UNIVERSIDAD SIMÓN BOLÍVAR

DEPARTAMENTO DE COMPUTACIÓN Y TECNOLOGÍA DE LA INFORMACIÓN

CI5437 – INTELIGENCIA ARTIFICIAL I

Trimestre: Enero – Marzo 2024

Profesor: Carlos Infante

Informe del Proyecto #3

CNF – SAT

Estudiantes:

José Matías González, 15-10627

Gabriel Chaurio, 17-10126

Ana Santos, 17-10602

Sartenejas, Marzo del 2023

ÍNDICE

DEFINICIÓN DEL PROBLEMA

El desarrollo de sistemas capaces de resolver problemas complejos de planificación y horarios es un estudio importante y que ha presentado un avance significativo dentro de la computación, pues su uso correcto y óptimo facilita el día a día de las personas ante una sociedad que pide orden y disciplina con los horarios y tareas asignadas. Para resolver problemas de esta índole, se ha creado la representación CNF que se basa en lógica proposicional para darle una estructura a los problemas , que de la mano de otras herramientas como lo son los SAT solvers, pueden solucionarse muchos problemas de planificación de forma sencilla y automatizada.

La tarea central de este proyecto es desarrollar un sistema que, dada una lista de participantes y un conjunto de restricciones temporales y logísticas, sea capaz de generar un horario de juegos para un torneo.

Además, el proyecto se enfoca en el uso de SAT solvers para abordar este problema. Los SAT solvers son herramientas poderosas en la resolución de problemas lógicos y han demostrado ser útiles en una variedad de aplicaciones. En este contexto, el desafío será traducir las restricciones del torneo a una forma que sea comprensible para un SAT solver, específicamente en formato CNF (Conjunctive Normal Form).

METODOLOGÍA

1. Teoría de las restricc iones de implementación

Las restricciones teóricas que se establecieron para determinar el establecimiento de cada partido siguen los siguientes índices $M_{i,j,d,s}$, donde M son los partidos (o matchs), i, j como la numeración de los equipos como local y visitante respectivamente, d para la cantidad de días y s para los espacios donde los equipos puedan enfrentarse.

Finalmente, las restricciones para el modelo son cada equipo debe jugar como local contra todos los otros equipos exactamente una vez, se puede máximo un jugo por día, no pueden haber más de dos partidos al mismo tiempo y un equipo no puede jugar de manera consecutiva como local ni como visitante dos días seguidos. A continuación, mostramos las ecuaciones lógicas respectivas.

$$\left(\vee i,j \mid i \neq j : \sum_d \sum_s M_{i,j,d,s} = 1\right),$$

$$\left(\vee i,d \mid \sum_{j,j\neq i} \sum_s M_{i,j,d,s} + M_{i,j,d,s} \leq 1\right),$$

$$\left(\vee d,s \mid \sum_i \sum_{i,j\neq i} M_{i,j,d,s} \leq 1\right),$$

$$\left(\vee i,d \mid \sum_{j,j\neq i} \sum_s M_{i,j,d,s} + M_{i,j,d,s} \leq 1\right),$$

$$\left(\vee j,d \mid \sum_{i,j\neq i} \sum_s M_{i,j,d,s} + M_{i,j,d,s} \leq 1\right),$$

2. Implementación de los Algoritmos de Árboles

La implementación que se utilizó para solucionar el problema planteado fue hecha en Python, donde se modela el problema del torneo en formato CNF (forma normal conjuntiva). Este modelo se utiliza para ser procesado por Glucose, un solucionador de problemas SAT. Los archivos y sus funciones son los siguientes:

a. Variable Generales

Esta clase es responsable de generar todas las posibles variables que representan partidos únicos entre equipos, teniendo en cuenta las restricciones de día y espacio. Utiliza un enfoque exhaustivo para considerar todas las combinaciones posibles, excluyendo los casos donde un equipo juega contra sí mismo. Además, proporciona métodos para generar variables específicas para restricciones como no jugar partidos sucesivos en casa o fuera, todo con el fin de cumplir las restricciones impuestas sobre el calendario.

b. Restricciones en CNF (sumGreaterOrEqual, sumLessOrEqual)

Los métodos sumGreaterOrEqual y sumLessOrEqual son clave para transformar las restricciones del torneo en cláusulas CNF. Estos métodos generan cláusulas que garantizan que la suma de ciertos subconjuntos de variables booleanas cumpla con los límites superior e inferior especificados, respectivamente. Esta técnica es crucial para modelar restricciones como "un juego por día" o "no jugar como visitante en días consecutivos".

c. Clase SatSolver

SatSolver es la clase principal que integra la generación de variables con las restricciones en CNF y la interacción con el SAT solver. Maneja la creación del archivo de entrada en formato DIMACS CNF para Glucose y procesa la salida para determinar si existe una solución viable al problema del torneo.

