Assigning Season Tickets Fairly

This article is presented by Mr. Thomas A. Grandine. It shows very good way to use mixed Integer Programming to assign the season tickets. Motivation for this came when six friend decided to purchase a pairs of season tickets for the Seattle Mariners Baseball games. They were faced with the problem of having to decide who would go to to each of 81 home games. The basic problem was the priorities of each individual for attending certain games.
So to solve this problem they used mixed integer programming approach to get the good distribution of the tickets.

Approach:
Consider a simple case of four game season with three participants A,B,and C.Suppose C wants all four tickets to two games , and that A and B each want two pairs of tickets .Suppose further that the order in which people pick is A,B and C.If A chooses Game1, B choosesGame2, and C chooses Game 3 in first round and A chooses Game 4 in second round.This leaves no game open from which C can choose remaining block of four tickets .To fis this problem C should be allowed to oick earlier in the draft than his turn.This problem can be solved by treating it is an assignment problem.

Set up for the original problem:
Allow each person to describe his ideal season ticket package. Thus, person i, selected two quantities, p_i and q_i, corresponding to the number of games for which he wanted a pair of tickets and the number of games for which he wanted all four tickets. So,
\[ \sum_{i=1}^{n} p_i + 2q_i = 162 \]  (this ensures that everyone of the 162 pairs of ticket will be allocated to somebody.
Now ,define a large no. of variables p_{ij} and q_{ij} and required each to have a value of either zero or one. Here p_{ij} = 1 means that the person I is assigned a pair of tickets to game I, while p_{ij} = 0 means that he does not get that pair of tickets. Similarly, q_{ij} = 1 means that person I gets all the four tickets to game j, while q_{ij} = 0 means that he doesn’t.
Thus , each person I had to have
\[ \sum_{j=1}^{81} p_{ij} = p_i \]  (this ensured that he got the mix of tickets he wanted.
\[ \sum_{j=1}^{81} q_{ij} = q_i \]
Furthermore , each game j had to satisfy
\[ \sum_{i=1}^{n} p_{ij} + 2q_{ij} = 2. \]
As two pairs of ticket had to be allocated for each game.
Now each person to place restrictions on the tickets he received. For example, person I could limit the number of tickets he was allocated on a nine –game home stand that started with game j by imposing
\[ \sum_{k=0}^{8} p_{i,j+k} + q_{i,j+k} \leq 1 \]
which ensured that I got tickets to at most one game during that particular home stand.

Now the final step in setting up this problem was to establish criteria for evaluating the goodness of any particular allocation. This can be achieved by having each person assign weights to each of the games, i.e., person I assigned a weight \( \alpha_{ij} \) to each game j. Each person’s satisfaction with his allocation of tickets then had the measure.
\[ \sum_{j=1}^{81} \alpha_{ij} (p_{ij} + 2q_{ij}) \]

where small measures were assumed to be more satisfactory than larger ones.

Putting all these together, will give an assignment problem that allocated the available tickets by minimizing the dissatisfaction of the person who was most dissatisfied:

Minimize \( e \)

Subject to

\[ \sum_{j=1}^{81} p_{ij} = p_i \quad \text{for each } i \]

\[ \sum_{j=1}^{81} q_{ij} = q_i \quad \text{for each } i \]

\[ \sum_{j=1}^{81} \alpha_{ij} (p_{ij} + 2q_{ij}) \leq e \quad \text{for each } i \]

\[ \sum_{i=1}^{n} p_{ij} + 2q_{ij} = 2 \quad \text{for each } j \]

various personal constraints

\( p_{ij}, q_{ij} = 0 \) or \( 1 \)

for some constant \( c \).

Hence this can be solved to get the assignment for the matches. But there is one big problem in this, i.e. one person could use different weighting schemes to bias the results in one’s favor. In particular, using weights of \( 1 \) for the games one wants to attend and weights of \( 1+\epsilon \) for the games one doesn’t want to attend would result in great dissatisfaction. The optimization code would then struggle to improve this person’s lot, allocating the games he or she wanted to everyone else’s expense.

Fixing this required a change in the objective function.

The modified optimization problem is

Minimize \( ce + \sum_{i=1}^{n} \sum_{j=1}^{81} \alpha_{ij} (p_{ij} + 2q_{ij}) \)

Subject to

\[ \sum_{j=1}^{81} p_{ij} = p_i \quad \text{for each } i \]

\[ \sum_{j=1}^{81} q_{ij} = q_i \quad \text{for each } i \]

\[ \sum_{j=1}^{81} \alpha_{ij} (p_{ij} + 2q_{ij}) \leq e \quad \text{for each } i \]

\[ \sum_{i=1}^{n} p_{ij} + 2q_{ij} = 2 \quad \text{for each } j \]

various personal constraints

\( p_{ij}, q_{ij} = 0 \) or \( 1 \)

for some constant \( c \).

Now this optimization problem can be solved by mixed integer programming solving code. In actual practice it was solved by using CPLEX. It did compute a near optimal solution. Hence this provides a good approach and shows a good use of Mixed integer programming in real world problems.