



**An Introduction to the National Football League Scheduling Problem using
Integer Programming**

-Amrith Deepak, Benjamin Teo, Yihao Yang

Introduction/Background (i)

Football is the biggest sport in the United States by measures of fan base and popularity. There are two main leagues broadcasting the sport (NCAA and NFL), with NCAA being a college league while National Football League being the professional one. We are interested in the NFL as it is the highest-grossing professional sports league in the worldⁱ, topping organizations such as the NBA (basketball) and Premier League (soccer).

The structure of the NFL is typical of most professional sports leagues, with 32 individually operated and owned “member” teams as well as a central governance organization. However, while the corporate governance structure is financed by the member teams collectively, it acts independently with regards to aspects such as bylaw enforcement and season schedulingⁱⁱ.

The NFL is an extremely lucrative league, with over \$12 billion in revenue for 2015ⁱⁱⁱ. Websites like Forbes and ESPN provide an income breakdown for the League, and broadcasting contracts have consistently represented over 60% of all revenue in the past half a decade. For example, in 2015, TV contracts were worth \$7.5 billion, while ticket and merchandising sales, as well as sponsorships, collectively brought in \$4.5 billion for the league^{iv}. Thus, if TV contracts are the dominant revenue stream, then the schedule of each NFL season significantly affects the league’s overall revenue, as well as each member team’s success.

In this paper, we aim to understand the current NFL season scheduling problem, from objective to constraints. After examining the current method, we will attack the problem by formulating and implementing our own model using methods learned in Operations Research, such as integer programming, to create a potentially “better”, or more “optimal”, solution.

We will begin the paper with a discussion of the current NFL scheduling methods in section (ii), including process and implementation, followed by presenting our own model in section (iii). We will then compare the two schedules in section (iv) using measurements such as overall welfare for the

League and “fairness” of the schedule. Lastly, in section (v), we will perform sensitivity analysis and test variations on the standard constraints, aiming to gain insight into how robust our solution is and how influential the various constraints are to the objective function.

Current NFL Scheduling Algorithm (ii)

Until this decade, the NFL had scheduled the entire season completely by hand^v. Perhaps in an effort to keep the integrity of its seasonal schedule, the organization is very secretive with regards to the specific method and algorithm for scheduling games. However, multiple sources confirm that the organization does employ an integer programming method, albeit not completely^{vi}.

The scheduling team at the NFL first hard-codes the marquee games (slots) of the year, such as the Thanksgiving night game. Then, the team uses computers to generate over 200 possible schedules from over 824 trillion game combinations satisfying the “hard-constraints” which we will discuss in the following section^{vii}. The objective, in this case, is to maximize the overall “ratings” of the match-ups, a direct indicator of profit^{viii} as network contracts are dependent on ratings.

After generating a list of possible schedules, the team assesses the strengths and weaknesses of each individual schedule. It is here that various “soft-constraints”, including “fairness”, are examined. For example, while the scheduling algorithm normally would not take the distance traveled between games into account since each team plays only once a week, if the group sees that one team must play consecutive away coast-to-coast games, they would consider it to be an unfair schedule, and try to permute accordingly. However, permutating the schedule by hand is usually a lost cause, as NFL’s Senior Vice President of Broadcasting, Howard Catz, put it: “You fix one team [problem], and 12 more problems come up.”^{ix} Ultimately, the scheduling team presents 3 to 4 viable solutions to the NFL executive committee, while scrapping the rest^x.

One thing to note is that the preferences for the schedules can change dramatically from year-to-year based on qualitative factors. For example, when generating the 2014 schedule, the team placed more emphasis on scheduling the in-division games towards the end of the season, as it has been shown to be a trend well-received by audiences and networks^{xi}.

Our Method (iii)

We use a strict integer-programming method to generate our schedule, believing that if the correct values are assigned to each team, the nature of the algorithm would assign the most valuable games to the most valuable slots, as to maximize overall revenue. Thus, we are letting the algorithm perform all the assignments, in a departure from the usual NFL methods.

