό την ελαχιστοποίηση του αριθμού των δύο συνεχόμενων εντός κι εκτός έδρας αγώνων για όλες τις ομάδες.

5. Αποτελέσματα

Το πρόβλημα αυτό είναι αρκετά πολύπλοκο με πολλούς περιορισμούς και στάλθηκε για επίλυση στον NEOS SERVER [8]. Αφού αποσταλεί το πρόβλημα στον server ο μέγιστος χρόνος που μπορεί να διατεθεί προς την επίλυσή του είναι 8 ώρες. Δυστυχώς για τον αρχικό αριθμό ομάδων (20) που έχει το αγγλικό πρωτάθλημα και λόγων των πολλών περιορισμών και μεταβλητών που προέκυψαν ο server δεν κατάφερε να δώσει κάποια βέλτιστη λύση. Επίσης δεν αποδόθηκε λύση για το ίδιο πρόβλημα με αριθμό ομάδων (8, 10, 12, 14, 16, 18).

Δοκιμάστηκε η επίλυση του προβλήματος με αριθμό ομάδων 6 και χωρίς τον τρίτο επιπλέον περιορισμό προγράμματος στην παράγραφο 3.3 σε έναν υπολογιστή με επεξεργαστή Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz και μνήμη 8 GB. Το αποτέλεσμα που προέκυψε από την επίλυση του προβλήματος φαίνεται στην παρακάτω φωτογραφία.

```
Objective value:                4.00000000
Enumerated nodes:               42726
Total iterations:               2581706
Time (CPU seconds):             168.10
Time (Wallclock seconds):       168.10

Option for printingOptions changed from normal to all
Total time (CPU seconds):       168.12   (Wallclock seconds):       168.12

Week: 1  LeicesterCity vs. NewcastleUnited
Week: 1  Liverpool vs. ManchesterUnited
Week: 1  ManchesterCity vs. Everton

Week: 2  Everton vs. Liverpool
Week: 2  ManchesterUnited vs. LeicesterCity
Week: 2  NewcastleUnited vs. ManchesterCity

Week: 3  LeicesterCity vs. ManchesterCity
Week: 3  Liverpool vs. NewcastleUnited
Week: 3  ManchesterUnited vs. Everton

Week: 4  Everton vs. LeicesterCity
Week: 4  ManchesterCity vs. Liverpool
Week: 4  NewcastleUnited vs. ManchesterUnited

Week: 5  Liverpool vs. LeicesterCity
Week: 5  ManchesterUnited vs. ManchesterCity
Week: 5  NewcastleUnited vs. Everton

Week: 6  Everton vs. NewcastleUnited
Week: 6  LeicesterCity vs. Liverpool
Week: 6  ManchesterCity vs. ManchesterUnited

Week: 7  LeicesterCity vs. Everton
Week: 7  Liverpool vs. ManchesterCity
Week: 7  ManchesterUnited vs. NewcastleUnited

Week: 8  Everton vs. ManchesterUnited
Week: 8  ManchesterCity vs. LeicesterCity
Week: 8  NewcastleUnited vs. Liverpool

Week: 9  LeicesterCity vs. ManchesterUnited
Week: 9  Liverpool vs. Everton
Week: 9  ManchesterCity vs. NewcastleUnited

Week: 10  Everton vs. ManchesterCity
Week: 10  ManchesterUnited vs. Liverpool
Week: 10  NewcastleUnited vs. LeicesterCity
```

Η τιμή της αντικειμενική συνάρτησης δηλώνει πόσα 2 συνεχόμενα εντός έδρας παιχνίδια προέκυψαν για όλες τις ομάδες. Από το αποτέλεσμα φαίνεται ότι αυτό προέκυψε για την Leicester City τις εβδομάδες 6-7, για την Manchester United τις εβδομάδες 2-3, για την Manchester City τις εβδομάδες 8-9 και για την Newcastle United τις εβδομάδες 4-5. Άρα σύνολο 4 ομάδες θα παίξουν 2 συνεχόμενα παιχνίδια εντός έδρας 1 φορά η κάθε μία ομάδα.

