ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Instituto de Ciencias Matemáticas
Ingeniería en Logística y Transporte

TEMA

Desarrollo de una aplicación para calendarizar el Campeonato Ecuatoriano de Fútbol profesional por medio de una aproximación heurística utilizando Programación Entera.

TESIS DE GRADO

Previo a la obtención del Título de:
Ingeniero en Logística y Transporte

Presentado por
Jorge Luis Morales Espinoza

Guayaquil-Ecuador
2012

Dedicatoria

Este trabajo es dedicado a Dios por darme salud, vida y mucha sabiduría, a mis padres Jorge y Jannet, los cuales siempre me brindaron todo su apoyo incondicional, junto a muchos consejos constantes y a mi hermana Karla. Los amo con todas mis fuerzas, gracias por apoyarme en mis decisiones durante todo este tiempo universitario.

Agradecimiento

Le doy gracias a Dios, por darme vida junto a mis padres, los cuales siempre me brindaron su apoyo, amor y mucha motivación para llegar alcanzar y culminar esta meta, a mis grandes amigos Roberto, Jorge, y Fede, quienes siempre estuvieron apoyandome en momentos difíciles con sus consejos, adicionalmente le agradezco al Ing. Xavier Cabezas por ayudarme en esta investigación y por su apoyo profesional junto al Ing Daniel Agreda quien me abrió las puertas en la actividad profesional en su empresa, finalmente a cada compañero que tuve en el transcurso de esta etapa como: Irvin, Julio, Gio, Allan, Katherine entre otros; que gracias a su excelente amistad pasamos momentos muy agradables.

Declaración Expresa

La responsabilidad del contenido de esta Tesis de Pregrado, me corresponde exclusivamente; y el patrimonio intelectual de la misma al ICM (Instituto de Ciencias Matemáticas) de la ESPOL (Escuela Superior Politécnica Del Litoral).

Jorge Luis Morales Espinoza

TRIBUNAL DE GRADUACIÓN

Ing. Guillermo Baquerizo Palma
PRESIDENTE DEL
TRIBUNAL

M.Sc. Xavier Cabezas García
DIRECTOR DE TESIS

M.Sc. Fabricio Echeverria Briones
VOCAL

Índice general

Índice de figuras

Índice de tablas

Resumen

En este trabajo se presenta una aplicación de las técnicas de Investigación de Operaciones, el cual consiste en el desarrollo de una aplicación para calendarizar el Campeonato Ecuatoriano de Fútbol profesional por medio de una aproximación heurística utilizando Programación Entera, este documento presenta mucho interés personal en los Scheduling Problems, debido a que involucra un procedimiento heurístico con la programación lineal entera (ILP).

Dentro del documento se presenta restricciones basadas a las características actuales del campeonato con unas variantes, por ejemplo el aparecimiento de los recursos de las TVs, la cual se considera la más importante.

Este procedimiento para llegar a una solución del problema, está dividido en tres fases o partes, las cuales se las detallan en el contexto del documento. Se utilizo como partes de las herramientas computacionales los softwares GAMS y Wolfram Mathematica 7.0. Mathematica es utilizado para la generación de datos de entrada, las cuales recibe GAMS utilizando como solver XPRESS para poder ejecutar el modelo matemático propuesto para cada fase. Una vez culminado las tres fases se procedió a revisar los resultados encontrados para luego comparar con el calendario original.

Este trabajo tiene la posibilidad de generar no solo un calendario, sino varias alternativas, este método se lo encuentra explicado en los capítulos posteriores. La información

obtenida de las características del campeonato, fueron adquiridas por el personal de la Federación Ecuatoriana de Fútbol (FEF), los cuales son los organizadores del campeonato, con el objetivo primordial de que se llegase a plasmar este método para la elaboración de campeonatos posteriores.

Capítulo 1

INTRODUCCIÓN

1.1. Problemas Combinatorios

Un problema de optimización combinatoria es especificado por un conjunto P de las instancias que estan asociadas a la solución del problema, las cuales son de tipología de minimización o maximización.

Por *problema* se entiende, una pregunta general a ser contestada, teniendo muchas variables y parámetros con valores no especificados.

La *instancia* del problema, a su vez se entiende como un caso particular del problema, con valores especificos para todos los parámetros y variables. Dentro de un espacio combinatorio, es un par (V, f), donde V es el conjunto de soluciones factibles para la instancia y f representa la función objetivo o de costo, la cual se le asigna a cada solución factible un valor dentro de los reales, el objetivo fundamental es hallar una solución globalmente óptima, de tal forma que para un $s \in V$ satisfaga $f(s) \leq f(i), \forall i \in V$. Esta definición corresponde a un problema de minimización, en el caso de un problema de maximización, se tiene un planteamiento similar, excepto que la solución globalmente óptima seria: $f(s) \geq f(i), \forall i \in V$.

Es importante recalcar que las instancias no son dadas explícitamente, sino más bien, se expresan en un conjunto de valores relacionados directamente con los parámetros que han sido usados para modelar el problema. En algunos casos, además de encontrar una solución óptima, debemos hallar una solución que satisfaga un conjunto de restricciones dentro del espacio de solución, es decir, esta podría no ser válida, aunque sea la mínima o máxima globalmente.

Tomando en cuenta posibles restricciones que puedan presentarse dentro de un problema combinatorio, una *instancia* se la definiria como una terna (S, f, θ) donde S es el conjunto de soluciones candidatas para la instancia, θ es un conjunto de restricciones y f es la función objetivo o de costo, la cual asigna a cada solución candidata un valor dentro de los reales $f : S \Rightarrow \Re$. Se denomina soluciones factibles, a las soluciones que pertenezcan a $\hat{S}$ de manera que $\hat{S} \subseteq S$ y que cumplan con todas las restricciones dentro de θ.

El objetivo es hallar una solución globalmente óptima, de tal forma $s \in \hat{S}$ satisfaga $f(s) \leq f(i), \forall i \in \hat{S}$ para el caso de un problema de minimización, en presentarse un problema de maximización se tiene un planteamiento similar, en la cual la solución globalmente optima seria: $f(s) \geq f(i), \forall i \in \hat{S}$.

1.2. Metodología de las soluciones para problemas Combinatorios

1.2.1. Definición de Heurística y Metaheurística

El calificativo *heurístico* dentro de Inteligencia Artificial (IA), se emplea de forma muy simple, es decir, para aplicarlo a todos aquellos aspectos que tiene que ver con el empleo de conocimiento en la realización dinámica de tareas.

Se habla de *heurística* para referirse a una técnica o método inteligente de realizar alguna tarea que no es producto de un riguroso análisis formal, sino de conocimiento experto de la misma; este término, se lo usa más que todo para referirse a un determinado procedimiento, la cual trata de otorgar diferentes soluciones a un problema con un buen rendimiento, en base a la calidad de las soluciones y a los distintos recursos empleados.

En un problema de optimización, se califica de *heurístico* a un procedimiento para el que se tiene un alto grado de confianza en que encuentran soluciones de alta calidad con un costo computacional razonable, aunque no se garantice su optimalidad o su factibilidad, e incluso, en algunos casos, no se llegue a establecer lo cerca que se está de dicha situación.

En optimización matemática, se usa el término *heurístico* en contraposición a exacto, que se aplica los procedimientos a los que se les exige que la solución aportada sea óptima y factible. Además, es usual aplicar este término, cuando utilizando el conocimiento que se tiene del problema, se realizan modificaciones en el procedimiento de solución del problema, que aunque no afecta a la complejidad del mismo, mejoran el rendimiento en su comportamiento práctico.

De acuerdo a la literatura encontrada de esta terminología, puede encontrar dos tipos de heurísticas como:

- **Heurísticas Específicas.-** Estas deben ser diseñadas a próposito para cada problema, utilizando toda la información disponible y el análisis teórico del modelo. Suele tener un rendimiento significativamente más alto que las heurísticas generales.

- **Heurísticas Generales.-**Estas presentan ventajas como la sencillez, adaptabilidad o robustez.

El término *Mateherística* se obtiene de anteponer a *heurística* el sufijo "meta" que significa "más allá" o "a un nivel superior". Las *metaheurísticas* son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento. Las *metaheurísticas* se refieren al diseño de los tipos fundamentales de procedimientos heurísticos de solución de un problema de optimización.

Las principales *metaheurísticas* se refieren a métodos de relajación, procesos constructivos, búsquedas por entornos y procedimientos evolutivos.

- *Las metaheurísticas de relajación* se refieren a procedimientos de resolución de problemas que utilizan relajaciones del modelo original (es decir, modificaciones del modelo que hacen al problema más fácil de resolver), cuya solución facilita la solución del problema original.

- *Las metaheurísticas constructivas* se orientan a los procedimientos que tratan de la obtención de una solución a partir del análisis y selección paulatina de las componentes que la forman.

- *Las metaheurísticas de búsqueda* guían a procedimientos que usan transformaciones o movimientos para recorrer el espacio de soluciones alternativas y explorar las estructuras de entornos asociadas.

- *Las metaheurísticas evolutivas* están enfocadas a los procedimientos basados en conjunto de soluciones que evolucionan sobre el espacio de soluciones.

Capítulo 2

ESTADO DEL ARTE

2.1. Antecedentes

Dentro de la literatura de la *Programación Matemática*, encontramos mucha información sobre *Problemas de Calendarización*, de acuerdo a las necesidades que se presentan día a día dentro de las empresas nacionales e internacionales.

Los *Problema de Calendarización* han ido evolucionando proporcionalmente con los requerimientos de las personas, se entiende por *calendarización*, a una programación de actividades, tareas u horarios que depende del factor tiempo, y en algunos casos de los recursos que se posee inicialmente.

Estos tipos de problemas, han sido definido de una manera muy simple por parte de Jin-Kao Hao and Zhipeng Lu que lo describieron de la siguiente manera:

Definición 2.1.1. *[Calendarización, Jin-Kao Hao and Zhipeng Lu [18]] Asignar un número de eventos, cada uno con ciertas características, a un número limitado de recursos sujeto a restricciones.*

Definición 2.1.2. *[Calendarización, Chambers L. [23]] Se puede definir como aquello que describe, dónde y cuándo las personas y los recursos deben estar en un instante dado, sujeto a restricciones.*

Debido a estas definiciones, muchas veces la programación de horarios o calendarización de horarios, la secuenciación de tareas y la calendarización de actividades, se usan como sinónimos; sin embargo puede haber algunas diferencias entre estas, por ejemplo:

Una calendarización de horarios, nos muestra cuando ciertos eventos van a ocurrir, y no implica directamente a la asignación de recursos, así puede servirnos de ejemplo la calendarización de los horarios de transporte público, esta nos muestra cuando los viajes hacia una ruta o rutas van a efectuarse, y nada nos dice sobre los buses o choferes con los que se efectuará dichos viajes. La asignación de vehículos y choferes forma parte de la calendarización de actividades.

En un sentido más amplio, el objetivo general de *calendarización*, es el de resolver problemas prácticos, relacionados con la asignación de recursos dentro de un espacio de tiempo, el cual está sujeto a una serie de restricciones, haciendo uso o desarrollando todas las herramientas que sean necesarias.

En este trabajo se detalla ciertas caracteristicas de problemas particulares, que han sido de esta tipología *"Calendarización"* para poder interpretar el significado de ciertos términos utilizados en las definiciones anteriores 2.1.1 y 2.1.2.

La *Calendarización* se ha visto reflejada en muchos procesos industriales, como por ejemplo en sistemas de producción, el diseño de circuitos integrados, logística, las comunicaciones, entre otros. Esto demuestra la importancia de estudiar este tipo de problemas y de analizar métodos de solución que nos proporcionen soluciones aproximadas en tiempos razonables.

2.2. Problemas con Calendarización de Trabajos JSSP

Uno de los problemas con mayor relevancia en la industria especialmente en el área de manufactura y la Optimización, es el «Problema de Calendarización de Trabajos» mejor conocido *JSSP* (por sus siglas en inglés, *Job Shop Scheduling Problem*), debido a la administración, así como el manejo eficiente de los recursos, es de vital importancia en las empresas. Debido a su complejidad, *JSSP* ha sido uno de los problemas más estudiados durante las últimas cuatro décadas, motivo por lo que es el problema que ha obtenido más avances en el área de la Calendarización, y sirve como referencia para comparar las técnicas empleadas en diversos problemas clasificados como difíciles, como por ejemplo el caso del Agente Viajero, entre otros. Además ha sido clasificado dentro de la Teoría de la Complejidad como NP-Duro [24], también se conoce como uno de los problemas más difíciles de resolver en esta clasificación.

En el trabajo presentado por Beatriz Pérez Rojas [4] se abordó la solución del *Pro-*

blema de la Calendarización de Trabajos con Transferencia Cero o Flujo Continuo. Ella utilizó dos de las heurísticas más efectivas que existen actualmente «*Heurística de CHANDRASEKHARAN RAJENDRAN [5]*» y la «*Heurística NEH*» adicionalmente un algoritmo genético que utiliza cromosomas no binarios, cruza de subsecuencias, mutación inversa y alternación de generaciones.

2.2.1. Heurística de Chandrasekharan Rajendran

Esta heurística fue diseñada por Chandrasekharan [5] utilizando el concepto de retardo mínimo de la primera máquina entre el trabajo inicial i y k, con la condición de transferencia cero entre cada uno. Además de las variables ya descritas, se utiliza σ, el conjunto de trabajos que ha sido calendarizado e $[i]$, el trabajo que se encuentra en la i-ésima posición de una secuencia o arreglo. La duracion de la secuencia se calcula con:

$$M = \sum_{i=2}^{n} d_{[i-1][i]} + \sum_{j=1}^{m} t_{[n]j} \tag{2.1}$$

Un análisis de la ecuación 2.1 revela que la duración de los trabajos adyacentes que se encuentran dentro de la secuencia y el tiempo de procesamiento del trabajo que se encuentra en la última posición de la secuencia, tienen influencia en la duración de cualquier secuencia de trabajos. Con lo anterior es evidente que para minimizar la duración de una secuencia de trabajos se necesita:

- Que los trabajos adyacentes en una secuencia de trabajos sean lo más parecido posibles en tiempo de procesamiento para que se minimice el tiempo de espera entre los trabajos.
- El último trabajo debe tener tiempos de procesamiento cortos.

Una característica adicional para minimizar tiempos de procesamiento es tomado del problema de calendarización de dos máquinas de Johnson [19], en donde los trabajos con tendencia creciente (o tendencia positiva) en sus tiempos de procesamiento son procesados (o calendarizados) antes que los trabajos con tendencia decreciente (o tendencia negativa) en sus tiempos de procesamiento para minimizar la duración.

A continuación se describe el trabajo de la heurística mediante un algoritmo:

Paso .1. *Formar* $A = \{i \,| P_i \geq (1+m)/2, i \in J\}$ *y* $B = \{i \,| P_i < (1+m)/2, i \in J\}$

Paso .2. *Obtener* $A' = \left\{ [i] |\, T_{[i]} \leq T_{[i-1]}, \qquad i = 1, 2, ..., n(A) \qquad e\, [i] \in A \right\}$ *y* $B' = \left\{ [i] |\, T_{[i]} \geq T_{[i-1]}, i = 1, 2, ..., n(B) - 1e\, [i] \in B \right\}$*. Lo anterior se realiza ordenando los trabajos en* A *de forma ascendente de acuerdo al valor de* T_i *y en* B *en forma descendente de acuerdo al valor de* T_i*. Se unen los conjuntos* A *y* B *para obtener un arreglo de trabajos inicial* I*.*

Paso .3. *Remover el primer trabajo de* I *y tomarlo como trabajo semilla. De esta forma se forma la calendarización parcial* σ *con* n'*, el número de trabajos en* σ *es igual a 1. Actualizar el contenido de* I*.*

Paso .4. *Remover el primer trabajo que se encuentra en* I *e insertarlo en la p-ésima posición de la calendarización parcial* σ*, donde* $(n'+1)/2 \leq p \leq (n'+1)$*. Evaluar todas las calendarizaciones parciales resultantes en cuanto a su duración usando (4) y seleccionar la calendarización parcial con la duración menor. La calendarización parcial seleccionada es asignada a* σ *con* $n' = n' + 1$*:*

Paso .5. *Regresar al paso 4 si I no es un conjunto vacío, de otra manera parar y la calendarización parcial es considerada completa. Calcular la duración de ésta utilizando (4).*

2.2.2. Heurística de NEH

Una de las heurísticas clásicas es la creada por Nawaz, Enscore y Ham [27], llamada NEH con el objetivo de minimizar los retardos. El algoritmo es el siguiente:

Paso .1. *Ordenar los n trabajos en forma descendente de acuerdo al tiempo total de procesamiento de cada una en las máquinas.*

Paso .2. *Tomar los dos primeros trabajos y calendarizarlos de manera que se minimice la duración de la calendarización parcial.*

Paso .3. *Para* $k = 3$ *hasta* n *hacer*

Paso .4. *Insertar el k-ésimo trabajo en alguna posición de las k posiciones disponibles, de manera que el lugar en que se quede fijo minimice la duración de la calendarizacion parcial.*

La complejidad del paso (1) es $O(nlog(n))$; del paso (2) es $O(m)$. Cada calendarización parcial creada en el paso (4) se necesita $O(km)$ operaciones.

2.2.3. Algoritmo Genético de Cromosomas no Binarios

Las características del algoritmo genético utilizado fue tomado del Algoritmo Genético propuesto por Kobayashi, Ono y Yamamura [22]. El orden de los trabajos se codifica como una secuencia donde cada trabajo es representado por un número. Cada secuencia se representa como un cromosoma no-binario, lo cual lo hace diferente de los algoritmos genéticos clásicos.

La población es proporcional al tamaño de la secuencia de trabajos a calendarizar con una relación igual a n^2. Las secuencias de la población inicial se generan de forma aleatoria, resolviendo de esta manera, la parte combinatoria. El operador fundamental del algoritmo es el cruce por intercambio de subsecuencias, la cual se realiza si las subsecuencias a intercambiar contienen el mismo conjunto de trabajos. En la Figura 2.1 se muestra un ejemplo con cromosomas de longitud 6.

