

Inteligencia Artificial

Informe Final: Defibrillator Relocation Problem

Cristopher Jiménez Meza

29 de julio de 2023

Resumen

Usar un desfibrilador en los primeros 3 - 5 minutos en una persona que acaba de sufrir un evento de paro cardíaco puede producir una tasa de supervivencia de 50-70 %. Para lograrlo es necesario contar con desfibriladores externos automáticos cerca del área del evento. Es aquí donde entra Defibrillator Relocation Problem, que busca encontrar las posiciones de los desfibriladores externos automáticos que maximizan la cobertura de los eventos de paros cardíacos históricos con el fin de lograr cubrir nuevos eventos que se produzcan cerca de los históricos. Es por ello que resulta interesante estudiar los métodos y/o técnicas que se han utilizado en el último tiempo con el fin de identificar cuales son los que abordan el problema de mejor manera y poder saber la orientación que tomarán futuros estudios.

1. Introducción

El presente documento tiene como propósito realizar una investigación de las técnicas y/o métodos utilizados en los últimos 7 años que han sido ocupados para resolver el problema Defibrillator Relocation Problem, esto incluye: tipos de modelos matemáticos utilizados para plantear el problema, técnicas empleadas para resolverlos y resultados obtenidos. Lo anterior se ha realizado con el fin de comprender los enfoques que se han utilizado para enfrentar el problema, de esta manera será posible identificar los mejores para saber la orientación que posibles estudios utilizarán. Además se utilizará un algoritmo de tipo Hill Climbing con mejor mejora que puede resolver instancias de problemas con enfoque fijo y flexible para analizar los resultados obtenidos tras resolver dichas instancias y poder obtener conclusiones sobre los pros y contras de cada uno de dichos enfoques.

El presente documento tiene la siguiente estructura: resumen, introducción, definición del problema, estado del arte, modelo matemático, representación, descripción del algoritmo, experimentos, resultados, conclusiones y referencias. En la sección estado del arte, se analizarán algunos de los estudios realizados en el período que abarca el año 2013 hasta el año 2018, la sección definición del problema ayudará al lector a entender en que consiste el problema que se estudiará, en modelo matemático se mostrará uno de los tantos modelos que se utilizan para representar el problema en cuestión, las secciones posteriores se enfocan a tratar el tema del algoritmo Hill Climbing con mejor mejora que será utilizado para resolver las instancias de problemas con enfoque fijo y flexible, en la sección representación se mostrará la forma en la cual se representará la solución manejada por el algoritmo y las ventajas de usar dicha representación, en descripción del algoritmo se explicará mediante el uso de pseudocódigo la estructura general del algoritmo utilizado, en experimentos se mencionará cual es el ambiente utilizado para ejecutar el programa, que pruebas fueron realizada, cuales fueron los parámetros utilizados, como fueron variando, entre otros temas y finalmente en la sección resultados se mostrán los resultados obtenidos que posteriormente serán analizados para obtener conclusiones acerca de los dos enfoques en cuestión.

Defibrillator Relocation Problem es un problema de optimización que consiste en encontrar las ubicaciones de un conjunto de desfibriladores externos automáticos de tal manera que estos puedan cubrir la mayor cantidad de paros cardíacos históricos en una determinada área geográfica de interés. La idea detrás de ello es que se pueda cubrir la mayor cantidad de nuevos paros cardíacos que ocurran cerca de las posiciones de los paros cardíacos históricos.

Como menciona el consejo europeo de la resucitación (en inglés, European Resuscitation Council o ERC) aplicar un desfibrilador a una persona que sufre de un evento de paro cardíaco dentro del rango de los primeros 3 - 5 minutos puede producir tasas de supervivencia de 50 - 70 %. La desfibrilación temprana de un paciente se puede lograr através de proveedores de desfibriladores de acceso público ubicados en las cercanías del suceso. Se considera rentable ubicar un desfibrilador externo automático en áreas donde se puede esperar un evento de paro cardíaco cada 5 años [1]. Por estas razones resulta interesante estudiar el Defibrillator Relocation Problem, con el fin de investigar nuevos métodos que permitan encontrar las ubicaciones de los desfibriladores externos automáticos que permitan maximizar la cobertura de eventos de paro cardíacos con el fin de mejorar las tasas de supervivencia de las personas que los sufren y de esta manera poder salvar una mayor cantidad de vidas.

2. Definición del Problema

Defibrillator Relocation Problem consiste en obtener las posiciones de un conjunto de desfibriladores externos automáticos (en inglés, Automated External Defibrillator o AED) que puedan optimizar la cobertura de eventos de paros cardíacos (en inglés, Out-of-Hospital Cardiac Arrest o OHCA) de un determinado lugar geográfico.

Entre las variables que se pueden encontrar en este problema están:

- Saber si un OHCA esta cubierto por algún AED.
- Saber si se instala un nuevo AED en una determinada ubicación o ya existe uno instalado previamente.
- Saber si un AED se mueve a una nueva ubicación.
- Saber si un determinado OHCA esta cubierto por un AED específico.

El objetivo de este problema es encontrar las posiciones del conjunto de AEDs que maximizan la cobertura de OHCA's históricos, ya que la idea detrás de esto es que sea posible cubrir la mayor cantidad de nuevos OHCA's que se produzcan cerca de los OHCA's históricos.

