# Realignment in the National Football League

**Did They Do It Right?**

**John E. Mitchell**[1]

Mathematical Sciences
Rensselaer Polytechnic Institute
Troy, NY 12180
http://www.rpi.edu/~mitchj

November 22, 2000
Revised: February 11, 2003

**Abstract**

The National Football League (NFL) in the United States expanded to 32 teams in 2002 with the addition of a team in Houston. At that point, the league was realigned into eight divisions, each containing four teams. We describe a branch-and-cut algorithm for minimizing the sum of intradivisional travel distances. We consider first the case where any team can be assigned to any division. We also consider imposing restrictions, such as aligning the AFC (American Football Conference) and the NFC (National Football Conference) separately or maintaining traditional rivalries. We show that the alignment chosen by the NFL does not minimize the sum of intradivisional travel distances, but that it is close to optimal for an alignment that aligns the NFC and AFC separately and imposes some additional restrictions.

**Keywords:** Realignment, graph equipartition, branch-and-cut.

## 1 Introduction

The National Football League (NFL) in the United States expanded to 32 teams in 2002 with the addition of a team in Houston. At that point, the league was realigned into eight divisions, each containing four teams [30]. Each team plays sixteen games during the regular season. Each team plays each other team in its division twice, playing its remaining games against a subset of the teams outside its division. A team's opponents from outside its division depend on the team's results in the previous season and on a rotating choice of one other division. Many

[1]Research supported in part by NSF grant number CCR–9901822

Figure 1: The realignment that minimizes the sum of intradivisional travel distances.

arrangements of the divisions were under consideration. Since the choice of outside teams varies from season to season, one possibility was to choose the divisions so as to minimize the total intradivisional travel distance. This objective has the benefit of placing nearby cities in the same division, and such an arrangement will create natural rivalries. If this objective had been used, we show that the NFL could have reduced the sum of intradivisional travel distances by 45%. One advantage of our solution is that no two teams separated by more than one time zone appear in the same division. This optimal solution is shown in Figure 1 and it is detailed in Table 1 in §3.1.

Computational results are presented in §3. First, we consider the case where any team can be assigned to any division, in §3.1. The realignment chosen by the NFL was designed to maintain various historical rivalries (for example, between the Dallas Cowboys and the Washington Redskins). The NFL is divided into two conferences. Prior to the 2002 season, the AFC (American Football Conference) had 16 teams and the NFC (National Football Conference) had 15 teams. The Houston Texans could have been placed in the NFC, so each conference had 16 teams, and then each conference could be aligned separately. Instead, Houston was placed in the AFC and the Seattle Seahawks were moved from the AFC to the NFC, again with each conference being aligned separately. We consider aligning the NFL in the presence of various combinations of restrictions in §3.2. We show that the realignment chosen by the NFL is optimal if several historical rivalries are respected.

We model the realignment problem as a $k$-way equipartition problem, where the nodes of a graph are divided into $k$ equally sized sets in order to minimize an appropriate objective function (see §2 for details). Other application areas of the $k$-way equipartition problem include telecommunications [26], 3-dimensional integrated circuit design [7], and microaggregation for statistical disclosure control [12].

The network design application considered in [26] is to cluster customers into equally sized sets and then build a SONET ring for each cluster, in order to provide redundancy for communication between two customers that exchange a large amount of data.

Currently, printed circuit boards are almost exclusively fabricated on a single board, with transistors placed on each side of the board. It is relatively expensive to connect transistors on opposite sides of the board, so it is desirable to partition the set of transistors into two sets of approximately equal size. There is interest in developing three dimensional integrated circuits, consisting of many layers of transistors [7]. It would then be desirable to partition the transistors into $k$ sets of approximately equal cardinality.

Many datasets contain confidential data, with the raw material consisting of information from individual respondents. One way to protect confidentiality is to cluster the data from respondents with similar characteristics. In order to protect the confidentiality of the data, it is desirable to impose a minimum size on any of the clusters. In certain situations, the clusters have equal cardinality. This method of statistical disclosure control is known as microaggregation [12]. A different problem in statistical disclosure control is described by Fischetti and Salazar [18], who use an integer programming branch-and-cut approach.

Operations research has been used successfully in the past in several applications drawn from sports — we describe just a few examples. Nemhauser and Trick [29] developed a playing schedule for the ACC, a major college basketball conference. Trick [32] has examined the scheduling of other sports tournaments, including minor league baseball. Older work on sports scheduling includes that of Bean and Birge [4] on the NBA (National Basketball Association). The problem of determining which teams can still qualify for the playoffs, based on the remaining schedule, can be formulated as a linear programming problem. For a recent description of this problem, see Adler *et al.* [1]. Duckworth and Lewis [14] proposed a method for accommodating rain delays in determining the result of cricket matches. This method has been adopted by the International Cricket Council and is used throughout the world.

Saltzman and Bradford [31] modeled the NFL realignment problem as a modification of the quadratic assignment problem. They also proposed minimizing the sum of intradivisional travel distances. Their formulation leads to a quadratic programming problem with a nonconvex objective function. They used MINOS to find a local minimizer for this quadratic program. When the NFL expanded to 30 teams in 1996, they showed that the alignment resulting from their algorithm would have reduced the total intradivisional travel distance by 40%.

Before discussing our results, we describe the algorithm used in §2. It should be emphasized that our approach results in an optimal solution in each situation considered; that is, there is no possible arrangement of the cities that has a smaller sum of intradivisional travel distances.

## 2 The $k$-way equipartition problem

The teams in the NFL can be regarded as the vertices of a graph. An edge between two teams has weight equal to the distance between the two corresponding cities. Any partition of the teams into eight divisions each containing four teams is a valid alignment. The *total intradivisional travel distance* is the sum of the edge weights of all edges that have both endpoints in the same division. The objective we choose is to minimize this sum. This has the effect of placing teams that are close to one another in the same division.