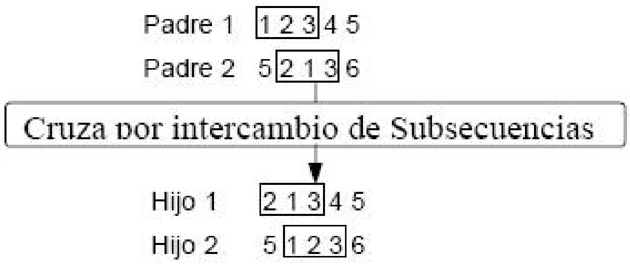


Figura 2.1: Ejemplo de cruza por intercambio de subsecuencias

2.3. Problemas Calendarización referente al Área Educativo

Referente al *Área Educativo* Xavier Cabezas [35] ha diseñado e implementado una heurística para resolver el Problema de Calendarización de horarios para Universidades, Colegios o Escuelas; consistía en la programación de clases de un conjunto de materias con un número dado de aulas y periodos de tiempo disponibles que satisfacen varias restricciones, ver Figura 2.2, en este problema se utilizaron restricciones consideradas "suaves" y "*duras*"; las restricciones [35] no pueden ser infringidas, por otro lado las restricciones [35] pueden relajarse de tal forma de buscar soluciones cercanas al óptimo que serán cotas (inferiores o superiores) para buscar nuevas alternativas de solución; luego realizó la representación de una solución con los procedimientos de Cruce, Mutación y Selección, obteniendo excelentes resultados con los problemas de prueba propuesto por SaTT (Scheduling and Timetabling Group) de la Universidad de Udine, Italia.

El caso Educativo es solo una de las variaciones de los **TTP's**, también existen otras aplicaciones en áreas muy distintas, tales como transporte (programación de salidas de buses), TV (Calendario de Programas de TV) o deportes (programación de calendarios de juegos) donde nos enfocaremos en este trabajo y veremos mas adelante unos trabajo realizados por ciertos autores, donde mostraremos un método de solución que utiliza una aproximación con programación lineal entera (**ILP**, por las siglas en inglés de *Integer Linear Programming*)

		Sistema de Planificación Académica		1er. Término 2009	Profesor: Xavier Cabezas	
		Lunes	Martes	Miércoles	Jueves	Viernes
7:30	8:30	Logística 1 - Par 1 BA18		Logística 1 - Par 1 BA18		
8:30	9:30					
9:30	10:30					Simulación - Par 2 Lab. OMEGA
10:30	11:30					
11:30	12:30	Procesos Estocásticos - Par1 ED25		Procesos Estocásticos - Par1 ED25		
12:30	13:30					

Figura 2.2: Ejemplo de Calendario Educativo

Para el caso Educacional se pueden clasificar en dos grupos: Calendarización para Exámenes (**ETT**, Exam Timetabling) y Calendarizacián para Materias (Clases Regulares) (**CTT**, Course Timetabling), este último a su vez puede ser subdividido en Calendarización por Materias Basado en Inscripciones (**EB-CTT**, Enrollment- Based Course Timetabling) y Calendarización por Materias Basado en Plan de Estudios (**CB-CTT**, Curriculum-Based Course Timetabling), Tabla 2.1

Tabla 2.1: Clasificación de Problemas de Calendarización Educativo

<table>
<tr><td colspan="3">Calendarización</td></tr>
<tr><td rowspan="3">Educacional</td><td colspan="2">Exámenes ETT</td></tr>
<tr><td>Materias
CTT</td><td>Basado en Inscripciones
EB-CTT</td></tr>
<tr><td>Materias
CTT</td><td>Basado en Plan de Estudios
CB-CTT</td></tr>
</table>

2.4. Enfoques de algunos métodos de soluciones para Calendarización

Los problemas de Calendarización en general han sido aplicados en diferentes casos, como: educacional, deportes, tareas de actividades, entre otras. Este trabajo trata de resolver la calendarización para deportes las cuales tienen características propias incluso diferentes entre las otorgadas por la Federación Ecuatoriana de Fútbol (FEF).

El esfuerzo (humano y computacional) de quien o quienes planifican los calendarios de juegos, ha hecho que el problema de calendarización para este caso particular sea considerado uno de los más importantes en la optimización combinatoria.

En muchos de los casos, las soluciones encontradas de forma manual suelen dejar muchas de las restricciones del problema sin cumplir, y el descontento de los actores que de una u otra forma están relacionados con la construcción del calendario final, se hace notar de diferentes maneras.

Un camino para resolver el problema es la Programación Lineal Entera (ILP) [6], que considera un conjunto de ecuaciones que representan las restricciones naturales de la elaboración de horarios, además de una función objetivo (FO). Formulaciones de programación lineal del problema y sus restricciones, se pueden encontrar a lo largo de la literatura, sin embargo muchas de ellas no contienen algunas restricciones de fuerte interés y que se esperan se cumplan en la planificación. Muchas veces, esto es debido a que algunas restricciones suelen ser difíciles de formular y limitan los problemas a instancias que no son aplicables a algunos de los casos reales. Aquí es donde se consideran alternativas de solución Heurísticas o Metaheurísticas como Algoritmos Genéticos [21], Búsqueda Tabú [18], Recocido Simulado, entre otros, que encuentran soluciones aproximadas, pero que con un buen diseño pueden estar muy cerca del óptimo.

Referencias específicas sobre los tres métodos presentados a continuación pueden encontrarse en Díaz [1].

2.4.1. Algoritmos Genéticos (AG):

John Holland [16] fue el primero en utilizar el término Algoritmo Genético, y que como su nombre lo indica, son procedimientos sistemáticos basados en la selección natural de los seres vivos y el paso de información genética de generación a generación. Holland basó su trabajo en la teoría de la evolución de las especies, propuesta por Darwin, simulando el proceso de selección de la naturaleza, con la esperanza de que así se consigan éxitos similares en relación con la capacidad de adaptación de un amplio número de ambientes diferentes.

Los primeros algoritmos de Holland eran simples, pero populares gracias a que podían resolver problemas que al menos en esa época eran considerados difíciles. Esta contribución de Holland, ha servido de inspiración para crear un campo de investigación que se ha desarrollado mucho más allá del trabajo original de los AG.

En terminos generales los Algoritmos Genéticos se definen de la siguiente manera:

Definición 2.4.1. *[Algoritmos Genéticos (AG).-] Un algoritmo Genético es una estructura de control que organiza o dirige un conjunto de transformaciones y operaciones diseñadas para simular los procesos de evolución.*

Procedimientos Principales en los AG's

- **Población Inicial**

  Antes de comenzar a buscar soluciones a un problema particular, es necesario contar una un conjunto de soluciones (Población Inicial) el cual pueda evolucionar generación tras generación. La Población Inicial puede ser construida de forma aleatoria o mediante algún procedimiento sistemático. Es muy importante considerar que este conjunto debe contener suficiente información para que su transformación procure una solución final óptima o al menos muy cerca del óptimo, una población muy homogénea podría tener como consecuencia una convergencia hacia una solución no necesariamente buena. Debería tener un tamaño adecuado para evitar un bajo rendimiento del algoritmo, por otro lado, un tamaño de la población muy grande podría llevarnos a un tiempo de proceso fuera de los límites de lo esperado.

- **Selección para una Nueva Generación**

  Cuando se tiene una población inicial diversa y de calidad se debe escoger los candidatos a quienes se aplicarán los procedimientos de cruce y/o mutación, se requieren que estas soluciones deben ser individuos fuertes, lo que simula la elección natural de las especies. Existen varios métodos para seleccionar soluciones entre los que se encuentran dos de las más conocidas: Torneo y Rueda de la Fortuna (Roulette Wheel).

  En el método de selección por *Torneo* se eligen aleatoriamente desde la población dos individuos (soluciones), y de estos dos se selecciona aquel que al evaluarlo en la función objetivo obtenga el mejor resultado deseado (maximiza o minimiza).

Para el caso de la *Rueda de la Fortuna* los elementos de la nueva generación se obtienen por medio de la generación de una variable aleatoria discreta que se define de la siguiente forma:

1. Se ordena la Población Actual de mayor a menor de acuerdo a la valoración de cada cromosoma en relación a la función objetivo.
2. Se suman las evaluaciones de la función objetivo de todos los cromosomas.
3. A cada cromosoma se le asigna una probabilidad de ser seleccionada de forma aleatoria, con: $Evaluacion = \frac{Evaluacion(i)}{\sum Evaluacion(Poblacion)}$.

Se selecciona un padre y una madre según el peso dado por la función de evaluación y la tabla de frecuencias relativas que se construye. Un ejemplo se muestra en la Figura 2.3, que se ha tomado desde Sóley [3], donde se tiene 5 individuos en la población, el individuo P1 tiene un valor de fitness de f1, P2 de f2, etc. y donde se observa que si la ruleta diera vuelta se detendría con más probabilidad en P3 y que con menos frecuencia se obtendrá P4.

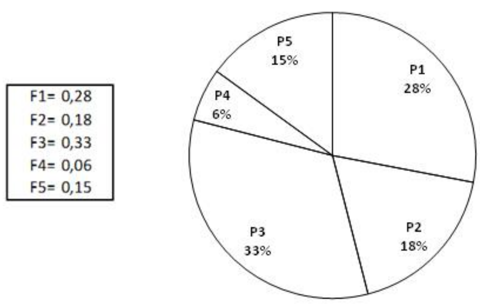


Figura 2.3: Ejemplo de selección *Ruleta de la Fortuna*

- **Criterio de Parada**

  Además de los procesos de cruce y mutación que ya se han mencionado, es necesario contar con algún criterio de parada del algoritmo. Es claro, que mientras avanza el tiempo de ejecución del AG, las nuevas generaciones serán cada vez más homogéneas, por este motivo un criterio natural de parada es detener el procedimiento cuando un gran número de cromosomas de la población sean iguales. Otro criterio podría ser definir un tiempo de ejecución fijo y luego de la última iteración elegir aquella solución representada por un cromosoma con mejor función de evaluación (fitness).

Un seudocódigo clásico y simple del AG se muestra a continuación:

```
Procedimiento Algoritmo Genético
     Generar población inicial
     Evaluar población
While NO Criterio de Parada
     Selección_población
     Cruzar población
     Mutar población
     Evaluar población
end While
```

Figura 2.4: Seudocódigo del Algoritmo Genético

2.4.2. Búsqueda Tabú (Tabu Search):

Es un tipo de búsqueda por entornos, el cual guía un procedimiento de búsqueda local para explorar el espacio de soluciones más allá del óptimo local. La búsqueda Tabú selecciona de una manera agresiva el mejor de los movimientos posibles a cada paso. Al contrario que en la búsqueda local que siempre se mueve al mejor de su entorno y finaliza con la llegada a un óptimo local, la búsqueda Tabú permite moverse a una solución de su entorno o vecindad que no sea tan buena como la actual, de tal manera que pueda tener oportunidad de salir de óptimos locales y continuar estratégicamente la búsqueda de soluciones aún mejores.

Para evitar ciclos clasifica un determinado número de los más recientes movimientos como "movimientos Tabú". Por lo tanto el escape de los óptimos locales se produce de forma sistemática y no aleatoria. En otras palabras, la Búsqueda Tabú mantiene una memoria de eventos. La filosofía de esta técnica es la creencia de que la elección de una mala estrategia sistemática de búsqueda es mejor que una buena elegida al azar.

La característica que distingue a la Búsqueda Tabú de las otras metaheurísticas de búsqueda es el uso de la memoria la cual tiene una estructura basada en una lista tabú y unos mecanismos de selección del siguiente movimiento.

La *lista tabú* es donde se registran aquellas soluciones o atributos de soluciones que no deben ser elegidas.

Una forma sencilla de construir una lista tabú consiste en que cada vez que se realiza un movimiento, se introduce su inverso (si se pasó de x_0 a x_1 , el inverso es x_1 a x_0) en una lista circular, de forma que los elementos de dicha lista están penalizados durante un cierto tiempo. Por tanto, si un movimiento está en la lista tabú no será aceptado, aunque aparentemente sea mejor solución que la solución actual.

Metodología de la búsqueda tabú

La búsqueda tabú procede como cualquier algoritmo de búsqueda: Dada una solución x se define un entorno o vecindario $N(x)$, se evalúa y se "mueve" a una mejor solución pero, en lugar de considerar todo el entorno o vecindario la búsqueda tabú define el entorno reducido $N^*(x)$ como aquellas soluciones disponibles (no tabú) del entorno de x.

```
Generar solución inicial x0
    k := 1.
    x= x0.                                        (x es la solución actual)

MIENTRAS la condición de finalización no se encuentre
HACER:
    Identificar N(x).                             (Vecindario de x)
    Identificar T(x,k).                           (Lista Tabú )
    Identificar  A(s,k).                          (Conjunto de Aspirantes)
    Determinar N*(x,k) = {N(x) – T(x,k)} ∪A(x,k).   (Vecindario reducido)
    Escoger la mejor x ∈ N*(x,k)
    "Guardar" x si mejora la mejor solución  conocida      xk := x.
    Actualizar la lista tabú
    k := k+1.
FIN MIENTRAS
```

Figura 2.5: Algoritmo de la Búsqueda Tabú Simple

En la tabla 2.2 se presentan los componentes del algoritmo de Búsqueda tabú que deben estar determinados con claridad y precisión.

Componente	**Símbolo**	**Descripción**
Función Objetivo	f(x)	La función que nos permite identificar la calidad de la solución, generalmente es maximizar o minimizar una expresión lineal o no lineal
Espacio de Soluciones	S	El conjunto de soluciones posibles, puede estar dado explicitamente o estar dada la estructura de la Solución generalmente en R^n.
Vecindario	N(x)	Las posibles soluciones que pueden seguir a x. En el método simplex sería todos los arreglos tal que una variable actualmente básica pasa a tener un valor cero y una no básica para a ser básica $N(x) \subset S$
Lista Tabú	T(x,k)	Lista de las posibles soluciones que no pueden ser elegidas en la actual iteración. Pueden ser las "L" soluciones anteriores o los últimos movimientos
Conjunto de aspitantes o candidatos	A(x,k)	Soluciones que tienen algún atributo que las hace elegibles aún si estuvieran en la lista tabú. El conocimiento del "experto" es importante para determinar que soluciones
Número máximo de iteraciones	MaxIter	Para evitar que el algoritmo entre en un proceso sin fin, es aconsejable establecer una cota superior de iteraciones posibles

Tabla 2.2: Componente del Algoritmo Búsqueda Tabú

Un ejemplo de una aplicacion del algoritmo de Búsqueda Tabú para la calendarización de examenes fue presentado por Emilio González [10] en su tema "Un algoritmo de Búsqueda Tabú para generar el calendario de exámenes de la nueva Facultad de Ciencias de la Educación"

2.4.3. Recocido Simulado (Simulated Annealing):

Se dice que este método es la más vieja entre las metaheurísticas, y seguramente uno de los primeros algoritmos que tienen una estrategia explícita para evitar mínimos locales. La idea fundamental es permitir movimientos que resulten en soluciones de peor calidad que la solución actual, con el fin de escapar de mínimos locales. La probabilidad de hacer tal movimiento es disminuido durante la búsqueda.

Este es un método heurístico que guarda relación con un campo muy diferente al de la optimización, la termodinámica. En cada iteración una vecindad es generada, este vecino es aceptado como una solución candidata, si se considera que tiene baja penalidad. Si por otra parte, este mismo vecino tiene una alta penalización, ésta podría ser aceptada como solución actual, es decir como un calendario (horario) acorde a una probabilidad relacionada con un parámetro de control denominado temperatura.

La temperatura y la probabilidad de que vecinos con alta penalidad sean aceptados, va disminuyendo en cada iteración o más usualmente después de un número de iteraciones (este número puede ser constante o puede incrementarse de acuerdo a como disminuye la temperatura).

A continuación en la figura 2.6 se presenta el seudocódigo del Recocido Simulado:

RS

1. Escoja una secuencia inicial S
2. Halle la temperatura inicial $T = T_0 > 0$
3. Repita:
    a. Escoja una nueva secuencia S'
    b. Sea $\Delta E = E(S)' - E(S)$, donde $E(S)$ es la energía (*Makespan*) de la secuencia S
    c. Si $\Delta E \leq 0$, acepte la nueva secuencia, haga S=S'
    d. De lo contrario, Si $exp(-\Delta E/T) \geq rand(0,1)$ acepte la nueva secuencia
    e. De lo contrario rechace la secuencia
    f. Reduzca la temperatura de acuerdo al enfriamiento programado
4. Hasta la condición de parada.

Figura 2.6: Algoritmo del Recocido Simulado

Ventajas del Recocido Simulado

En este método, se ha demostrado que converge a una solución muy cercana a la óptima de un problema.

Una implementación sencilla del algoritmo, hace que sea muy fácil de adaptar un método de búsqueda local a un algoritmo de recocido simulado, usualmente aplicando la búsqueda local con resultados mucho mejores.

Desventajas del Recocido Simulado

Aunque se ha demostrado que el algoritmo converge al óptimo, este converge en un tiempo infinito. No solo por esta razón, si no debido a que se debe enfriar lentamente, el algoritmo no es usualmente más rápido que sus contemporáneos.

Otras metodologías de solución son muy novedosas, como: El Algoritmo de Búsqueda Dispersa (SSA por sus siglas en inglés Scatter Search Algorithm) [25], GRASP (Greedy

Randomized Adaptive Search Procedure) y Redes Neuronales que suelen considerar restricciones poco comunes en relación a los modelos estándares.

2.5. Problemas de Calendarización de Deportes

Adicionalmente Xavier Cabezas interpretó el documento de Della-Croce acerca de "Scheduling the Italian Football League: an ILP-based approach", donde se presenta un procedimiento para aproximar una solución del problema de calendarizar los partidos de la Serie A de la Liga Italiana de Fútbol, uno de los principales motivos por el que escogió ese tema fue por la mezcla de un procedimiento heurístico con la programación lineal entera (ILP). En el paper orginal se probó el procedimiento a datos reales de la Serie A, para los años 2001-2002, 2002-2003 y 2003-2004. En todos los casos se trató con 4 equipos cabezas de serie de un total de 18 equipos. Además en los años 2001-2002 y 2002-2003 hubieron 3 pares de equipos en la misma ciudad, Lazio/Roma, Inter/Milan y Juventus/Torino. Una de las principales restricciones eran sobre los requerimientos y recursos de las televisoras. En la figura 2.7 se muestra el calendario propuesto por medio de este método de optimización referente a la Liga Italiana 2003/2004.