Variable Representation

- Each team has a corresponding triplet representation (i, j, k) as well as a numerical one. Team (i, j, k) represents conference $i \in \{0,1\}$ ($AFC = 0, NFC = 1$), division $j \in \{0,1,2,3\}$ ($East = 0, North = 1, South = 2, West = 3$), Rank $k \in \{0,1,2,3\}$ ($1 = 0, 2 = 1, 3 = 2, 4 = 3$).
 - The triplet representation is a bit more complicated when we translate the problem into Gurobi[®]. Thus, we created a bijection between the triplet representation to $i \in \{0,31\}$.
- There are 17 weeks in the regular NFL season, with 6 slots in each week barring special occasions: Thursday Night, Sunday Early Afternoon, Sunday 4 pm, Sunday 4:25 pm, Sunday Night, and Monday Night. Each slot has a duplet representation (k, l) , $k \in \{0:16\}$ and $l \in \{0:5\}$ corresponding to the week number and time slot, respectively. For example, time slot $(14,3)$ would correspond to the Sunday 4 pm slot of week 15.
 - The duplet representation, once again, is a bit more complicated when we transfer to Gurobi[®]. Thus, we created a bijection between the duplet representation to $i \in \{0,101\}$.

- x_{ijklh} is the variable of concern, representing team i (represented by the triplet or single value) playing team j (represented by the triplet or single value) at week k time slot l and home or away.
- v_{ijklh} is the value we assign the variable in which team i plays team j at week k , time slot l , and home or away h . The method of valuation is described below.

Value of The Game

As discussed earlier, network and broadcasting contracts account for a majority of revenue for the NFL. Hence, we are focusing on the TV-revenue for determining the optimal schedule for a season.

We first need to understand the influential factors for determining the value of each game

One of the most important factors in determining ratings (revenue) is the winning record of the participating team. People are more likely to watch games between top teams than ones between teams performing poorly. Thus, looking at the winning percentages is undoubtedly an important factor.

However, winning record alone is not sufficient for determining the value of each team. People's "interest" in the participating teams are often a direct predictor of ratings. Fortunately, websites such as CBS^{©xii} and USAToday^{©xiii} have already researched and quantified each team's "interest-level" with its fans, giving us another crucial set of data for determining the value of a game.

In addition to the two factors mentioned above, Teams located in big cities such as New York, Chicago, and San Francisco often have more valuable games in the eyes of a broadcaster, despite not having the best record. This is due to the larger market and fan base of these high net-worth teams. Such teams can command higher-ratings, despite perhaps generating less competitive games.

The last critical factor in determining the value of any game is the time slot assigned. NFL games are played on Thursday evening, Sunday early afternoon, 2 slots on Sunday late afternoon, Sunday evening, and Monday evening. We are approximating the value of each slot through the price of a 30-second commercial during that time. With this method, we find that the most valuable time slot is

Sunday evening, as a half-a-minute commercial costs \$700,000, twice as much as that of the least valuable slot, Sunday early afternoon^{xiv}. Using the same method, we value Sunday late afternoon and Monday night as 1.2 times Sunday afternoon, and Thursday night as 1.5 times Sunday afternoon.

Having found the contributing factors for determining the value of a team, what logically follows is the assignment of weights to accurately account for all these factors. Our scaling and weighting are based on research by similar papers. Multiple papers and sports analysts believe that for such a problem, the winning record is twice as important as the popularity of a team (measured in fan base)^{xv}. Thus, to account for that fact, we scale the net worth of the team by a factor of 1 billion and multiply that by 4, multiply the percentage of wins by 120, and normalize people's interest in the game to average around 30. This gives a 60% weight to winning percentage of the team, 30% weight to people's interest, and 10% weight to the value of the team, fitting with the aforementioned research. Finally, to find the value for a game, we use the sum of the two teams' values, multiplied by the value of the time slot. While there are other metrics available, they will all yield similar results due to the scaling of all the variables.