The realignment problem is an example of the $k$-way equipartition problem. Given a graph $G = (V, E)$ with vertices $V$ and edges $E$, where each edge $e \in E$ has weight $c_e$, the aim of the $k$-way equipartition problem is to divide the graph into $k$ sets of vertices, each of the same size, so as to minimize the total weight of the edges which have both endpoints in one of the sets. We will call these $k$ sets *divisions*. We assume $|V|$ is an integer multiple of $k$. This can be regarded as a clustering of the vertices, with the additional condition that each cluster must contain the same number $|V|/k =: S$ of elements. In what follows we assume $G$ is the complete graph on $|V|$ vertices, as is the case for the NFL realignment problem.

We use a branch-and-cut algorithm to solve the problem. This constructs an integer programming formulation of the problem. It then solves a linear programming (LP) relaxation of the problem. The solution to the LP relaxation can be used to generate feasible integer solutions. The LP relaxation is tightened by adding strong cutting planes, that is, inequalities that are high-dimensional faces of the convex hull of feasible solutions to the integer program, and are violated by the solution to the LP relaxation. Once the LP relaxation has been tightened, it can be solved again and the process repeated. If good cutting planes cannot be found and optimality has not yet been proved, then branching can be used to finish off the solution. The algorithm is described in more detail in §2.3. First, in §2.2, we discuss the polyhedral theory of the $k$-way equipartition problem, and the inequalities that are used as cutting planes.

We define a binary variable $x_{ij}$, which takes the value 1 if $i$ and $j$ are in the same division and 0 otherwise. Our formulation is:

$$\begin{array}{lll} \text{min} & \sum_{e \in E} c_e x_e & \\ \text{subject to} & \sum_{e \in \delta(v)} x_e \;=\; S - 1 & \forall v \in V \\ & x \text{ is the incidence vector of a clustering} & \end{array}$$

where $\delta(v)$ denotes the set of edges incident to vertex $v$.

The set of incidence vectors of clusterings consists of all binary vectors satisfying certain triangle inequalities, given in (3)–(5) below. Finding the clustering to minimize a linear objective function is itself an $NP$-hard problem. A complete description of the convex hull of incidence vectors of clusterings is not known and would be exponential in size.

We define $Q(kS)$ and $\bar{Q}(kS)$ as follows:

$$\begin{aligned} Q(kS) \;&:=\; \{x \in \{0,1\}^n : \sum_{e \in \delta(v)} x_e = S - 1 \;\; \forall\, v \in K_{kS}, \\ &\qquad\qquad x \text{ is the incidence vector of a clustering}\} \\ \bar{Q}(kS) \;&:=\; \{x \in [0,1]^n : \sum_{e \in \delta(v)} x_e = S - 1 \;\; \forall\, v \in K_{kS}\}, \end{aligned}$$

where $n = kS(kS-1)/2$ and $K_q$ denotes the complete graph on $q$ vertices. We want to minimize the objective function over $Q(kS)$. Our initial linear programming relaxation will have feasible region $\bar{Q}(kS)$.

In addition to being a constrained version of the clustering problem, the $k$-way equipartition problem is closely related to the classical graph partition problem, another $NP$-hard problem. This requires partitioning the vertices into two equally sized sets $U_1$ and $U_2$ so as to minimize the total weight of the edges that either have both endpoints in $U_1$ or both in $U_2$.

Approaches to the $k$-way equipartition problem include the multilevel approach of Karypis and Kumar [24] and the eigenvalue approach of Donath and Hoffman [13], tested computationally by Areibi and Vannelli [2] and by Falkner *et al.* [15]. If $k = 2$ we have the equipartition problem, which has been studied extensively. Branch-and-cut approaches to this problem include the work of Conforti *et al* [6, 10, 11] and Chopra [8]. Ferreira *et al.* [16, 17] have developed a branch-and-cut algorithm for the related node capacitated graph partitioning problem. Semidefinite programming approaches for graph equipartition were investigated by Frieze and Jerrum [19] and Ye *et al.* [5, 34]. Semidefinite programming approaches for the $k$-way equipartition problem were developed by Karisch and Rendl [23] and Wolkowicz and Zhao [33]. Lisser and Rendl [26] have described a network design problem that can be modeled as a $k$-way equipartition problem and they have investigated semidefinite and polyhedral approaches to it.

If the sets are not all constrained to be of the same cardinality, we have a clustering or partition problem. This problem has been investigated by Grötschel and Wakabayashi [21, 22]. Chopra and Rao [9] have investigated a polyhedral approach for the partition problem for graphs that are sparse; their approach defines variables for both the edges and the vertices.

When $S = 2$, we have a matching problem, so the problem is polynomially solvable. For larger choices of $S$, the problem is NP-complete, as shown in Garey and Johnson [20].

## 2.1 Notation

Given a subset $U \subseteq V$, we define $E(U)$ to be the edges with both endpoints in $U$, and we define $x(U) := \sum_{e \in E(U)} x_e$. Similarly, we define $\delta(U)$ to be the edges with exactly one endpoint in $U$. Given two disjoint subsets $U \subseteq V$ and $W \subseteq V$, we define $E(U, W)$ to be the edges with exactly one endpoint in $U$ and exactly one endpoint in $W$; further, we define $x(U, W) := \sum_{e \in E(U,W)} x_e$. All vectors will be column vectors. The transpose of a matrix $M$ will be written $M^T$. If $C$ denotes a cycle then $E(C)$ denotes the edges of the cycle and $x(E(C)) := \sum_{e \in E(C)} x_e$.

Our definition of $x$ agrees with the work of Grötschel and Wakabayashi [21, 22] on the clustering problem. Note that it is the opposite of that in literature on the MAXCUT problem, where typically $x_e = 1$ if edge $e$ appears in the cut.

## 2.2 Polyhedral theory for the $k$-way equipartition problem

We discuss the dimension of the $k$-way equipartition problem polytope and give some facet-defining and some valid inequalities for it, restricting our attention to inequalities that are used in the branch-and-cut algorithm described later. For more details, see Mitchell [28]. The polyhedral theory for this problem is related to that of the equipartition problem and the clustering problem.