Se puede demostrar que la aplicación de técnicas de optimización a la programación de ligas deportivas ha recibido mucha atención; el interés práctico y la difícil resolución a instancias de tamaño real de los problemas de programación en deportes han cautivado la atención de académicos en la Investigación de Operaciones (IO) y tareas relacionadas.

semana 1	semana 2	semana 3	semana 4	semana 5	semana 6
ANC-*MOD*	BOL-SAM	*BRE*-REG	*ANC*-SAM	*BRE*-LAZ	*ANC*-PAR
CHI-REG	*BRE*-*CHI*	*CHI*-*EMP*	BOL-UDI	*CHI*-JUV	BOL-*CHI*
MIL-*BRE*	*EMP*-PAR	MIL-BOL	*EMP*-*BRE*	MIL-*MOD*	**INT-MIL**
PAR-LEC	INT-*PER*	PAR-LAZ	INT-SIE	PAR-INT	JUV-*BRE*
PER-BOL	JUV-UDI	*PER*-*ANC*	JUV-PAR	*PER*-SAM	LAZ-*EMP*
ROM-*EMP*	LAZ-*ANC*	ROM-JUV	LAZ-*CHI*	REG-*EMP*	LEC-REG
SAM-INT	LEC-MIL	SAM-*MOD*	LEC-ROM	ROM-*ANC*	*MOD*-SIE
SIE-JUV	*MOD*-ROM	SIE-LEC	*MOD*-*PER*	SIE-BOL	*PER*-UDI
UDI-LAZ	REG-SIE	UDI-INT	REG-MIL	UDI-LEC	SAM-ROM

semana 7	semana 8	semana 9	semana 10	semana 11	semana 12
BRE-BOL	*ANC*-SIE	*BRE*-INT	*ANC*-MIL	BOL-*ANC*	*ANC*-*BRE*
CHI-SAM	BOL-REG	*CHI*-*ANC*	BOL-*EMP*	*BRE*-SAM	**INT-JUV**
EMP-LEC	INT-*CHI*	*EMP*-*MOD*	INT-REG	*CHI*-PAR	*MOD*-BOL
MIL-JUV	JUV-*EMP*	LAZ-LEC	**JUV-LAZ**	*EMP*-*PER*	PAR-REG
PAR-*MOD*	**LAZ-MIL**	MIL-*PER*	LEC-*CHI*	**LAZ-INT**	*PER*-LEC
REG-LAZ	LEC-*BRE*	PAR-BOL	*MOD*-*BRE*	LEC-JUV	**ROM-LAZ**
ROM-INT	*MOD*-UDI	REG-JUV	*PER*-PAR	MIL-ROM	SAM-MIL
SIE-*PER*	*PER*-ROM	SIE-ROM	ROM-UDI	REG-*MOD*	SIE-*CHI*
UDI-*ANC*	SAM-PAR	UDI-SAM	SAM-SIE	SIE-UDI	UDI-*EMP*

semana 13	semana 14	semana 15	semana 16	semana 17
BOL-INT	*ANC*-JUV	*BRE*-*PER*	*ANC*-*EMP*	*BRE*-UDI
BRE-ROM	INT-*EMP*	*CHI*-*MOD*	BOL-LAZ	*CHI*-ROM
CHI-*PER*	*MOD*-LEC	*EMP*-SAM	MIL-*CHI*	*EMP*-MIL
EMP-SIE	PAR-*BRE*	JUV-BOL	*MOD*-INT	INT-*ANC*
JUV-SAM	*PER*-REG	LAZ-SIE	*PER*-JUV	JUV-*MOD*
LAZ-*MOD*	ROM-BOL	LEC-INT	ROM-REG	LAZ-*PER*
LEC-*ANC*	SAM-LAZ	MIL-UDI	SAM-LEC	LEC-BOL
MIL-PAR	SIE-MIL	PAR-ROM	SIE-*BRE*	PAR-SIE
REG-UDI	UDI-*CHI*	REG-*ANC*	UDI-PAR	REG-SAM

Figura 2.7: Uno de los calendarios generados por la HEURÍSTICA para la Serie A de la Liga Italiana 2003/2004

Nemhauser [28] en 1998 introdujó el uso de patrones (local-visita) para reducir el tiempo de resolución de un modelo de programación entera en un campeonato universitario de baloncesto de los Estados Unidos, su objetivo era de reducir breaks y satisfacer requerimientos de empresas televisoras. Despues de unos años Henz [15] utilizó Constraint Programming (CSP) para reducir considerablemente los tiempos de resolución del problema presentado por Nemhauser y Trick [28].

El uso de técnicas de programación en deportes, en ligas de fútbol del mundo real es relativamente reciente. La primera aplicación es reportada en 1992 por Schreuder [33], donde se muestra la programación de la liga de fútbol de Holanda en la temporada 1989/1990. Esta liga constaba de 18 equipos, que se enfrentaban todos contra todos una vez, es decir, en un torneo todos contra todos. Fueron consideradas diferentes condiciones, como por ejemplo aspectos comerciales, deportivos y de organización. Las restricciones son divididas en duras, que deben cumplirse obligatoriamente, y blandas, que deseablemente deben cumplirse, a las cuales se asignaba un peso que representaba su grado de importancia. La función objetivo buscaba maximizar el peso total que logra la solución. El problema fue resuelto mediante la implementación de una heurística constructiva. La solución encontrada solo alcanzo a cumplir el 80 % de las condiciones requeridas para el torneo de la temporada 1989/1990.

Una aplicación basada al campeonato Chileno, lo realizó Durán [9], donde incorpora la reducción de breaks, distancias recorridas y tenia como principal objetivo promover el interés de los aficionados y distribuir los ingresos a través del torneo. La Primera División constaba entonces de 20 equipos, divididos en cuatro grupos de cinco equipos cada uno. El torneo se componía de dos fases. La primera fase estaba organizada como un round robin, es decir los 20 equipos jugaban todos contra todos una vez. A la segunda fase avanzaban los dos mejores equipos de cada grupo, en total ocho equipos, que se enfrentaban en series de playoffs para determinar al campeón. Las condiciones que el programa debía cumplir incluían distintos aspectos, tales como restricciones geográficas, atractivo para los simpatizantes y requerimientos de la TV. El enfoque de resolución se basaba en programación lineal entera y procedimientos básicos para la asignación de patrones de localías a los equipos. Esta metodología fue aplicada durante los años 2005 a 2007.

Rasmussen programó la principal liga de fútbol de Dinamarca en la temporada 2006/2007 [31]. La liga consistía de 12 equipos, que se enfrentaban entre sí tres veces, constituyendo así un torneo triple todos contra todos. Una variedad de condiciones fueron consideradas, tales como requerimientos geográficos y restricciones asociadas a los equipos principales. Estas condiciones son divididas en duras y blandas. La función objetivo busca minimizar las penalizaciones de no cumplir ciertas restricciones. El enfoque de solución usa descomposición de Benders y técnicas de generación de columnas, y conduce a soluciones buenas en poca cantidad de tiempo.

Goossens and Spieksma programaron la principal liga de fútbol de Bélgica en la temporada 2007/2008 [14]. El torneo consistió de 18 equipos que jugaron un doble *roundrobin* espejado. El número total de breaks (repetición de la condición de local o visitante de un equipo en fechas consecutivas) debía ser mínimo. Varias condiciones más fueron consideradas, tales como requerimientos de la TV y balance de los efectos de $carry - over$ (un equipo i recibe una unidad de efecto $carry - over$ del equipo j, si i juega en la fecha $k + 1$ contra el mismo rival que j enfrentó en la fecha k). Las restricciones del problema son clasificadas en grupos de acuerdo a su nivel de prioridad, asignando penalizaciones. El enfoque de solución utiliza un modelo de programación entera mixta que asigna equipos a un fixture canónico preestablecido (generado para balancear los efectos de $carry - over$), satisfaciendo las restricciones de fixture espejado y minimizando el número de breaks. Luego, procedimientos de búsqueda local son utilizados para mejorar la solución, en términos de minimizar la penalidad total de las restricciones violadas.

Más recientemente, otros dos casos de aplicaciones han sido reportados. Flatberg et al. programaron la liga de fútbol de Noruega usando un modelo de programación entera [11], que considera en su objetivo los efectos de $carry - over$. Esta liga juega un doble round robin no espejado de 14 equipos. Ribeiro y Urrutia programaron la liga de fútbol de Brasil en 2009 [32], utilizando también un enfoque de programación entera que busca maximizar el número de partidos atractivos entre equipos de elite que pueden ser transmitidos por televisión abierta. La liga de Brasil está compuesta de 20 equipos que juegan un torneo doble todos contra todos.

Hasta lo que conocemos, los trabajos citados en los párrafos anteriores son las únicas aplicaciones reportadas en la literatura que han sido usadas en la práctica en ligas de fútbol del mundo real. Hay otros artículos contextualizados en ligas de fútbol, pero que han permanecido a nivel teórico. También suponemos que hay otras aplicaciones en el mundo real que no han sido reportadas en artículos académicos. Mediante comunicación personal, sabemos que técnicas de optimización han sido utilizadas en la programación de ligas de fútbol en República Checa, desde la temporada 2002/2003 a la fecha [13]; Polonia, en las temporadas 2007/2008 y 2008/2009 [26]; y Honduras, en el torneo 2010.

Existe también una extensa literatura en la programación de torneos en otros deportes. Para una revisión extensa a la literatura, incluyendo aspectos teóricos y prácticos, referimos al lector a un par de artículos recientes ([30],[20]). Instancias de prueba para distintos torneos del mundo real pueden encontrarse en [29].

Las aplicaciones de IO a la programación de torneos de fútbol parecen ser pocas, considerando que actualmente asociaciones de 208 países están aliadas a la Federación Internacional de Fútbol Asociado (FIFA). Es presumible entonces que el uso de técnicas manuales aún impera en la programación de muchas ligas alrededor del mundo.

2.5.1. Emparejamientos aplicados a la elaboración de calendarios deportivos

Uno de los métodos de solución para el problema de Calendarización de deportes, fue presentado en el documento realizado por Aida Días, María Guerrero y Javier Rodrigo [2], que consitía en el uso de *Emparejamientos*, lo cual pertenece a la Teoría de Grafos que es parte de la Matemática Discreta. Los *emparejamientos* que fueron realizados entre los vértices del grafo (los equipos) eran disjuntos y de utilizar todas las aristas del grafo, puesto a que los equipos se tenían que enfrentar todos contra todos sin repetir partidos. De este modo hay diversas formas de realizar este tipo de emparejamientos, las cuales fueron expuestos. Este análisis fue realizado con datos de los equipos de la primera división del fútbol español de la temporada 2004-2005.

Conceptos de términos usados en esta solución

- **Grafo.-** un grafo G está formado por un conjunto finito no vacío V = V(G) de puntos (o vértices, o nodos) y por un conjunto E de pares no ordenados de puntos u, v de V. Cada par de puntos en E es una arista (o arco, o lado) de G y se dice que u, v une u con v, u y v son los extremos de la arista y son adyacentes. Un grafo con p puntos y q aristas se llama grafo (p, q). El grafo (1,0) es trivial, es decir, un solo punto aislado.

- **Vértice aislado.-** es aquel vértice al cual no se puede acceder desde ningún otro vértice.

- **Grado de un vértice.-** es el número de aristas que inciden en dicho vértice.

- **Cardinal de V.-** es el número de vértices del grafo y se representa como —V—. Si V tiene n vértices, entonces —V— = n.

- **Emparejamientos.-** Un emparejamiento en un grafo G = (V,A) es un subconjunto M de A, tal que dos aristas cualesquiera de M no tienen un extremo común. Figura 2.8.

2.5.2. Entidades que participan en la Calendarización de Deportes TTP

De acuerdo a la literatura de documentos desarrollados por otros autores, he encontrado una serie de presentaciones para la formulación de este problema, lo cual es debido a que no es posible escribir una que contenga todos los casos posibles que pueden presentarse

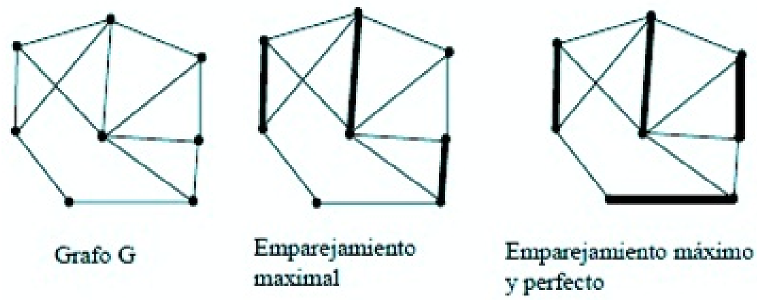


Figura 2.8: Ejemplo de emparejamiento

en la vida real, cada federación de distintos países poseen características diferentes que hacen que el problema se vuelva muy particular.

Con el fin de tener una descripción general de este problema se ha tomado como referencia diferentes propuestas que contienen las restricciones más populares (duras y suaves) del problema.

Primero se comienza con definir las entidades que participan en el problema:

- **Esquema (Pattern).-** Cadena de símbolos que pertenecen al conjunto H,A indicando una sucesión de partidos de Local (H, por Home) y Visitante (A, por Away) de un equipo en el torneo.

- **Quiebre (Break).-** Un Quiebre ocurre cuando dos partidos consecutivos se juegan o bien de local o bien de visitante.

- **Conjunto de esquemas (Pattern Set).-** Conjunto de esquemas diferentes con cardinalidad igual al número de equipos del torneo.

- **Televisoras.-** Son las distintas empresas televisivas que poseen los derechos de transmisión de los diferentes equipos.
- **Esquemas Complementarios.-** Son los esquemas correspondientes a los partidos de vuelta por cada equipo.
- **Semanas.-** Es el número total de encuentros realizados por cada equipo.
- **Emparejamiento.-** Es la asignación de los equipos a los esquemas factibles.
- **Round Robin.-** Es el tipo de torneo cuando todos los equipos jugaban todos contra todos una sola vez por cada etapa.

Capítulo 3

DESCRIPCIÓN Y DEFINICIÓN DEL PROBLEMA

3.1. Campeonato Ecuatoriano de Fútbol 2011

El Campeonato Ecuatoriano de Fútbol 2011, cuyo nombre comercial es Copa Credife 2011 desde el año 2009 por razones comerciales, es un torneo oficial de fútbol de primera división realizado en Ecuador desde 1957, organizado anualmente por la Federación Ecuatoriana de Fútbol (FEF).

Este campeonato es considerado uno de los mejores del mundo de acuerdo con la IFFHS (International Fed for Football History and Statistics). Actualmente, ocupa el undécimo puesto en el ranking oficial de dicha institución.

El sistema del Campeonato Ecuatoriano de Fútbol 2011 fue definido el 15 de Diciembre de 2010, por los dirigentes de los clubes que participan en el campeonato de Serie A en conjunto con los directivos de la Federación Ecuatoriana de Fútbol (FEF), los cuales analizarón las diferentes propuestas del sistema de campeonato por parte de los directivos de las diferentes instituciones en una sesión ordinaria del Comité Ejecutivo de la FEF. Este sistema del torneo fue aprobado por el Congreso de la FEF el 7 de enero de 2011.

3.1.1. Sistema del Campeonato Ecuatoriano de Fútbol 2011

El Campeonato Ecuatoriano de Fútbol de 2011, se jugó con la misma modalidad que el año 2010, de acuerdo a lo que decidieron los dirigentes de la Federación Ecuatoriana de Fútbol.

El Campeonato Ecuatoriano de Fútbol de 2011, según lo establecido en el Reglamento 2011 otorgado por la Federación Ecuatoriana de Fútbol, éste se jugó con 12 equipos que se disputarón el título en tres etapas. De las cuales dos etapas fue de todos contra todos y la tercera etapa fué para definir al campeón y a los mejores equipos de la Tabla Acumulada, que no eran los finalistas, los cuales pelearón por el tercer lugar que otorgó un cupo a la Primera Fase (repesca) de la Copa Libertadores.

Etapa .1. *Se desarrolló entre el 30 de enero y el 12 de junio de 2011, consistió de 22 jornadas que por disposición de los directivos, fue con 17 domingos y cinco miércoles. Debido a la Copa América 2011 que se desarrolló en Argentina. La modalidad fue de todos contra todos, el ganador de esa etapa clasificó a la final del torneo, a la Copa Sudamericana 2011 como Ecuador 1 y a la Copa Libertadores 2012. El Segundo puesto de esta etapa jugó la Copa Sudamericana como Ecuador 2, debido a que el año 2010 en* *la Primera Etapa se entregó tres cupos a la Copa Sudamericana, pero para el 2011 sólo hubo dos, porque Liga Deportiva Universitaria de Quito (LDU Q) ya se habia adjudicado como Ecuador 3 por haber ganado la Segunda Etapa del 2010. En caso de que LDU Q hubiese terminado en alguno de los 2 primeros puestos en esta fase, el tercer cupo era al tercer mejor ubicado.*

Etapa .2. *Se realizó desde el 24 de julio hasta el 4 de diciembre de 2011. La modalidad fue de todos contra todos, consistió de 22 jornadas disputadas los días domingo, excepto el 24 de agosto y el 21 de septiembre (ambos en miércoles).*

El ganador clasificó a la tercera etapa donde disputó la final del campeonato con el ganador de la primera etapa, adicionalmente clasificó a la Copa Sudamericana 2012 como Ecuador 1 y a la Copa Libertadores 2012.

Etapa .3. *Se llevó a cabo los dias 11 y 18 de diciembre del 2011. Se enfrentarón los ganadores de la primera y segunda etapa por la disputa del campeonato y subcampeonato. Los dos equipos que terminaron en la tabla acumulada en la tercera y cuarta posición, jugarón un repechaje y el ganador seria el tercer representante al fútbol ecuatoriano (Ecuador 3) a la Copa Libertadores del próximo año.*

En caso de que el ganador de la segunda etapa, hubiese sido el mismo equipo que ganó la primera, se convertía automáticamente en el campeón del fútbol ecuatoriano.

Los otros dos cupos para la Copa Sudamericana de 2012 se disputarán en la primera etapa del Campeonato Nacional del 2012. Los dos equipos que sumaron menor puntaje en la tabla acumulada, perdierón la categoría y jugarán en el 2012 en la Serie B.

Cabe recalcar que todos los equipos desean o aspiran jugar en la Serie A del campeonato ecuatoriano y adicionalmente llegar a obtener algunna clasificación a una copa internacional, ya sea la Copa Libertadores o la Copa Sudamericana; pero a pesar de esto, se ha demostrado en el transcurso de los años que jugar en la Serie A es la más popular entre los simpatizantes (hinchas) y, por lo tanto, la más atractiva para los auspiciantes y la TV.