	1	2	3	4	5	6	7	8	9	10
LeicesterCity	NewcastleUnited	ManchesterUnited	ManchesterCity	Everton	Liverpool	Liverpool	Everton	ManchesterCity	ManchesterUnited	NewcastleUnited
NewcastleUnited	LeicesterCity	ManchesterCity	Liverpool	ManchesterUnited	Everton	Everton	ManchesterUnited	Liverpool	ManchesterCity	LeicesterCity
Liverpool	ManchesterUnited	Everton	NewcastleUnited	ManchesterCity	LeicesterCity	LeicesterCity	ManchesterCity	NewcastleUnited	Everton	ManchesterUnited
ManchesterUnited	Liverpool	LeicesterCity	Everton	NewcastleUnited	ManchesterCity	ManchesterCity	NewcastleUnited	Everton	LeicesterCity	Liverpool
ManchesterCity	Everton	NewcastleUnited	LeicesterCity	Liverpool	ManchesterUnited	ManchesterUnited	Liverpool	LeicesterCity	NewcastleUnited	Everton
Everton	ManchesterCity	Liverpool	ManchesterUnited	LeicesterCity	NewcastleUnited	NewcastleUnited	LeicesterCity	ManchesterUnited	Liverpool	ManchesterCity

Με τη βοήθεια του κώδικα στο αρχείο exportSolution.py δημιουργήθηκε το παραπάνω excel αρχείο όπου βοηθάει στην καλύτερη οπτική παρατήρηση του προγράμματος. Στην πρώτη γραμμή φαίνονται οι εβδομάδες του πρωταθλήματος και στην πρώτη στήλη οι ομάδες του πρωταθλήματος. Με κίτρινο έχουν χρωματιστεί τα κελιά όπου η ομάδα της γραμμής παίζει εντός έδρας εναντίων της ομάδας που αναγράφεται στο συγκεκριμένο κελί τη συγκεκριμένη εβδομάδα. Με άσπρο έχουν χρωματιστεί τα κελιά όπου η ομάδα της γραμμής παίζει εκτός έδρας εναντίων της ομάδας που αναγράφεται στο συγκεκριμένο κελί τη συγκεκριμένη εβδομάδα.

6. Έλεγχος Αποτελεσμάτων

Δημιουργήθηκε επίσης ένα πρόγραμμα σε Python το οποίο επιβεβαιώνει ότι η λύση που προέκυψε από τον solver ικανοποιεί όλους τους περιορισμούς.

Η προηγούμενη λύση αποθηκεύτηκε σε ένα αρχείο και έχει την εξής μορφή:

```
x_Everton_LeicesterCity_1 0
x_Everton_Liverpool_1 0
x_Everton_ManchesterCity_1 0
x_Everton_ManchesterUnited_1 0
x_Everton_NewcastleUnited_1 0
x_LeicesterCity_Everton_1 0
x_LeicesterCity_Liverpool_1 0
x_LeicesterCity_ManchesterCity_1 0
x_LeicesterCity_ManchesterUnited_1 0
x_LeicesterCity_NewcastleUnited_1 1
x_Liverpool_Everton_1 0
x_Liverpool_LeicesterCity_1 0
x_Liverpool_ManchesterCity_1 0
x_Liverpool_ManchesterUnited_1 1
x_Liverpool_NewcastleUnited_1 0
x_ManchesterCity_Everton_1 1
x_ManchesterCity_LeicesterCity_1 0
x_ManchesterCity_Liverpool_1 0
x_ManchesterCity_ManchesterUnited_1 0
x_ManchesterCity_NewcastleUnited_1 0
x_ManchesterUnited_Everton_1 0
x_ManchesterUnited_LeicesterCity_1 0
x_ManchesterUnited_Liverpool_1 0
x_ManchesterUnited_ManchesterCity_1 0
x_ManchesterUnited_NewcastleUnited_1 0
x_NewcastleUnited_Everton_1 0
x_NewcastleUnited_LeicesterCity_1 0
x_NewcastleUnited_Liverpool_1 0
x_NewcastleUnited_ManchesterCity_1 0
x_NewcastleUnited_ManchesterUnited_1 0
x_Everton_LeicesterCity_2 0
x_Everton_Liverpool_2 1
x_Everton_ManchesterCity_2 0
x_Everton_ManchesterUnited_2 0
x_Everton_NewcastleUnited_2 0
x_LeicesterCity_Everton_2 0
x_LeicesterCity_Liverpool_2 0
x_LeicesterCity_ManchesterCity_2 0
x_LeicesterCity_ManchesterUnited_2 0
x_LeicesterCity_NewcastleUnited_2 0
x_Liverpool_Everton_2 0
x_Liverpool_LeicesterCity_2 0
x_Liverpool_ManchesterCity_2 0
x_Liverpool_ManchesterUnited_2 0
```

