uc3m | Universidad Carlos III de Madrid

HEURÍSTICA Y OPTIMIZACIÓN

Jorge Mejías Donoso 100495807

Alberto Menchen Montero 100495692

ÍNDICE

Introducción

En esta segunda práctica relacionada con la planificación y la optimización en el ámbito aeronáutico, vamos a utilizar otro sistema de resolución de problemas como son la resolución por Satisfacción de restricciones (CSP) y Búsqueda Heurística. El objetivo principal de dicha práctica es desarrollar un conjunto de soluciones válidas capaz de resolver las particiones del problema que se plantea.

La práctica se divide en dos partes:

1. **Mantenimiento de flota de aviones:** Consiste en asignar talleres o parkings a aviones en franjas horarias determinadas, respetando restricciones como la capacidad de los talleres, las tareas necesarias para cada avión y las que pueden desempeñar los talleres, todo ello sujeto a unas reglas que se deben respetar para una perfecta organización del problema.

2. **Planificación de Rodaje de Aviones:** Implica modelar y resolver un problema de movimiento en un aeropuerto, optimizando el tiempo necesario para que los aviones alcancen sus pistas asignadas sin percances.

Todo ello, pone en práctica los conocimiento obtenidos en los últimos temas de la asignatura para organizar y optimizar problemas reales de forma eficiente con modelados por CSP y uso de algoritmos de A* para búsqueda heurística.

Descripción del modelo Parte 1:

En esta sección pasamos a analizar detalladamente el modelo desarrollado para dar solución al problema de mantenimiento de flota de aviones usando la técnica de Satisfacción De Restricciones (CSP). El objetivo principal es asignar talleres y parkings a los aviones que haya como variables en el problema y las franjas horarias que se determinen, cumpliendo con una serie de restricciones operativas. Cabe recalcar que la complejidad del modelo está determinada por el número de variables que se introduzcan, número de aviones y franjas horarias.

1. Planteamiento del problema como CSP

Descripción del modelo: Un problema CSP se define como un conjunto (X, D, C) en donde tenemos:

- $X = \{X_{i,t}\},\ 1 \leq i \leq m,\ 1 \leq t \leq n,\ m * n = tamaño\, matriz$, el cual es el conjunto de variables del problema (la asignación de aviones en ubicaciones en las distintas franjas horarias)

- $D = \{D_{i,t}\},\ i, t \in \mathbb{N}$, el cual representa el dominio de cada variables en cada instante de tiempo
- $C = \{C_j\}\ 1 \leq j \leq m\ ,\ m = n^{\underline{o}}restricciones$, el cual es el conjunto de restricciones del problema

Variables: Cada avión en cada franja horaria es una variable, con sus atributos correspondientes: ID, tipo de avión, RESTR, nº actividades tipo estándar (T1) y nº de actividades tipo especialista (T2).

$$X = \{x_{i,t}\} \mid i \in A\ ,\ t \in T$$

Aviones: $A = \{a_1, a_2, ..., a_n\}$

Franjas Horarias: $T = \{t_1,\ t_2,\ ...,\ t_m\}$

Variable de decisión: $a_i = \ (ID_i,\ TIPO_i,\ RESTR_i,\ T1_i,\ T2_i)$

Dominio: El dominio de cada variable está compuesto por las posiciones de los talleres estándar, los talleres especialistas y los parkings disponibles.

$$D(x_{i,t}) = \{PRK \cup STD \cup SPC\}$$

$$i \in Aviones\ ,\ t \in Franjas$$

donde PRK, STD y SPC son posiciones en la matriz

Restricciones: Reglas establecidas en el enunciado para garantizar que las asignaciones sean válidas.

$$C = \{Restricciones\}$$

2. Restricciones del problema

a. Asignación completa

- Cada avión debe estar asignado a una posición válida en cada franja horaria.

$$\forall i \in A,\ \forall t \in T,\ x_{i.t} \neq \emptyset$$

b. Capacidad máxima de talleres:

- Cada taller puede atender un máximo de 2 aviones por franja horaria.

$$\forall (x, y) \in Matriz,\ \forall t \in T,\ card\{x_{i,t} = (x, y)\} \leq 2$$

- No puede haber más de un avión Jumbo en un taller en la misma franja horaria.