A distintos canales de TV se les hace muy atractivo el torneo, por lo cual ha cautivado el interés por parte de ellos, adicionalmente los equipos han realizado contratos con distintas empresas de TV desde hace algunos años como una base primordial de la economía de los clubes.

Una de las tareas de la FEF es la programación de los partidos de cada torneo de la Primera y Segunda División en general. Hace algunos años este era uno de los mas altos problemas que tenía de FEF, debido a que por lo general cada año cambiaban la modalidad (características) del campeonato. Una de las modalidades que tenía inicialmente la FEF era de elegir aleatoriamente unos numeros los cuales eran asignados a los equipos participantes en el torneo, de esta forma generaban el calendario utilizado en los torneos. Esta modalidad les permitía realizar una programación que cumplía los requerimientos básicos del torneo, pero no permitía considerar una serie de otros criterios, lo que conllevaba a múltiples quejas de parte de los equipos.

En base a este tipo de problemas que se presenta en la elaboración del calendario en este tipo de torneos, ya sean estos de Fútbol, Basket, etc., es creado este documento, debido a que se presenta un mecanismo para la elaboración de los mismo (calendarios de juegos), en el cual nos basamos a las caracteristicas básicas de un torneo *todos contra todos* mediante técnicas de Investigación de Operaciones (IO). Adicionalmente agregarón restricciones basadas en los derechos de televisivos de las TVs, para de una u otro forma definir los recursos de estas (TVs); ésta es una de las caracteristicas más importantes que presentamos en el desarrollo del calendario, la cual ademaás incorporó requerimientos de la FEF y de los clubes. Las condiciones consideradas integraron múltiples aspectos tales como partidos considerados clásicos (equipos de la misma ciudad), partidos de mucho interés para el público (aficionados), entre otros.En los siguientes secciones describiremos las características más importantes de la aplicación.

3.1.2. Equipos Participantes

Existieron 12 equipos en total afiliados a la Federación Ecuatoriana de Fútbol (FEF), los cuales participarón en el torneo. Los clubes que jugarón el campeonato 2011 estaban conformados por los 10 mejores equipos que disputaron el torneo de Serie A de 2010 y se complementó la lista con los 2 mejores equipos del torneo de Serie B de 2010, en este caso fueron la Liga de Loja y el Imbabura SC.

En términos geográficos, 3 clubes pertenecían a la región litoral (costa), mientras que los 9 restantes tendían sus sedes en provincias de la región interandina (sierra). Con lo que respecta a las grandes ciudades, en Guayaquil se encuentraban 2 equipos, en Quito se concentraban la mayoría de los clubes participantes del torneo con un total de 4 equipos, aunque uno de ellos (Espoli) jugó sus partidos en Santo Domingo, además el

Independiente del Valle tuvo su sede en el Estadio Municipal General Rumiñahui, en la localidad de Sangolquí, Valle de Los Chillos, en el área metropolitana de la capital; mientras que en Cuenca solo jugó el Deportivo Cuenca. Además, hubieron otras cuatro ciudades con un solo equipo las cuales fuerón: la ciudad de Riobamba como sede del Olmedo, Ibarra como sede del Imbabura SC, Loja como sede de la Liga de Loja y Manta como sede del Manta FC.

Equipos Participantes		
Equipo	**Ciudad**	**Estadio**
Barcelona	Guayaquil	E. Monumental Banco Pichincha
Emelec	Guayaquil	E. George Capwell
Deportivo Quito	Quito	E. Olimpico Atahualpa
El Nacional	Quito	E. Olimpico Atahualpa
Espoli	Quito	E. Olimpico Municipal Etho Vega (en Santo Domingo)
Liga de Quito	Quito	E. Casa Blanca
Deportivo Cuenca	Cuenca	E. Alejandro Serrano Aguilar
Imbabura S.C.	Ibarra	E. Olimpico de Ibarra
Independiente del Valle	Sangolquí	E. Municipal General Rumiñahui
Liga de Loja	Loja	E. Federativo Reina del Cisne
Manta F.C.	Manta	E. Jocay
Olmedo	Riobamba	E. Olimpico de Riobamba

Tabla 3.1: Equipos Participantes. **Fuente:** Federación Ecuatoriana de Fútbol

Equipos por Provincia		
Provincia	**No.**	**Equipos**
Pichincha	5	El Nacional, Independiente del Valle, Liga de Quito, Espoli y Deportivo Quito
Guayas	2	Barcelona y Emelec
Azuay	1	Deportivo Cuenca
Chimborazo	1	Olmedo
Imbabura	1	Imbabura S.C.
Loja	1	Liga de Loja
Manabí	1	Manta F.C.

Tabla 3.2: Equipos por Provincia

3.2. Definición del problema

Cuando el problema de calendarización de un torneo todos contra todos (con un número par de equipos) no tiene restricciones, su construcción puede ser fácilmente obtenido en tiempo lineal respecto al número de partidos y tiene una interpretación teórica sobre grafos, porque corresponde al problema de encontrar la 1-factorización de un grafo completo no dirigido con un número par de vértices.

Con respecto a las restricciones de partidos sucesivos de local (visitante), se introduce la siguiente notación, la cual se encuentra previa y análogamente establecida en el popular trabajo de Nemhauser [28].

Definición 3.2.1. *[Esquema (Pattern)] Cadena de símbolos que pertenecen al conjunto H,A indicando una sucesión de partidos de Local (H, por Home) y Visitante (A, por Away) de un equipo en el torneo.*

Definición 3.2.2. *[Quiebre (Break)] Un Quiebre ocurre cuando dos partidos consecutivos se juegan o bien de local o bien de visitante.*

Definición 3.2.3. *[Conjunto de esquemas (Pattern Set)] Conjunto de esquemas diferentes con cardinalidad igual al número de equipos del torneo.*

El problema de calendarización de deportes consiste básicamente en:

Dados un conjunto de equipos, un conjunto de televisoras, un conjunto de esquemas; en asignar los equipos a las respectivas televisoras que poseen sus derechos de transmisión con un determinado esquema específico disminuyendo la cantidad de quiebres, todo esto están sujetos a un conjunto de restricciones que son consideradas suaves y duras. La mayor dificultad de este problema consiste en encontrar soluciones que involucren un emparejamiento de los equipos con las televisoras y sus respectivo esquema para una determinada semana.

Debido a estas características este problema estaría dentro de la categoría de los problemas de optimización combinatoria. Por este motivo no siempre es posible dar una solución que sea del todo aceptable para todas las partes involucradas.

3.2.1. Restricciones del problema

Dado un conjunto N de n equipos provenientes de diferentes ciudades, crearemos un calendario de juegos para la primera y segunda etapa, con la modalidad de todos contra todos (DOUBLE ROUND ROBIN TOURNAMENT), cada equipo jugará dos veces contra un equipo, uno de local y el otro de visitante.

Las restricciones que asumiremos para este problema son las siguientes:

- Esta es basada a la posición que los equipos del campeanato obtuvieron en a temporada inmediata anterior. Los mejores equipos de la seria A del año anterior los consideraremos *cabezas de serie* para el siguiente año y para estos equipos se sugiere que no jueguen entre ellos las primeras y últimas γ semanas (asumiremos $\gamma = 2$).

Equipos cabeza de serie
Emelec
Liga de Quito
Barcelona
Deportivo Quito

Tabla 3.3: Equipos Cabezas de Serie

- Un restricción típica que consideraremos es relacionado a la suceción de partidos de local y visitante, es decir no se permitirá más de dos partidos consecutivos de local(visitante) para cada equipo.

- Como este es un caso particular de las aplicaciones de TTP para deportes, debemos tener particularidades en su formulación. En lo presente, tenemos equipos que pertenecen a las mismas ciudades como equipos representantes, es decir que será necesario que los equipos de la misma ciudad deberían tener programaciones de juegos local-visitante complementarios, por ejemplo Barcelona y Emelec son únicos representantes de Guayaquil, lo que implica que si Barcelona juega de local en una determinada fecha, Emelec jugariá de visitante en la misma jornada.

Equipos de una misma ciudad
Barcelona / Emelec
Liga de Quito / Deportivo Quito / El Nacional

Tabla 3.4: Equipos de una misma ciudad

- De la misma forma que para los equipos cabezas de serie, los equipos entre equipos de una misma ciudad no se les permitirá jugar ente ellos en las primeras y últimas k semanas, asumiremos $k = 2$. Table 3.4

- Como suele suceder en todos los paises, existen equipos que son únicos representantes de una ciudad, es comunmente preferible que estos equipos no jueguen un partido de local cuando exista en la ciudad algún evento festivo obligatorio, como fiestas de fundación,entre otras. Esta restricción se considera *suave*.

- Finalmente desde hace algunos años las compañias de TV poseen derechos de transmisión de los partidos del Campeonato Nacional Copa Credifé. Los respectivos equipos son asignados a estas empresas como: Ecuavisa, Teleamazonas y GamaTv; lo que significa que la asignación de un equipo j a la televisora p implica que p puede transmitir en vivo los partidos de j.

Equipos asignados a cada Televisora		
TV1	**TV2**	**TV3**
Emelec	Liga de Quito	Barcelona S.C.
El Nacional	Dep. Quito	Independiente
Deportivo Cuenca	Liga de Loja	
Olmedo	Manta F.C.	
	Espoli	
	Imbabura	

Tabla 3.5: Televisoras con los derechos de transmisión

- En este trabajo se impone una restricción de balance en la asignación de equipos a las televisoras, donde los partidos de local en todas las semanas son proporcionalmente divididos. Esto es dividido a que al menos los 2/3 de los partidos se juegan al mismo tiempo. Es decir, un buen calendario balanceado deberia minimizar los requerimientos de personal y equipo disponible en paralelo.

3.3. Modelo Matemático del Problema

El modelo matemático que se presenta a continuación está basado en la información referente al problema de calendarización de deportes, aunque no todas las restricciones que serán tomadas en este documento se encuentran presentes, sino las más relevantes que serán analizadas para el caso de estudio.

Conjunto de datos

Las entidades correspondientes a este tema, son descritas de la siguiente manera:

- **Equipos**

  Representa a cada equipo participante en el campeonato Ecuatoriano de fútbol de la primera categoría organizado por la Federación Ecuatoriana de Fútbol. El dominio de esta variable $E = E1, E2, E3, \ldots, E12$, es el conjunto de todos los equipos inscritos en la Federación Ecuatoriana de Fútbol (FEF).

- **Televisoras**

  Representa a cada TV que posee los derechos de transmisión de los diferentes equipos, las cuales están registradas en la F.E.F., el dominio de esta variable $TV = TV1, TV2, TV3$, es el conjunto de empresas de televisión que trasmitirán los partidos correspondientes a sus derechos, las cuales están asociadas a los equipos participantes.

3.3.1. Procedimiento para llegar a la solución (Calendario)

En el siguiente contexto se detalla el mecanismo o procedimiento para llegar a la solución del problemase, ésta se divide en tres fases que se enuncia de la siguiente manera:

1. Búsqueda de conjuntos de esquemas factibles en Base a los recursos de las televisoras.

2. Búsqueda de calendarios factibles basados a los esquemas factibles.

3. Emparejamiento de equipos a esquemas para la elaboración del calendario final.

Primera fase: Búsqueda de conjuntos de esquemas factibles.

En la primera fase se busca un número suficientemente grande de diferentes conjuntos de esquemas formados por esquemas complementarios los cuales están balanceados respecto a la cobertura de televisión, además de minimizar el número total de quiebres.

Modelo de Programación Entera

Matrices de Entrada:

Las entidades matemáticas descritas a continuación, son generadas en $Mathematica$:

p_i = Número de *quiebres* del *esquema* i (que toma valores de $p_i = 0$, 1 o 2) que corresponde a cero *quiebres*, un *quiebre* y dos *quiebres* por cada torneo todos contra todos y sus complementos, respectivamente. (Tabla 4.2)

NTV_k = Número total de equipos asignados (partidos de local) a la televisora k.

$C_{i,l}$ = Conjunto que contiene los pares de *esquemas complementarios*.(Tabla 4.1)

$$A_{i,j} = \begin{cases} 1 & \text{Si el } \textit{esquema } i \text{ juega de visitante en la semana } j \\ 0 & \text{Si no (Tabla 6.1)} \end{cases}$$

Y por último, definimos una variable binaria, la cual se describe de la siguiente manera:

$$x_{i,k} = \begin{cases} 1 & \text{Si el esquema } i \text{ es seleccionado a la televisora } k \\ 0 & \text{Si no} \end{cases}$$

En ésta programación matemática el objetivo se basa principal es minimizar el número de *quiebres*, la cual se la define de la siguiente manera:

$$Min \qquad z = \sum_i \sum_k p_i x_{i,k}$$

Este modelo matemático está sujeto a las siguientes restricciones:

- Indica el número total de partidos de local asignados a cada televisora.

$$\sum_i A_{i,j} x_{i,k} = \frac{N_k}{2}; \qquad k = 1,2 \qquad \forall j, \tag{3.1}$$

- El número total de *esquemas* asignados a cada televisora.

$$\sum_i x_{i,k} = N_k; \qquad k = 1,2, \tag{3.2}$$

- Si un *esquema* es seleccionado, entonces su *esquema complementario* también es seleccionado.

$$x_{i,k} - x_{j,k} = 0; \qquad \forall (i,j) \in C_{i,l}, \qquad k = 1,2, \tag{3.3}$$

- Cada *esquema* puede ser seleccionado solo una vez.

$$\sum_k x_{i,k} \leq 1; \qquad \forall i, \tag{3.4}$$

- Especificación de variables binarias.

$$x_{i,k} \in \{0,1\}\,; \qquad \forall i,k. \tag{3.5}$$

Para obtener un segundo conjunto de *esquemas*, como para generar un nuevo calendario o para buscar esquemas factibles para uso posterior,se repite esta fase agregando la siguiente restricción:

$$\sum_{(i,k)\in S} x_{i,k} \leq \frac{n}{2}; \tag{3.6}$$

donde S es el conjunto de *esquemas* seleccionados por la solución inmediatamente anterior. La restricción 3.6 asegura que al menos el 50 % del conjunto de variables en la solución previa tomarán el valor de cero, es decir se obtendrá una nueva solución significativamente diferente a la anterior.

La repetición de este procedimiento permite obtener en pocos segundos un número grande de conjuntos de esquemas diferentes.

Segunda fase: Búsqueda de calendarios factibles.

En esta fase, para cada *conjunto de esquemas* generados en la *fase 1*, se busca un *calendario* factible y consistente con este.

Modelo de Programación Entera

En ésta segunda fase interviene la siguiente variable binaria, la cual se la define de la sigiente manera:

$$x_{i,j,t} = \begin{cases} 1 & \text{Si el } \textit{esquema } i \text{ es emparejado al } \textit{esquema } j \text{ en la } \textit{semana } t \\ 0 & \text{Si no} \end{cases}$$

$d_{i,j,t}$ = Conjunto de *esquemas* que tienen juego de local (visitante) en la semana t donde el *esquema* i juega de visitante (local)

Como estrategia de modelización para esta fase usaremos como objetivo minimizar una constante (función de costos ficticia), lo que quiere decir que la podemos ejecutar con cualquier motor de optimización (Motor utilizado XPRESS o CPLEX en GAMS).

Las restricciones consideradas son:

- Indica que para cada semana y *esquema* de local (visitante) exactamente un *esquema* de oponente visitante (local) deberia ser seleccionado.

$$\sum_{j \in d_{i,j,t}} x_{i,j,t} = 1; \qquad \forall i, t, \tag{3.7}$$

- Si un *esquema* j es el oponente de un *esquema* i en la semana t, entonces, de forma correspondiente, el *esquema* i debería ser el oponente del esquema j sobre la semana t.

$$x_{i,j,t} = x_{j,i,t}; \qquad \forall i, j > i, t, \tag{3.8}$$

- Cada par de *esquemas* i, j debería ser seleccionada para solo una semana.

$$\sum_t x_{i,j,t} = 1; \qquad \forall i, j \neq i, \tag{3.9}$$

- Especificación de las variables binarias.

$$x_{i,j,t} \in \{0, 1\}; \qquad \forall i, j, t. \tag{3.10}$$

Tercera fase: Emparejamiento de equipos a esquemas.

En esta fase se recibe como datos de entrada cada calendario factible generado por la fase 2. **Es posible que en la fase 2 no se generen calendarios factibles para algun *conjunto de esquemas***, para estos casos la fase 3 no se ejecutaría. Se asignarán *equipos* reales a cada calendario factible cumpliendo las restricciones de los cabeza de serie y de aquellos que se localizan en la misma ciudad.

Modelo de Programación Entera

El modelo de programación entera para esta fase va definida de la siguiente manera:

Sea x una variable binaria descrita como:

$$x_{i,j} = \begin{cases} 1 & \text{Si el esquema } i \text{ es emparejado al } \textit{equipo } j \\ 0 & \text{Si no} \end{cases}$$

$C_{i,h}$ Conjunto que contiene los pares de *esquemas* complementarios. Ver en la tabla 4.1.

$D_{j,j}$ Conjunto que contiene los pares de *equipos* que juegan en la misma ciudad. Ver en la tabla 3.4.

$E_{i,r}$ = Conjunto de *esquemas* que no pueden ser emparejados a un equipo cabeza de serie β si el *esquema* i ya ha sido emparejado con otro equipo cabeza de serie α. Esta variable ayuda a cumplir la restricción de que los partidos de los equipos cabeza de serie no pueden jugarse en las primeras y últimas γ semanas.

F_i = Conjunto de *esquemas* que no pueden ser emparejados a un equipo dado δ si el *esquema* i es emparejado a otro equipo dad ϵ el cual esta localizado en la misma ciudad de δ.Esta variable ayuda a cumplir la restricción de que los partidos de los equipos cabeza de la misma ciudad no pueden jugarse en las primeras y últimas γ semanas.

