

Proyecto de Curso

El Problema de la Planificación del Calendario de Torneos Deportivos

Escuela de Ingeniería de Sistemas y Computación
Facultad de Ingeniería



Prof. Juan Francisco Díaz Frias

Mayo de 2023

1. El Problema de la Planificación del Calendario de Torneos Deportivos (CalDep)

1.1. El problema

Aunque a simple vista no lo parezca, la creación de calendarios adecuados para torneos deportivos es un problema de investigación ¹ alrededor del cual se genera mucha actividad académica.

La idea principal consiste en confeccionar calendarios que además de tener en cuenta hechos básicos como:

- el número de equipos es par,
- un *partido* enfrenta dos equipos, uno de ellos es *local* y el otro *visitante*,
- una *fecha* es un conjunto de partidos donde cada equipo juega una y solo una vez,
- un *calendario a una ronda*² es una secuencia de fechas tal que en el conjunto de todos los partidos cada equipo se enfrenta una sola vez a cada otro equipo,
- un *calendario a dos rondas*³ es un conjunto de fechas tal que en el conjunto de todos los partidos cada equipo se enfrenta exactamente dos veces a cada otro equipo; en estos dos enfrentamientos los equipos se alternan la localía. Al primer partido entre dos equipos se le llama el *partido de ida* y al segundo se le llama el *partido de vuelta*.

¹Ver artículo *The Traveling Tournament Problem Description and Benchmarks* de K. Easton, G. Nemhauser y M. Trick

²conocido como Round-Robin

³conocido como Doble Round-Robin

También tenga en cuenta restricciones como:

- alternar, en la medida de lo posible, partidos de local con partidos de visitante para cada equipo, de una fecha a la siguiente,
- no programar partidos de vuelta hasta tanto no se hayan programado todos los partidos de ida,
- cuando a un equipo se le programan varias fechas seguidas como visitante, tratar que los desplazamientos, en términos de distancias recorridas por el equipo visitante, se minimicen,
- a ningún equipo le gusta tener todos los partidos difíciles seguidos,
- a ningún equipo le conviene que todos sus partidos de local se programen en semana y los de visitante los fines de semana,

y otras parecidas.

Lograr confeccionar un calendario que satisfaga todas las restricciones y condiciones no es una tarea sencilla. Y un calendario preconcebido donde solo haya que asociar etiquetas a equipos no puede tener en cuenta la restricción de minimización de costos de desplazamientos.

Por otro lado, los torneos nacionales de fútbol, por ejemplo, cuentan con 16 a 20 equipos participantes, un campeonato mundial de fútbol con 24 o 32 o 36 equipos. Un campeonato de béisbol o baloncesto en los USA cuenta con 32 participantes. Lograr soluciones correctas para problemas con entradas de estos tamaños, en tiempos adecuados, es un problema computacionalmente muy interesante, y será el reto de este proyecto.

1.2. Formalización

Dados n equipos (n par) un *torneo todos contra todos (ttt)* es un torneo en el que cada par de equipos juega un partido entre ellos; y un *torneo todos contra todos ida y vuelta (tttiv)* es un torneo en el que cada par de equipos juega dos partidos entre ellos: el de ida y el de vuelta.

Un calendario a una ronda para un *ttt* consiste de $n - 1$ fechas, cada una conteniendo $\frac{n}{2}$ partidos.

Un calendario a dos rondas para un *tttiv* consiste de $2(n - 1)$ fechas, cada una conteniendo $\frac{n}{2}$ partidos. En cada juego un equipo se denomina *local* y el otro *visitante*. Los dos juegos entre cada par de equipos deben alternar la localía.

En este ejercicio vamos a suponer que debemos planificar un torneo *tttiv*. Para representar el calendario de manera sencilla, se utilizará una matriz Cal de dimensión $2(n - 1) \times n$ tal que $Cal[i, j] = k$ significa que en la fecha i el equipo j juega de local contra el equipo k y $Cal[i, j] = -k$ significa que en la fecha i el equipo j juega de visitante contra el equipo k . Como consecuencia de esta representación un calendario cumple las siguientes propiedades:

- $Cal[i, j] = k$ si y solo si $Cal[i, k] = -j$.
- $\{|Cal[i, j]| : 1 \leq j \leq n\} = \{1, 2, \dots, n\}$, para todo $1 \leq i \leq 2(n - 1)$.
- $\frac{n}{2} = |\{Cal[i, j] > 0 : 1 \leq j \leq n\}| = |\{Cal[i, j] < 0 : 1 \leq j \leq n\}|$, para todo $1 \leq i \leq 2(n - 1)$.
- $\forall j \in [1..n], \forall k \neq j, \exists i_1, i_2 \in [1..2(n - 1)] : Cal[i_1, j] = k \wedge Cal[i_2, j] = -k$.

