Universitat politècnica de Catalunya

Màster en enginyeria informàtica - MEI

Facultat d'informàtica de Barcelona

Constrained Minimal Support Set

Yang, le Danny

Torres Alfonso, Pol

Índex

1 Introducción

El presente documento corresponde al informe de la primera práctica de la asignatura Computació i sistemes intel·ligents (CSI) cursada durante el año académico 2018 - 2019 en la Facultat d'informàtica de Barcelona (FIB).

El objetivo principal de esta primera práctica ha sido encontrar una solución para el problema Çonstrained Minimal Support Set", para hallar aquellas patologias que dada una enfermedad permiten caracterizar su padecimiento o no. Para ello se ha pedido realizar varias actividades:

- Una fóruma en CNF para hallar los support set.
- Un programa en MiniZinc para la fórmula en CNF.
- Un generador de datos para comprobar el buen funcionamiento de la fórmula.
- Una valoración final de los resultados obtenidos y del funcionamiento de la fórmula.

2 Estructura del proyecto

El proyecto está estructurado en cuatro directorios, en cada uno de ellos se encuentra uno de los elementos solicitados en el enunciado.

- **datagenerator:** Contiene el generador de datos en Python.
- **dataset:** Contiene los tests que se han utilizado para comprobar el rendimiento del programa.
- **docs:** Contiene el enunciado y el informe de la práctica.
- **solvercode:** Contiene la formula codificada en lenguaje MiniZinc.

3 Fórmula CNF

La formula en CNF para general el support set S debe cumplir el conjunto de restricciones siguientes:

1. El tamaño del support set debe ser $\leq$ a un número 'k'.
2. S debe ser un support set para $\Omega+$ y $\Omega-$.
3. S debe cumplir la restricción AtMostOne(A) para el conjunto de restricciones A1, A2, ..., Ac.

3.1 El tamaño del support set debe ser $S \leq K$

Se define al conjunto de elementos del support S set como {S1, S2, ... St}, donde la 't' tiene los valores contenidos entre 1 i 't', siendo t el número de literales que contiene un vector de $\Omega+$, y $\Omega-$. Los elementos Si con valor = 1, indican que el valor i se encuentra en el support set. Los elementos con valor = 0, indican que el elemento no se encuentra en el support set. Para cumplir esta condición podemos usar la restricción AtMost(k, S), de modo que el conjunto de S tendrá como máximo k elementos a 1.

Para ello definimos las siguientes variables:

- Conjunto S = {S1, S2, S3 ... St}

Para que S tenga como máximo un tamaño 'k' hay que prohibir que todos los sub conjuntos de S de tamaño k + 1 tengan sus valores a 1. De este modo como mucho habrá 'k' elementos a 1.

$$\neg(S1 \wedge S2 \wedge S3 \wedge ... \wedge Sk) \wedge \neg(S1 \wedge S2 \wedge S4 \wedge ... \wedge Sk) \wedge \neg(S1 \wedge S3 \wedge S4 \wedge ... \wedge Sk) \wedge \neg(S2 \wedge S4 \wedge S5 \wedge ... \wedge Sk) \wedge \neg(S1 \wedge S2 \wedge S5 \wedge ... \wedge Sk) \wedge ... \wedge \neg(S1 \wedge S3 \wedge S5 \wedge ... \wedge Sk-1)$$

Tamaño: $O(K\binom{t}{k+1})$

3.2 S debe ser un support set para $\Omega+$ y $\Omega-$

Para que S sea un support set de $\Omega+$ y $\Omega-$ la proyección de S sobre $\Omega+$ y Ω' debe ser disyunta. Para ello se consideran aquellos elementos del support set con valor a cierto (que pertenecen al support set) y se comprueba que, los valores de las posiciones coincidentes con los elementos a cierto del support set, de los vectores que forman $\Omega+$ y los vectores que forman $\Omega-$ sean disjuntos.