Entre las restricciones asociadas a este problema, se pueden encontrar:

- No es posible sobrepasar el presupuesto con el que se cuenta.
- Para que un OHCA sea cubierto debe haber a lo menos un AED que se encuentre localizado a una distancia igual o menor a 100 metros (esto se debe a que existe un rango de tiempo crítico en el cual se debe suministrar el AED al afectado).
- En caso de tener que mover AEDs existentes estos solo pueden ser movidos a lugares candidatos para instalar un AED.
- Se puede colocar o mover un AED a una determinada ubicación, pero estas dos operaciones no pueden ser realizadas al mismo tiempo sobre la misma ubicación.

Entre las variantes más conocidas de este problema se pueden encontrar:

- Ubicación fija: Esta variante del problema considera que el conjunto de AEDs que se encuentran instalados previamente no se pueden reubicar, por lo tanto solo es posible agregar nuevos AEDs (comprarlos) en los lugares candidatos, esta es la variante más sencilla de Defibrillator Relocation Problem.
- Ubicación flexible: Esta variante del problema permite relocalizar AEDs previamente instalados, además de instalar nuevos AEDs. En esta variante los costos unitarios de mover un AED previamente instalado es mucho menor que el costo unitario de instalar un nuevo AED.

Un problema relacionado a Defibrillator Relocation Problem es el de Ambulance Relocation Problem [2], este problema consiste en ubicar los vehículos de emergencia de tal forma que se pueda cubrir la demanda de estos de manera efectiva (en otras palabras se busca posicionar las ambulancias de tal forma que se pueda cubrir la mayor cantidad de emergencias posibles). Los vehículos se ubican de tal forma que en caso de existir algún llamado de emergencia estos puedan llegar lo más rápido posible al sitio del suceso, además un paciente se considera cubierto si existe una ambulancia que puede llegar a él dentro de un determinado límite de tiempo.

3. Estado del Arte

El Defibrillator Relocation Problem nace de la necesidad de que un transeúnte (espectador) administre el tratamiento relacionado con utilizar un AED sobre una persona que esta sufriendo un evento OHCA lo más rápido posible, ya que el tiempo promedio entre la llamada realizada a los servicios médicos de emergencia y la llegada de estos va desde los 5 a 8 minutos a lo anterior hay que agregarle el tiempo que se demoran en aplicar la primera descarga al paciente, aumentando el tiempo mencionado anteriormente a un rango de 8 a 11 minutos [1]. Es por ello que es de vital importancia que los transeúntes sean capaces de administrar el tratamiento lo más rápido posible, ya que aplicar un desfibrilador a una persona que sufre de un evento de paro cardíaco dentro del rango de los primeros 3 - 5 minutos puede producir tasas de supervivencia de 50 - 70 % tal como menciona el consejo europeo de la resucitación (en inglés, European Resuscitation Council o ERC).

Chan et al. (2013) [3]. Tuvieron como objetivo desarrollar un método geoespacial que utiliza optimización matemática para identificar grupos de paros cardíacos y priorizar ubicaciones públicas para el despliegue de AEDs. El modelo fue probado usando datos de OHCA de la ciudad de Toronto, Canadá, se observó un aumento de la cobertura de OHCA de un 23 % a un 32 % y la distancia promedio desde un OHCA hasta el AED más cercano bajó de 281 metros a 262 metros.

Sun et al. (2016) [4]. Desarrollaron un modelo de optimización para el despliegue de AEDs que tenía en consideración el acceso espacial y temporal al AED, dicho modelo buscaba determinar sus ubicaciones para maximizar la cobertura real de los OHCA. Se probó contra otro que sólo consideraba el acceso espacial, para ambos se utilizaron datos de Toronto, Canadá, el modelo espacio - temporal logró un aumento relativo del 25,3 % en la cobertura real en comparación con el modelo que sólo consideraba el acceso espacial.

Chan et al. (2016) [5]. Abordaron el problema mediante modelos de optimización que generalizan el problema de ubicación de cobertura máxima (MCLP), los modelos resultantes tratan el problema Defibrillator Relocation Problem como un MCLP pero añaden restricciones basadas en una función de probabilidad utilizada para saber el nivel de cobertura de OHCA, los autores consideraron tres casos realistas por ello generaron tres modelos cuya diferencia radica en la función de probabilidad utilizada, estos modelos cubren los casos: 1. Respondedor múltiple (representa una situación en la que varios transeúntes se encuentran frente a un OHCA y buscan de manera independiente un AED y lo llevan al sitio del suceso), 2. Un respondedor en el peor caso (se da cuando se produce un OHCA y un solo transeúnte busca un AED, este encuentra el más lejano siendo que existen AEDs cercanos y lo lleva al sitio del suceso), 3. Un respondedor en el mejor caso (se da cuando se produce un OHCA y un solo transeúnte busca un AED, este encuentra el más cercano y lo lleva al sitio del suceso). Se probaron los modelos con datos de la ciudad de Toronto, Canadá, se obtuvieron 1669 AEDs registrados en Toronto y se simuló un conjunto de 5000 OHCA, al realizar la reubicación de AEDs se obtuvo una cobertura media para el modelo 1 de 28,4 % sobre los 100 conjuntos de prueba, para obtener la misma cobertura teniendo en cuenta los 1669 AEDs iniciales y sin reposicionamiento, sería necesario agregar 900 AEDs.

Chan et al. (2017) [6]. Presentan un modelo de optimización robusto para la implementación de AED que tiene en cuenta la incertidumbre en la distribución espacial de los paros cardíacos además se ha aplicado el algoritmo de generación de fila y columna para resolverlo. Se utilizaron datos reales de paros cardíacos ocurridos en la ciudad de Toronto, Canadá, los resultados mostraron que tener en cuenta la incertidumbre en las ubicaciones de un paro cardíaco puede reducir la distancia entre las víctimas de paro cardíaco y el AED más cercano entre 9 - 15 %.