$$\forall (x, y) \in Matriz,\ \forall t \in T$$
$$TIPO_i = JMB \Longrightarrow card\{x_{i,t} = (x, y) \land TIPO_k = JMB\} \leq 1$$

- Hemos asumido que las combinaciones posibles de aviones en un taller son las siguientes: avión STD solo, avión STD con avión STD, avión STD con avión JMB y por último avión JMB solo

c. Compatibilidad entre talleres y tareas:

- Los talleres estándar sólo pueden realizar tareas de tipo 1.

$$\forall i \in A, \forall t \in T,\ x_{i,t} \in STD \Longrightarrow T2_i(t) = 0$$

- Los talleres especialistas pueden realizar tareas de tipo 1 y tipo 2 .

$$\forall i \in A,\ \forall t \in T,\ x_{i,t} \in SPC \Longrightarrow\ T2_i(t) \geq 0 \land T1_i(t) \geq 0$$

- Si un avión requiere tareas de tipo 2 antes que las de tipo 1, como indicaría el parámetro RESTR, estas tienen prioridad sobre las estándar.

$$\forall i \in A,\ RESTR_i = T \Longrightarrow \exists\ t_1,\ t_2 \in T\ con\ t_1 < t_2\ ,$$
$$T2_i(t_1) > 0 \Longrightarrow T1_i(t_2) = 0.$$

d. Restricciones de maniobrabilidad:

- Al menos, uno de los talleres o parkings adyacentes a una posición ocupada deben de estar vacíos para garantizar que los aviones tengan rango de maniobra.

$$\forall (x, y) \in Matriz,\ \forall i \in A,\ \forall t \in T$$
$$x_{i,t} = (x, y) \Longrightarrow \exists (x', y') \in Vecinos(x, y)\ tal\ que\ x_{k,t} = (x', y')$$
$$Vecinos(x, y) = \{(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1)\}$$

- Dos aviones Jumbo no pueden estar en talleres adyacentes en la misma franja horaria.

$$\forall i\ , k \in A$$
$$TIPO(a_i) = TIPO(a_k) = JMB \Longrightarrow Distancia(x_{i,t}, x_{k,t}) > 1,\ \forall\ t \in T$$

e. **Cunmplimieto íntegro de tareas:**

- Se deben cumplimentar todas las tareas asignadas a cada avión dentro de las franjas horarias.

$$\forall i \in A, \sum_{j \in T} T1_i(j) = T1_i^{total} \wedge \sum_{j \in T} T2_i(j) = T2_i^{total}$$

3. Análisis de Resultados

Para verificar el correcto funcionamiento de la implementación conseguida, se han desarrollado una serie de casos de prueba, que se pueden visualizar en la carpeta CSP-tests. Vamos a analizar uno para demostrar que se cumplen todas las restricciones íntegramente.

Número de franjas horarias: `3`
Tamaño de la matriz: `2x3`
Disposición de los talleres en la matriz:

```
STD:(0,1) (1,0)
SPC:(0,2)
PRK:(0,0) (1,1) (1,2)
```

Características de los aviones:

```
1-JMB-T-1-1
2-STD-F-2-0
3-STD-T-1-2
```

Para este ejemplo el problema ha encontrado 6696 soluciones distintas que cumplen todas las restricciones, analicemos algunas de ellas.

Solución 1:

```
1-JMB-T-1-1: SPC(0, 2) SPC(0, 2) SPC(0, 2)
2-STD-F-2-0: STD(1, 0) STD(1, 0) SPC(0, 2)
3-STD-T-1-2: SPC(0, 2) SPC(0, 2) STD(0, 1)
```

Como se puede observar en esta solución el avión 1 hace sus 2 tareas especiales en las 2 primeras franjas y luego se queda reposando en dicho taller. El avión 2 cumple sus dos tareas estándar en un taller estándar y se va a descansar al taller spc puesto que se ha determinado que también los talleres sirven como parking. El tercer y último avión cumple sus 2 tareas especiales antes que la tarea estándar, como indica una de las restricciones del problema cuando cuentan con el parámetro “REST” en “True”, y lo hace en un taller especial. Después, procede a irse a un taller estándar para hacer su tarea de tipo 1.