Las distancias entre las ciudades sedes de cada equipo se representan por medio de una matriz D de dimensión $n \times n$. Cuando el equipo i juega de visitante ante el equipo j debe desplazarse una distancia d_{ij} . Si en el calendario el equipo i juega k partidos consecutivos de visitante ante los equipos j_1, \dots, j_k , se dice que el equipo i está de *gira*. La *gira* es de tamaño k y su costo es $d_{ij_1} + d_{j_1j_2} + d_{j_2j_3} + \dots + d_{j_{k-1}j_k} + d_{j_ki}$.

Por otro lado, si en el calendario el equipo i juega k partidos consecutivos de local se dice que el equipo i está en *permanencia* de tamaño k .

El problema de la Planificación del Calendario de Torneos Deportivos (CalDep)

Entrada: n , el número de equipos, D la matriz de distancias de dimensión $n \times n$, Min y Max dos enteros positivos.

Salida: Una matriz Cal de dimensión $2(n-1) \times n$ representando un calendario válido tal que el tamaño de toda *gira* y toda *permanencia* es mayor o igual a Min y menor o igual a Max , y tal que la suma total de los costos de todas las *giras* es mínimo. Adicionalmente no se permite que en dos fechas consecutivas se repita un mismo partido.

1.3. ¿Entendimos el problema?

Entrada: $n = 4, Min = 1, Max = 3$,

$$D = \begin{bmatrix} 0 & 745 & 665 & 929 \\ 745 & 0 & 80 & 337 \\ 665 & 80 & 0 & 380 \\ 929 & 337 & 380 & 0 \end{bmatrix}$$

Salida: $Cal = \begin{bmatrix} 3 & 4 & -1 & -2 \\ 2 & -1 & 4 & -3 \\ 4 & -3 & 2 & -1 \\ -3 & -4 & 1 & 2 \\ -2 & 1 & -4 & 3 \\ -4 & 3 & -2 & 1 \end{bmatrix}$

Note que en esta solución, el equipo 1 tiene una permanencia de 3 partidos y una gira de 3 partidos; el equipo 2 tiene una permanencia de 1 partido, una gira de 3 partidos y una permanencia de 2 partidos; el equipo 3 tiene una gira de 1 partido, una permanencia de 3 partidos y una gira de 2 partidos; y el equipo 4 tiene una gira de 3 partidos y una permanencia de 3 partidos.

El costo de la gira del equipo 1 es $d_{13} + d_{32} + d_{24} + d_{41} = 665 + 80 + 337 + 929 = 2011$.

El costo de la gira del equipo 2 es $d_{21} + d_{13} + d_{34} + d_{42} = 745 + 665 + 380 + 337 = 2127$.

El costo de las giras del equipo 3 es $d_{31} + d_{13} = 1330$ y $d_{34} + d_{42} + d_{23} = 380 + 337 + 80 = 797$.

El costo de la gira del equipo 4 es $d_{42} + d_{23} + d_{31} + d_{14} = 337 + 80 + 665 + 929 = 2011$.

El costo total de la solución, que corresponde a la suma de los costos de todas las giras, es $2011 + 2127 + 1330 + 797 + 2011 = 8276$.

Por otro lado, note que cualquier otro calendario solución, tiene que corresponder a una permutación de las filas de Cal .

Por ejemplo $Cal_1 = \begin{bmatrix} 3 & 4 & -1 & -2 \\ 2 & -1 & 4 & -3 \\ -3 & -4 & 1 & 2 \\ 4 & -3 & 2 & -1 \\ -2 & 1 & -4 & 3 \\ -4 & 3 & -2 & 1 \end{bmatrix}$

es un calendario que cumple con todas las restricciones solicitadas. Como ejercicio de entendimiento, calcule el costo de Cal_1 y compárelo con el costo de Cal .

También hay permutaciones de Cal que no son soluciones factibles porque no cumplen con todas las restricciones. Por

ejemplo $Cal_2 = \begin{bmatrix} 3 & 4 & -1 & -2 \\ -2 & 1 & -4 & 3 \\ 2 & -1 & 4 & -3 \\ -3 & -4 & 1 & 2 \\ 4 & -3 & 2 & -1 \\ -4 & 3 & -2 & 1 \end{bmatrix}$

no cumple con una de las restricciones solicitadas (¿con cual?).

Por último, note también que si en la entrada el valor de Max es 2 y no 3, entonces Cal, Cal_1, Cal_2 no son soluciones factibles (¿por qué?). Como ejercicio de asimilación, encuentre una solución para ese caso.