Tierney et al. (2018) [7]. Proponen resolver el problema usando un modelo flexible. Luego de probar los modelos de población, fijo y flexible en datos de la localidad de Ticino y utilizando restricciones de presupuesto que consistían en colocar 20, 40, 60, 80 y 100 AEDs (donde cada AED cuesta 6,000 euros) el porcentaje de eventos OHCA que logró cubrir cada modelo para los presupuestos 120,000 y 600,000 euros fueron: modelo de población 23,59 - 27,01 %, modelo fijo 26,40 - 34,51 %, modelo flexible 37,54 - 49,85 %. El modelo flexible logró cubrir una cantidad mayor de eventos en comparación con los otros modelos, para el caso de presupuesto 120,000 euros el modelo reubicó 171 AEDs y para el caso de 600,000 euros el modelo reubicó 604 AEDs e instaló 29, mediante la combinación de redistribución e instalación se pudo mejorar la cobertura de una manera significativa.

Nazarian (2018) [8]. Aplica el problema generalizado de ubicación de máxima cobertura (GMCLP) para enfrentar el problema de despliegue de AEDs. Además propone métodos heurísticos de optimización que comprometen una heurística codiciosa de manera eficiente y efectiva además de un algoritmo híbrido más complejo basado en una combinación del procedimiento de búsqueda adaptativa aleatoria codiciosa (GRASP) y el algoritmo de recocido simulado (SA). Se utilizaron datos reales de North Holland y de Twente regiones de los países bajos, se reubicaron AEDs existentes en 43 municipalidades dentro del área de estudio, se mostró que la proporción promedio de instancias del problema donde se puede recuperar un AED dentro de los primeros 6 minutos críticos se puede mejorar de un 47,2 % a un 68,5 %, además usando una función de decaimiento de la cobertura más realista, la cobertura de futuros paros cardíacos mejora en un 73,5 %.

4. Modelo Matemático

Modelo basado en ubicación flexible

El presente modelo ha sido mostrado por Gallardo S. (2020) [9]. este a su vez se basa en la notación de Chan et al. (2016).

Parámetros

- J : Conjunto de eventos OHCA.
- I_c : Conjunto de lugares candidatos a instalar un AED.
- I_e : Conjunto de lugares con un AED previamente instalado.
- $I = I_c \cup I_e$. Conjunto de lugares que incluye tanto sitios donde se han ubicado previamente AEDs como sitios candiatos a instalar AEDs. Hay que tener en cuenta que $I_c \cap I_e = \emptyset$.
- a_{ij} : Vale 1 si el evento OHCA j esta a menos de 100 metros del lugar o locación i . $i \in I$, $j \in J$. 0 en caso contrario.
- CE: Costo unitario de mover un AED de una posición en I_e a otra en I_c .
- CN: Costo unitario de instalar un nuevo AED en una posición en I_c .
- B: budget, es decir, presupuesto total para la instalación de AEDs.

Variables

- x_j : Vale 1 si el evento OHCA j esta cubierto por algún AEd, 0 en caso contrario. $j \in J$.
- y_i : Vale 1 si se instala un nuevo AED en la ubicación i , o bien ya existe uno instalado previamente, 0 en caso contrario. $i \in I$.
- v_{ik} : Vale 1 si un AED en la ubicación i se mueve a la ubicación k , 0 en caso contrario. $i, k \in I$.
- Z_{ij} : Vale 1 si un nuevo evento OHCA j esta cubierto por el AED en la posición i , vale 0 en caso contrario. $i \in I, j \in J$.

Función objetivo

$$\text{máx } Z_{drp} = \sum_{j \in J} x_j$$

Restricciones

- El costo total de instalación no supera el budget.

$$\sum_{i, k \in I, i \neq k} CE \times v_{ik} + \sum_{i \in I_c} CN \times y_i \leq B$$

- Un AED en la posición i puede cubrir un evento OHCA j si y solo si está dentro del rango de los 100 metros.

$$z_{ij} \geq a_{ij} \cdot y_i + a_{ij} \cdot \sum_{k \in I_e, i \neq k} v_{ki} \quad \forall i \in I, j \in J$$

- Se setea x_j si esta cubierto por algún AED

$$x_j = \sum_{i \in I} z_{ij} \quad \forall j \in J$$

- y_i , cuando $i \in I_e$, vale 1 si y solo si no se ha movido el AED asociado. Esta restricción controla y_i en el caso de AEDs previamente existentes.

$$y_i = 1 - \sum_{k \in I_c} v_{ik} \quad \forall i \in I_e$$

- Solo se puede mover AEDs desde posiciones en I_e a posiciones candidatas (I_c)

$$\begin{aligned} v_{ik} &= 0 & \forall k \in I_e, i \in I \\ v_{ki} &= 0 & \forall k \in I_c, i \in I \end{aligned}$$

- Se puede instalar un nuevo AED en la posición i , o mover alguno a la misma posición, pero no ambas al mismo tiempo.

$$y_i + \sum_{k \in I, i \neq k} w_{ki} \leq 1 \quad \forall i \in I$$

Dominio de variables

$$x_j, y_i, z_{ij}, v_{ik} \in \{0, 1\} \quad \forall i, k \in I, i \neq k, j \in J$$

5. Representación

El algoritmo Hill Climbing con mejor mejora diseñado para resolver las variantes flexible y fijo del problema Defibrillator Relocation Problem proporcionadas para realizar este trabajo, utiliza la siguiente representación matemática para manejar las soluciones del problema en cuestión:

Representación matemática

$$\vec{X}: X_i \in \{0, 1\} \text{ e } 1 \leq i \leq N$$

- N : Representa la cantidad de eventos OHCA que se deben cubrir, proporcionados en las diferentes instancias del problema que se debe resolver.
- i : Representa un evento OHCA específico, por lo tanto el anterior vector representa todos los eventos OHCA que deberían ser cubiertos.
- X_i : Indica la existencia de un AED ubicado en el evento OHCA i . Si X_i vale 1, quiere decir que existe un AED cubriendo el evento OHCA i , en caso contrario X_i vale 0.