$T(j)$ = Conjunto de equipos cabezas de serie. Ver en la tabla 3.3

La función objetivo maximiza la función de costos relacionada a los requerimientos mencionados en el punto 5 de la sección 2.

$$min \qquad z = \sum_i \sum_j x_{i,j}$$

sujeto a las siguientes restricciones:

- Cada *esquema* es emparejado con exactamente un *equipo*.

$$\sum_i x_{i,j} = 1; \qquad \forall j, \tag{3.11}$$

- Cada *equipo* es emparejado con exactamente un *esquema*.

$$\sum_{j} x_{i,j} = 1; \qquad \forall i, \tag{3.12}$$

- Los partidos entre los equipos cabezas de serie, no pueden ocurrir en las primeras y últimas γ semanas del calendario.

$$x_{i,j} + \sum_{r \in E_{i,r}} x_{l,m} \leq 1; \qquad \forall i, m \in (T/j), \tag{3.13}$$

- Los partidos entre los equipos de una misma ciudad, no pueden ocurrir en las primeras y últimas γ semanas del calendario.

$$x_{i,j} + \sum_{r} x_{r,m} \leq 1; \qquad \forall i, (j,m) \in D, r \in F_{i,r}, \tag{3.14}$$

- Cada equipo localizado en la misma ciudad requiere *esquemas* complementarios.

$$x_{i,j} - x_{l,m} = 0; \qquad \forall (i,l) \in C, (j,m) \in D, \tag{3.15}$$

- Especificación de variables binarias.

$$x_{i,j} \in o, 1 \qquad \forall i, j \tag{3.16}$$

Capítulo 4

Implementación computacional

4.1. Entorno GAMS

El Sistema General de Modelaje Algebraico (GAMS) está diseñado específicamente para modelar problemas de optimización tanto lineales, no lineales o de enteros mezclados. El sistema es especialmente útil para problemas que sean grandes y complejos. GAMS está disponible en versiones para computadores personales, estaciones de trabajo, bases de datos y súper computadores.

GAMS le permite al usuario concentrarse en el problema a modelar haciendo que el planteamiento del problema sea simple. El sistema se toma el trabajo en los detalles que consumen más tiempo de maquinas especificas e implementación de software.

GAMS es especialmente útil para problemas únicos que sean grandes y complejos que pueden necesitar muchas revisiones antes de establecer el modelo final. El sistema modela problemas en una manera compacta y natural. El usuario puede cambiar la formulación del problema con facilidad, cambiar de un tipo de solucionador a otro y hasta convertir el problema de lineal a no lineal sin problemas.

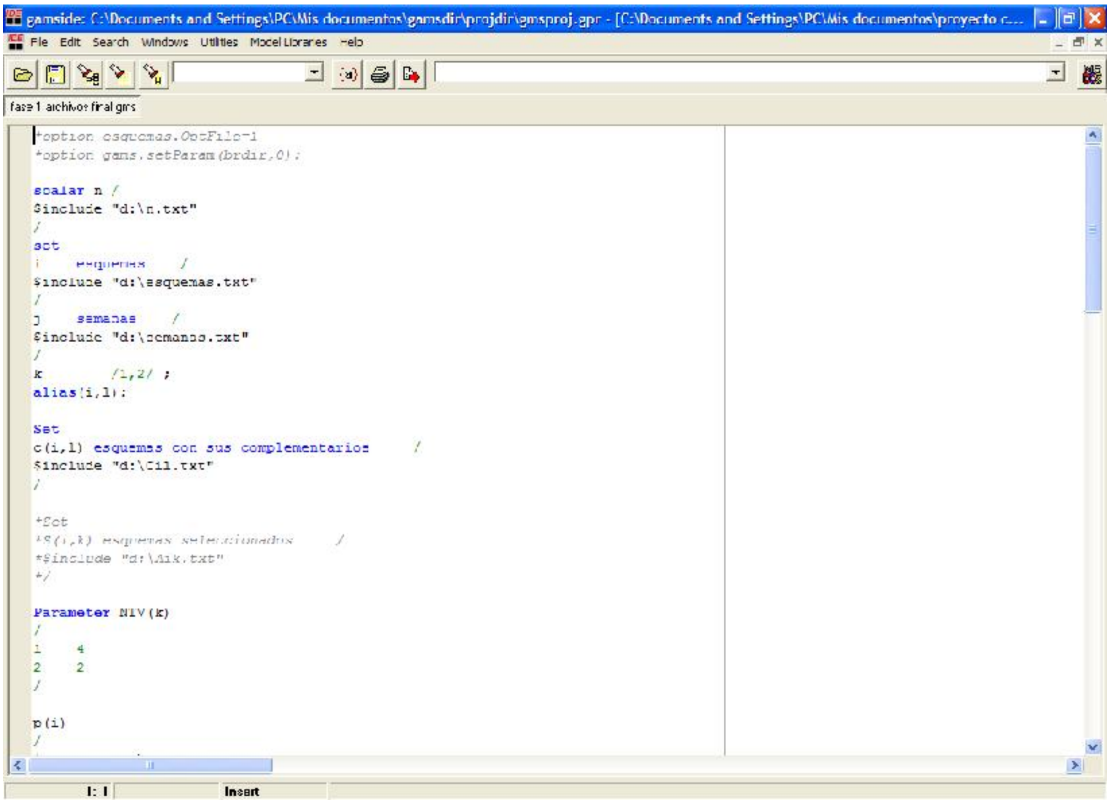


Figura 4.1: Entorno de GAMS

4.1.1. Vista General del Sistema y Características Importantes

El modelado y optimización de *GAMS* se basa en una arquitectura abierta, que permita comunicación libre de irregularidades con constitución integrados (por ejemplo, solucionadores de optimización) y sistemas externos.

Las aplicaciones de GAMS son portátiles a través de diferentes plataformas (incluyendo Windows de 32/64bit, Linux, Mac OS X, AIX, HP-UX, Solaris).

GAMS provee un portafolio único en su género de solucionadores con la última tecnología en solucionadores y también con solucionadores especiales para procesos estocásticos y optimización global.

Características Importantes

- Tecnología de modelaje robusta y dimensionable.
- Hecho a la medida para aplicaciones complejas, de modelaje a gran escala.
- Aumento de productividad a través de un ambiente de desarrollo eficiente.
- Una ancha red académica y comercial.
- Más de 30 años de experiencia en la industria y academia.

Tipos de Modelos Básicos

- Programas de Enteros Lineales/Cuadráticos Mezclados (MIP/MIQCP).
- Programa de Enteros No Lineales Mezclados (MINLP).
- Problemas Complementarios Mezclados (MCP).
- Programas Matemáticos con Equilibrio de Constantes (MPEC).
- Sistemas No Lineales constreñidos (CNS).

La arquitectura abierta de GAMS asegura una integración fácil de los modelos de optimización en toda clase de ambientes aplicación.

Interfaces y herramientas de Conectividad

- Ejecución de modelos interactiva y orientada.
- Ejecución distribuida (computación de gradillas).
- Intercambio de datos con DBMS, MS-Office, Matlab, etc.
- Componentes de librería con interfaces para C++, Java, .NET, etc.

4.2. Entorno Mathematica

Mathematica es una herramienta especializada en análisis numérico y cálculo simbólico, que incorpora un potente lenguaje de programación propio y una interfaz externa que permite salidas a C, Fortran y TEX, además de otras potentes comunicaciones con otros paquetes mediante MathLink.

4.2.1. Vista General del Sistema y Características Importantes

Visión General

Ingenieros, científicos, analistas financieros, investigadores, profesores y estudiantes de enseñanza superior, usan en todo el mundo Mathematica para desarrollar sus cálculos

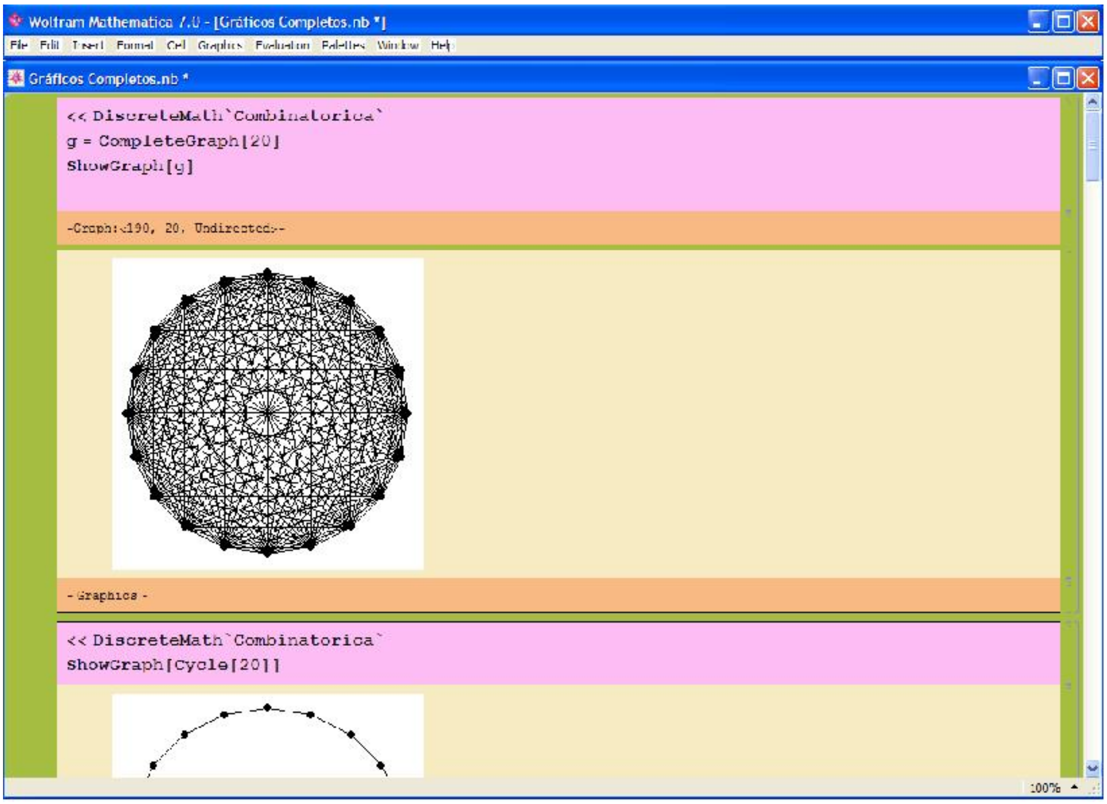


Figura 4.2: Entorno de Mathematica

de precisión en proyectos críticos. Mathematica es el primer programa para la computación y visualización numérica, simbólica y gráfica. Mathematica ofrece a sus usuarios una herramienta interactiva de cálculo y un versátil lenguaje de programación para una rápida y precisa solución a problemas técnicos.

Los documentos electrónicos de Mathematica, llamados notebooks le permiten organizar de forma fácil sus textos, cálculos gráficos y animaciones para impresionantes informes técnicos, courseware, presentaciones o registro de su trabajo. Y además puede usar el protocolo de comunicación de Mathematica, MathLink, para intercambiar información entre Mathematica y otros programas.

Características Principales

Realización de cálculos y simulaciones de cualquier nivel de complejidad mediante el uso de la amplia librería de funciones matemáticas y computacionales. Rápida y fácil importación y exportación de datos, que incluye imágenes y sonido, en más de veinte formatos.

Generación de documentos interactivos, independientes de la plataforma, con textos, imágenes, expresiones matemáticas, botones e hyperlinks.

Entrada de expresiones a través del teclado o de la paleta (programable) más adecuada. Construcción de complejas expresiones y fórmulas con formato automático y ruptura de líneas.
Exportación de los "notebooks" a formato HTML para presentaciones web o LaTeX para publicaciones especiales.

Áreas de Aplicación

Por sus amplias capacidades de cálculo, gráficos y sonidos, tiene interesantísimas aplicaciones en los siguientes campos:

- *Ciencias Físicas:* Física teórica y experimental, química, ciencias de los materiales y ciencias de la tierra.

- *Ciencias de la Computación:* Ingeniería informática, desarrollo de software, gráficos.

- *Ciencias Matemáticas:* Matemática pura, matemática aplicada, estadística, investigación operativa, etc. Negocios y Finanzas: Análisis financiero, economía, ciencias actuariales, gestión.
- *Ciencias de la Salud:* Investigación médica, biología, bioquímica, psicología y ciencias del entorno.

Versiones

La última versión de Mathematica es Mathematica 7.0. Para conocer detalladamente las características y precios de esta última versión visite la información correspondiente a través del enlace en la parte inferior de esta página.

Información del fabricante

Mathematica ha sido desarrollado por la empresa Wolfram Research Inc., fundada en 1987 por Stephen Wolfram. Wolfram Research Inc. se ha caracterizado durante toda su historia por ser una compañía líder en el desarrollo de herramientas de gran calidad para el cálculo científico y técnico y por la incorporación de una tecnología de computación propia e innovadora. Addlink Software Científico, desde su fundación en 1991 ostenta el título de distribuidor oficial certificado de los productos de Wolfram Research. Esta calificación garantiza un alto nivel de calidad en: asesoramiento comercial que permite una optimización de la inversión gracias a un profundo conocimiento de los productos, tipos de configuración, licencias y precios; soporte técnico y atención al cliente, a través de un equipo técnico altamente cualificado en los productos, sistemas y áreas de conocimiento.

4.3. Proceso de Ejecución usando los Software Gams y Wolfram Mathematica

En la realización para encontrar una solución al problema de la calendarización de deportes, hemos definido tres modelos matemáticos, los cuales estan descritos en el capitulo anterior. Una vez definido el modelo procedemos al uso de las herramientas computacionales, para estas usaremos las siguientes herramientas:

- Sistema Operativo: Microsoft Windows XP Profesional Service Pack 3
- Equipo: Inter(R) Pentium(R) 4, CPU 2.80GHz, 704MB de RAM
- Gams (Motor de Optimización)
- Wolfram Mathematica 7.

Como se ha descrito en secciones anteriores, la solución se ha dividido en tres fases, las cuales de igual manera la realizaremos usando los software.

4.3.1. Método de ejecución de la FASE 1

Para llevar a cabo esta fase, tuve que utilizar de antemano *Wolfram Mathematica 7* para poder elaborar una función con parámetro n que me permita generar el listado de todos los diferentes *esquemas* de acuerdo al número de *equipos*, además esta función elabora los datos de entrada para uso de Gams, como el conjunto los pares de esquemas complementarios $C_{i,l}$ (Tabla 4.1), la matriz $A_{i,j}$ (Tabla 6.1) y la tabla p_i (Tabla 4.2) que contiene el número de *quiebres* de cada *esquema*.

En la figura 4.3 se muestra una parte de la programación realizada en Mathematica.

(Ver ANEXOS)

```
FASE1[n_] := (
conjunto = Take[Flatten[Table[{"A", "A", "H"},
{i, 1, 2^10}]], n - 1];
M = Flatten[ Table[{"A", "H"},
{StringCount[StringJoin[conjunto], {"A"}]}]];
Esquemas = Permutations[M, {n - 1}];
   visitante = {};

For[i = 1, i <= Length[Esquemas]/2, i++,
      If[StringCount[StringJoin[Esquemas[[i]]], {"AAA", "HHH"}] == 0 &&
      StringCount[StringJoin[Esquemas[[i]]], {"AA", "HH"}] <= 2,
     p = Esquemas[[i]]; visitante = Append[visitante, p]]];
                    .
                    .
                    .
```

Figura 4.3: Programación de la Fase 1 en Wolfram Mathematica

DATOS DE SALIDA

La tabla 4.1 a continuación, representa el conjunto de esquemas junto a su complementario, por ejemplo, **1.48** significa, que el esquema 1 tiene como complementario al esquema 48.

Esquemas Complementarios

1	.	48	13	.	60	25	.	72	37	.	84
2	.	49	14	.	61	26	.	73	38	.	85
3	.	50	15	.	62	27	.	74	39	.	86
4	.	51	16	.	63	28	.	75	40	.	87
5	.	52	17	.	64	29	.	76	41	.	88
6	.	53	18	.	65	30	.	77	42	.	89
7	.	54	19	.	66	31	.	78	43	.	90
8	.	55	20	.	67	32	.	79	44	.	91
9	.	56	21	.	68	33	.	80	45	.	92
10	.	57	22	.	69	34	.	81	46	.	93
11	.	58	23	.	70	35	.	82	47	.	94
12	.	59	24	.	71	36	.	83			

Tabla 4.1: Tabla de datos $C_{i,l}$ generados en Wolfram Mathematica

En la tabla 4.2, contiene el número total de *quiebres* para cada esquema i, como se puede observar en la misma, el esquema **94** tiene **2** *quiebres*.

1	0	**20**	2	**39**	2	**58**	1	**77**	2
2	1	**21**	1	**40**	2	**59**	2	**78**	2
3	1	**22**	2	**41**	2	**60**	2	**79**	2
4	2	**23**	2	**42**	2	**61**	2	**80**	2
5	1	**24**	2	**43**	1	**62**	1	**81**	2
6	2	**25**	2	**44**	2	**63**	2	**82**	1
7	1	**26**	2	**45**	2	**64**	2	**83**	2
8	2	**27**	1	**46**	2	**65**	2	**84**	2
9	2	**28**	2	**47**	2	**66**	2	**85**	2
10	2	**29**	2	**48**	0	**67**	2	**86**	2
11	1	**30**	2	**49**	1	**68**	1	**87**	2
12	2	**31**	2	**50**	1	**69**	2	**88**	2
13	2	**32**	2	**51**	2	**70**	2	**89**	2
14	2	**33**	2	**52**	1	**71**	2	**90**	1
15	1	**34**	2	**53**	2	**72**	2	**91**	2
16	2	**35**	1	**54**	1	**73**	2	**92**	2
17	2	**36**	2	**55**	2	**74**	1	**93**	2
18	2	**37**	2	**56**	2	**75**	2	**94**	2
19	2	**38**	2	**57**	2	**76**	2		

Tabla 4.2: Tabla de datos P_i generados en Wolfram Mathematica

Una vez obtenido los datos de entrada para GAMS, se procedio a introducir el modelo matemático analizado en el capitulo anterior, con las respectivas sintaxis que se emplea en Gams para poder seleccionar los conjuntos de esquemas factibles; a continuación presento los datos de salida y una pequeña parte de la programcación realizada en GAMS (Ver ANEXOS):

```
*Fase 1 Creado por Jorge Morales
scalar n /
$include "d:\n.txt"
/
set  i    esquemas    /
$include "d:\esquemas.txt"
/
j    semanas    /
$include "d:\semanas.txt"
/
k         /1,2,3/ ;
alias(i,l);
Set
c(i,l) esquemas con sus complementarios     /
$include "d:\Cil.txt"
/
*Set
*S(i,k) esquemas seleccionados     /
*$include "d:\Aik.txt"
*/
Parameter NTV(k) /
1    4
2    6
3    2
/
p(i)    /
$include "d:\pi.txt"
/
                    .
                    .
                    .
```

Figura 4.4: Programación de la Fase 1 en GAMS

DATOS DE SALIDA

Los esquemas que serán seleccionados a las TVs se muestra en la figura 4.5, por ejemplo, el par **5.3** indica que el esquema **5** es asignado a la TV # 3.