Objective and Constraints^{xvi}

$$\text{Objective: } \max_{i,j,k} \sum_{i=0}^{32} \sum_{j>i}^{32} \sum_{k=0}^{16} \sum_{l=0}^5 \sum_{h=0}^1 v_{ijklh} * x_{ijklh}$$

Maximizing the Overall Revenue

Subject To:

$$\sum_{l=0}^5 x_{ijklh} = 1 \quad \forall i \in \{0: 32\}, j > i, k \in \{0: 2, 12: 16\}$$

Each team plays once per week for first 3 weeks and last 5 weeks

$$\sum_{l=0}^5 x_{ijklh} \leq 1 \quad \forall i \in \{0:32\}, j > i, k \in \{3:11\}$$

Between weeks 4 and 12, each team can play at most once per week

$$M_l \leq \sum_{i=0}^{32} \sum_{j>i}^{32} x_{ijklh} \leq N_l \quad \forall k \in \{0,16\}, l \in \{0,5\}, h \in \{0,1\}$$

Each time slot has an upper N_l and lower bound M_l on the number of games

For the following constraints, please refer to attached code and reference line number

Teams can't play on Thursday if there was a game the Monday prior (Line 103)

Each team plays no more than 1 short week per season (Line 132)

(Short week: If a team played on Sunday and then again on the following Thursday)

Avoid consecutive away games for each team during short weeks (Line 106)

No more than 3 consecutive home or away (Line 118)

Implementation (Excel)

We are currently using the Gurobi[®] package for Python to create schedules due to its more flexible constraints and capabilities compared with Excel-based solvers such as Open Solver. However, for this portion of the paper and explanation of our implementation, it is simpler to represent in Excel, due to its easily understandable interface and set-up. Note that our set-up in Gurobi[®] is extremely similar, albeit being more comprehensive and complete.

The set-up for our problem in Excel is complicated, due to the larger size of the program and the number of variables needed for accurate representation. Thus, we will detail a simpler problem, which is very similar to the set-up for the full problem. The full spreadsheet and Python code can be found in the Appendix.

Sample Problem:

- Objective: maximize the revenue of an NFL season comprised of 3 teams $i \in \{0,2\}$ (Patriots=0, Bills=1, Dolphins=2), 3 time slots $t \in \{0,2\}$, and 3 games (*Patriots v Bills*, *Patriots v Dolphins*, *Bills v Dolphins*).
- Constraints:
 - Each game must be played exactly once
 - Each time slot must have exactly one game

| Team# | | Team# | | 1 | 2 | 3 | | Value of Each Variable | | |
|----------|--|----------|--|---|---|---|--|------------------------|---|---|
| Patriots | | Bills | | A | F | G | | a | f | g |
| Dolphins | | Bills | | B | E | H | | b | e | h |
| Patriots | | Dolphins | | C | D | I | | c | d | i |

Here, the decision variables are A, B, \dots, I and corresponding values are a, b, \dots, i .

To satisfy the game constraint, we can require the sum of each row of decision variables to equal exactly 1.

| Team# | | Team# | | 1 | 2 | 3 | | Value of Each Variable | | |
|----------|--|----------|--|---|---|---|----|------------------------|---|---|
| Patriots | | Bills | | A | F | G | =1 | a | f | g |
| Dolphins | | Bills | | B | E | H | =1 | b | e | h |
| Patriots | | Dolphins | | C | D | I | =1 | c | d | i |

To satisfy the time slot constraint, we can require the sum of each column of decision variables to equal exactly 1.

| Team# | | Team# | | 1 | 2 | 3 | | Value of Each Variable | | |
|----------|--|----------|--|----|----|----|--|------------------------|---|---|
| Patriots | | Bills | | A | F | G | | a | f | g |
| Dolphins | | Bills | | B | E | H | | b | e | h |
| Patriots | | Dolphins | | C | D | I | | c | d | i |
| | | | | =1 | =1 | =1 | | | | |

Now, we can add our objective cell and our final problem in Excel as follows:

| Team# | | Team# | | 1 | 2 | 3 | | Value of Each Variable | | |
|----------------|--|----------------------------------|--|----|----|----|----|------------------------|---|---|
| Patriots | | Bills | | A | F | G | =1 | a | f | g |
| Dolphins | | Bills | | B | E | H | =1 | b | e | h |
| Patriots | | Dolphins | | C | D | I | =1 | c | d | i |
| | | | | =1 | =1 | =1 | | | | |
| Objective Cell | | =SUM(Aa+Bb+Cc+Dd+Ee+Ff+Gg+Hh+Ii) | | | | | | | | |

One aspect of generating the full schedule is the inclusion of a “fairness” constraint. For example, a team should not have more than 2 away games in a row or a team cannot play more than 1 “short week”, where a team plays on a Sunday, then on Thursday of the following week. Such penalties were hard to implement in Excel, one of the few factors for switching to Gurobi®.