Brunetta *et al.* [6] have developed a branch-and-cut algorithm for the equicut problem. Their work is based on that of Conforti *et al.* [10, 11], who developed a great deal of polyhedral theory for the equipartition problem.

They proved the following result regarding the dimension of the equicut problem:

**Lemma 1** *([10], Lemma 3.5.) The dimension of the equicut polytope on $2S$ vertices is* $\begin{pmatrix} 2S \\ 2 \end{pmatrix} - 2S$.

Mitchell [28] extended this result to the $k$-way equipartition problem , showing that the dimension of $Q(kS)$ is $d(kS)$ provided $S > 2$, where we define

$$d(q) := \begin{pmatrix} q \\ 2 \end{pmatrix} - q. \tag{1}$$

Among the families of cutting planes that Conforti *et al.* describe for the equipartition problem on a graph with $2S$ vertices are the facet defining *cycle inequalities*:

> *Cycle inequalities ([11], Theorem 6.2):* For every cycle $C$ of length $S + 1$, we have
> $$x(E(C)) \leq S - 1. \tag{2}$$

The cycle inequalities are facet defining for the $k$-way equipartition problem [28].

In the $NP$-hard clustering problem, we are given a set of $p$ observations, each of which possesses $k$ characteristics. The objective is to divide the observations into clusters where the observations within each cluster are similar to one another. For example, the observations could consist of different types of computers, and the characteristics could include the speed of the computer, the amount of RAM of the computer and the size of the hard disk of the computer. There are no *a priori* constraints on the number of clusters or on the number of elements in a cluster. The $k$-way equipartition problem is a version of the clustering problem where all the clusters are required to have the same prescribed size.

Grötschel and Wakabayashi [21, 22] described a cutting plane algorithm for the clustering problem. The set of incidence vectors of feasible clusters with $p$ observations are given by the solutions to the following set of constraints:

$$-x_{ij} + x_{il} + x_{jl} \leq 1 \quad \text{for } 1 \leq i < j < l \leq p \tag{3}$$

$$x_{ij} - x_{il} + x_{jl} \leq 1 \quad \text{for } 1 \leq i < j < l \leq p \tag{4}$$

$$x_{ij} + x_{il} - x_{jl} \leq 1 \quad \text{for } 1 \leq i < j < l \leq p \tag{5}$$

$$x_{ij} = 0 \text{ or } 1, \ 1 \leq i < j \leq p$$

where we interpret $x_{ij} = 1$ to mean that $i$ and $j$ are in the same cluster, and $x_{ij} = 0$ to mean that they are in different clusters. The constraints (3), (4), and (5) are called *triangle inequalities*. Constraint (3) corresponds to the logical condition that if $i$ and $j$ are in different clusters then $l$ can not be in the same cluster as both $i$ and $j$; constraints (4) and (5) have similar interpretations. All these inequalities define facets of the convex hull of the set of feasible solutions to the clustering problem. We used these inequalities as cutting planes in our algorithm for the $k$-way equipartition problem. The triangle inequalities are also presented in [6] and they are well known in the literature for the MAX-CUT problem; see, for example, Barahona and Mahjoub [3].

Other classes of facets for this problem are known, but a complete description of the convex hull is not currently known — see Grötschel and Wakabayashi [21, 22] for more details. We have the following theorem.

**Theorem 1** *([22], Theorem 4.1.) For every pair of nonempty disjoint subsets $U, W \subseteq V$, the 2-partition inequality*

$$x(U, W) - x(U) - x(W) \leq \min\{|U|, |W|\} \tag{6}$$

*defines a facet of the clique partitioning polytope, provided $|U| \neq |W|$.*

There are several families of inequalities that we have used in our cutting plane approach, beyond those discussed in earlier work on the clustering problem or the equipartition problem. Some of these families are defined on the vertices of an equipartition polytope, although they are not facet defining for the equipartition problem. Other families use more than $2S$ vertices. The next four theorems are proved by the author and presented in another paper [28].

The first inequality uses the fact that there is a limit on the number of edges that can be used from any complete subgraph.

**Theorem 2** *Let $U \subseteq V$ with $|U| = pS + q$, with $1 \leq p < k$ and $1 \leq q < S$. The following is a valid inequality:*

$$x(U) \leq \begin{cases} p\begin{pmatrix} S \\ 2 \end{pmatrix} + \begin{pmatrix} q \\ 2 \end{pmatrix} & \text{if } q \geq 2 \\ p\begin{pmatrix} S \\ 2 \end{pmatrix} & \text{if } q = 1. \end{cases} \tag{7}$$

For the equipartition problem, (7) with $p = 1$ is implied by the cycle constraints if $q = 1$ and by the degree constraints if $q > 1$. Nonetheless, this inequality is violated by some points in the LP relaxation of the $k$-way equipartition problem for $k > 2$. For example, if $k = 3$ and $S = 4$, divide the twelve vertices into two equal sets. Set $x_e = 0.6$ for each edge with both endpoints within one set, and take $x_e = 0$ otherwise. This point satisfies the degree constraints, the triangle constraints, and the cycle constraints but it violates (7).

In the next theorem, we give a valid inequality for the equipartition problem that extends to the $k$-way equipartition problem. It is not implied by the constraints considered earlier.

**Theorem 3** *Let $U$ and $W$ be two disjoint subsets of $V$ with $|U| = |W| = S - 1$. The following is a valid inequality:*

$$(S-1)\, x(U) + (S-1)\, x(W) + (S-2)\, x(U, W) \leq (S-2)(S-1)^2. \tag{8}$$

For example, take $S = 4$, so $|U| = |W| = 3$. Let

$$x_e = \begin{cases} \frac{4}{5} & \text{if } e \in E(U) \text{ or } e \in E(W) \\ \frac{11}{45} & \text{if } e \in E(U, W) \end{cases}$$

This point takes the value $18\frac{4}{5}$ for the left hand side of (8), while the right hand side is equal to 18. The point satisfies all the other constraints. It satisfies (7) at equality for the set of all the six vertices.