Para que se cumpla la condición de disyunción entre dos vectores deben existir almenos una posición en ambos vectores para los que su valor no sea igual. La condición de disyunción debe cumplirse para todos los vectores resultantes de la proyección de S sobre $\Omega+$ y Ω'. Teniendo en cuenta la siguientes consideraciones:

- S conjunto de elementos del support set S ={S1, S2, ... Sj}, $1 \leq k \leq t$ donde 't' es el numero de elementos de un vector
- Pik es el conjunto de vectores de $\Omega+$. Donde 'i' representa el numero del vector, y 'j' el literal.
- Nhk es el conjunto de vectores de $\Omega-$. Donde 'i' representa el numero del vector, y 'j' el literal.
- $1 \leq i \leq n$ donde 'n' es el numero de vectores de $\Omega+$.
- $1 \leq j \leq m$ donde 'm' es el numero de vectores de $\Omega-$.

Para cada vector 'i' de $\Omega+$, hay que comprobar que este sea disjunto con todos los vectores de 'h' $\Omega+$. Para ello al menos un elemento 'j' perteneciente al conjunto de elementos que forman el support set debe ser diferente en ambos vectores. Todo ello puede expresarse mediante la siguiente formula:

$$(S1 \wedge ((\neg P11 \vee \neg N11)) \wedge (P11 \vee N11)) \vee ... \vee (St \wedge ((\neg P1t \vee \neg N1t)) \wedge (P1t \vee N1t))$$

$$\wedge ... \wedge$$

$$(S1 \wedge ((\neg P11 \vee \neg Nm1)) \wedge (P11 \vee Nm1)) \vee ... \vee (St \wedge ((\neg P1t \vee \neg Nmt)) \wedge (P1t \vee Nmt))$$

$$\wedge ... \wedge$$

$$(S1 \wedge ((\neg Pn1 \vee \neg N11)) \wedge (Pn1 \vee N11)) \vee ... \vee (St \wedge ((\neg P1t \vee \neg N1t)) \wedge (P1t \vee N1t))$$

$$\wedge ... \wedge$$

$$(S1 \wedge ((\neg Pn1 \vee \neg Nm1)) \wedge (Pn1 \vee Nm1)) \vee ... \vee (St \wedge ((\neg Pnt \vee \neg Nmt)) \wedge (Pnt \vee Nmt))$$

Tamaño: $O(m\binom{n}{t})$

3.3 S debe cumplir la restricción AtMostOne(A)

El conjunto de sub sets de A representa la totalidad de las restricciones AtMostOne que debe cumplir el support set S para ser válido. Este tipo de restricción implica que el support set no puede contener más de un elemento que también se encuentre en alguno de los sub sets de A. Un ejemplo de ello es:

Para estos valores de S y de A la restricción no se cumple.

S = (1, 1, 0) A = (1, 1, 0)

En cambio para estos valores de S y de A la restricción si se cumple.

S = (1, 0, 1) A = (1, 0, 0)

Todo ello puede expresarse con la siguiente formula, en la que para todos los sub conjuntos que forman S y A se prohíbe el hecho de que exista más de un valor perteneciente a S que también se encuentre en A. Un valor pertenece al conjunto S o al A si evalua a cierto.

$\neg(S1 \wedge A1 \wedge S2 \wedge A2) \wedge \neg(S1 \wedge A1 \wedge S3 \wedge A3) \wedge \neg(S2 \wedge A2 \wedge S3 \wedge A3)$

Generalizando para el caso del proyecto la fórmula CNF resultante es la siguiente:

- Para cada Ar = {Ar1, Ar2, ... Arpt} puede haber como máximo 1 elemento de S perteneciente al support set, $1 \leq r \leq C$.
- Support set S = {S1, S2, ..., St}.

$(\neg S1 \vee \neg A11 \vee \neg S2 \vee \neg A12) \wedge (\neg S1 \vee \neg A11 \vee \neg S3 \wedge \neg A13) \wedge (\neg S2 \vee \neg A12 \vee \neg S3 \vee \neg \vee A13) \wedge (\neg S1 \vee \neg A21 \vee \neg S3 \vee \neg \vee A23) \wedge ... \wedge (\neg St-1 \vee \neg Act-1 \vee \neg St \vee \neg \vee Act)$

De este modo si en alguna de las restricciones Ai no se cumple, la restricción AtMostOne no se cumplirá, y el support set no será válido.