Implementation (Gurobi®)

While our implementation in Excel made it easy to visualize our results, it was also very cumbersome to use. The decision variables had to be represented as a single matrix of cells, and so each time we decided to increase the “dimension” of our decision variables (by “dimension” we mean the number of factors that determine the value of the decision variable) it forces an expansion of this matrix (and consequently also make several adjustments across the multiple spreadsheets to which this matrix was linked). As the size of the matrix grew, Excel’s value as a convenient tool for providing an intuitive visual representation of the problem quickly diminished. In addition, we spoke to fellow classmates about the runtime of their linear solvers and realized that Open Solver was far less efficient than other linear solving packages tied to popular programming languages such as MATLAB and Python. The combination of these factors compelled us to find better alternatives for computationally representing and solving our problem. After some exploration, we decided to write our problem formulation in Python and use the popular mathematical programming solver Gurobi to compute the solution.

Gurobi® offers us much more flexibility because we can increase the “dimension” of our decision variables or impose additional constraints with minimal changes to the rest of our program. The only information that we need to pass to our program are one Excel sheet specifying the predetermined

matchups between different pairs of teams as well as whether or not that game is “home” or “away” with respect to the first team, and another Excel sheet specifying some measure of net worth for each team.

Our program reads in these two sheets as data frames and creates a dictionary mapping teams to the games that they have to play. It then creates a “five-dimensional” (team 1, team 2, week, timeslot, home/away status with respect to team 1) binary decision variable for each matchup (team 1, team 2, home/away status with respect to team 1) and the possible time (week, timeslot) that the game might be played at, and constrains the problem by specifying a series of inequalities/equalities involving various linear combinations of the decision variables before solving for optimality.

In Gurobi, the hard constraints that we imposed were:

- 1) Each game is played once.
- 2i) Each team plays once a week for the first 3 weeks, and once a week for the last 5 weeks.
- 2ii) Each team plays at most once a week for the remaining weeks.
- 3) Each timeslot has an upper and lower bound on the number of games that can be played.
- 4) Teams are not allowed to play on Thursday if their previous game was on the Monday just before.
- 5) Each team plays no more than 3 consecutive home or away games.
- 6) No team has to play consecutive away games that are also separated by a short week

These constraints were easily expressed as linear constraints, and the program was able to solve for optimality in a span of about 10-15 seconds (a great improvement over Open Solver, which took about 10 minutes to run to completion). However, further constraint that we initially did have problems imposing was:

7) Each team plays no more than 1 game separated by a short week per season

This constraint could only be exactly captured as a quadratic constraint, and made it necessary for us to first solve the relaxation of the problem before applying the branch-and-bound method to gradually constrain all the decision variables to be binary variables. Due to the sheer number of quadratic variables that we ended up having, it took an unfeasibly long amount of time for our the branch-and-bound process to reach stopping criteria.

The key to resolving this issue is based on 2 observations. First, for a team to play a short week it has to play a Thursday game. Second, there are only 17 Thursday games (since there is only 1 game each Thursday, and that is the 8:30 pm game) and 32 teams, so on average no particular team ought to have to play too many Thursday games and hence short weeks. Based on these 2 observations we came up with a generalized version of a soft constraint intended to serve as a surrogate for constraint 7:

7*) Each team plays at most x Thursday games per season

We started off by letting x equal to 1, but this proved to yield no feasible solutions in combination with the constraints 1 – 6, so we increased x to 2 and this time it did have feasible solution that was not too far off from in terms of objective value compared to the solution achieved with only constraints 1 – 6. Thus the last constraint that we added to our model was:

7**) Each team plays at most 2 Thursday games per season

This is undoubtedly weaker than 7) but at least it provides a guarantee that each team plays at most 2 short weeks per season, and importantly generates an actual feasible solution.