Una vez que Glucose encuentra una solución, el sistema la parsea y la traduce a un calendario en formato iCalendar. Este proceso implica convertir las asignaciones de variables en fechas y horas de partidos, respetando todas las reglas del torneo.

d. Main

El script principal (__main__) actúa como cliente, gestionando el flujo de trabajo completo. Comienza leyendo la configuración del torneo desde un archivo JSON, calcula el número total de días y espacios disponibles según las fechas y horas del torneo, y luego llama

a SatSolver para resolver el problema. Finalmente, genera el archivo .ics con el calendario del torneo si la solución es satisfactoria.

3. Especificaciones del Equipo usado

Todas las pruebas se corrieron en el mismo equipo, el cual cuenta con las siguientes especificaciones:

- Procesador: Intel i5-12500H, 12 Cores.
- RAM: 64GB, 3100MHz.
- SSD

RESULTADOS Y ANÁLISIS

Se evaluaron 11 archivos de prueba los cuales varían en complejidad, siendo esta proporcional a la cantidad de equipos, días de juego y horas disponibles para jugar por día. La idea de estas pruebas además de aumentar el tamaño de las variables es modificar también valores individualmente mientras otros se mantienen constantes para observar el comportamiento. A continuación, se muestra la tabla donde se especifican los datos de cada archivo de prueba junto con los resultados obtenidos.

P Name	N Equipos	N Días	N Partidos / Dia	N Horas / Dia	Ejecución (Segundos)
p0	4	14	1	3	0.077
p1	6	14	3	7	0.701
p2	6	18	2	5	0.489
p3	6	14	4	9	1.418
p4	8	18	4	9	7.632
p5	8	21	3	7	5.173
p6	8	28	2	5	4.585
p7	10	28	4	9	43.730
p8	10	31	3	7	31.323
p9	12	31	5	11	152.344
p10	12	31	7	14	306.640

Tabla 1. *Resultados obtenidos al implementar el SatSolver*

En la tabla anterior, se puede observar el comportamiento del código ante modificaciones en las variables.

En primera instancia, podemos observar que mientras más grande es el problema, mayor es el tiempo de ejecución. Esto es evidente pues el modelo debe procesar, verificar y

organizar para cada par de equipos tomando en cuenta cada una de las restricciones del problema.

Por otro lado, podemos ver que el aumento del tiempo de ejecución aumenta exponencialmente ante un rango mayor en las variables, lo cual a pesar de ser contraintuitivo (por el hecho de que deberia ser mas complicado organizar un horario más “apretado” por así decirlo con menor soltura para la organización), cuando se ve detenidamente la manera en la que se teclea el problema, se le encuentra sentido, esto debido a que el algoritmo debe revisar más espacios de tiempo para colocar los partidos tomando en cuenta todas las restricciones.

También, ante el crecimiento presentado, es evidente que si sigue aumentando el tamaño de las variables, en un punto llega a ser ineficiente e incluso inutilizable el algoritmo ante un número grande de equipos, fechas y horas disponibles (como lo presenta un torneo de liga profesional de fútbol).

CONCLUSIONES

El modelado en CNF resulta ser un método práctico para la creación de eventos pequeños, ante un conjunto de restricciones sencillas se pueden crear calendarios de forma muy práctica y rápida si se posee el código que realice la tarea. Herramientas como Glucose facilita mucho el trabajo pues ya están optimizadas y funcionan de maravilla para problemas en forma de CNF.

Gracias al proyecto, pudimos profundizar el conocimiento de CFN y se aprendió a programar soluciones en dicho formato a problemas de la vida real. Se practicó la programación y planteamiento de soluciones en forma de restricciones y cláusulas CNF por lo que aprendimos de primera mano cómo crear y utilizar este recurso en horizontes tan grandes o mayores a los presentados en el proyecto.