Debido a que la representación matemática elegida ha sido un vector, se ha elegido la estructura de datos de tipo vector para representarla computacionalmente.

Los beneficios de utilizar la representación anterior son:

- Manejo de restricciones: Se pueden manejar dos restricciones.
 1. Colocar AEDs en posiciones válidas: Una de las restricciones es que los AEDs deben ser ubicados en las posiciones donde existe un evento OHCA. Debido a que el vector utilizado contiene todos los eventos OHCA que deben ser cubiertos, los AEDs siempre serán colocados en posiciones válidas.
 2. No colocar más de un AED en la misma posición: Debido a que el vector está compuesto solo por 1's y 0's que representan la existencia de un AED en una determinada ubicación, significa que en una misma posición puede o no existir un único AED.
- Fácil de usar: Realizar operaciones tales como movimiento, función de evaluación, generación de soluciones iniciales resulta bastante fácil, por ejemplo:
 - Para el caso de movimiento, este algoritmo realiza un flip bit.
 - Para el caso de función de evaluación, lo que se hace es operar sobre el bit cambiado (en caso de agregar un AED) u operar sobre todos los bit existentes (en caso de quitar un bit).
 - Para el caso de generar soluciones iniciales, el algoritmo genera soluciones con 1's y 0's aleatorios.

La relación entre la estructura de datos utilizada y la representación matemática es que ambas representan un vector de eventos OHCA donde se deben colocar AEDs, los cuales se representan por un 1 o un 0 que indican si existe un AED o no respectivamente, también las dos poseen un largo igual a la cantidad de eventos OHCA que deben ser cubiertos.

6. Descripción del algoritmo

A continuación se presentan los pseudocódigos de las partes más importantes del algoritmo y sus correspondientes explicaciones:

Algorithm 1 Generador de coberturas

Require: *coords_x*: posición en el eje x de los eventos OHCA, *coords_y*: posición en el eje y de los eventos OHCA, *radio*: radio de cobertura de cada AED.

Ensure: *coberturas*: Vector de cobertura de cada OHCA

```
1: procedure OBTENER_COBERTURAS(coords_x, coords_y, radio)
2:   coberturas.inicializar()
3:   for x_act, y_act in coords_x, coords_y do
4:     cobertura.inicializar()
5:     for x_sig, y_sig in coords_x, coords_y do
6:       if calcular_radio(x_act, x_sig, y_act, y_sig)  $\leq$  radio then
7:         cobertura.insertar((x_sig, y_sig))
8:       end if
9:     end for
10:    coberturas.insertar(cobertura)
11:  end for
12:  return coberturas
13: end procedure
```

El generador de coberturas se basa en la siguiente idea: si se coloca un AED en un determinado evento OHCA, existe un conjunto de otros eventos OHCA que se encuentran dentro del radio de cobertura del AED instalado. Es por ello que para cada evento OHCA se genera un conjunto de todos aquellos eventos OHCA (posiciones) que se encuentran dentro del radio de cobertura dado y esto es agregado a una estructura de tipo vector (donde cada índice del vector representa un determinado evento OHCA que es el centro de su radio de cobertura). La estructura de datos generada se utiliza en la función de evaluación para consultar la cobertura total de las soluciones.

Algorithm 2 Generador de soluciones iniciales para enfoque fijo

Require: *largo*: cantidad de eventos OHCA que se deben cubrir, *presupuesto*: cantidad de recursos disponibles para colocar AEDs, *n_aeds*: número de AEDs iniciales (esta variable se usa en caso de que el problema sea de tipo flexible).

Ensure: *solucion*: vector de 1's y 0's que representa una solución candidata inicial.

```
1: procedure GENERAR_SI_FIJO(largo, presupuesto, n_aeds)
2:   solucion.inicializar(largo, 0)
3:   presupuesto  $\leftarrow$  presupuesto * (1 + ((random() % 4) / 10))
4:   for c in [0, ..., largo - 1] do
5:     hay_aed  $\leftarrow$  ((random() % 101) / 100)  $\geq$  0,9
6:     if presupuesto = 0 then
7:       break
8:     else if hay_aed then
9:       solucion[c]  $\leftarrow$  1
10:      presupuesto  $\leftarrow$  presupuesto - 1
11:    end if
12:  end for
13:  return solucion
14: end procedure
```

Para que el algoritmo diseñado pueda trabajar las soluciones de mejor manera, se ha acotado la cantidad de AEDs existentes a un rango que se mueve entre presupuesto y presupuesto + 40 %, esto se debe a que las soluciones factibles tendrán una cantidad máxima de AEDs igual a la cantidad de presupuesto con el cual se cuenta, es por ello que partir con soluciones que tengan una cantidad de AEDs un poco mayor al presupuesto dado, acota bastante el trabajo que el algoritmo debe realizar.