Tamaño: $O(C\binom{t}{4})$

4 Rendimiento del MiniZinc

El rendimiento del MiniZinc resolviendo el problema propuesto se ha evaluado con un conjunto de datos generados aleatoriamente, concretamente se han ejecutado un total de treinta y nueve pruebas.

La evaluación se ha dividido en dos partes, una de ellas enfocada al tiempo de ejecución con la que el MiniZinc resuelve los casos, y la otra aplicada a la satisfacibilidad que logra el programa.

4.1 Tiempo de cómputo

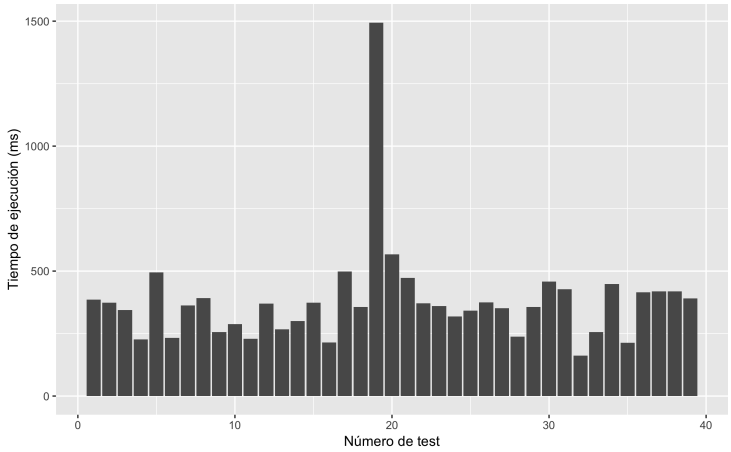

Figura 1: Rendimiento: Tiempo de cómputo

El tiempo de cómputo observado oscila entre los 150 ms y los 1500 ms. Por lo general el tiempo de las ejecuciones se encuentra entre los 200 y los 500 ms. De modo que para los valores que se han asignado para las t, k, n, m, c no se observa una gran distinción en el tiempo de cómputo, a excepción de un valor (1500 ms) para el que la m y la n eran significativamente más grandes que para el resto de tests.

4.2 Satisfacibilidad

Una vez ejecutadas todas las pruebas, el total de tests con valor satisfactorio han estado 23, frente a 16 insatisfactorios.

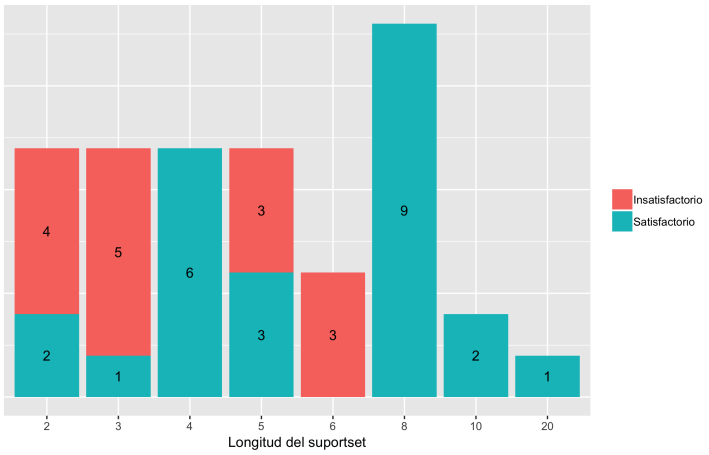

Figura 2: Rendimiento: Longitud del Support set

Longitud del support set

Como se observa en la gráfica aquellos support set que presentan una longitud reducida tienden a provocar que el resultado de la ejecución sea insatisfactorio, eso es debido a que, como menor sea la longitud del support set, es más difícil que los sub conjuntos generados por la proyección de este con omega positivo, y con omega negativo sean disjuntos.