1 . 1	48 . 1
3 . 2	50 . 2
5 . 3	52 . 3
11 . 2	58 . 2
21 . 1	68 . 1
35 . 2	82 . 2

Figura 4.5: Esquemas Seleccionados a las TVs

La secuencia de cada esquema i, se presenta en la figura 4.6, sea 1 si el esquema juega de local, y 0 si juega de visitante.

	1	2	3	4	5	6	7	8	9	10	11
1	1	0	1	0	1	0	1	0	1	0	1
3	1	0	1	0	1	0	1	0	1	1	0
5	1	0	1	0	1	0	1	0	0	1	0
11	1	0	1	0	1	0	0	1	0	1	0
21	1	0	1	0	0	1	0	1	0	1	0
35	1	0	0	1	0	1	0	1	0	1	0
48	0	1	0	1	0	1	0	1	0	1	0
50	0	1	0	1	0	1	0	1	0	0	1
52	0	1	0	1	0	1	0	1	1	0	1
58	0	1	0	1	0	1	1	0	1	0	1
68	0	1	0	1	1	0	1	0	1	0	1
82	0	1	1	0	1	0	1	0	1	0	1

Figura 4.6: Esquemas Seleccionados

4.3.2. Método de ejecución de la FASE 2

Para esta fase trabajaremos solo con conjuntos relacionados a los esquemas seleccionados en la fase 1, en esta fase al correr el modelo matemático en Gams, da la posibilidad de que resulte o exista infactibilidad, por lo tanto se deberia introducir la restricción 3.6 la que nos ayudará a generar otro conjunto de esquemas factibles, de cierta manera diferente a la anterior.

En Mathematica se generó la Matriz $d_{i,j,t}$ basada a los esquemas factibles generados en la Fase 1, para luego introducirlo en el modelo matemático y sea ejecutado en Gams con el optimizador XPRESS o CPLEX; a continuación presento una parte de la progrmación realizada en Mathematica 4.7 y Gams 4.8(Ver Anexos) junto con los datos de salida de Gams en esta Fase 2.

```
FASE2[n_] := (
   seleccionados = Import["d:seleccionados.txt", "Table"];
   jlme = {};
   For[i = 1, i <= Length[AH], i++,
       For[j = 1, j <= n, j++,
           If[i == seleccionados[[j]][[1]], jorge = AH[[i]];
       jlme = Append[jlme, jorge]]]]
    espol = Prepend[Transpose[ Prepend[Transpose[jlme],
    Flatten[Take[seleccionados, -n, 1]]]], Range[0, n - 1]];
   espol[[1, 1]] = Null;
                 .
                 .
                 .
```

Figura 4.7: Programación de la Fase 2 en Wolfram Mathematica

```
*FASE 2 Creado por Jorge Morales
Sets
i    esquemas    /
$include "d:\seleccionados.txt"
/
t    semanas    /
$include "d:\semanas.txt"
/
alias (i,j);
sets
d(i,j,t)      /
$include "d:\dijt.txt"
/

Variables
x(i,j,t) si el esquema i es emparejado con el esquema j en la semana t
z2       artificial ;
Binary Variable x ;
                    .
                    .
                    .
```

Figura 4.8: Programación de la Fase 2 en GAMS

4.3.3. Método de ejecución de la FASE 3

En esta última fase el objetivo principal es de emparejar los esquemas seleccionados con anterioridad a los respectivos equipos, unos de los problemas que tuve en esta fase, era de definir los parámetros que se requieren, los cuales indican el número de las primeras y últimas fechas que no se pueden enfrentar ciertos equipos, como por ejemplo Barcelona vs Emelec, entre otros.

Una vez obtenido un calendario factible en la Fase 2, el siguiente paso definir que esquema se le asigna a los distintos equipos. En esta fase el dato principal de salida de GAMS es dicha asignación la cual se la presenta en la figura 2.8.

Adicionalmente podemos mostrar parte de esta programación en la figura 4.9 y 4.10.

```
FASE3[n_, Lambda_, Beta_] := (
   Cih = {};
   For[i = 1, i <= Length[Cil], i++,
       For[j = 1, j <= Length[seleccionados], j++,
     If[Cil[[i]][[1]] == seleccionados[[j]][[1]], tomar = Cil[[i]];
      Cih = Append[Cih, tomar]]]];
   calendario = Import["d:calendario.txt", "Table"];
   Eir = {};
   Fir = {};
   For[i = 1, i <= Length[calendario], i++,
      For[j = 1, j <= Lambda, j++,
        If[calendario[[i]][[5]] == j || calendario[[i]][[5]] == n - j,
        sel = calendario[[i]]; Eir = Append[Eir, sel]]]]
                    .
                    .
                    .
```

Figura 4.9: Programación de la Fase 3 en Wolfram Mathematica

```
*FASE 3 Creado por Jorge Morales
Sets
i   esquemas      /
$include "d:\seleccionados.txt"
/
j      equipos  / BARCELONA, EMELEC, DEP_QUITO, EL_NACIONAL, ESPOLI,
                  LIGA_DE_QUITO, DEP_CUENCA, IMBABURA, INDEPENDIENTE,
                  LIGA_DE_LOJA, MANTA, OLMEDO/
T(j)   Equipos cabeza de serie / EMELEC, LIGA_DE_QUITO, BARCELONA,
 DEP_QUITO/
k        TVS's         / 1,2,3 /
alias (i,h)
alias (i,r)
alias (j,m)
Set
A(i,k)   Conjunto de esquemas asignados a la televisora k        /
$include "d:\Aik.txt"
/
                    .
                    .
                    .
```

Figura 4.10: Programación de la Fase 3 en GAMS

Finalmente en la tabla 4.3, se observa el emparejamiento de los equipos a sus esquemas correspondientes.

Descripción
Barcelona - 1
Emelec - 50
Dep. Quito - 52
El Nacional - 1
Espoli - 82
LDU Q - 5
Dep. Cuenca - 48
Imbabura - 68
Independiente - 21
LDU Loja - 11
Manta - 35
Olmedo - 58

Tabla 4.3: Emparejamiento de Equipos a esquemas

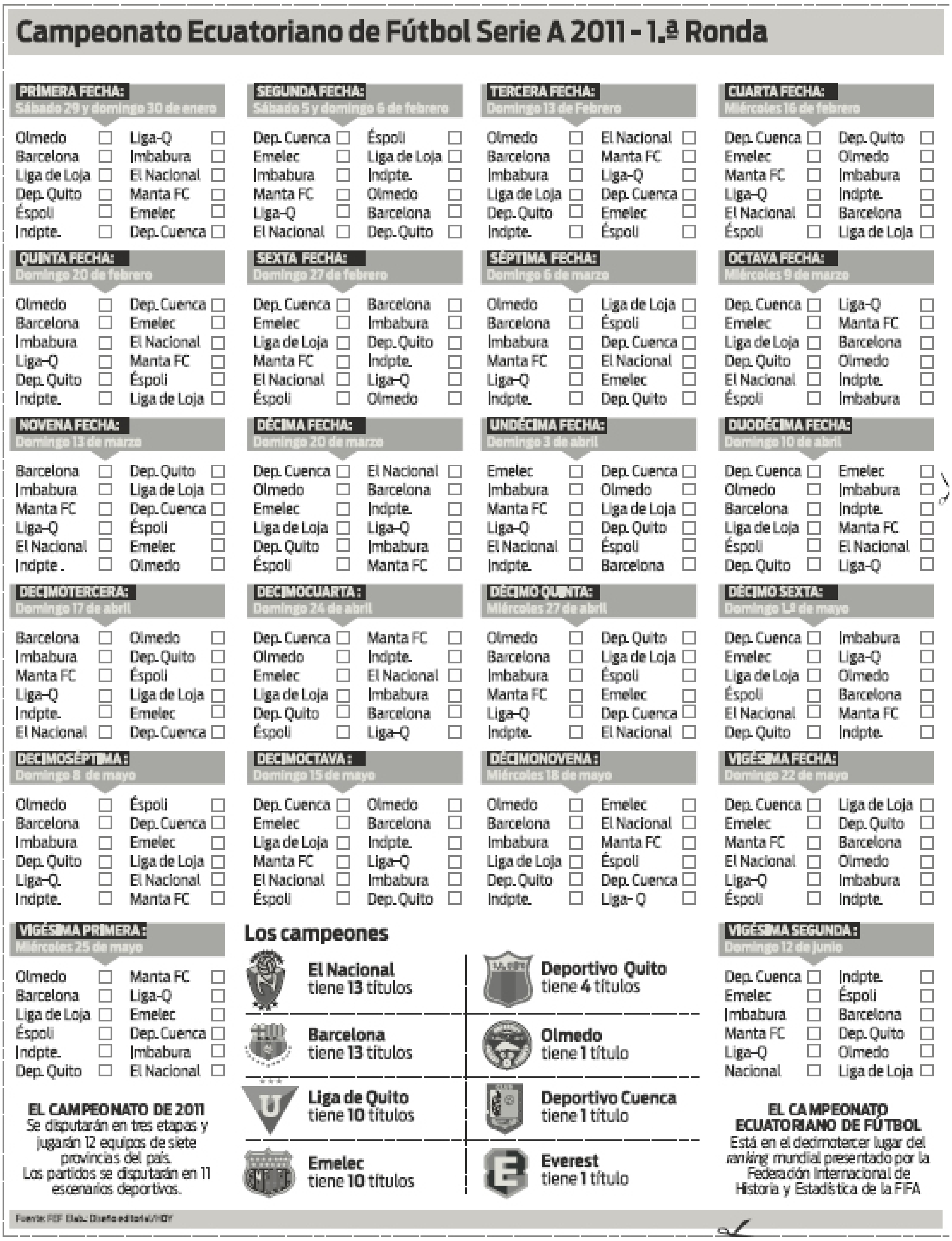
4.4. Comparación del Calendario Propuesto vs Original

En esta sección se campara la eficiencia de este trabajo en base al calendario que fue utilizado en el año 2011.

De antemano se presenta el Calendario Propuesto, bajo la metodología desarrollada en este trabajo (Ver Figura 4.11), junto al Calendario Oficial del Campeonato Ecuatoriano de Fútbol Serie A 2011 (Ver Figura 4.12), correspondientes a las 22 fechas que se desarrolló; a partir de la fecha o semana 12 representa los partidos de revancha (de vuelta) de la primera ronda.

Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6
El Nac vs Espoli	D. Cuenca vs Indep	El Nac vs D. Cuenca	Manta vs L. Loja	El Nac vs D. Quito	Indep vs Bar
Bar vs Olm	Eme vs El Nac	Bar vs Eme	D. Cuenca vs LDU Q	Bar vs D. Cuenca	Manta vs Espoli
LDU Q vs Imba	D. Quito vs Manta	LDU Q vs D. Quito	Eme vs Indep	LDU Q vs Manta	D. Cuenca vs L. Loja
L. Loja vs Eme	Olm vs LDU Q	L. Loja vs Olm	D. Quito vs Bar	L. Loja vs Indep	Eme vs LDU Q
Indep vs D. Quito	Imba vs Bar	Indep vs Manta	Olm vs Espoli	Imba vs Olm	D. Quito vs Imba
Manta vs D. Cuenca	Espoli vs L. Loja	Espoli vs Imba	Imba vs El Nac	Espoli vs Eme	Olm vs El Nac
Semana 7	**Semana 8**	**Semana 9**	**Semana 10**	**Semana 11**	**Semana 12**
El Nac vs Manta	L. Loja vs LDU Q	El Nac vs L. Loja	Bar vs El Nac	El Nac vs LDU Q	LDU Q vs El Nac
Bar vs L. Loja	Indep vs El Nac	Bar vs LDU Q	LDU Q vs Espoli	Eme vs Manta	Manta vs Eme
LDU Q vs Indep	Manta vs Bar	D. Quito vs Eme	L. Loja vs D. Quito	D. Quito vs D. Cuenca	D. Cuenca vs D. Quito
Olm vs Eme	D. Cuenca vs Espoli	Olm vs D. Cuenca	Indep vs Imba	Olm vs Indep	Indep vs Olm
Imba vs D. Cuenca	Eme vs Imba	Imba vs Manta	Manta vs Olm	Imba vs L. Loja	L. Loja vs Imba
Espoli vs D. Quito	D. Quito vs Olm	Espoli vs Indep	D. Cuenca vs Eme	Espoli vs Bar	Bar vs Espoli
Semana 13	**Semana 14**	**Semana 15**	**Semana 16**	**Semana 17**	**Semana 18**
El Nac vs Bar	L. Loja vs El Nac	LDU Q vs L. Loja	Manta vs El Nac	Bar vs Indep	D. Quito vs El Nac
Espoli vs LDU Q	LDU Q vs Bar	El Nac vs Indep	L. Loja vs Bar	Espoli vs Manta	D. Cuenca vs Bar
D. Quito vs L. Loja	Eme vs D. Quito	Bar vs Manta	Indep vs LDU Q	L. Loja vs D. Cuenca	Manta vs LDU Q
Imba vs Indep	D. Cuenca vs Olm	Espoli vs D. Cuenca	Eme vs Olm	LDU Q vs Eme	Indep vs L. Loja
Olm vs Manta	Manta vs Imba	Imba vs Eme	D. Cuenca vs Imba	Imba vs D. Quito	Olm vs Imba
Eme vs D. Cuenca	Indep vs Espoli	Olm vs D. Quito	D. Quito vs Espoli	El Nac vs Olm	Eme vs Espoli
Semana 19	**Semana 20**	**Semana 21**	**Semana 22**		
L. Loja vs Manta	D. Cuenca vs El Nac	Indep vs D. Cuenca	Espoli vs El Nac		
LDU Q vs D. Cuenca	Eme vs Bar	El Nac vs Eme	Olm vs Bar		
Indep vs Eme	D. Quito vs LDU Q	Manta vs D. Quito	Imba vs LDU Q		
Bar vs D. Quito	Olm vs L. Loja	LDU Q vs Olm	Eme vs L. Loja		
Espoli vs Olm	Manta vs Indep	Bar vs Imba	D. Quito vs Indep		
El Nac vs Imba	Imba vs Espoli	L. Loja vs Espoli	D. Cuenca vs Manta		

Figura 4.11: Calendario Propuesto en este trabajo



Campeonato Ecuatoriano de Fútbol Serie A 2011 - 1.ª Ronda

PRIMERA FECHA: Sábado 29 y domingo 30 de enero

Olmedo	Liga-Q
Barcelona	Imbabura
Liga de Loja	El Nacional
Dep. Quito	Manta FC
Éspoli	Emelec
Indpte.	Dep. Cuenca

SEGUNDA FECHA: Sábado 5 y domingo 6 de febrero

Dep. Cuenca	Éspoli
Emelec	Liga de Loja
Imbabura	Indpte.
Manta FC	Olmedo
Liga-Q	Barcelona
El Nacional	Dep. Quito

TERCERA FECHA: Domingo 13 de Febrero

Olmedo	El Nacional
Barcelona	Manta FC
Imbabura	Liga-Q
Liga de Loja	Dep. Cuenca
Dep. Quito	Emelec
Indpte.	Éspoli

CUARTA FECHA: Miércoles 16 de febrero

Dep. Cuenca	Dep. Quito
Emelec	Olmedo
Manta FC	Imbabura
Liga-Q	Indpte.
El Nacional	Barcelona
Éspoli	Liga de Loja

QUINTA FECHA: Domingo 20 de febrero

Olmedo	Dep. Cuenca
Barcelona	Emelec
Imbabura	El Nacional
Liga-Q	Manta FC
Dep. Quito	Éspoli
Indpte.	Liga de Loja

SEXTA FECHA: Domingo 27 de febrero

Dep. Cuenca	Barcelona
Emelec	Imbabura
Liga de Loja	Dep. Quito
Manta FC	Indpte.
El Nacional	Liga-Q
Éspoli	Olmedo

SÉPTIMA FECHA: Domingo 6 de marzo

Olmedo	Liga de Loja
Barcelona	Éspoli
Imbabura	Dep. Cuenca
Manta FC	El Nacional
Liga-Q	Emelec
Indpte.	Dep. Quito

OCTAVA FECHA: Miércoles 9 de marzo

Dep. Cuenca	Liga-Q
Emelec	Manta FC
Liga de Loja	Barcelona
Dep. Quito	Olmedo
El Nacional	Indpte.
Éspoli	Imbabura

NOVENA FECHA: Domingo 13 de marzo

Barcelona	Dep. Quito
Imbabura	Liga de Loja
Manta FC	Dep. Cuenca
Liga-Q	Éspoli
El Nacional	Emelec
Indpte .	Olmedo

DÉCIMA FECHA: Domingo 20 de marzo

Dep. Cuenca	El Nacional
Olmedo	Barcelona
Emelec	Indpte.
Liga de Loja	Liga-Q
Dep. Quito	Imbabura
Éspoli	Manta FC

UNDÉCIMA FECHA: Domingo 3 de abril

Emelec	Dep. Cuenca
Imbabura	Olmedo
Manta FC	Liga de Loja
Liga-Q	Dep. Quito
El Nacional	Éspoli
Indpte.	Barcelona

DUODÉCIMA FECHA: Domingo 10 de abril

Dep. Cuenca	Emelec
Olmedo	Imbabura
Barcelona	Indpte.
Liga de Loja	Manta FC
Éspoli	El Nacional
Dep. Quito	Liga-Q

DECIMOTERCERA: Domingo 17 de abril

Barcelona	Olmedo
Imbabura	Dep. Quito
Manta FC	Éspoli
Liga-Q	Liga de Loja
Indpte.	Emelec
El Nacional	Dep. Cuenca

DECIMOCUARTA: Domingo 24 de abril

Dep. Cuenca	Manta FC
Olmedo	Indpte.
Emelec	El Nacional
Liga de Loja	Imbabura
Dep. Quito	Barcelona
Éspoli	Liga-Q

DÉCIMO QUINTA: Miércoles 27 de abril

Olmedo	Dep. Quito
Barcelona	Liga de Loja
Imbabura	Éspoli
Manta FC	Emelec
Liga-Q	Dep. Cuenca
Indpte.	El Nacional

DÉCIMO SEXTA: Domingo 1.º de mayo

Dep. Cuenca	Imbabura
Emelec	Liga-Q
Liga de Loja	Olmedo
Éspoli	Barcelona
El Nacional	Manta FC
Dep. Quito	Indpte.