Αποθηκεύτηκε κάθε μεταβλητή απόφασης με την αντίστοιχη τιμή της που δηλώνει τον αγώνα τη συγκεκριμένη εβδομάδα και ποια ομάδα παίζει εντός και ποια εκτός έδρας.

```
with open(solutionFile) as f:
    answer = f.readlines()
    answer = answer[:-1]
```

Οι πρώτες εντολές αποτελούνται από το διάβασμα του αρχείου εκτός της τελευταίας γραμμής που είναι κενή.

```
schedule = []
for game in answer:
    if game[-2] == "1":
        game = game.strip()
        game = game.split(" ")[0]
        game = game.split("_")[1:]
        schedule.append(game)
```

Στη συνέχεια γίνεται μια προ-επεξεργασία ώστε να αποθηκευτούν μόνο οι αγώνες που όντως συνέβησαν.

```
1  T = []
2  for game in schedule:
3      for team in game[:-1]:
4          if team not in T:
5              T.append(team)
6
7  T
```

```
['LeicesterCity',
 'NewcastleUnited',
 'Liverpool',
 'ManchesterUnited',
 'ManchesterCity',
 'Everton']
```

Έπειτα με αυτό το κομμάτι του κώδικα γίνονται γνωστές οι ομάδες που θα συμμετέχουν στο πρόγραμμα.

```
1  teams = {team: [] for team in T}
2  for game in schedule:
3      teams[game[0]].append('h')
4      teams[game[1]].append('a')
5
6  teams
```

```
{'LeicesterCity': ['h', 'a', 'h', 'a', 'a', 'h', 'h', 'a', 'h', 'a'],
 'NewcastleUnited': ['a', 'h', 'a', 'h', 'h', 'a', 'a', 'h', 'a', 'h'],
 'Liverpool': ['h', 'a', 'h', 'a', 'h', 'a', 'h', 'a', 'h', 'a'],
 'ManchesterUnited': ['a', 'h', 'h', 'a', 'h', 'a', 'h', 'a', 'a', 'h'],
 'ManchesterCity': ['h', 'a', 'a', 'h', 'a', 'h', 'a', 'h', 'h', 'a'],
 'Everton': ['a', 'h', 'a', 'h', 'a', 'h', 'a', 'h', 'a', 'h']}
```