Discussion of Results and Solution (iv)

Our solution seems to conform with a sensible NFL schedule. For example, high-profile matchups between high-popularity or high-ranked teams such as Cowboys and Patriots have the prime-time slot of Sunday night, fitting with our objective of maximizing overall revenue.

To quantify how “optimal” our solution is, we can compare our generated schedule to the one NFL currently uses. We used our method of values of each game to get the total value of all games played during the 2016 NFL season. We then compare this to that of our own generated schedule. Our solution gives a value of 68779.9, whereas the actual 2016 games give a value of 59986 using our values of the game. The values are very close to each other, but the fact that we get a slightly higher value using our schedule suggests that the NFL is looking for some extra criteria like team preferences, or how to split fairly between the different TV networks. In addition, the NFL could have a different, or more sophisticated method of valuing each game, having access to data not publicly available.

The games we scheduled for Thursday night and the actual games played on Thursday night differ considerably. The only overlap is the game between the Patriots and Ravens. The other games are divided between Sunday early afternoon, Sunday late afternoon, Sunday night, and Monday night, with relative proportions approximately equal to the number of games played on those nights. Similarly, the only overlap for Mondays’ games is a game between the Bills and Seahawks. The other games are similarly divided as with the Thursday night games. These variations are conceivable as our valuations for these time-slots are quite alike.

In contrast with the other slots, there is an overlap of more than half the games for the prime-time slot (Sunday Night). This reasonably good agreement is probably due to the high value of Sunday night games and the algorithm matching high-value games with the high-value slot. In a similar fashion, our Sunday early afternoon and Sunday late afternoon games agree with the actual schedule with more than half the games overlapping. The remaining games are played in the other time slots. Looking at

this, it seems like the overall values are similar, as the very high-value games like Cowboys vs. Eagles and Seahawks vs. Patriots are played on Sunday night. The less valuable Sunday night, Monday night, and Thursday night games being mixed or the Sunday early afternoon and late afternoon games being different contribute to the variation in overall value, but comparing the values as a whole, there is not a huge difference in the objective considering the subjective weights on different factors as well as imperfect information.

Sensitivity Analysis (v)

One of the afterthoughts we had upon successful generation of the full schedule from our Gurobi implementation was to examine how sensitive the value of the maximized objective was to the removal/inclusion of various constraints. Intuitively, the extent to which the maximized objective decreases in value upon the addition of a given constraint is indicative of how restrictive the constraint is on the solution space of tenable and profitable schedules. We wanted to gain a better insight as to which constraints seem to “matter more” as far as the overall profit of the schedule is concerned. Below is a simple summary of our preliminary findings:

| | |
|--|----------------------|
| No constraints | 6881490 |
| Constraint 1 | 102479 |
| Constraints 1, 2 | 102479 |
| Constraints 1, 2, 3 | 68808.2 |
| Constraints 1, 2, 3, 4 | 68808.2 |
| Constraints 1, 2, 3, 4, 5 | 68808.2 |
| Constraints 1, 2, 3, 4, 5, 6 | 68779.9 |
| All constraints (1-7** without relaxation) | No Feasible Solution |
| All constraints (1 – 7**) | 68779.9 |

While it is evident and reasonable that constraint 3 (Each timeslot has an upper and lower bound on the number of games that can be played) places a significant restriction on the solution space, it is less intuitive yet still plausible that constraints 4 and 5 do not reduce the value of the maximized objective. Furthermore, the constraints involving short weeks (i.e. 6 and 7**) appear to have only a small effect in decreasing the value of the maximized objective, so perhaps this suggests the relative importance of other constraints (that we have not explored in our formulation) such as the predetermined time scheduling for particular marquee games in restricting the profitability of the overall schedule. This would be an interesting aspect of the problem to consider for further research.

Another possible path for further research is with variations in the problem. Currently, we are maximizing the overall revenue, conforming to the NFL's scheduling problem. However, it could also be interesting to consider an objective that tries to maximize the fairness of the game, either from the teams' or networks' perspective. These are all interesting and deep problems for further integer programming research.