Another useful inequality for the $k$-way equipartition problem is given in the next theorem. This inequality is implied by the triangle inequalities and the degree constraints when $k = 2$.

**Theorem 4** *Let $U_1 \subseteq V$ and $U_2 \subseteq V$ be two disjoint sets with $|U_1| = S - 3$ and $|U_2| = S + 1$. Let $v$ be a vertex from $V \setminus (U_1 \cup U_2)$. The following is a valid inequality:*

$$(S-2)\, x(v, U_1 \cup U_2) + x(U_2) \leq \frac{3S^2 - 9S + 8}{2}. \tag{9}$$

For example, with $S = 4$, the set $U_1$ is a singleton and $U_2$ contains five vertices. The right hand side of (9) is equal to 10. Let

$$x_e = \begin{cases} 0.5 & \text{if } e \in E(v, U_1 \cup U_2) \\ 0.45 & \text{if } e \in E(U_2) \end{cases}$$

This point violates (9) by 0.5. Only 15 components of $x$ have been specified. If $k \geq 4$, we can choose values for the remaining components so that $x$ satisfies all the earlier inequalities.

The following two theorems consider a subset $U$ of the vertices of cardinality $2S+2$. In each case, some of the edges in $E(U)$ are not included in the left hand side of the inequality.

**Theorem 5** *Let $U \subseteq V$ with $|U| = 2S+2$. Let $\bar{E} \subseteq E(U)$ and let $\bar{G} := (U, \bar{E})$ with vertices $U$ and edges $\bar{E}$. If $\bar{E}$ is such that for any two vertex disjoint copies of $K_S$ in $\bar{G}$ the remaining two vertices are not adjacent in $\bar{G}$, then the following is a valid inequality:*

$$\sum_{e \in \bar{E}} x_e \leq S(S-1). \tag{10}$$

For example, take $|S| = 4$ and consider the set $U$ of ten vertices in Figure 2. The total edge weight of all the edges in $\bar{E}$ is 12.6, while the right hand side of (10) is 12. Provided $k \geq 4$, this arrangement can be extended to a point $x$ that satisfies all the earlier constraints.

**Theorem 6** *Let $U_1 \subseteq V$ and $U_2 \subseteq V$ be two disjoint sets with $|U_1| = S$ and $|U_2| = S+1$. Let $v$ be a vertex from $V \setminus (U_1 \cup U_2)$. The following is a valid inequality:*

$$\begin{aligned} S\,x(U_1) + (S-1)\,x(U_2) + (S-1)\,x(U_1, U_2) + (S-1)\,x(v, U_1) \\ \leq \frac{S(S-1)(2S-1)}{2}. \end{aligned} \tag{11}$$

For example, take $|S| = 4$, so $|U_1| = 4$ and $|U_2| = 5$. Let

$$x_e = \begin{cases} 0.8 & \text{if } e \in E(U_1), \text{ so } x(U_1) = 4.8 \\ 0.6 & \text{if } e \in E(U_2), \text{ so } x(U_2) = 6 \\ 0.3 & \text{if } e \in E(v, U_1), \text{ so } x(v, U_1) = 1.2 \\ \frac{3}{50} & \text{if } e \in E(U_1, U_2), \text{ so } x(U_1, U_2) = 0.6 \end{cases}$$

If $k \geq 4$, this can be extended to a solution $x$ that satisfies all the earlier constraints. We get a violation of 0.6 in (11), with a left hand side of 42.6 and a right hand side of 42.

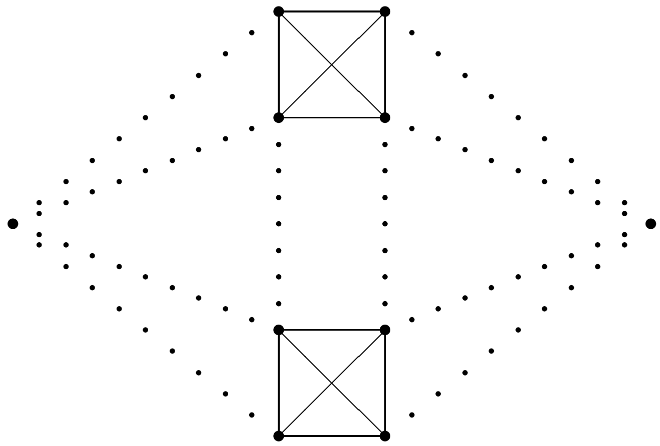

Figure 2: Example for Theorem 5. Solid edges have $x_e = 0.8$, dotted edges have $x_e = 0.3$. Edges shown are all in $\bar{E}$. Violation is 0.6.

### 2.3 Branch-and-cut

Our branch-and-cut algorithm for the $k$-way equipartition problem is described in [28]. It is as follows:

1. **Initialize**: The feasible region for the initial linear programming relaxation is $\bar{Q}(kS)$. The initial incumbent integer feasible solution is a random assignment of teams to divisions.

2. **Solve** the current **LP relaxation**.

3. If the **gap** between the value of the LP relaxation and the value of the incumbent integer solution is sufficiently small, **STOP** with optimality.

4. Use a variant of the Kernighan-Lin heuristic [25] to **round** the fractional solution to the LP relaxation into a good integer feasible solution. Replace the current incumbent solution with this solution if it is an improvement.

5. Use the **separation routine** (defined below) to find violated cutting planes.

6. If enough violated cutting planes with sufficiently large violation are found, **return to Step 2**. Otherwise, call the branch-and-cut solver in the commercial solver CPLEX to attempt to verify that the incumbent integer solution is optimal and then **STOP**.

An interior point method was used in step 2 to approximately solve the LP relaxations, with the tolerance gradually tightened at each iteration. For more details on interior point cutting plane algorithms of this type, see [27]. In all the experiments considered later, the data was integer. Thus, the algorithm terminates in Step 3 if the gap is smaller than one.

In step 6, we provide the set of constraints from the final LP relaxation, together with the integrality restrictions, to the branch-and-cut solver. Since this set of constraints does not contain all of the triangle inequalities, the solution to this integer program may not be feasible in the $k$-way equipartition problem. If the integer solution returned by the branch-and-cut solver has the same value as the solution found in step 4 then we know that it is feasible, and this confirms that the incumbent integer solution is optimal. This was the case in all the experiments described later.