Σ’ αυτό το κομμάτι του κώδικα δημιουργείται ένα λεξικό teams με κλειδί την κάθε ομάδα και με τιμή μια λίστα με τους αγώνες που παίζει ως εντός ή εκτός έδρας κατά τη διάρκεια του πρωταθλήματος.

```python
x = {}
for game in schedule:
    x[int(game[2])] = {}

for game in schedule:
    x[int(game[2])][game[0]] = game[1]

x
```

```
{1: {'LeicesterCity': 'NewcastleUnited',
  'Liverpool': 'ManchesterUnited',
  'ManchesterCity': 'Everton'},
 2: {'Everton': 'Liverpool',
  'ManchesterUnited': 'LeicesterCity',
  'NewcastleUnited': 'ManchesterCity'},
 3: {'LeicesterCity': 'ManchesterCity',
  'Liverpool': 'NewcastleUnited',
  'ManchesterUnited': 'Everton'},
 4: {'Everton': 'LeicesterCity',
  'ManchesterCity': 'Liverpool',
  'NewcastleUnited': 'ManchesterUnited'},
 5: {'Liverpool': 'LeicesterCity',
  'ManchesterUnited': 'ManchesterCity',
  'NewcastleUnited': 'Everton'},
```

Ακόμα δημιουργείται ένα νέο λεξικό όπως αυτό που δημιουργεί η βιβλιοθήκη pulp ώστε να είναι πιο εύκολος ο έλεγχος των παρακάτω περιορισμών για να είναι γνωστό ποια ομάδα παίζει με ποια και σε τι έδρα η κάθε μία.

```python
# Check if each team plays a game each week
for team in T:
    if len(teams[team]) != 2*len(T)-2:
        print("Team " + team + " does not play a game each week.")

```

✓ 0.0s

Εδώ ελέγχεται κάθε ομάδα αν ο αριθμός των αγώνων που παίζει είναι ίσος με τις εβδομάδες του πρωταθλήματος. Αφού αυτό το κομμάτι κώδικα δεν έδωσε αποτέλεσμα σημαίνει ότι ικανοποιείται ο περιορισμός.

```
# Check that each team plays each other team exactly once in the first round:
for team in T:
    count = 0
    for team2 in T:
        if team != team2:
            for week in range(1, len(T)):
                try:
                    if x[week][team] == team2:
                        count += 1
                except:
                    try:
                        if x[week][team2] == team:
                            count += 1
                    except:
                        pass
    if count != len(T)-1:
        print("Team " + team + " does not play each other team exactly once in the first round.")
```

Εδώ ελέγχεται για τον πρώτο γύρο του πρωταθλήματος αν κάθε ομάδα παίζει με όλες τις άλλες είτε εντός είτε εκτός. Το κλειδί x[week][team] δηλώνει ότι ομάδα team παίζει εντός έδρας εναντίων την τιμή αυτού του κλειδιού. Αφού αυτό το κομμάτι κώδικα δεν έδωσε αποτέλεσμα σημαίνει ότι ικανοποιείται ο περιορισμός.

```
# Check that each team plays each other team exactly once in the second round:
for team in T:
    count = 0
    for team2 in T:
        if team != team2:
            for week in range(len(T), 2*len(T)-1):
                try:
                    if x[week][team] == team2:
                        count += 1
                except:
                    try:
                        if x[week][team2] == team:
                            count += 1
                    except:
                        pass
    if count != len(T)-1:
        print("Team " + team + " does not play each other team exactly once in the second round.")
```

Εδώ ελέγχεται για το δεύτερο γύρο του πρωταθλήματος αν κάθε ομάδα παίζει με όλες τις άλλες είτε εντός είτε εκτός. Αφού αυτό το κομμάτι κώδικα δεν έδωσε αποτέλεσμα σημαίνει ότι ικανοποιείται ο περιορισμός.

```
# Check if team i played at home against team j in the first half,
# they will play away against team j in the second half, and vice versa.

for team in T:
    for team2 in T:
        if team != team2:
            countha = 0
            countah = 0
            for week in range(1, len(T)):
                try:
                    if x[week][team] == team2:
                        countha += 1
                except:
                    try:
                        if x[week][team2] == team:
                            countah += 1
                    except:
                        pass
            for week in range(len(T), 2*len(T)-1):
                try:
                    if x[week][team] == team2:
                        countah += 1
                except:
                    try:
                        if x[week][team2] == team:
                            countha += 1
                    except:
                        pass
            if countha == 1 and countah == 1:
                print("Team " + team + " does not play at home against team " + team2 + " in the first half, and away in the second half, or vice versa.")
```