DECIMOSÉPTIMA: Domingo 8 de mayo

Olmedo	Éspoli
Barcelona	Dep. Cuenca
Imbabura	Emelec
Dep. Quito	Liga de Loja
Liga-Q.	El Nacional
Indpte.	Manta FC

DECIMOCTAVA: Domingo 15 de mayo

Dep. Cuenca	Olmedo
Emelec	Barcelona
Liga de Loja	Indpte.
Manta FC	Liga-Q
El Nacional	Imbabura
Éspoli	Dep. Quito

DÉCIMONOVENA: Miércoles 18 de mayo

Olmedo	Emelec
Barcelona	El Nacional
Imbabura	Manta FC
Liga de Loja	Éspoli
Dep. Quito	Dep. Cuenca
Indpte.	Liga- Q

VIGÉSIMA FECHA: Domingo 22 de mayo

Dep. Cuenca	Liga de Loja
Emelec	Dep. Quito
Manta FC	Barcelona
El Nacional	Olmedo
Liga-Q	Imbabura
Éspoli	Indpte.

VIGÉSIMA PRIMERA: Miércoles 25 de mayo

Olmedo	Manta FC
Barcelona	Liga-Q
Liga de Loja	Emelec
Éspoli	Dep. Cuenca
Indpte.	Imbabura
Dep. Quito	El Nacional

VIGÉSIMA SEGUNDA: Domingo 12 de junio

Dep. Cuenca	Indpte.
Emelec	Éspoli
Imbabura	Barcelona
Manta FC	Dep. Quito
Liga-Q	Olmedo
Nacional	Liga de Loja

Los campeones

El Nacional tiene 13 títulos

Deportivo Quito tiene 4 títulos

Barcelona tiene 13 títulos

Olmedo tiene 1 título

Liga de Quito tiene 10 títulos

Deportivo Cuenca tiene 1 título

Emelec tiene 10 títulos

Everest tiene 1 título

EL CAMPEONATO DE 2011
Se disputarán en tres etapas y jugarán 12 equipos de siete provincias del país. Los partidos se disputarán en 11 escenarios deportivos.

EL CAMPEONATO ECUATORIANO DE FÚTBOL
Está en el decimotercer lugar del *ranking* mundial presentado por la Federación Internacional de Historia y Estadística de la FIFA

Fuente: FEF Elab.: Diseño editorial/HOY

Figura 4.12: Calendario Original del Campeonato Ecuatoriano de Fútbol Serie A

En las figuras 4.13 y 4.14 se muestra la cantidad de quiebres de ambos calendarios, con el objetivo de escoger el mejor.

En ambos calendarios se ha podido observar que poseen el mismo número de quiebres (partidos consecutivos), con un total de **10**, con la ventaja que el calendario propuesto cubre los recursos de las televisoras (TVs), cuestión que se considera primordial por los posibles cuestionamientos que puedan generar los aficionados de cada equipo.

Matriz de Esquemas	Quiebres del Calendario Original											
	1	2	3	4	5	6	7	8	9	10	11	# Quiebres
Olmedo	1	0	1	0	1	0	1	0	0	1	0	1
Barcelona	1	0	1	0	1	0	1	0	1	0	0	1
Liga de Loja	1	0	1	0	0	1	0	1	0	1	0	1
Dep. Quito	1	0	1	0	1	0	0	1	0	1	0	1
Espoli	1	0	0	1	0	1	0	1	0	1	0	1
Indepte	1	0	1	0	1	0	1	0	1	0	1	0
LDU Q	0	1	0	1	1	0	1	0	1	0	1	1
Imbabura	0	1	1	0	1	0	1	0	1	0	1	1
El Nacional	0	1	0	1	0	1	0	1	1	0	1	1
Manta	0	1	0	1	0	1	1	0	1	0	1	1
Emelec	0	1	0	1	0	1	0	1	0	1	1	1
Dep. Cuenca	0	1	0	1	0	1	0	1	0	1	0	0
								Numero Total de Quiebres				10

Figura 4.13: Números de Quiebres presentados en el Calendario Original

Matriz de Esquemas	Quiebres del Calendario Propuesto											
	1	2	3	4	5	6	7	8	9	10	11	# Quiebres
El Nacional	1	0	1	0	1	0	1	0	1	0	1	0
Barcelona	1	0	1	0	1	0	1	0	1	1	0	1
LDU Q	1	0	1	0	1	0	1	0	0	1	0	1
Liga de Loja	1	0	1	0	1	0	0	1	0	1	0	1
Indepte	1	0	1	0	0	1	0	1	0	1	0	1
Manta	1	0	0	1	0	1	0	1	0	1	0	1
Dep. Cuenca	0	1	0	1	0	1	0	1	0	1	0	0
Emelec	0	1	0	1	0	1	0	1	0	0	1	1
Dep. Quito	0	1	0	1	0	1	0	1	1	0	1	1
Olmedo	0	1	0	1	0	1	1	0	1	0	1	1
Imbabura	0	1	0	1	1	0	1	0	1	0	1	1
Espoli	0	1	1	0	1	0	1	0	1	0	1	1
								Numero Total de Quiebres				10

Figura 4.14: Números de Quiebres presentados en el Calendario Propuesto

En la Figura 4.15 podemos constatarnos que en un promedio de 0.361 seg nos tomaría generar los datos correspondientes a la fase 1, en un promedio de 0.149 seg nos tamaria para ejecutar y extraer los datos correspondientes de la Fase 2 y finalmente en 0.050 seg en promedio para obtener los datos de la fase 3; es decir solo generando los valores de entrada para cada fase nos tomaría 0.560 seg con una desviación estándar de 0.036 seg en total.

	Ejecuciones realizadas en Wolfram Mathematica											
Tiempo de Corrida(seg)	**Iter 1**	**Iter 2**	**Iter 3**	**Iter 4**	**Iter 5**	**Iter 6**	**Iter 7**	**Iter 8**	**Iter 9**	**Iter 10**	**Promedio**	**Desviación Estandar**
FASE 1	0,422	0,329	0,312	0,343	0,374	0,344	0,406	0,344	0,391	0,344	0,361	0,036
FASE 2	0,157	0,141	0,156	0,156	0,157	0,125	0,156	0,156	0,125	0,157	0,149	0,013
FASE 3	0,032	0,047	0,047	0,062	0,032	0,063	0,063	0,062	0,047	0,047	0,050	0,012
Tiempo Total	***0,611***	***0,517***	***0,515***	***0,561***	***0,563***	***0,532***	***0,625***	***0,562***	***0,563***	***0,548***	***0,560***	***0,036***

Figura 4.15: Tiempos de Ejecución de las fases en Wolfram Mathematica

Capítulo 5

Conclusiones y Recomendaciones

5.1. Conclusiones

1. Ésta metodología, permite generar en cuestiones de minutos nuevos calendarios, disponibles para ser presentado como propuesta en la Federación Ecuatoriana de Fútbol (FEF), enfocadas en restricciones vistas en el Capítulo 3.

2. Basándonos a los recursos de las empresas de TV, podemos concluir que esta metodología les permite planificar sus actividades relacionadas a los derechos de transmisión sin ningún imprevisto de horarios.

3. Por medio de ésta heurística, se permite elaborar calendarios de juegos, dependiendo del número de equipos participantes en el torneo. Cabe señalar, que el nivel de dificultad depende del número de restricciones que ésta contenga.

4. Esta aplicación generaría un impacto positivo tanto a nivel cuantitativo como cualitativo. A pesar de que la medición del impacto cuantitativo no es directa y es prácticamente imposible aislarlo de efectos exógenos, algunas observaciones pueden ser realizadas. Un factor relevante, que significaría ahorros para las TVs

es en la logística que ellos plantearían cuando tiene que cubrir más de dos partidos de local en una determinada fecha (semana).

5.1.1. Recomendaciones

1. Implementar esta metodología, inicialmente en torneos pequeños, para realizar un análisis del mismo, para luego, llegar a plasmarlo en la calendarización de la 1era y 2da División del Campeonato Nacional de Fútbol y otros torneos de interés.

2. Evaluar el impacto que genera esta metodología en base a los recursos de las empresas de TV, tomando en cuenta los ingresos que estas podrían generar por las transmisiones de los partidos y su aumento en competitividad.

3. Teniendo en cuenta, que cada equipo participante en el torneo (Clubes), presentan su propuesta en la calendarización en base a sus propios beneficios, se recomienda que se lleve a cabo una sesión ordinaria del Comite Ejecutivo de la Federación Ecuatoriana de Fútbol, junto a los representantes de cada club, con el fin de realizar en conjunto características propias del torneo, para luego, ingresarlas computacionalmente mediante el método propuesto en este documento.

Capítulo 6

ANEXOS

6.1. Código Fase 1 Mathematica y GAMS

6.1.1. Programación Fase 1 Mathematica

```
FASE1[n_] := (
conjunto = Take[Flatten[Table[{"A", "A", "H"},
{i, 1, 2^10}]], n - 1];
M = Flatten[ Table[{"A", "H"},
{StringCount[StringJoin[conjunto], {"A"}]}]];
Esquemas = Permutations[M, {n - 1}];
   visitante = {};
For[i = 1, i <= Length[Esquemas]/2, i++,
      If[StringCount[StringJoin[Esquemas[[i]]], {"AAA", "HHH"}] == 0 &&
      StringCount[StringJoin[Esquemas[[i]]], {"AA", "HH"}] <= 2,
     p = Esquemas[[i]]; visitante = Append[visitante, p]]];
r1 = {};
For[i = 1, i <= Length[visitante], i++,
 r1 = Append[r1, StringJoin[visitante[[i]]]]];
   complementarios = StringReplace[r1, {"A" -> "H", "H" -> "A"}];
   complementarios = Join[r1, complementarios];
AH = {};
   For[i = 1, i <= Length[complementarios], i++,
     AH = Append[AH, Characters[complementarios[[i]]]]];
pi = Table[If[StringCount[StringJoin[AH[[i]]], {"AA", "HH"}] == 0,
    "0",If[StringCount[StringJoin[AH[[i]]],
  {"AA", "HH"}] == 1, "1","2"]], {i, 1, Length[AH]}];
pi = Transpose[
     Prepend[Transpose[Transpose[{pi}]], Range[1, Length[AH]]]];
   Datos = StringReplace[complementarios, {"A" -> "1", "H" -> "0"}];
   Aij = {};
   For[i = 1, i <= Length[complementarios], i++,
     Aij = Append[Aij, Characters[Datos[[i]]]]];
    Aij = Transpose[
     Prepend[Transpose[Prepend[Aij, Range[1, n - 1]]],
      Range[0, Length[Aij]]]];
   Aij[[1, 1]] = Null;
   NumEsquema = Table[i, {i, Length[pi]}];
```

```
   Semanas = Flatten[Table[i, {i, n - 1}]];
   Cil = Table[{i, ".", i + Length[AH]/2}, {i, Length[AH]/2}];
   Export["d:\\Aij.txt", Aij, "Table"];
   Export["d:\\pi.txt", pi, "Table"];
   Export["d:\\esquemas.txt", NumEsquema, "Table"];
   Export["d:\\Semanas.txt", Semanas, "Table"];
   Export["d:\\Cil.txt", Cil, "Table"];
   Export["d:\\n.txt", n, "Table"];)

Fase1[2]
Time1=TimeUsed[]
```

6.1.2. Programación Fase 2 GAMS

```
*Fase 1 Creado por Jorge Morales
scalar n /
$include "d:\n.txt"
/
set  i    esquemas    /
$include "d:\esquemas.txt"
/
j    semanas    /
$include "d:\semanas.txt"
/
k        /1,2,3/ ;
alias(i,l);
Set
c(i,l) esquemas con sus complementarios     /
$include "d:\Cil.txt"
/
*Set
*S(i,k) esquemas seleccionados     /
*$include "d:\Aik.txt"
*/
Parameter NTV(k) /
1    4
2    6
3    2
/
p(i)    /
$include "d:\pi.txt"
/
table a(i,j)
$include "d:\Aij.txt";
variable
z, x(i,k)
binary variable x;
equation
obj      minimizar
r1 ,r2,r3,r4;
*r5 ;
obj..                    z=e=sum((i,k),p(i)*x(i,k));
r1(k,j)..                sum(i,a(i,j)*x(i,k))=e=NTV(k)/2;
r2(k)..                  sum(i,x(i,k))=e=NTV(k);
r3(i,l,k)$(c(i,l))..     x(i,k)-x(l,k)=e=0;
r4(i)..                  sum(k,x(i,k))=l=1;
*r5..                    sum((i,k)$S(i,k),x(i,k))=l=n/2;
model esquemas / obj, r1, r2, r3, r4
*r5
/
Solve esquemas using mip minimizing z;
display x.l, c

FILE SOLUCION/ d:\Aik.TXT/;
PUT SOLUCION;
LOOP((i,k),
IF (x.l(i,k) eq 1,PUT i.TL:5,put ".":5,put k.tl:5/););

FILE Esquemas_Seleccionados/ e:\seleccionados.TXT/;
PUT Esquemas_Seleccionados;
LOOP((i,k),
IF (x.l(i,k) eq 1,PUT i.TL:5/););

*Ejecución en HTML

FILE html /'Seleccionados_a_las_TV.html'/;
```

```
PUT  html;
PUT  '<H2 ALIGN=CENTER><font color="Red">ESQUEMAS A LAS TV</H2>'/;
PUT  '<H3 ALIGN=CENTER>CANTIDAD DE QUIEBRES: </FONT>',Z.l:8:0/;
PUT  '<p>'/;
PUT  '<table ALIGN=CENTER border="10" cellpadding="20"
cellspacing="0">'/;
PUT  '<tr><th></th>'/;
PUT '<th BGCOLOR=#23238E><FONT COLOR=WHITE>','TV':15,'</th>'/;
PUT  '</tr>'/;
LOOP ((i,k),
 IF (x.L(i,k)EQ 1,PUT '<tr><th BGCOLOR=#23238E><FONT COLOR=WHITE>',
  i.tl,'</th>'/);
 IF ((x.L(i,k)>0),PUT '<th ALIGN=center>',k.TL:15,'</th>'/);
  PUT '</tr>'/;
);

PUT '<table ALIGN=CENTER border="5" cellpadding="14" cellspacing="0">'/;
PUT '<tr><th></th>'/;
LOOP(j, PUT '<th BGCOLOR=#996699><FONT COLOR=WHITE>',j.tl,'</th>'/);
PUT '</tr>'/;
LOOP((i,k),
 IF (x.l(i,k)eq 1,PUT '<tr><th BGCOLOR=#996699><FONT COLOR=WHITE>',
  i.tl,'</th>'/);
    LOOP(j,
    IF ((x.l(i,k)>0),PUT '<td ALIGN=center BGCOLOR=#AAABD7>',
    a(i,j):8:0,'</td>'/);
    );
    PUT '</tr>'/;);
PUT '<H3 ALIGN=CENTER><font color="Blue">ESQUEMAS</H3>'/;
PUT '<p>'/;
PUTCLOSE;
EXECUTE '=shellexecute Seleccionados_a_las_TV.html';
```

6.2. Código Fase 2 Mathematica y GAMS

6.2.1. Programación Fase 2 Mathematica

```
FASE2[n_] := (
   seleccionados = Import["d:seleccionados.txt", "Table"];
   jlme = {};
   For[i = 1, i <= Length[AH], i++,
       For[j = 1, j <= n, j++,
           If[i == seleccionados[[j]][[1]], jorge = AH[[i]];
       jlme = Append[jlme, jorge]]]]
    espol = Prepend[Transpose[
        Prepend[Transpose[jlme], Flatten[Take[
seleccionados, -n, 1]]]],Range[0, n - 1]];
   espol[[1, 1]] = Null;
   jlme2 = {};
   For[i = 2, i <= Length[Aij], i++,
        For[j = 1, j <= n, j++,
            If[(i - 1) == seleccionados[[j]][[1]], jorge2 = Aij[[i]];
       jlme2 = Append[jlme2, jorge2]]]]
    jlme2 = Prepend[jlme2, Range[0, n - 1]];
   dijt = {};
   For[k = 2, k <= n + 1, k++,
      For[j = 2, j <= n + 1, j++,
          For[i = 2, i <= n, i++,
           If[jlme2[[k]][[i]] != jlme2[[j]][[i]],
             gye = {jlme2[[k]][[1]], ".", jlme2[[j]][[1]], ".",
             jlme2[[1]][[i]]}; dijt = Append[dijt, gye]]]]]
    jlme = Prepend[jlme, Range[1, n - 1]];
   jlme = Transpose[Prepend[Transpose[jlme], Transpose[jlme2][[1]]]];
   Export["d:\\dijt.txt", dijt, "Table"];)
(*Ejecución*)
Fase2[2]
Time2=TimeUsed[]
```

6.2.2. Programación Fase 2 GAMS

```
*FASE 2 Creado por Jorge Morales
Sets
i    esquemas    /
$include "d:\seleccionados.txt"
/
t    semanas    /
$include "d:\semanas.txt"
/
alias (i,j);
sets
d(i,j,t)      /
$include "d:\dijt.txt"
/
Variables
x(i,j,t)  si el esquema i es emparejado con el esquema j en la semana t
z2        artificial ;
Binary Variable x ;
Equations
FO              función objetivo
r1(i,t)
r2(i,j,t)
r3(i,j)
;
FO..                                 z2  =e=  1 ;
r1(i,t)..                            sum(j$d(i,j,t), x(i,j,t))=e=1;
r2(i,j,t)$(ord(j) > ord(i)and d(i,j,t))..  x(i,j,t) - x(j,i,t)=e=0;
r3(i,j)$(ord(j) ne ord(i))..         sum(t$d(i,j,t), x(i,j,t))=e=1;

Model fase2 /all/ ;
Solve fase2 using mip min z2 ;
Display x.l, i, t,d,Z2.L ;

FILE SOLUCION/ d:\calendario.TXT/;
PUT SOLUCION;
LOOP((i,j,t),
IF (x.l(i,j,t) eq 1,PUT i.TL:5,put ".":5,put j.tl:5,put ".":5,
                    put t.tl:5/);););
*Ejecución en HTML
FILE html /'calendario.html'/;
PUT html;
PUT html;
PUT '<H2 ALIGN=CENTER><font color="ORANGE">CALENDARIO FACTIBLE</H2>'/;
PUT '<H2 ALIGN=CENTER>RESULTADOS</H2>'/;
PUT '<p>'/;
PUT '<table ALIGN=CENTER border="5" cellpadding="10" cellspacing="0">'/;
PUT '<tr><th></th>'/;
LOOP(t, PUT '<th BGCOLOR=#996699><FONT COLOR=WHITE>',t.tl,'</th>'/);
PUT '</tr>'/;
LOOP((i,j),
  IF (ord(j) ne ord(i),PUT '<tr><th BGCOLOR=#996699>
  <FONT COLOR=WHITE>', i.tl,'</th>'/);
  LOOP((t),
  IF ((x.l(i,j,t) eq 1),PUT '<td ALIGN=center BGCOLOR=#AAABD7>',
                           i.tl'vs'j.tl,'</td>'/);
  IF ((x.l(i,j,t)=0.0),PUT '<td ALIGN=center>','','</td>'/);
  );
  PUT '</tr>'/;);
```

```
PUT '</table>'/;
PUTCLOSE;
EXECUTE '=shellexecute calendario.html';
```