Algorithm 3 Generador de soluciones iniciales para enfoque flexible

Require: *largo*: cantidad de eventos OHCA que se deben cubrir, *presupuesto*: cantidad de recursos disponibles para colocar AEDs, *n_aeds*: número de AEDs iniciales (esta variable se usa en caso de que el problema sea de tipo flexible).

Ensure: *solucion*: vector de 1's y 0's que representa una solución candidata inicial.

```
1: procedure GENERAR_SI_FLEXIBLE(largo, presupuesto, n_aeds)
2:   solucion.inicializar(largo, 0)
3:   for c in [0, ..., largo - 1] do
4:     hay_aed  $\leftarrow$  ((random() % 101) / 100)  $\geq$  0,7
5:     if n_aeds = 0 then
6:       break
7:     else if hay_aed then
8:       solucion[c]  $\leftarrow$  1
9:       n_aeds  $\leftarrow$  n_aeds - 1
10:    end if
11:  end for
12:  return solucion
13: end procedure
```

La idea es similar a la expuesta en Generador de soluciones iniciales para enfoque fijo, la diferencia es que en este tipo de problemas se cuenta con una cantidad de AEDs inicialmente posicionados, es por ello que las soluciones generadas deben tener esa misma cantidad de AEDs, además se ha aumentado la probabilidad de agregar AEDs a la solución inicial para que calcen con la cantidad de AEDs iniciales

Algorithm 4 Función de costo para enfoque fijo

Require: *aeds_iniciales*: vector de 0's y 1's que indican en que evento se encontraba posicionado un determinado AED inicialmente (utilizado para los problemas de tipo flexible), *solucion_candidata*: vector de 0's y 1's que indican en que evento se encuentra cada AED de la solución candidata.

Ensure: costo asociado a la solución.

```
1: procedure CCC_FIJO(aeds_iniciales, solucion_candidata)
2:   return acumulado(solucion_candidata)
3: end procedure
```

Para este caso, debido a que todos los AEDs agregados cuestan 1, basta solamente con sumar todos los AEDs encontrados en la solución para obtener su costo.

Algorithm 5 Función de costo para enfoque flexible

Require: *aeds_iniciales*: vector de 0's y 1's que indican en que evento se encontraba posicionado un determinado AED inicialmente (utilizado para los problemas de tipo flexible), *solucion_candidata*: vector de 0's y 1's que indican en que evento se encuentra cada AED de la solución candidata.

Ensure: costo asociado a la solución.

```
1: procedure CCC_FLEXIBLE(aeds_iniciales, solucion_candidata)
2:   aeds_movidos  $\leftarrow$  0
3:   n_aeds_iniciales  $\leftarrow$  acumulado(aeds_iniciales)
4:   n_aeds_sol_candidata  $\leftarrow$  acumulado(solucion_candidata)
5:   aeds_agregados  $\leftarrow$  n_aeds_sol_candidata - n_aeds_iniciales
6:   if aeds_agregados  $\leq$  0 then
7:     aeds_agregados  $\leftarrow$  0
8:   end if
9:   for aed_inicial, aed_sol_cand in aeds_iniciales, solucion_candidata do
10:    if aed_inicial = 1 and aed_sol_cand = 0 then
11:      aeds_movidos  $\leftarrow$  aeds_movidos + 1
12:    end if
13:  end for
14:  return aeds_agregados + 0,2 * aeds_movidos
15: end procedure
```

Debido a que se pueden agregar y mover AEDs, primero se hace la diferencia entre los AEDs de la solución candidata y los posicionados inicialmente (entregados en la instancia del problema) solamente si la diferencia es positiva significa que han sido agregados nuevos AEDs, por lo tanto esa diferencia es el costo de haber instalado nuevos AEDs. Luego se revisan los AEDs posicionados inicialmente y los de la solución candidata, si existía un AED cubriendo un evento inicialmente pero no aparece cubriendo el mismo evento en la solución candidata, significa que dicho AED ha sido movido (o eliminado que para este algoritmo también lo considera como movido) y se actualiza la variable que registra dicha información, cuando ya se han examinado todos los AEDs se retorna el costo de la solución.

Algorithm 6 Cobertura total inicial

Require: *coberturas*: vector con las coberturas de cada evento OHCA, *solucion_candidata*: vector de 0's y 1's que indican en que evento se encuentra cada AED de la solución candidata.

Ensure: *ResultadoCT*: Objeto que contiene la información asociada a la cobertura de la solución inicial.

```
1: procedure COBERTURA_TOTAL_INICIAL(coberturas, solucion_candidata)
2:   eventos_cubiertos.inicializar()
3:   for cob, sol in coberturas, solucion_candidata do
4:     if sol = 1 then
5:       eventos_cubiertos.insertar(cob)
6:     end if
7:   end for
8:   return ResultadoCT(eventos_cubiertos.largo(), eventos_cubiertos)
9: end procedure
```

Por cada AED posicionado en un determinado evento OHCA se calcula la cobertura inicial, tomando todos los conjuntos de eventos OHCA cubierto por cada AED, luego se unen y se almacena el largo del resultado (que corresponde a la cantidad de eventos cubiertos) y el conjunto resultante.