Solución 39:

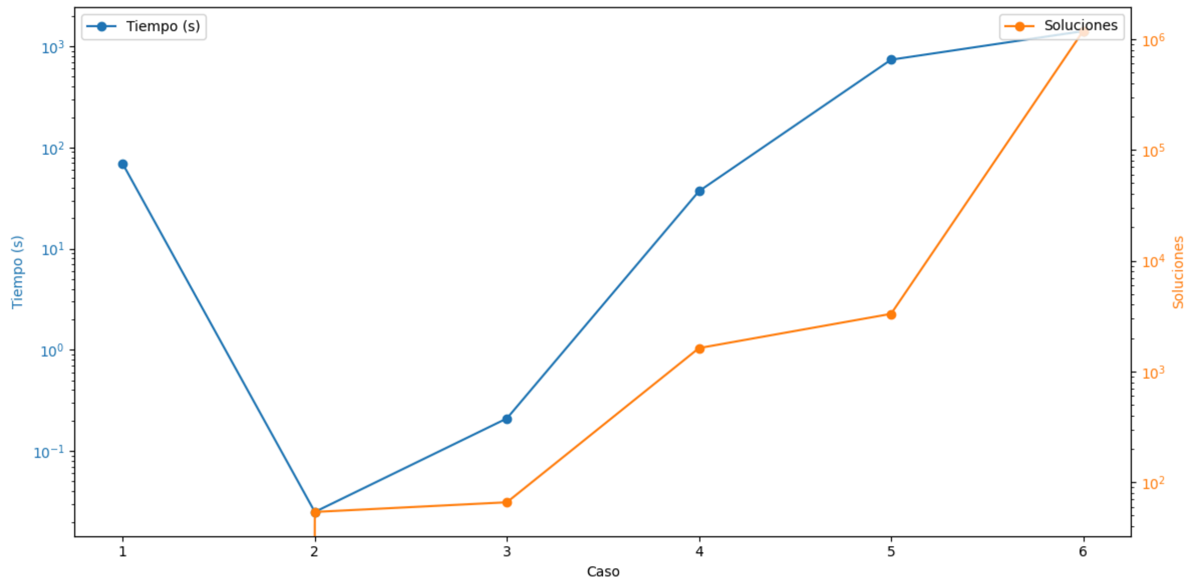
```
1-JMB-T-1-1: SPC(0, 2) SPC(0, 2) SPC(0, 2)
2-STD-F-2-0: STD(1, 0) PRK(1, 2) STD(0, 1)
3-STD-T-1-2: SPC(0, 2) SPC(0, 2) STD(0, 1)
```

En esta otra solución que es muy similar a la anterior, podemos ver que el avión 2 hace una tarea, se va a descansar a un parking y después vuelve a otro taller para realizar su tarea restante. Lo usamos como demostración de que los parkings también son usados por los aviones.

Nota: Los casos de prueba son indefinidos ya que se dispone de un algoritmo general que pasa las restricciones para una amplia variedad de datos. Indicamos uno de ellos para demostrar el perfecto funcionamiento del sistema. Además, indicamos la escalabilidad de los resultados mediante un gráfico representativo de cómo, aumentando el número de variables del problema, aumenta exponencialmente el número de soluciones y tiempo requerido para obtenerlas.

Resumen de Casos de Prueba

Caso	Franjas Horarias	Tamaño de Matriz	Aviones	Tiempo (s)	Soluciones
1	3	3x3	3	69.0	0
2	2	2x2	3	0.025	54
3	2	3x3	3	0.21	66
4	3	3x2	3	37.0	1626
5	3	3x3	3	738.0	3312
6	3	3x3	3	1415.0	1180164



Estos gráficos son un resumen de nuestros casos de prueba, podemos ver que el primero de todos tarda 69s en ejecutarse y no tiene solución ya que se tratan de 3 aviones JMB en los que deben de hacer tareas especiales y los 3 talleres SPC están en la misma columna por lo que no consiguen encontrar solución por la restricción de la adyacencia, tarda tanto ya que tiene 3 franjas horarias disponibles. El resto son casos normales en lo que se va incrementando en número de tareas y franjas horarias y vemos como crece exponencialmente el tiempo y las soluciones posibles.

Parte 2: Planificación con Búsqueda Heurística

4. Descripción del modelo

El problema se formula como un modelo de búsqueda basado en el algoritmo A estrella (A*), donde cada estado representa las posiciones de los aviones en un momento dado.
El objetivo de esta parte es encontrar una secuencia de movimientos que lleve a cada avión desde la posición de la que parte hasta su pista asignada, minimizando el tiempo total mientras se evitan posibles colisiones.