2. El proyecto: Modelamiento e Implementación

Usted como ingeniero ha sido contratado para resolver el problema y debe:

- Proponer un modelo genérico para solucionar el problema CalDep. El modelo debe ser incluido en formato pdf y debe contener: parámetros, variables, restricciones, función objetivo. El modelo debe utilizar notación formal para que soporte cualquier instancia con la entrada definida en la Sección 2.1.
- Generar 5 instancias para retar a otros proyectos. Para cada instancia debe incluir la entrada y la salida esperada (el valor del óptimo, o por lo menos el valor de la mejor solución que su grupo haya encontrado)
- Implementar el modelo genérico en MiniZinc (**CalDep.mzn**).
- Incluir una tabla con pruebas realizadas sobre las instancias que se proveen con el proyecto y las 5 instancias creadas por su grupo de trabajo. Realice un análisis sobre los resultados obtenidos (incluya el análisis en el informe con el modelo).
- Desarrollar una interfaz gráfica con la tecnología de su predilección que permita configurar o leer una entrada para el problema (la entrada deberá convertirse a formato dzn para poder ser ejecutada por el modelo cumpliendo con las características de la entrada definida en la Sección 2.1) y visualizar la salida. Esta interfaz junto con el modelo sería el entregable para el cliente y será utilizada por algún operario. La interfaz debe incluir un botón que al presionarlo:
 - Cree un archivo **DatosCalDep.dzn** con los datos proporcionados en la interfaz
 - Ejecute el modelo genérico **CalDep.mzn** sobre los datos proporcionados
 - Despliegue los resultados de la solución
- Incluya los archivos fuente de su implementación gráfica en un directorio llamado **CalDepGUIFuentes**
Para mayor información sobre la forma de ejecutar un modelo MiniZinc a través de línea de comandos visite:
 - Modelamiento básico en MiniZinc: <https://www.minizinc.org/doc-2.2.3/en/modelling.html>
 - Modelos más complejos: <https://www.minizinc.org/doc-2.2.3/en/modelling2.html>
- Hacer un vídeo de máximo 4 minutos donde muestre su aplicación funcionando. Debe explicar brevemente los componentes y realizar por lo menos dos pruebas con distintas configuraciones donde se muestren las soluciones en la interfaz gráfica. **Se debe incluir un enlace al video en el archivo pdf del informe.**

2.1. Entrada

La entrada se leerá de un archivo *.dzn con la siguiente información:

1. La primera línea contiene un entero indicando el número de equipos del campeonato (es decir, conteniendo n).
2. La segunda línea contiene un entero indicando el tamaño mínimo de cada gira o permanencia (es decir, conteniendo Min).
3. La tercera línea contiene un entero indicando el tamaño máximo de cada gira o permanencia (es decir, conteniendo Max).
4. Las siguientes n líneas contienen las filas de la matriz de distancias entre las ciudades sedes (es decir, contienen $D_i, i = 1..n$). Cada fila contiene n enteros separados por espacios en blanco.

El archivo correspondiente a la entrada del ejemplo de la sección 1.3 sería:

```

4
1
3
0 745 665 929
745 0 80 337
665 80 0 380
929 337 380 0

```

En el campus se compartirán ejemplos de entradas con sus respectivas salidas (Nota: es posible que en algunos casos haya varias soluciones que tengan el mismo valor para la función objetivo)

2.2. Sobre el Informe...

El grupo deberá entregar un informe del proyecto, en formato pdf, que contenga, al menos, los siguientes aspectos:

- El modelo: una descripción del modelo (parámetros, variables y dominios, restricciones, función objetivo) y una justificación de su adecuación al problema planteado.
- Detalles importantes de implementación: lo más relevante de la implementación, en su concepto, sin incluir código.
- La búsqueda: La descripción de las diferentes estrategias de búsqueda exploradas.
- Pruebas: descripción de las pruebas realizadas a su implementación.
- Análisis: de los resultados de las pruebas realizadas, buscando responder a los diferentes criterios de evaluación definidos en la rúbrica. Desarrolle y soporte su análisis utilizando los métodos apropiados (tablas, gráficos, indicadores estadísticos), donde puedan apreciarse las variaciones de acuerdo al tamaño y naturaleza de los datos de entrada. Explique claramente el significado de sus datos y cómo se analizaron.
- Un enlace al video explicatorio de la interfaz gráfica
- Conclusiones: Esta es una de las partes más interesantes del trabajo (pero no por ello la que más vale). En ella se espera que usted analice los resultados obtenidos y **justifique** claramente sus afirmaciones.