Algorithm 7 Cobertura total

Require: *coberturas*: vector con las coberturas de cada evento OHCA, *eventos_cubiertos*: conjuntos con las coordenadas de todos los eventos OHCA que son cubiertos actualmente, *solucion_candidata*: vector de 1's y 0's que representa la solución candidata, *posicion*: entero que indica el evento OHCA donde se ha agregado o quitado un AED, *agregado*: booleano que indica si se ha agregado o quitado un AED

Ensure: *ResultadoCT*: Objeto que contiene la información asociada a la cobertura actual.

```
1: procedure COBERTURA_TOTAL(coberturas, eventos_cubiertos, solucion_candidata,
   posicion, agregado)
2:   nuevos_eventos_cubiertos  $\leftarrow$  eventos_cubiertos
3:   if not agregado then
4:     nuevo.inicializar()
5:     for count in [0, ..., largo - 1] do
6:       if solucion_candidata[count] then
7:         nuevo.insertar(coberturas[count])
8:       end if
9:     end for
10:    return ResultadoCT(nuevo.largo(), nuevo)
11:   end if
12:   nuevos_eventos_cubiertos.insertar(coberturas[posicion])
13:   return ResultadoCT(nuevos_eventos_cubiertos.largo(), nuevos_eventos_cubiertos)
14: end procedure
```

Debido a que el movimiento utilizado por el algoritmo diseñado consiste en agregar o quitar un solo AED a la vez, para obtener la cobertura total de una solución candidata, se realizan dos operaciones dependiendo de si se agrega o quita un AED.

1. Si se ha quitado un AED: se realiza la unión de todos los eventos OHCA cubiertos por los AEDs existentes.
2. Si se ha agregado un AED: se realiza una unión entre el conjunto de todos los eventos OHCA cubiertos por la mejor solución actual y el conjunto de todos los eventos OHCA

cubiertos por el AED agregado.

Finalmente se retorna la información asociada a la cobertura.

Algorithm 8 Función de evaluación

Require: *coberturas*: vector con las coberturas de cada evento OHCA, *aeds_iniciales*: vector de 1's y 0's que indican si existe un AED posicionado en un evento OHCA, *solucion_candidata*: vector de 1's y 0's que representa una solución candidata generada tras aplicar el movimiento flip bit a la mejor solución actual, *eventos_cubiertos*: conjunto con los eventos cubiertos por la mejor solución manejada por Hill Climbing, *posicion*: entero que indica el evento OHCA donde se ha agregado o quitado un AED, *agregado*: booleano que indica si se ha agregado o quitado un AED, *cost*: función de costo que depende del tipo de problema a resolver (fijo o flexible), *presupuesto*: flotante que representa el presupuesto disponible para colocar o mover AEDs.

Ensure: *ResultadoFEv*: objeto que contiene la información asociada al costo y cobertura de la solución candidata.

```
1: procedure F_EV(coberturas, aeds_iniciales, solucion_candidata, eventos_cubiertos,  
   posicion, agregado, cost, presupuesto)  
2:   costo  $\leftarrow$  cost(aeds_iniciales, solucion_candidata)  
3:   cobertura  $\leftarrow$  cobertura_total(coberturas, eventos_cubiertos, solucion_candidata, posicion, agregado)  
4:   return ResultadoFEv(costo, cobertura.obtenerCobertura(), cobertura.obtenerEventosCubiertos())  
5: end procedure
```

Esta función se encarga de llamar a la función *cost* que sirve para calcular el costo de la solución candidata (dependiendo del tipo de problema es si se usa la función de costo fijo o flexible), además se llama a la función *cobertura_total* para calcular la cobertura y obtener el conjunto de todos los eventos OHCA cubiertos por la solución candidata.

Algorithm 9 Hill Climbing mejor mejora

Require: *coberturas*: vector con las coberturas de cada evento OHCA, *aeds_iniciales*: vector de 1's y 0's que indican si existe un AED posicionado en un evento OHCA, *solucion_inicial*: vector de 1's y 0's que representa una solución inicial, *radio*: entero que representa el radio de cobertura de los AEDs, *presupuesto*: flotante que representa el presupuesto disponible para colocar o mover AEDs, *cost*: función de costo que depende del tipo de problema a resolver (fijo o flexible).