5. Estructura del modelo

Estado inicial: Posición inicial de todos los aviones y el tiempo de inicio.

Estado meta: Se alcanza cuando todos sus aviones llegan a su destino.

Funciones de transición: Los aviones pueden desplazarse hacia posiciones adyacentes o esperar si su posición actual se lo permite.

Restricciones: Se aplican las restricciones necesarias para evitar colisiones directas o cruzadas y respetar las celdas no transitables o de paso obligatorio.

Modelado del problema

El problema de planificación de rodaje de aviones se modela como un problema de búsqueda basado en estados. A continuación, se presenta su formalización:

1. **Definición del estado:**
    - Un estado s se define como:

$$s = (p, t, m)$$

Donde:

- $p = \{p_1, p_2, ..., p_n\}$ Representa las posiciones actuales de los aviones.
- t es el tiempo acumulado o makespan.
- $m = \{m_1, m_2, ..., m_n\}$ Es el conjunto de movimientos realizados por cada avión hasta el estado s.

2. **Estado inicial:**

$$s_0 = (p_{init},\ 0,\ m_{init})$$

Donde:

- pinit son las posiciones iniciales de los aviones.
- minit registra las posiciones iniciales como el primer movimiento.

3. **Estado objetivo:**

$$s_{goal} = (p_{goal},\ t,\ m)$$

Donde:

- $p_{goal} = \{g_1, g_2, ..., g_n\}$ son las metas de cada avión.
- t y m son el makespan y los movimientos realizados para alcanzar dichas metas.

4. **Espacio de estados:**
   - El espacio de estados está compuesto por todos los estados alcanzables desde s_0 hasta s_{goal}, considerando movimientos válidos para cada avión.

5. Transición entre estados:

Dado un estado $s = (p, t, m)$, las transiciones generan nuevos estados s′ mediante movimientos válidos:

$$s' = (p', t + 1, m')$$

Donde:

- p′ son las nuevas posiciones de los aviones.
- m′ incluye los movimientos realizados en t+1.

6. Función de costo:

- El costo de un estado s se calcula como:

$$f(s) = g(s) + h(s)$$

Donde:

- $g(s) = t$ es el tiempo acumulado (makespan).
- h(s) es la estimación heurística del costo restante.

6. Heurísticas Implementadas

Se han usado dos heurísticas para la resolución del problema:

H1: Heurística de Manhattan: Calcula la suma de las distancias Manhattan de cada avión a su meta. Es admisible porque siempre subestima el costo restante.

$$h_{suma}(p,g) = \sum_{i=1}^{n} \left(\left| (p_i^x - g_i^x) \right| + \left| (p_i^y - g_i^y) \right| \right)$$

H2: Máxima Manhattan: Tiene la misma funcionalidad que la primera, pero calcula la máxima distancia entre los aviones y sus metas. Es admisible de la misma manera y determina una aproximación más agresiva al estado objetivo.

$$h_{max}(p,g) = max \left| (p_i^x - g_i^x) \right| + \left| (p_i^y - g_i^y) \right|$$

7. Restricciones en la implementación

Manejo de colisiones:

- **Directas:** Se evita que dos aviones ocupen dos celdas al mismo tiempo

- **Cruzadas:** Se prohíbe el intercambio de posiciones entre dos aviones en pasos consecutivos.

Respetar mapa de movimiento: Las celdas de color gris no son transitables y las amarillas obligan a pasar sin deternerse.

Manejo de prioridades
Se utiliza una cola de prioridades que organiza los estados según costo acumulado más la heurística. Esto garantiza que los estados más prometedores se evalúan primero.

Movimientos válidos: Se garantiza que los aviones efectuarán movimientos hacia posiciones accesibles o permanecieron en su lugar (w) si alcanzaron el destino.

Posiciones meta: Todas las posibles soluciones son alcanzadas desde su posición inicial hasta la final.
Tiempo optimizado: rendimiento mejorado con uso de makespan para cada una de las heurísticas para garantizar el camino óptimo.