6.3. Código Fase 3 Mathematica y GAMS

6.3.1. Programación Fase 3 Mathematica

```
FASE3[n_, Lambda_, Beta_] := (
   Cih = {};
   For[i = 1, i <= Length[Cil], i++,
       For[j = 1, j <= Length[seleccionados], j++,
     If[Cil[[i]][[1]] == seleccionados[[j]][[1]], tomar = Cil[[i]];
      Cih = Append[Cih, tomar]]]];
   calendario = Import["d:calendario.txt", "Table"];
   Eir = {};
   Fir = {};
   For[i = 1, i <= Length[calendario], i++,
        For[j = 1, j <= Lambda, j++,
          If[calendario[[i]][[5]] == j || calendario[[i]][[5]] == n - j,
       sel = calendario[[i]]; Eir = Append[Eir, sel]]]]
    For[i = 1, i <= Length[calendario], i++,
        For[j = 1, j <= Beta, j++,
          If[calendario[[i]][[5]] == j || calendario[[i]][[5]] == n - j,
       sel1 = calendario[[i]]; Fir = Append[Fir, sel1]]]]
    Export["d:\\Eir.txt", Take[Eir, -Length[Eir], 3], "Table"];
   Export["d:\\Fir.txt", Take[Fir, -Length[Fir], 3], "Table"];
   Export["d:\\Cih.txt", Cih, "Table"];)

(*Ejecución*)
Fase3[12,2,2]
Time3=TimeUsed[]
```

6.3.2. Programación Fase 3 GAMS

```
*FASE 3 Creado por Jorge Morales
Sets
i   esquemas      /
$include "d:\seleccionados.txt"
/
j       equipos  / BARCELONA, EMELEC, DEP_QUITO, EL_NACIONAL, ESPOLI,
                   LIGA_DE_QUITO, DEP_CUENCA, IMBABURA, INDEPENDIENTE,
                   LIGA_DE_LOJA, MANTA, OLMEDO/
T(j)     Equipos cabeza de serie / EMELEC, LIGA_DE_QUITO, BARCELONA,
 DEP_QUITO/
k        TVS's          / 1,2,3 /
alias (i,h)
alias (i,r)
alias (j,m)
Set
A(i,k)   Conjunto de esquemas asignados a la televisora k        /
$include "d:\Aik.txt"
/
B(j,k)   Conjunto de equipos asignados a la televisora k
      / EMELEC.1, EL_NACIONAL.1, DEP_CUENCA.1, LIGA_DE_QUITO.2,
        DEP_QUITO.2, LIGA_DE_LOJA.2, MANTA.2, ESPOLI.2, IMBABURA.2,
        BARCELONA.3, INDEPENDIENTE.3 /
C(i,h)   Conjunto de esquemas complementarios      /
$include "d:\Cih.txt"
/
D(j,j)   Equipos que pertenecen a la misma ciudad
         / BARCELONA.EMELEC, DEP_QUITO.EL_NACIONAL,
         DEP_QUITO.LIGA_DE_QUITO/
E(i,r)   Esquemas que no seran emparejados a un equipo cabeza de serie/
$include "d:\Eir.txt"
/
F(i,r)   nomatchtown        /
$include "d:\Fir.txt"
/

Variables
x(i,j)    si el esquema i es emparejado con el equipo j
z3        función objetivo ;
Binary Variable x ;
Equations
FO          función objetivo
r1(j)
r2(i)
r3(i,j,m)
r4(i,j,m)
r5(i,h,j,m)
;
FO..                               z3  =e=sum((i,j), x(i,j)) ;
r1(j)..                            sum(i,x(i,j))  =e=  1 ;
r2(i)..                            sum(j,x(i,j))  =e=  1 ;
r3(i,j,m)$(T(j)and T(m)and(ord(m)ne ord(j)))..
     x(i,j) + sum(r$E(i,r),x(r,m))=l=1;
r4(i,j,m)$(D(j,m))..               x(i,j)+sum(r$F(i,r),x(r,m))=l=1 ;
r5(i,h,j,m)$(C(i,h) and D(j,m))..  x(i,j)-x(h,m) =e= 0;
```

```
Model fase3 /all/ ;
Solve fase3 using mip min z3 ;
Display x.l,i,C,D,E,F ;

FILE html /'EQUIPOS_ESQUEMAS.html'/;
PUT html;
PUT html;
PUT '<H2 ALIGN=CENTER><font color="ORANGE">EQ CON SU RESP ESQ</H2>'/;
PUT '<H2 ALIGN=CENTER>RESULTADOS</H2>'/;
PUT '<p>'/;
PUT '<table ALIGN=CENTER border="5" cellpadding="14" cellspacing="0">'/;
PUT '<tr><th></th>'/;
PUT '<th BGCOLOR=#996699><FONT COLOR=WHITE>','ASIGNACIONES','</th>'/;
LOOP((j),
 PUT '<tr><th BGCOLOR=#996699><FONT COLOR=WHITE>','Partidos','</th>'/;
 LOOP((i),
 IF ((x.l(i,j) eq 1),PUT '<td ALIGN=center BGCOLOR=#AAABD7>',
     j.tl '- 'i.tl,'</td>'/);
 );
 PUT '</tr>'/;);
PUT '</table>'/;
PUTCLOSE;
EXECUTE '=shellexecute EQUIPOS_ESQUEMAS.html';
```

6.4. Matriz $A_{i,j}$

En la tabla 6.1 se muestra todos los esquemas generados en Mathematica, para luego seleccionar 12 ellos por GAMS.

Tabla Aij

	1	2	3	4	5	6	7	8	9	10	11
1	1	0	1	0	1	0	1	0	1	0	1
2	1	0	1	0	1	0	1	0	1	0	0
3	1	0	1	0	1	0	1	0	1	1	0
4	1	0	1	0	1	0	1	0	0	1	1
5	1	0	1	0	1	0	1	0	0	1	0
6	1	0	1	0	1	0	1	1	0	0	1
7	1	0	1	0	1	0	1	1	0	1	0
8	1	0	1	0	1	0	1	1	0	1	1
9	1	0	1	0	1	0	0	1	1	0	1
10	1	0	1	0	1	0	0	1	0	1	1
11	1	0	1	0	1	0	0	1	0	1	0
12	1	0	1	0	1	0	0	1	0	0	1
13	1	0	1	0	1	1	0	0	1	0	1
14	1	0	1	0	1	1	0	1	0	0	1
15	1	0	1	0	1	1	0	1	0	1	0
16	1	0	1	0	1	1	0	1	0	1	1
17	1	0	1	0	1	1	0	1	1	0	1
18	1	0	1	0	0	1	1	0	1	0	1
19	1	0	1	0	0	1	0	1	1	0	1
20	1	0	1	0	0	1	0	1	0	1	1
21	1	0	1	0	0	1	0	1	0	1	0
22	1	0	1	0	0	1	0	1	0	0	1
23	1	0	1	0	0	1	0	0	1	0	1
24	1	0	1	1	0	0	1	0	1	0	1
25	1	0	1	1	0	1	0	0	1	0	1
26	1	0	1	1	0	1	0	1	0	0	1
27	1	0	1	1	0	1	0	1	0	1	0
28	1	0	1	1	0	1	0	1	0	1	1
29	1	0	1	1	0	1	0	1	1	0	1
30	1	0	1	1	0	1	1	0	1	0	1
31	1	0	0	1	1	0	1	0	1	0	1
32	1	0	0	1	0	1	1	0	1	0	1
33	1	0	0	1	0	1	0	1	1	0	1
34	1	0	0	1	0	1	0	1	0	1	1
35	1	0	0	1	0	1	0	1	0	1	0
36	1	0	0	1	0	1	0	1	0	0	1
37	1	0	0	1	0	1	0	0	1	0	1
38	1	0	0	1	0	0	1	0	1	0	1
39	1	1	0	0	1	0	1	0	1	0	1
40	1	1	0	1	0	0	1	0	1	0	1
41	1	1	0	1	0	1	0	0	1	0	1
42	1	1	0	1	0	1	0	1	0	0	1
43	1	1	0	1	0	1	0	1	0	1	0
44	1	1	0	1	0	1	0	1	0	1	1
45	1	1	0	1	0	1	0	1	1	0	1
46	1	1	0	1	0	1	1	0	1	0	1
47	1	1	0	1	1	0	1	0	1	0	1
48	0	1	0	1	0	1	0	1	0	1	0
49	0	1	0	1	0	1	0	1	0	1	1
50	0	1	0	1	0	1	0	1	0	0	1
51	0	1	0	1	0	1	0	1	1	0	0
52	0	1	0	1	0	1	0	1	1	0	1
53	0	1	0	1	0	1	0	0	1	1	0
54	0	1	0	1	0	1	0	0	1	0	1
55	0	1	0	1	0	1	0	0	1	0	0
56	0	1	0	1	0	1	1	0	0	1	0
57	0	1	0	1	0	1	1	0	1	0	0
58	0	1	0	1	0	1	1	0	1	0	1
59	0	1	0	1	0	1	1	0	1	1	0
60	0	1	0	1	0	0	1	1	0	1	0
61	0	1	0	1	0	0	1	0	1	1	0
62	0	1	0	1	0	0	1	0	1	0	1
63	0	1	0	1	0	0	1	0	1	0	0
64	0	1	0	1	0	0	1	0	0	1	0
65	0	1	0	1	1	0	0	1	0	1	0
66	0	1	0	1	1	0	1	0	0	1	0
67	0	1	0	1	1	0	1	0	1	0	0
68	0	1	0	1	1	0	1	0	1	0	1
69	0	1	0	1	1	0	1	0	1	1	0
70	0	1	0	1	1	0	1	1	0	1	0
71	0	1	0	0	1	1	0	1	0	1	0
72	0	1	0	0	1	0	1	1	0	1	0
73	0	1	0	0	1	0	1	0	1	1	0
74	0	1	0	0	1	0	1	0	1	0	1
75	0	1	0	0	1	0	1	0	1	0	0
76	0	1	0	0	1	0	1	0	0	1	0
77	0	1	0	0	1	0	0	1	0	1	0
78	0	1	1	0	0	1	0	1	0	1	0
79	0	1	1	0	1	0	0	1	0	1	0
80	0	1	1	0	1	0	1	0	0	1	0
81	0	1	1	0	1	0	1	0	1	0	0
82	0	1	1	0	1	0	1	0	1	0	1
83	0	1	1	0	1	0	1	0	1	1	0
84	0	1	1	0	1	0	1	1	0	1	0
85	0	1	1	0	1	1	0	1	0	1	0
86	0	0	1	1	0	1	0	1	0	1	0
87	0	0	1	0	1	1	0	1	0	1	0
88	0	0	1	0	1	0	1	1	0	1	0
89	0	0	1	0	1	0	1	0	1	1	0
90	0	0	1	0	1	0	1	0	1	0	1
91	0	0	1	0	1	0	1	0	1	0	0
92	0	0	1	0	1	0	1	0	0	1	0
93	0	0	1	0	1	0	0	1	0	1	0
94	0	0	1	0	0	1	0	1	0	1	0

Tabla 6.1: Tabla de datos $A_{i,j}$ generados en Wolfram Mathematica

Bibliografía

[1] Adenso Díaz (1996). *Optimización Heurística y Redes Neuronales* Addison Wesley.

[2] Aida Díaz Fernández, Javier Rodrigo, María Luisa Guerrero Lerma *Emparejamientos aplicados a la elaboración de calendarios deportivos.*

[3] Áslaug Sóley Bjarnadóttir (2004). *Solving the vehicle routing problem with genetic algorithms.* Informatics and Mathematical Modelling, IMM.

[4] Beatriz Pérez Rojas. *Análisis Comparativo de Heurísticas para el Problema de Calendarización de Trabajos con Transferencia Cero.* Departamento de Sistemas y Computación. Instituto Tecnológico de Puebla.

[5] Chandrasekharan (1994). *A No-wait Flowshop Sheduling Heuristic to Minimize Makespan,* Journal of Operations Reserach, vol. 45, No.4, p. 472-478.

[6] Chin-Yen Chen (2008). *Using integer programming to solve the school timetabling problem at chin-min institute of technology.* American Academy of Business.

[7] D. Oliveri and F. Della Croce (2004). *Scheduling the italian football league: an ILP-based approach.* ELSEVIER.

[8] David Álvarez Martínez, Eliana toro Ocampo, Ramón Gallego Rendón (2008). *Algoritmo Recocido Simulado aplicado al problema de secuenciamiento de tareas en sistemas de producción lineal Flow-Shop* Scientia et Technica Año XIV, No 40. Universidad Tecnológica de Pereira

[9] Durán, G., M. Guajardo, J. Miranda, D. Sauré, S. Souyris, A. Weintraub, 2007. *Scheduling the Chilean soccer league by Integer Programming*, Interfaces 37(6) 539-552

[10] Emilio González Caro. Universidad de Sevilla *Un algoritmo de Búsqueda Tabú para generar el calendario de exámenes de la nueva Facultad de Ciencias de la Educación.* Departamento de Organización Industrial y Gestión de Empresas.

[11] Flatberg, T., E. J. Nilssen, M. Stlevik. (2009). *Scheduling the topmost football leagues of Norway. 23rd European Conference on Operational Research, Book of Abstracts. Bonn, Germany, p.240.*

[12] Fred Glover and Gary A. *Kochenberger. Handbook of Methaheuristics. Kluwers International Series in Operation Research and Management Science.*

[13] Froncek, D., M. Meszka (2010). Personal communication.

[14] Goossens, D., F. Spieksma. (2009). *Scheduling the Belgian Soccer League.*

[15] Henz, M. (2001) *Scheduling a major college basketball conference.* Operations Research.

[16] J.H. Holland (1992)(1975). University of Michigan *Adaptation in natural and artificial systems.* Press, Ann Arbor, Michigan; re-issued by MIT Press.

[17] Javier Fiallos. (2010) *Calendarización del Campeonato de Liga de Fútbol Profesional de Honduras por medio de Programación Entera.* Departamento de Ingeniería Industrial. UNITEC, Tegucigalpa, Honduras.

[18] Jin-Kao Hao and Zhipeng Lu (2008). *Adaptive tabu search for course timetabling.* ELSEVIER.

[19] Johnson, S.M (1954). *Two and three-satage production schedules with set-up times included* Naval Res. Logist. Vol. 1, p. 61-68.

[20] Kendall, S. Knust, C. C. Ribeiro, S. Urrutia. (2009). *Scheduling in sports: An annotated bibliography.* Computers and Operations Research 37(1):1-19.

[21] Ki-Seok Sung and Enzhe Yu 2007. *A genetic algorithm for a university weekly courses timetabling problem.* Blackwell Publishers.

[22] Kobayashy, Ono, Yamamura (1995). *An Efficient Genetic Algorithm for Job Shop Scheduling Problems* Proceedings of the Sixth International Conference on Genetic Algorithms : University of Pittsburg, p. 506-511.

[23] Lance D. Chambers (1999). *Practical Handbook of GENETIC ALGORITHMS, Volumen III.* CRC Press LLC.

[24] M. Garey, R. Jonson, D. S. and SEIT (1976). *The Complexity of Flor Shop and Job Shop Scheduling,* in Mathematics of Operation Research, Vol. 1, No. 2. pp. 117-129.

[25] Manuel Laguna Rafael Martí. *Scatter Search: Diseño Básico y Estrategias Avanzadas.*

[26] Meszka, M. (2010). Personal communication

[27] Nawaz, M., Enscore Jr. and Ham (1983). *A heuristic algorithm for the mmachine, n-job flow-shop sequencing problem,* OMEGA, The international Journal of Management Science Vol. 11, No. 1 p.91-95.

[28] Nemhauser GL and Trick MA (1998). *Scheduling a major college basketball conference.* Operations Research.

[29] Nurmi K., T. Bartsch, F. Bonomo, D. Briskorn, G. Durán, D. Goossens, J. Kyngas, J. Marenco, C. Ribeiro, F. Spieksma, S. Urrutia, R. Wolf. (2010). *A Framework for a Highly Constrained Sports Scheduling Problem.* Proceedings of the International MultiConference of Engineers and Com- puter Scientists Vol III, IMECS 2010, March 17-19, Hong Kong, 1991-1997.

[30] Rasmussen, R. V., M. A. Trick (2008). *Round robin scheduling - a survey.* European Journal of Operational Research 188(3):617-636.

[31] Rasmussen, R. V. (2008) *Scheduling a triple round robin tournament for the best Danish soccer league.* European Journal of Operational Research.

[32] Ribeiro, C., S. Urrutia. (2009). *Scheduling the Brazilian soccer tournament by integer programming maximizing audience shares under fairness constraints.* 23rd European Conference on Operational Research, Book of Abstracts. Bonn, Germany, p.240.

[33] Schreuder, J. (1992). *Combinatorial aspects of construction of competition Dutch professional football leagues.* Discrete Applied Mathematics.

[34] Xavier Cabezas G. (2009). *Calendarización de la Liga Italiana de Fútbol: Una aproximación (heurística) basada en ILP.* Escuela Superior Politécnica del Litoral.

[35] Xavier Cabezas, Fernando Sandoya. (2010) *Un algoritmo evolutivo para resolver el problema de calendarización para universidades.* Escuela Superior Politécnica del Litoral.