Ensure: *ResultadoHCMM*: contiene la información asociada a la mejor solución, mejor costo y mejor cobertura.

```
1: procedure HC_MM(coberturas, aeds_iniciales, solucion_inicial, radio, presupuesto, cost)
2:   mejor_costo  $\leftarrow$  cost(aeds_iniciales, solucion_inicial)
3:   mejor_costo_act  $\leftarrow$  mejor_costo
4:   resultado_cobertura  $\leftarrow$  cobertura_total_inicial(coberturas, solucion_inicial)
5:   mejor_cobertura  $\leftarrow$  resultado_cobertura.obtenerCobertura()
6:   mejor_cobertura_act  $\leftarrow$  mejor_cobertura
7:   mejor_eventos_cubiertos_act  $\leftarrow$  resultado_cobertura.obtenerEventosCubiertos()
8:   mejor_eventos_cubiertos  $\leftarrow$  mejor_eventos_cubiertos_act
9:   mejor_solucion  $\leftarrow$  mejor_solucion_act
10:  solucion_candidata  $\leftarrow$  mejor_solucion_act
11:  aed_agregado
12:  hay_mejor_solucion  $\leftarrow$  true
13:  while hay_mejor_solucion do
14:    hay_mejor_solucion  $\leftarrow$  false
15:    for count in  $[0, \text{mejor\_solucion\_act.largo}() - 1]$  do
16:      if solucion_candidata[count] then
17:        solucion_candidata[count]  $\leftarrow$  0
18:      else
19:        solucion_candidata[count]  $\leftarrow$  1
20:      end if
21:      aed_agregado  $\leftarrow$  solucion_candidata[count]
22:      calidad  $\leftarrow$  f_ev(coberturas, aeds_iniciales, solucion_candidata, mejor_eventos_cubiertos_act,
23:        count, aed_agregado, cost, presupuesto)
24:      costo_act  $\leftarrow$  calidad.obtenerCosto()
25:      cobertura_act  $\leftarrow$  calidad.obtenerCobertura()
26:      if (mejor_costo_act > presupuesto and costo_act < mejor_costo_act) or
        (costo_act  $\leq$  presupuesto and cobertura_act > mejor_cobertura_act) then
27:        mejor_solucion  $\leftarrow$  mejor_solucion_act
28:        mejor_costo  $\leftarrow$  costo_act
29:        if mejor_costo > presupuesto then
30:          mejor_cobertura  $\leftarrow$  0
31:        else
32:          mejor_cobertura  $\leftarrow$  cobertura_act
33:        end if
34:        mejor_eventos_cubiertos  $\leftarrow$  calidad.obtenerEventosCubiertos()
35:        hay_mejor_solucion  $\leftarrow$  true
36:      end if
37:      if solucion_candidata[count] then
38:        solucion_candidata[count]  $\leftarrow$  0
39:      else
40:        solucion_candidata[count]  $\leftarrow$  1
41:      end if
42:    end for
43:    mejor_costo_act  $\leftarrow$  mejor_costo
44:    mejor_cobertura_act  $\leftarrow$  mejor_cobertura
45:    solucion_candidata  $\leftarrow$  mejor_solucion
46:    mejor_eventos_cubiertos_act  $\leftarrow$  mejor_eventos_cubiertos
47:  end while
48:  return ResultadoHCMM(mejor_solucion, mejor_costo, mejor_cobertura)
49: end procedure
```

Las características más destacadas del presente algoritmo Hill Climbing con mejor mejora son:

1. Movimiento: Debido a la representación utilizada, se ha decidido aplicar el movimiento flip bit para generar el vecindario.
2. Función de evaluación: La calidad de la solución viene determinada por dos factores: la cobertura total y el costo asociado, es por ello que la función de evaluación utilizada calcula y retorna esos dos componentes.
3. Criterio de aceptación de la solución: Las soluciones aceptadas por el presente algoritmo pueden ser factibles y no factibles (el algoritmo se mueve en zonas no factibles y factibles para encontrar soluciones) esto se logra de la siguiente forma:
 - Si el costo de la mejor solución actual sobrepasa el presupuesto y el costo de la solución candidata tiene un costo menor al de la mejor solución actual, entonces dicha solución candidata se vuelve mejor solución (caso no factible y factible): Cabe tener en cuenta que existe la posibilidad de que la solución inicial no cumpla con la restricción del presupuesto y que su vecindario tampoco lo haga, es por ello que para lidiar con dichos escenarios se ha decidido aceptar soluciones que no cumplan con la restricción del presupuesto para poder explorar otras zonas del espacio de búsqueda y encontrar soluciones factibles. Debido a que estas soluciones no cumplen con la restricción de presupuesto, se penalizan haciendo su cobertura 0.
 - Si el costo de la solución candidata esta dentro del presupuesto y su cobertura es mejor a la de la mejor solución actual, entonces la solución candidata se vuelve mejor solución (caso factible): En este caso las soluciones aceptadas solo son de tipo factible, como la restricción del presupuesto esta satisfecha, lo que se busca es aumentar la cantidad de eventos OHCA cubiertos por las soluciones factibles.

7. Experimentos

Los experimentos realizados en el presente trabajo fueron realizados en un computador con las siguientes características:

- Sistema operativo windows 10.
- Ubuntu bash instalado.
- Modelo LENOVO ideapad 330S.
- Procesador inter(R) Core(TM) i7-8550U CPU. 1.80 GHz, 2001 Mhz, 4 procesadores principales, 8 procesadores lógicos.
- RAM 8 GB

La metodología utilizada para realizar los experimentos consistía en resolver las diferentes instancias proporcionadas del problema (de tipo fijo y flexible), haciendo variar el número de restart que realizaría el algoritmo y la semilla utilizada para generar las soluciones aleatorias. Los números de restart utilizados fueron: 1, 2 y 3, las semillas utilizadas fueron: 120, 300, 500 y 1230.

Para el caso de enfoque fijo, se resolvieron instancias del problema con las siguientes características:

- Instancia con 324 eventos OHCA, presupuesto 3 y radio de cobertura de 800.
- Instancia con 402 eventos OHCA, presupuesto 3 y radio de cobertura de 800.
- Instancia con 500 eventos OHCA, presupuesto 3 y radio de cobertura de 800.
- Instancia con 500 eventos OHCA, presupuesto 7 y radio de cobertura de 800.
- Instancia con 708 eventos OHCA, presupuesto 5 y radio de cobertura de 800.
- Instancia con 708 eventos OHCA, presupuesto 7 y radio de cobertura de 800.
- Instancia con 708 eventos OHCA, presupuesto 11 y radio de cobertura de 800.

Para todas las instancias anteriores se hicieron variar el número de restart y las semillas (parámetros ingresados para que el algoritmo resuelva el problema), la forma en la que se hizo fue primero dejar fijo el número de restart haciendo variar las semillas y luego se cambia el número de restart y se vuelve a variar la semilla.