8. Análisis de Resultados

Soluciones del caso de prueba

Mapa del Caso

```
n° aviones: 2

Inicio-Fin avión 1:(3,3) (0,2)
Inicio-Fin avión 2:(0,2) (3,3)

Dimensión: 4x4

B;B;B;B
B;G;G;G
A;B;G;G
A;A;B;B
```

Heurística 1 (Manhattan Simple):

- **Avión 1:** `(3,3) ← (3,2) ← (3,1) ↑ (2,1) w (2,1) ← (2,0) ↑ (1,0) ↑ (0,0) → (0,1) → (0,2)`
- **Avión 2:** `(0,2) ← (0,1) ← (0,0) ↓ (1,0) ↓ (2,0) ↓ (3,0) → (3,1) → (3,2) → (3,3) w (3,3)`

Ambos aviones logran llegar a sus destinos, pero con un `makespan` de 9, ya que el Avión 1 realiza una espera en `(2,1)`.

Heurística 2 (Manhattan Máximo):

- **Avión 1:** `(3,3) w (3,3) ← (3,2) ← (3,1) ↑ (2,1) ← (2,0) ↑ (1,0) ↑ (0,0) → (0,1) → (0,2)`
- **Avión 2:** `(0,2) ← (0,1) ← (0,0) ↓ (1,0) ↓ (2,0) ↓ (3,0) → (3,1) → (3,2) w (3,2) → (3,3)`

El `makespan` también es de 9. Sin embargo, el Avión 1 realiza una espera en `(3,3)` al principio, mientras que el Avión 2 espera en `(3,2)` antes de llegar a su destino.

Estadísticas

Heurística 1:

- **Tiempo total:** `0.0056` segundos.
- **Makespan:** 9.
- **Heurística inicial:** 8.
- **Nodos expandidos:** `96`.

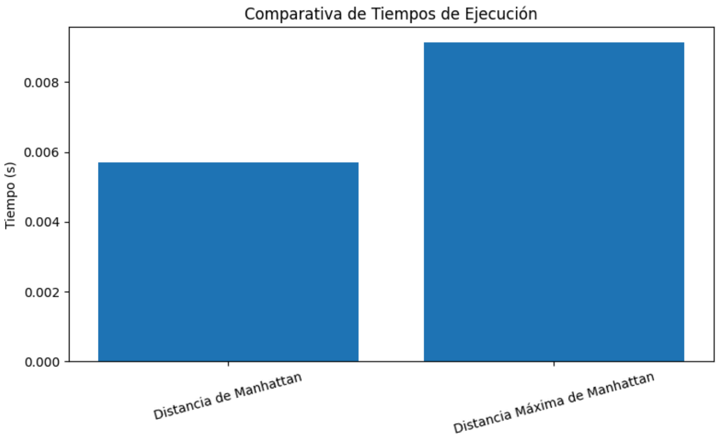
Heurística 2:

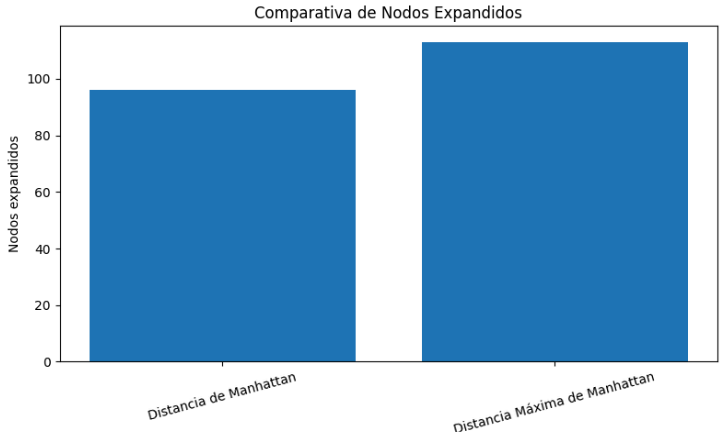
- **Tiempo total:** `0.0081` segundos.
- **Makespan:** 9.
- **Heurística inicial:** 4.
- **Nodos expandidos:** `113`.