The separation routine consists of the following parts:

5–i Search for **triangle inequalities** (3), (4), and (5), using complete enumeration. Inequalities are bucket sorted by the size of the violation. Starting with inequalities in the most violated bucket, a subset are added, ensuring that no two of these added inequalities share an edge. There is an upper bound on the number of constraints added; further, the violation of the last constraint added is restricted to be no smaller than a certain multiple of the violation of the first constraint added.

5–ii A routine similar to that described in Grötschel and Wakabayashi [21, 22] is used to find violated **2-partition inequalities** (6), with $|U| = 1$ or 2 and $|W| \geq 3$.

5–iii In order to find violated **cycle inequalities** (2), the algorithm builds up cliques of vertices using a breadth first search approach, based on adding vertices $v$ to the test set $U$ if $x_{uv}$ is sufficiently large for one or more vertices in $U$. Cliques of size $S+1$ with weight $x(U) > \frac{S(S-1)}{2}$ must contain violated cycle inequalities, so if we find a violated clique inequality of this form, we check the corresponding cycle inequalities.

5–iv **Clique inequalities** of the form (7) with $p = 1$ and $q = 2$ and of the form (8) are checked using the same breadth first search routine as in Step 5–iii.

5–v **Cycle inequalities** (2) are searched for directly, using a depth first search approach.

5–vi The final part of the separation routine is to add **nine specific inequalities** of the forms (7), (9), (10), and (11), chosen especially for realignment problems with these cities, based on examination of the solutions to the LP relaxations.

If one step does not produce enough violated cutting planes or if the cutting planes are not sufficiently violated, the separation routine moves to the next step. Otherwise, it returns to the main algorithm.

For the problems considered in §3, we moved to Step 5–ii if fewer than 20 triangle inequalities had been added, and we moved to each of Steps 5–iii, 5–iv, and 5–v only if fewer than five constraints had been added by the earlier steps. We only moved to Step 5–vi if no constraints were added in Steps 5–i to 5–v. At the beginning, over a hundred violated triangle inequalities may be found, some violated by almost one. As the algorithm proceeds, the number violated shrinks and the maximum violation shrinks. The number added depends on the maximum violation, as well as on the number violated. This is controlled by limiting the number of buckets that are examined using a formula similar to that discussed in [27]. Limiting the number of constraints added speeds up the solution of the LP relaxations.

# 3 Computational results

The situation where any team can be placed in a division with any other team is discussed in §3.1. The optimal solution has a sum of intradivisional travel distances of 27957km.

Such an alignment would be unacceptable to the NFL at present, due to the desire to maintain historical rivalries. Currently, the NFL is divided into two conferences. Prior to the 2002 season, the AFC had 16 teams and the NFC had 15 teams. The Houston Texans were placed in the AFC and the Seattle Seahawks were moved to the NFC. The optimal alignment of the NFC and AFC for this choice of teams is 18% better than that chosen by the NFL, because various additional historical rivalries were maintained. When the conferences are specified, each conference can be aligned separately, so the problem becomes a lot easier to solve. We also looked at the additional restrictions in the setting of an otherwise unrestricted problem, that is, without dividing the teams into two conferences. Some of these problems are harder to solve than the original unrestricted problem, because the restrictions force distant cities to be in the same division, which affects the neighbours of these distant cities. Problems that respect historical rivalries are the subject of §3.2.

The locations of the cities are taken from US Geographical Survey data. There are two teams from New York City, playing at the same stadium, so the distance between these two teams is zero. There are 32 teams, divided into eight divisions each of four teams, so $|V| = 32$, $k = 8$, and $S = 4$. The city-to-city distance matrix used in this paper, along with more information about its calculation, is available from

http://www.rpi.edu/~mitchj/generators/realign/

This website also contains the solutions for various versions of the problem.

The cutting plane algorithm was written in Fortran 77 and implemented on a Sun Enterprise workstation. CPLEX was run on a Sun 20/71 workstation. It took an mps file as input, containing a description of the final LP relaxation as well as integrality restrictions.

## 3.1 The unrestricted problem

In this section, we assume that any team can be assigned to any division. The minimum sum of intradivisional travel distances is 27957km, achieved by the arrangement illustrated in Figure 1 and listed in Table 1. The solution to the final relaxation before invoking branch and bound is shown in Figure 3. The numbers in Figure 3 correspond to those given in Table 1.

| Division | Teams | | | |
|---|---|---|---|---|
| 1 | Atlanta(1) | Carolina(5) | Cincinnati(7) | Tennessee(18) |
| 2 | Baltimore(2) | Buffalo(4) | Pittsburgh(25) | Washington(31) |
| 3 | New England(3) | New York Giants(20) | New York Jets(21) | Philadelphia(23) |
| 4 | Chicago(6) | Cleveland(8) | Detroit(11) | Indianapolis(12) |
| 5 | Dallas(9) | Denver(10) | Arizona(24) | Houston(32) |
| 6 | Jacksonville(13) | Miami(15) | New Orleans(19) | Tampa Bay(30) |
| 7 | Kansas City(14) | Green Bay(16) | Minnesota(17) | St Louis(26) |
| 8 | Oakland(22) | San Diego(27) | San Francisco(28) | Seattle(29) |

Table 1: Optimal alignment for the unrestricted problem

The cutting plane code required 57 seconds and CPLEX required a further 57 seconds to confirm that the solution was optimal.