Εδώ ελέγχεται αν στον πρώτο γύρο του πρωταθλήματος μια ομάδα έπαιξε εντός με μία άλλη τότε στο δεύτερο γύρο του πρωταθλήματος αν έπαιξε εκτός με την ίδια. Στο πρώτο for loop αυξάνουμε την κατάλληλη μεταβλητή counter αναλόγως ποια σειρά συναντήσαμε πρώτα, δηλαδή αν η team έπαιξε πρώτα ως εντός έδρας με την team2 ως εκτός έδρας αυξάνεται η μεταβλητή countha. Το ίδιο συμβαίνει και για το δεύτερο γύρο του πρωταθλήματος. Αν μία από τις δύο μεταβλητές έχει τιμή 2 τότε σημαίνει ότι ικανοποιείται ο περιορισμός, αν έχουν και οι δύο τιμή 1 τότε σημαίνει ότι και οι δύο αγώνες μεταξύ των δύο ομάδων έγιναν εντός έδρας για τη μία κι εκτός έδρας για την άλλη ομάδα. Αφού αυτό το κομμάτι κώδικα δεν έδωσε αποτέλεσμα σημαίνει ότι ικανοποιείται ο περιορισμός.

```
# Check if first two and last two games cannot be consecutive away or home games
for team in T:
    if teams[team][0] == teams[team][1]:
        print("Team " + team + " plays two consecutive " + teams[team][0] + " games.")
    if teams[team][-1] == teams[team][-2]:
        print("Team " + team + " plays two consecutive " + teams[team][-1] + " games.")
```

Εδώ ελέγχεται αν οι δύο πρώτοι και οι δύο τελευταίοι αγώνες είναι συνεχόμενοι εντός ή εκτός έδρας για κάθε ομάδα. Αφού αυτό το κομμάτι κώδικα δεν έδωσε αποτέλεσμα σημαίνει ότι ικανοποιείται ο περιορισμός.

```
1  # Check if the sequencing is satisfied
2  sequencing_rule = 0
3  for team in T:
4      for w in range(1, (2*len(T) - 1) - 5):
5          h = 0
6          a = 0
7          for i in range(5):
8              if teams[team][w+i] == 'h':
9                  h += 1
10             else:
11                 a += 1
12         if (h != 2 and a != 3) and (h != 3 and a != 2):
13             sequencing_rule += 1
14             print(h, a, w, team)
15
16 if (not sequencing_rule):
17     print("The sequencing rule is satisfied!")
✓ 0.0s
The sequencing rule is satisfied!
```

Εδώ ελέγχεται αν ικανοποιείται ο περιορισμός του sequencing. Για κάθε ομάδα επιλέγεται κάθε φορά ένα διάστημα 5 εβδομάδων κι ελέγχεται ο αριθμός των εντός κι εκτός έδρας αγώνων. Αν δεν ικανοποιείται ο περιορισμός στη γραμμή 12 τότε δεν ισχύει ο περιορισμός του sequencing. Από το αποτέλεσμα φαίνεται ότι ικανοποιείται πλήρως.

```python
# Teams from the same city cannot play both home in the same week
home_games = 0
for week in range(1, 2*len(T)-1):
    if (teams['Liverpool'] == teams['Everton']):
        home_games += 1

if (home_games):
    print("Teams from Liverpool play both home in the same week!")

home_games = 0
for week in range(1, 2*len(T)-1):
    if (teams['ManchesterCity'] == teams['ManchesterUnited']):
        home_games += 1

if (home_games):
    print("Teams from Manchester play both home in the same week!")
```