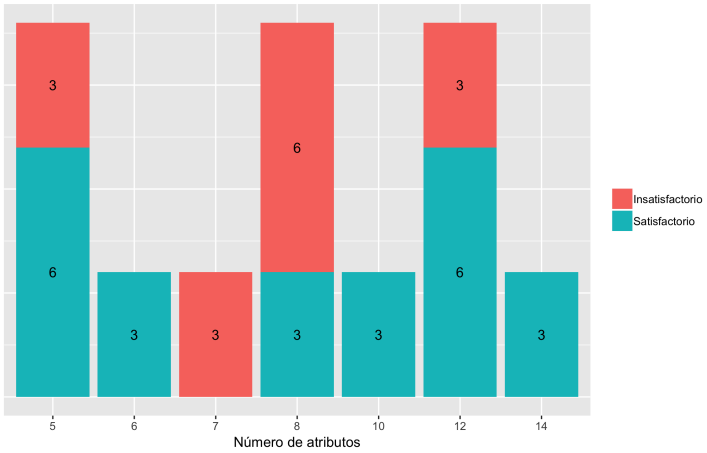

Figura 3: Número de atributos

Número de literales

El número de literales por si solo no representa un factor determinante para la satisfacibilidad, dado que no se observa una diferenciación significativa entre los diferentes valores.

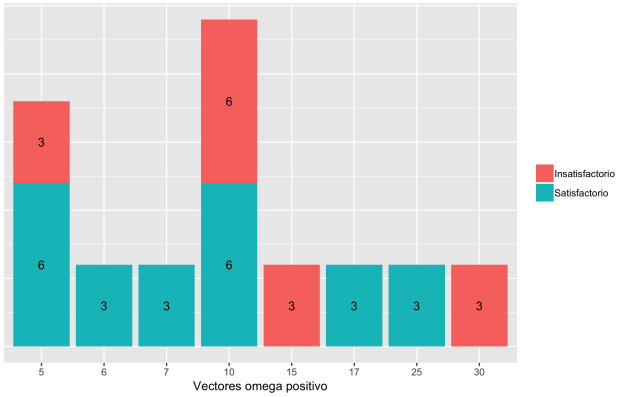

Figura 4: Rendimiento: Vectores de omega positivo

Número de vectores de omega positivo

El numero de vectores si es un factor determinante cuando se liga junto a la longitud de support set y el numero de literales, ya que a pesar de que la longitud del support set sea grande, si hay muchos vectores es complicado que despues de proyectar el support set todos ellos sean disjuntos.

5 Generador de datos

El generador con el cual se han generado el data set, sobre el que se han ejecutado las pruebas de programa, se ha realiza en el lenguaje de programación Python, concretamente con la versión 3.7. El programa se encuentra en el directorio 'MinizincDataGenerator/datagenerator.py'.

La principal funcionalidad del programa es generar archivos de datos aleatorios .dnz con el formato especificado en el enunciado de la práctica. Adicionalmente se le ha añadido la posibilidad de generar ficheros de datos aleatorios balanceados mediante un factor 'bias'.

La ejecución del programa puede realizarse desde la terminal utilizando el comando estándar de ejecución de python 'python datagenerator.py args*', donde args es el conjunto de parámetros que acepta el script. La lista de parámetros es la siguiente:

- **Obligatorios**
  - -t: Numero máximo de literales.
  - -k: Máxima longitud del "support set".
  - -n: Numero máximo de instancias positivas.
  - -m: Numero máximo de instancias negativas.
  - -c: Numero de restricciones atMostOne.
- **Opcionales**
  - -b: Valor del factor bias
  - -o: Nombre del fichero de datos (si hay mas de un fichero, se añade. automáticamente un numero distintivo al final del nombre).
  - -p: Directorio en el que guardar los ficheros de datos.
  - -n: Numero de ficheros a generar.

Ejemplo de uso

```
python datagenerator.py -t 10 -k 5 -n 5 -m 5 -c 4 -nf 10
```