2.3. Grupos de trabajo

El proyecto puede ser desarrollado por grupos de máximo 3 personas.

3. Entrega, sustentación y evaluación

3.1. Entrega

La entrega se debe realizar vía el campus virtual en las fechas previstas para ello, por uno sólo de los integrantes del grupo. **La fecha de entrega límite es el 20 de Junio de 2023 a las 17:00. La sustentación será en la sesión del 4 de Julio de 2023 (o en otras si hay un cambio de última hora).** Se debe subir al campus virtual en el enlace correspondiente a este proyecto un archivo comprimido **.zip** que siga la convención *CódigoEstudiante1-CódigoEstudiante2-CódigoEstudiante3-CódigoEstudiante4-Proyecto2-AdaII.2023-I.zip*. El comprimido deberá contener:

1. Archivo **Readme.txt** que describa todos los archivos entregados y las instrucciones para ejecutar la aplicación.
2. Archivo **Informe.pdf** acorde a la Sección 2.2. Recuerde incluir el link al video explicatorio.

3. Archivo **CalDep.mzn** con la implementación del modelo
4. Directorio **DatosCalDep** con los datos con que fue probado su modelo.
5. Directorio **CalDepGUIFuentes** con los archivos fuente de la implementación de la interfáz gráfica
6. Directorio **MisInstancias** con las 5 instancias generadas por su equipo de trabajo para retar a otros proyectos que resuelvan el mismo problema.

3.2. Evaluación

La evaluación de cada proyecto se hará de acuerdo a la rúbrica publicada en el campus virtual, diseñada para observar los indicadores de logro asociados a este proyecto. En específico se calificará según los siguientes criterios:

- Definición de variables de los modelos (15)

Se espera que las variables de los modelos estén precisamente definidas (su significado), y sus dominios estén precisamente definidos. Nunca se confunden variables con parámetros. Además de estar precisamente definidos, los dominios de las variables son apropiados para los problemas que modelan y no contienen valores que desde el inicio se sabe que no pueden tomar.
- Definición de las restricciones de los modelos (25)

Se espera que al definir las restricciones de los problemas, se tengan en cuenta todas las restricciones mencionadas en el problema, modelando todos los aspectos relevantes. Las restricciones se definen independientemente de la implementación, en un lenguaje formal preciso y claro, y se justifica su corrección o adecuación, cuando esta no es evidente. Las restricciones, en general, no están condicionadas por los valores que puedan tomar las variables; este tipo de restricciones sólo se dejan en caso de no haber otra opción.
- Claridad para modelar la función objetivo (15)

Se espera que se defina precisamente la función objetivo y el criterio de optimización (maximizar o minimizar) y que sean coherentes con el modelo y el problema.
- Calidad de las pruebas (10)

Se espera que cada modelo se haya probado con un suficiente número de pruebas para poder analizar a profundidad el modelo. Por ello este criterio lo que evalúa es si esa batería de pruebas es suficiente para verificar la corrección del modelo y para explorar las diferentes estrategias de búsqueda y compararlas. También se espera que los resultados de las pruebas se sistematicen de alguna manera (tablas, gráficas, ...) y se presenten en el informe.
- Implementación coherente con el modelo y con los análisis presentados en el informe (25)

Se espera que todos los modelos descritos hayan sido implementados coherentemente en MiniZinc y que los análisis correspondan con las pruebas de esos modelos en MiniZinc.
- Redacción del informe (10)

Se espera que el informe sea completo, claro, preciso, fácil de leer, bien presentado, contenga los análisis más importantes y las conclusiones.

3.3. Sustentación

El trabajo debe ser sustentado por los autores en día y hora por definir. En todos los casos la sustentación será pilar fundamental de la nota individual asignada a cada integrante del grupo. En la sustentación se demuestra la capacidad del grupo de navegar en el código y realizar cambios rápidamente en él, así como la capacidad de responder con solvencia a las preguntas que se le realicen.

Cada persona de cada grupo, después de la sustentación, tendrá asignado un número real (el factor de multiplicación) entre 0 y 1, correspondiente al grado de calidad de su sustentación. **Su nota definitiva será la calificación del proyecto para el grupo, multiplicada por ese valor.** Si su asignación es 1, su nota será la del proyecto. Pero si su asignación es 0.9, su nota será 0.9 por la nota del proyecto. La no asistencia a la sustentación tendrá como resultado una asignación de un factor de 0.

La idea es que lo que no sea debidamente sustentado no vale así funcione muy bien!!! Y que, del trabajo en grupo, es importante que todos aprendan, no sólo algunos.

Éxitos!!!