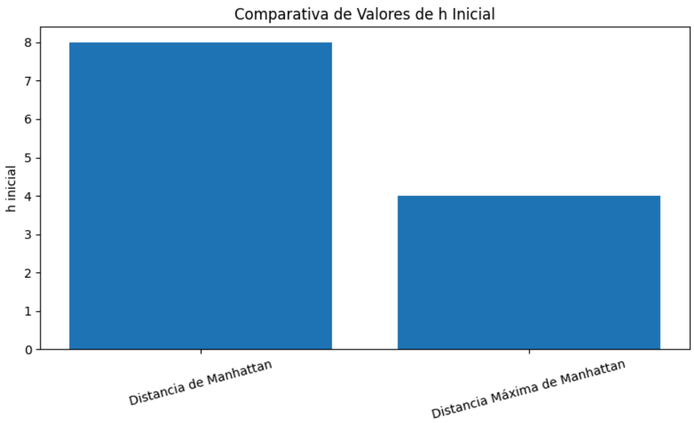
Comparación de las heurísticas

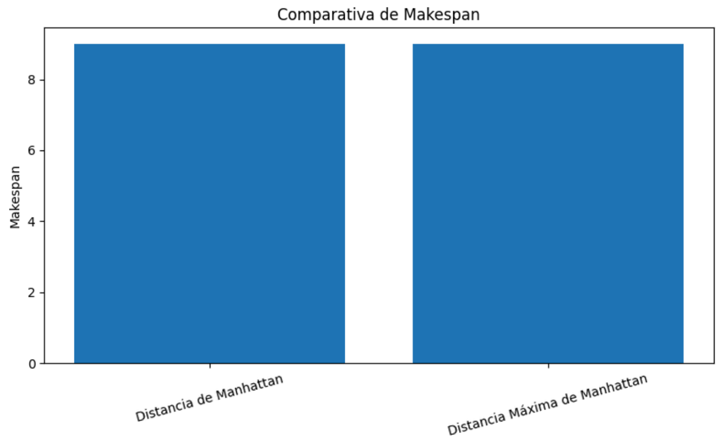
1. **Calidad de la solución (Makespan):** Ambas heurísticas alcanzan un `makespan` de 9, lo que indica que ninguna es mejor en términos de tiempo total en este caso.

2. **Costo heurístico inicial:**
    - Heurística 1: Valor inicial más alto (8), posiblemente una mejor aproximación al coste real.
    - Heurística 2: Valor más bajo (4), lo que podría reflejar una subestimación.
3. **Nodos expandidos:**
    - Heurística 1 expande menos nodos (96), siendo más eficiente.
    - Heurística 2 expande más nodos (113), lo que sugiere que requiere más exploración.
4. **Tiempo de ejecución:**
    - Heurística 1: Más rápida (`0.0056s`).
    - Heurística 2: Más lenta (`0.0081s`).









Conclusión

- **Eficiencia:** La **Heurística 1** es más eficiente en términos de tiempo y nodos expandidos.
- **Calidad:** Ambas heurísticas encuentran soluciones con el mismo `makespan`.
- **Solución óptima:** Para este tipo de problemas, **Heurística 1 (Manhattan Simple)** es más factible para un menor costo computacional.

Conclusión

El desarrollo de esta práctica representa la integridad de los temas de satisfacibilidad y búsqueda heurística aplicados para resolver problemas complejos de planificación y optimización. A lo largo de las dos partes que componen el proyecto, se han modelado, implementado y analizado soluciones efectivas que satisfacen las restricciones impuestas y optimizan lo máximo posible todos los recursos utilizados.

En la primer parte, mediante la formulación de un modelo de Satisfacción de restricciones CSP, se ha logrado abordar el problema del mantenimiento de una flota de aviones de manera estructurada y modular, capaz de asignar talleres y parkings respetando restricciones operativas como la capacidad, la secuencia de tareas y la maniobrabilidad. Los resultados mostraron que el modelo es sólido y cumple con las restricciones exigidas, aunque el tiempo de resolución se incrementa significativamente en escenarios con alta densidad de variables.
En la segunda parte, la implementación del algoritmo A* permitió encontrar rutas seguras y eficientes para aviones en movimiento, destacando la integración de restricciones de colisión y maniobrabilidad. El análisis comparativo de heurísticas mostró que la heurística basada en la suma de distancias Manhattan prioriza la rapidez en encontrar soluciones iniciales.

El trabajo refleja la capacidad de integrar técnicas avanzadas para resolver problemas prácticos, destacando la importancia del diseño de restricciones claras y heurísticas bien formadas. Como mejoras que se podrían añadir a la implementación, se sugiere optimizar el manejo de la combinación en CSP y añadir algunas heurísticas adicionales para mejorar el rendimiento y escalar el problema para tratar con un mayor abanico de datos y variables de entrada.