Five of the optimal divisions appear reasonably clearly in the optimal solution to the LP relaxation in Figure 3. Even within these divisions, many of the edge variables

**Cutting plane solution:**

Lower bound: 27938 km

Final gap: 0.07%

Key:

$0.90 < x_e \leq 1.0$

$0.65 < x_e \leq 0.90$

$0.40 < x_e \leq 0.65$

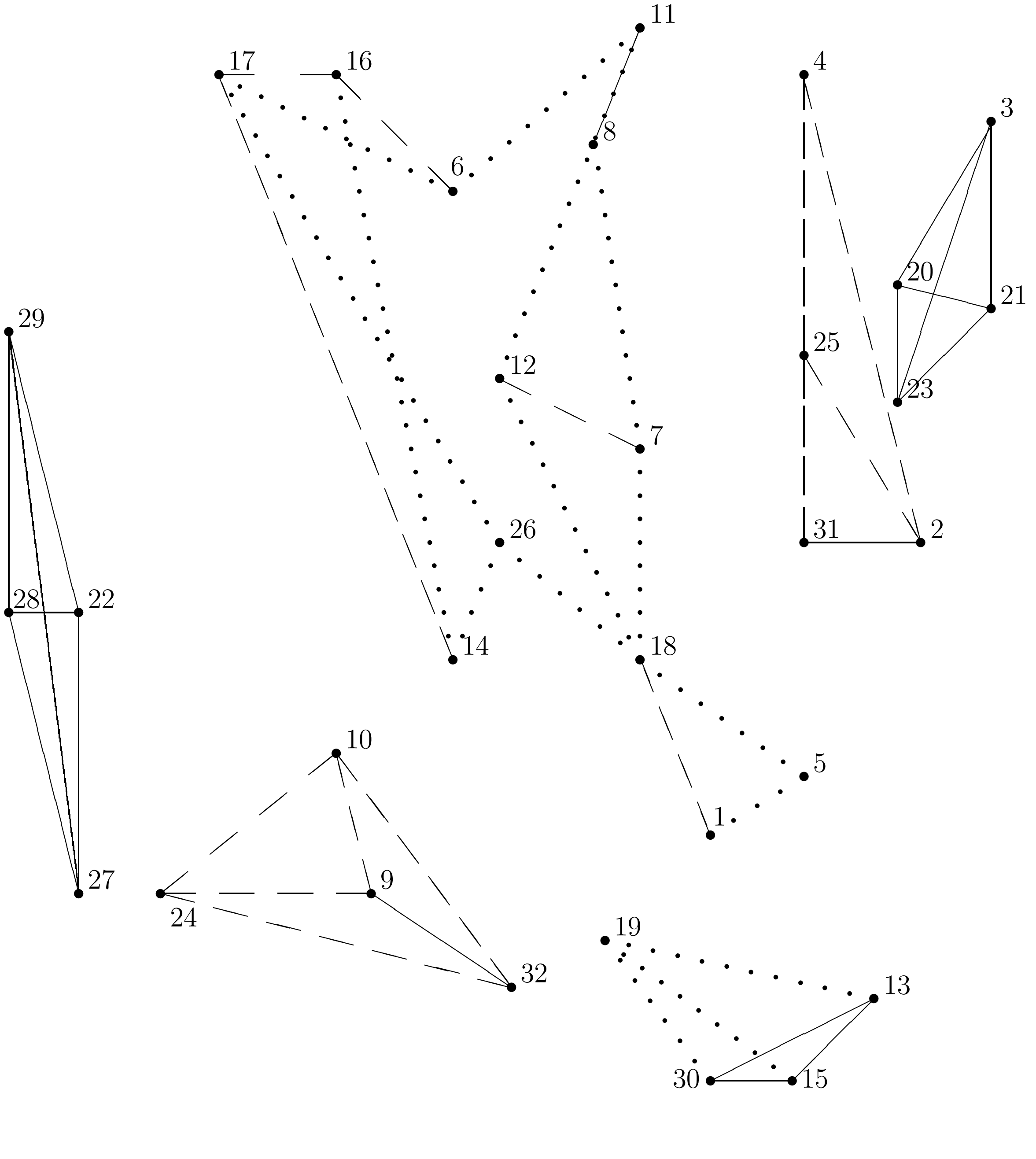


Figure 3: Final LP cutting plane solution for unrestricted problem

take pronounced fractional values. The cutting plane code added 323 constraints altogether in 36 stages. These constraints comprised 130 triangle constraints, 43 cycle inequalities, 23 inequalities of the form (8), 76 2-partition inequalities, 42 clique inequalities, and the nine additional inequalities mentioned in Step 5–vi. The branch and bound routine in CPLEX confirmed optimality in two nodes of the tree.

The number of possible alignments of the teams is

$$\frac{1}{8!}\binom{32}{4}\binom{28}{4}\binom{24}{4}\binom{20}{4}\binom{16}{4}\binom{12}{4}\binom{8}{4} \approx 6 \times 10^{24},$$

which is too large for enumeration to be an attractive option.

### 3.2 Traditional rivalries

There are several traditional rivalries in the NFL that have developed over many years from teams being in the same division. For example, the Washington Redskins and Dallas Cowboys, the Kansas City Chiefs and Oakland Raiders, and the Chicago Bears, Detroit Lions, Green Bay Packers, and Minnesota Vikings are great rivals. Some of these rivalries are not maintained by the optimal arrangement given in §3.1. In this section, we examine realignments that maintain some of these rivalries, and we also look at alignments that closely correspond to the current separation of teams into the AFC and NFC. The solution chosen by the NFL is presented in this section, and we show how it corresponds to an optimal solution to a constrained version of our formulation of the problem.

Prior to the inclusion of the Houston Texans, the 31 teams in the NFL were split into two conferences. The AFC contained 16 teams and there were 15 teams in the NFC. The NFL placed Houston in the AFC and moved Seattle to the NFC. Their chosen alignment is given in Table 2 and illustrated in Figures 4 and 5. This has a sum of intradivisional travel distances of 50480km, which is 80% greater than the alignment given in §3.1.

The NFL choice is designed to keep Washington and Dallas in the same division, to keep Minnesota, Detroit, Green Bay, and Chicago together, and to keep Buffalo, Miami, New England, and the New York Jets in a single division. Imposing these conditions and optimizing is sufficient to generate the NFL choice. The number of alignments of the AFC that preserve the AFC East is 5775. The number of NFC alignments meeting the requirements is 1575.