-
- ⁱ <https://howmuch.net/articles/sports-leagues-by-revenue>
- ⁱⁱ https://en.wikipedia.org/wiki/National_Football_League#Corporate_structure
- ⁱⁱⁱ <http://www.investopedia.com/articles/personal-finance/062515/how-nfl-makes-money.asp>
- ^{iv} <http://www.cnbc.com/2014/09/04/how-the-nfl-makes-the-most-money-of-any-pro-sport.html>
- ^v Urena, Ivan. *Pro Football Schedules: A Complete Historical Guide from 1933 to the Present*. Jefferson, NC: McFarland &, 2014. Print.
- ^{vi} <http://mmqb.si.com/mmqb/2016/04/14/2016-nfl-schedule-release>
- ^{vii} <http://mmqb.si.com/2014/04/24/making-of-2014-nfl-schedule>
- ^{viii} <http://bleacherreport.com/articles/1606598-how-is-the-nfl-schedule-created#>
- ^{ix} <http://mmqb.si.com/2014/04/24/making-of-2014-nfl-schedule>
- ^x <http://mmqb.si.com/mmqb/2016/04/14/2016-nfl-schedule-release>
- ^{xi} <http://mmqb.si.com/2014/04/24/making-of-2014-nfl-schedule>
- ^{xii} <http://www.cbssports.com/nfl/news/nfl-fan-base-rankings-cowboys-beat-out-steelers-packers-fans-for-top-spot/>
- ^{xiii} <http://ftw.usatoday.com/2016/07/ranking-all-32-nfl-teams-by-likability>
- ^{xiv} <http://adage.com/article/media/ad-pricing-chart-sunday-night-football-empire-broadcasts-most-expensive-ad-buys/300516/>
- ^{xv} <http://www.theharrispoll.com/sports/Americas-Favorite-Football-Team-2015.html>
- ^{xvi} <http://operations.nfl.com/the-game/creating-the-nfl-schedule/>

Appendix:

Resources:

<https://howmuch.net/articles/sports-leagues-by-revenue>
https://en.wikipedia.org/wiki/National_Football_League#Corporate_structure
<http://www.investopedia.com/articles/personal-finance/062515/how-nfl-makes-money.asp>
<http://www.cnbc.com/2014/09/04/how-the-nfl-makes-the-most-money-of-any-pro-sport.html>
Urena, Ivan. *Pro Football Schedules: A Complete Historical Guide from 1933 to the Present*. Jefferson, NC: McFarland &, 2014. Print.
<http://mmqb.si.com/mmqb/2016/04/14/2016-nfl-schedule-release>
<http://mmqb.si.com/2014/04/24/making-of-2014-nfl-schedule>
<http://bleacherreport.com/articles/1606598-how-is-the-nfl-schedule-created#>
<http://www.cbssports.com/nfl/news/nfl-fan-base-rankings-cowboys-beat-out-steelers-packers-fans-for-top-spot/>
<http://ftw.usatoday.com/2016/07/ranking-all-32-nfl-teams-by-likability>
<http://adage.com/article/media/ad-pricing-chart-sunday-night-football-empire-broadcasts-most-expensive-ad-buys/300516/>
<http://www.theharrispoll.com/sports/Americas-Favorite-Football-Team-2015.html>
<http://operations.nfl.com/the-game/creating-the-nfl-schedule/>
<http://www.gurobi.com/company/news/the-national-football-league-chooses-gurobi>
<http://blogs.sas.com/content/operations/2016/02/05/detecting-anomalies-in-the-nfl-schedule/>

Reference Papers:

<http://www.sloansportsconference.com/wp-content/uploads/2015/02/SSAC15-RP-Finalist-Alleviating-competitive-imbalance-in-NFL-schedules.pdf>
<http://pubsonline.informs.org/doi/pdf/10.1287/ited.5.1.10>
http://www.fico.com/en/wp-content/uploads/NFL_Success_2495CS.pdf
https://courses.cs.washington.edu/courses/cse521/07wi/prj/sam_scott.pdf
http://mat.gsia.cmu.edu/trick/tourn_final.pdf
<https://pdfs.semanticscholar.org/4af6/5185dbf7f201aa1171ba46ada4cb02f1dd36.pdf>