Εδώ ελέγχεται αν οι ομάδες από την ίδια πόλη παίζουν εντός έδρας την ίδια εβδομάδα. Αφού αυτό το κομμάτι κώδικα δεν έδωσε αποτέλεσμα σημαίνει ότι ικανοποιείται ο περιορισμός.

```python
# Check which teams play consecutive home and away games
for team in T:
    for w in range((2*len(T) - 2) - 1):
        if teams[team][w] == teams[team][w + 1]:
            if (teams[team][w] == 'h'):
                print("Team " + team + " plays consecutive home games in week " + str(w + 1))
            else:
                print("Team " + team + " plays consecutive away games in week " + str(w + 1))

```

```
Team LeicesterCity plays consecutive away games in week 4
Team LeicesterCity plays consecutive home games in week 6
Team NewcastleUnited plays consecutive home games in week 4
Team NewcastleUnited plays consecutive away games in week 6
Team ManchesterUnited plays consecutive home games in week 2
Team ManchesterUnited plays consecutive away games in week 8
Team ManchesterCity plays consecutive away games in week 2
Team ManchesterCity plays consecutive home games in week 8
```

Εδώ ελέγχεται ποιες ομάδες παίζουν 2 συνεχόμενα εντός κι εκτός έδρας παιχνίδια. Η αντικειμενική συνάρτηση προσπαθεί να ελαχιστοποιήσει τον αριθμό των 2 συνεχόμενων παιχνιδιών εντός έδρας κι έβγαλε προηγουμένως αποτέλεσμα 4. Εδώ μπορεί να διαπιστωθεί αφού οι αριθμός των 2 συνεχόμενων εντός έδρας παιχνιδιών είναι 4 και συμβαίνει μία φορά για 4 ομάδες. Επίσης αφού συνέβη 4 φορές να υπάρχουν 2 συνεχόμενα εντός έδρας παιχνίδια αναγκαστικά για τις ίδιες ομάδες θα συμβεί να υπάρχουν και 2 συνεχόμενα εκτός έδρας παιχνίδια.

Βιβλιογραφία

[1] https://en.wikipedia.org/wiki/Premier_League

[2] https://www.premierleague.com/news/419044

[3] F. Della Croce, D. Oliveri, (2004) Scheduling the Italian Football League: an ILP-based approach, Computers & Operations Research, 33, 1963-1974.

[4] Kendall, G., McCollum, B., Cruz, F.R.B., McMullan, P., While, L.: Scheduling English Football Fixtures: Consideration of Two Conflicting Objectives. In: Talbi, E.-G. (ed.) Hybrid Metaheuristics. SCI, vol. 434, pp. 369–385. Springer, Heidelberg (2013)

[5] https://arxiv.org/abs/1704.04879

[6] https://www.python.org/

[7] https://coin-or.github.io/pulp/

[8] https://neos-server.org/neos/

Αρχεία Κώδικα

Το βασικό αρχείο κώδικα main.py μοντελοποιεί το πρόβλημα και δημιουργεί ένα .lp αρχείο το οποίο στη συνέχεια μπορεί να τροφοδοτηθεί σε έναν solver ώστε να επιλύσει το πρόβλημα.

Στην παράγραφο 4 τα αποσπάσματα του κώδικα είναι από το αρχείο solvableProblem.py το οποίο μπορεί να εκτελεστεί αρκετά εύκολα από έναν υπολογιστή. Στην παράγραφο 5 με τη βοήθεια του αρχείου exportSolution.py δημιουργείται ένα αρχείο excel (PremierLeagueSchedule2022-2023.xlsx) το οποίο χρησιμοποιείται για την καλύτερη οπτικοποίηση των αποτελεσμάτων. Τέλος στην παράγραφο 6 χρησιμοποιείται το αρχείο validateSolution.py για την επικύρωση των αποτελεσμάτων και των έλεγχο των περιορισμών που έχουν οριστεί στην παράγραφο 3.

Η βασική βιβλιοθήκη που χρειάζεται να εγκατασταθεί είναι η pulp. Μόνο για την εκτέλεση του αρχείου exportSolution χρειάζεται επίσης να εγκατασταθούν οι βιβλιοθήκες pandas και openpyxl.