Perhaps surprisingly, if the requirement that Minnesota be in the same division as Green Bay, Detroit, and Chicago is not imposed then the optimal solution for the

Figure 4: The realignment chosen by the NFL for the NFC.

Figure 5: The realignment chosen by the NFL for the AFC.

| NFC | | | | |
|---|---|---|---|---|
| South | Atlanta(1) | Carolina(5) | New Orleans(19) | Tampa Bay(30) |
| North | Chicago(6) | Detroit(11) | Green Bay(16) | Minnesota(17) |
| West | Arizona(24) | St Louis(26) | San Francisco(28) | Seattle(29) |
| East | Dallas(9) | NY Giants(20) | Philadelphia(23) | Washington(31) |
| AFC | | | | |
| East | New England(3) | Buffalo(4) | Miami(15) | NY Jets(21) |
| North | Baltimore(2) | Cincinnati(7) | Cleveland(8) | Pittsburgh(25) |
| West | Denver(10) | Kansas City(14) | Oakland(22) | San Diego(27) |
| South | Indianapolis(12) | Jacksonville(13) | Tennessee(18) | Houston(32) |

Table 2: Realignment chosen by the NFL

NFC is to swap the positions of Minnesota and St. Louis. This reduces the total intradivisional travel distance by 791km.

Given the selection of teams for the AFC and NFC, the optimal values without the additional conditions can be calculated.

- If the AFC and NFC are optimized with no additional restrictions, the optimal value is 41152km. In the optimal solution for the NFC, Detroit moves from the North to the East, Dallas moves from the East to the West, and St. Louis moves from the West to the North. In the optimal solution for the AFC, Miami moves from the East to the South, Indianapolis moves from the South to the North, and Baltimore moves from the North to the East. This realignment is shown in Figures 6 and 7.

- If the only restriction is that Dallas and Washington be in the same division, the optimal value is 45359km. Thus, this restriction adds 4208km.

- If, in addition, the composition of the AFC East is specified, the optimal value increases to 49689km, an increase of 4330km. This gives the NFL's choice for the AFC.

- Finally, specifying the makeup of the NFC North gives the NFL choice as the optimal solution, with value 50480km.

A team other than Seattle could have been moved from the AFC to the NFC. Given the three additional conditions imposed by the NFL, the best choice is to move

Figure 6: The optimal realignment for the NFC.

Figure 7: The optimal realignment for the AFC.

the Kansas City Chiefs. The optimal solution exchanges these two teams in Table 2, giving a value of 48296km. This is 4.3% less than the chosen solution.

If the Houston Texans had been placed in the NFC then the optimal value, with the additional conditions, is 48499km. The NFC alignment is as in Table 2, with Houston replacing Seattle, but the AFC alignment is quite different — see Table 3.

| Division | Teams | | | |
|---|---|---|---|---|
| AFC 1 | New England(3) | Buffalo(4) | Miami(15) | NY Jets(21) |
| AFC 2 | Baltimore(2) | Cleveland(8) | Jacksonville(13) | Pittsburgh(25) |
| AFC 3 | Denver(10) | Oakland(22) | San Diego(27) | Seattle(29) |
| AFC 4 | Cincinnati(7) | Indianapolis(12) | Kansas City(14) | Tennessee(18) |

Table 3: Optimal AFC alignment with additional restrictions, if Houston placed in NFC

If all three restrictions are imposed *without* assigning each team to a particular conference, the optimal solution has value 37126km, with the alignment given in Table 4.

| Division | Teams | | | |
|---|---|---|---|---|
| 1 | Atlanta(1) | Carolina(5) | Jacksonville(13) | Tampa Bay(30) |
| 2 | Chicago(6) | Detroit(11) | Green Bay(16) | Minnesota(17) |
| 3 | Denver(10) | Kansas City(14) | Arizona(24) | St Louis(26) |
| 4 | Dallas(9) | New Orleans(19) | Washington(31) | Houston(32) |
| 5 | New England(3) | Buffalo(4) | Miami(15) | NY Jets(21) |
| 6 | Baltimore(2) | NY Giants(20) | Philadelphia(23) | Pittsburgh(25) |
| 7 | Cincinnati(7) | Cleveland(8) | Indianapolis(12) | Tennessee(18) |
| 8 | Oakland(22) | San Diego(27) | San Francisco(28) | Seattle(29) |

Table 4: Optimal alignment with three restrictions

None of the experiments in this section required more than 5 seconds, and they were all solved in the cutting plane phase. Even the alignment given in Table 4 was found in 5 seconds, requiring 18 stages and the addition of 151 cutting planes to prove optimality. There are restrictions that can be imposed that make the problem harder to solve. One example is to impose the four requirements that Washington and Dallas be in the same division, that Miami and the New York Jets be in the same division (without further requirements on Buffalo and New England), that the

New York Giants and the New York Jets be in different divisions, and that Kansas City and Oakland be in the same division, with no assignment to conferences. The cutting plane code solved this problem to optimality in 84 seconds. If only the Washington/Dallas restriction is imposed, the cutting plane code leaves a duality gap of 169, in 17 seconds. This gap is closed by CPLEX in 2 nodes of the branch and bound tree.

# 4 Conclusions

The minimum possible value for the total intradivisional travel distance is 27957km (Table 1). Requiring that the alignment closely correspond to the current conferences gives values at least 40% larger than this. The alignment chosen by the NFL has a value of 50480km (Table 2), which is 80% larger. This choice maintains the current separation of the teams into the AFC and NFC, apart from moving Seattle to the NFC and placing Houston in the AFC. If the 32 teams are divided into two conferences in this manner, the smallest possible value is 41152km, so the NFL choice is 23% larger. The NFL choice is optimal subject to the chosen composition of the conferences and the three additional constraints of Washington and Dallas being in the same division, the AFC East being specified, and the NFC North being specified.

Advantages of the realignment proposed in Table 1 include the creation of new rivalries based on geographic proximity, such as San Francisco and Oakland, Washington and Baltimore, Dallas and Houston, Kansas City and St. Louis, the two New York City teams, and the three Florida teams together with New Orleans. At a slight extra cost, these rivalries can be created together with maintaining the traditional rivalry of Chicago, Detroit, Green Bay, and Minnesota. For an additional small cost, the two New York City teams can be kept in separate divisions.

The alignment given in Table 1 ensures that no teams separated by more than one time zone play in the same division. Any of the alignments that are based on the current NFC and AFC result in San Francisco being in the same division as at least one team that is two time zones away.

In baseball, the American League and National League play under different rules, making it politically difficult to perform such a dramatic realignment as proposed here for the NFL. The NFL has no such restriction. The National Hockey League and the National Basketball Association have both recently realigned along more geographical lines.

## Acknowledgements

The author would like to thank the referees and the editor for their helpful and constructive comments, and Dina Williams for help with the graphics.

## References

[1] I. Adler, A. L. Erera, D. S. Hochbaum, and E. V. Olinick. Baseball, optimization, and the world wide web. *Interfaces*, 32(2):12–22, 2002.

[2] S. Areibi and A. Vannelli. Advanced search techniques for circuit partitioning. In P. M. Pardalos and H. Wolkowicz, editors, *Quadratic Assignment and Related Problems*, volume 16 of *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, pages 77–96. AMS, 1994.

[3] F. Barahona and A. R. Mahjoub. On the cut polytope. *Mathematical Programming*, 36:157–173, 1986.

[4] J. C. Bean and J. R. Birge. Reducing travel costs and player fatigue in the NBA. *Interfaces*, 10:98–102, 1980.

[5] S. J. Benson, Y. Ye, and X. Zhang. Solving large-scale sparse semidefinite programs for combinatorial optimization. *SIAM Journal on Optimization*, 10(2):443–461, 2000.

[6] L. Brunetta, M. Conforti, and G. Rinaldi. A branch-and-cut algorithm for the equicut problem. *Mathematical Programming*, 78:243–263, 1997.

[7] V. W. C. Chan, P. C. H. Chan, and M. Chan. Multiple layers of CMOS integrated circuits using recrystallized silicon film. *IEEE Electron Device Letters*, 22(2):77–79, February 2001.

[8] S. Chopra. The graph partitioning polytope on series-parallel and 4-wheel free graphs. *SIAM Journal of Discrete Mathematics*, 7:16–31, 1994.

[9] S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59:87–115, 1993.

[10] M. Conforti, M. R. Rao, and A. Sassano. The equipartition polytope I: Formulations, dimension and basic facets. *Mathematical Programming*, 49:49–70, 1990.

[11] M. Conforti, M. R. Rao, and A. Sassano. The equipartition polytope II: Valid inequalities and facets. *Mathematical Programming*, 49:71–90, 1990.

[12] J. Domingo-Ferrer and J. M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Engineering*, 14(1):189–201, 2002.

[13] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17:420–425, 1973.

[14] F. C. Duckworth and A. J. Lewis. A fair method for resetting the target in interrupted one-day cricket matches. *Journal of the Operational Research Society*, 49(3):220–227, 1998.

[15] J. Falkner, F. Rendl, and H. Wolkowicz. A computational study of graph partitioning. *Mathematical Programming*, 66:211–239, 1994.

[16] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. A. Wolsey. The node capacitated graph partitioning problem: formulations and valid inequalities. *Mathematical Programming*, 74:247–267, 1996.

[17] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. A. Wolsey. The node capacitated graph partitioning problem: a computational study. *Mathematical Programming*, 81:229–256, 1998.

[18] M. Fischetti and J. J. Salazar. Models and algorithms for the 2-dimensional cell suppression problem in statistical disclosure control. *Mathematical Programming*, 84:283–312, 1999.

[19] A. Frieze and M. Jerrum. Improved approximation algorithms for max $k$-cut and max bisection. In *Proceedings of the Fourth IPCO Conference*, pages 1–13, 1995.

[20] M. R. Garey and D. S. Johnson. *Computers and intractibility: a guide to the theory of NP-completeness*. Freeman, 1979.

[21] M. Grötschel and Y. Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45:59–96, 1989.

[22] M. Grötschel and Y. Wakabayashi. Facets of the clique partitioning polytope. *Mathematical Programming*, 47:367–387, 1990.

[23] S. E. Karisch and F. Rendl. Semidefinite programming and graph equipartition. In P.M. Pardalos and H. Wolkowicz, editors, *Topics in Semidefinite and Interior-Point Methods*, volume 18 of *Fields Institute Communications*, pages 77–95. AMS, 1998.

[24] G. Karypis and V. Kumar. Multilevel $k$-way hypergraph partitioning. *VLSI Design*, 11(3):285–300, 2000.

[25] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

[26] A. Lisser and F. Rendl. Telecommunication clustering using linear and semidefinite programming. Technical report, Institüt für Mathematik, Universität Klagenfurt, A – 9020 Klagenfurt, Austria, November 2000.

[27] J. E. Mitchell. Computational experience with an interior point cutting plane algorithm. *SIAM Journal on Optimization*, 10(4):1212–1227, 2000.

[28] J. E. Mitchell. Branch-and-cut for the $k$-way equipartition problem. Technical report, Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180, January 2001.

[29] G. L. Nemhauser and M. A. Trick. Scheduling a major college basketball conference. *Operations Research*, 46:1–8, 1998.

[30] New century, new look for NFL. News release, May 22, 2001. The National Football League, http://www.nfl.com/news/010522realignment.html.

[31] R. Saltzman and R. M. Bradford. Optimal realignments of the teams in the National Football League. *European Journal of Operational Research*, 93(3):469–475, 1996.

[32] M. A. Trick. A schedule-and-break approach to sports scheduling. In E. Burke and W. Erben, editors, *PATAT2000*, volume 2079 of *Springer Lecture Notes in Computer Science*, pages 242–253. Springer-Verlag, 2000.

[33] H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Applied Mathematics*, 96/97:461–479, 1999.

[34] Y. Ye. A .699-approximation algorithm for max-bisection. *Mathematical Programming*, 90(1):101–111, 2001.