Para el caso de enfoque flexible, se resolvieron instancias del problema con las siguientes características:

- Instancia con 324 eventos OHCA, presupuesto 2.2 y radio de cobertura de 800.
- Instancia con 402 eventos OHCA, presupuesto 2.2 y radio de cobertura de 800.
- Instancia con 500 eventos OHCA, presupuesto 2 y radio de cobertura de 800.
- Instancia con 500 eventos OHCA, presupuesto 4 y radio de cobertura de 800.
- Instancia con 708 eventos OHCA, presupuesto 4.2 y radio de cobertura de 800.
- Instancia con 708 eventos OHCA, presupuesto 3.4 y radio de cobertura de 800.
- Instancia con 708 eventos OHCA, presupuesto 7 y radio de cobertura de 800.

Para variar el número de restart y las semillas, se realizó el mismo procedimiento mencionado anteriormente.

Los parámetros entregados al algoritmo son:

- nombre del archivo: texto que representa el nombre del archivo que contiene las instancias de los problemas a resolver.
- número de restart: entero que representa la cantidad de restart que realizará el algoritmo para realizar exploración.
- semilla: entero que representa la semilla que se utilizará para generar las soluciones iniciales

Todos los parámetros mencionados anteriormente son recibidos por el algoritmo a través de la línea de comandos. El proceso para modificar los parámetros es:

1. Se mantiene fijo el número de restart y se hace variar la semilla. Teniendo en cuenta que los restart pueden tomar el valor 1, 2 y 3, y que las semillas toman los valores 120, 300, 500 y 1230; se deja fijo primero el número de restart (inicialmente se parte con número de restart 1) y se hace variar la semilla para que tome los valores 120, 300, 500 y 1230.
2. Luego de terminar de iterar sobre todos los valores de las semillas, se cambia el valor del número de restart, luego de eso el número de restart queda fijo y se vuelve a repetir el paso 1 hasta que se hayan usado todos los valores de las semillas y todos los números de restart.

Las instancias entregadas al algoritmo tienen las siguientes características:

- La primera línea del archivo contiene la información relacionada al número de eventos OHCA que deben ser cubiertos, el presupuesto y el radio de cobertura de los AEDs. El número de eventos OHCA y el radio de cobertura son valores enteros y el presupuesto es un valor flotante.
- Luego de ello le siguen n líneas, donde n es igual al número de eventos OHCA que deben ser cubiertos, esas n líneas tienen el siguiente formato: coordenada X del evento OHCA, coordenada Y del evento OHCA y un valor que indica si existe un AED puesto en esas coordenadas. Donde coordenada X y coordenada Y son valores enteros y la variable que indica la existencia de un AED puesto en las coordenadas de un evento OHCA particular es un valor entero que puede ser un 1 o 0, donde 1 indica que existe un AED y 0 lo contrario.

La complejidad de las instancias mencionadas anteriormente aumenta a medida que el número de eventos OHCA a ser cubiertos y la cantidad de presupuesto disponible aumenta, además para el caso flexible y fijo, las instancias con menor y mayor dificultad son:

- Para el caso fijo: la instancia que presenta menor complejidad es la que posee 324 eventos OHCA, presupuesto 3 y radio de cobertura de 800, la que presenta mayor complejidad es la que posee 708 eventos OHCA, presupuesto 11 y radio de cobertura de 800.
- Para el caso flexible: la instancia que presenta menor complejidad es la que posee 324 eventos OHCA, presupuesto 2.2 y radio de cobertura de 800, la que presenta mayor complejidad es la que posee 708 eventos OHCA, presupuesto 7 y radio de cobertura de 800.

8. Resultados

A continuación se mostrarán los resultados de los experimentos realizados, se utilizará el formato número_eventos_OHCA-presupuesto para identificar las instancias y el formato números_AEDs_agregados/números_AEDs_movidos para indicar la cantidad de AEDs agregados y/o reposicionados, además como todas las instancias tienen el mismo radio (800) dicha información no será mostrada en los resultados de los experimentos.

Resultados experimentos enfoque fijo

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	3/0	0	12.14
402-3	390	97.01	3/0	0	22.53
500-3	336	67.20	3/0	0	37.85
500-7	425	85.00	7/0	0	123.01
708-5	706	99.72	5/0	0	135.33
708-7	706	99.72	7/0	0	132.03
708-11	707	99.86	8/0	3	267.65

Cuadro 1: Número de restart: 1 y Semilla: 120

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	3/0	0	17.90
402-3	380	94.53	3/0	0	16.17
500-3	461	92.20	3/0	0	26.58
500-7	499	99.80	7/0	0	26.43
708-5	643	90.82	5/0	0	76.02
708-7	684	96.61	7/0	0	84.27
708-11	707	99.86	9/0	2	161.08

Cuadro 2: Número de restart: 1 y Semilla: 300

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	318	98.15	3/0	0	22.97
402-3	396	98.51	3/0	0	43.61
500-3	417	83.40	3/0	0	77.34
500-7	485	97.00	7/0	0	358.18
708-5	682	96.33	5/0	0	260.08
708-7	702	99.15	7/0	0	250.62
708-11	707	99.86	10/0	1	721.19

Cuadro 3: Número de restart: 1 y Semilla: 500

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	318	98.15	3/0	0	18.80
402-3	359	89.30	3/0	0	33.20
500-3	361	72.20	3/0	0	53.80
500-7	499	99.80	5/0	2	55.33
708-5	381	53.81	5/0	0	134.63
708-7	652	92.09	7/0	0	123.89
708-11	707	99.86	10/0	1	524.53

Cuadro 4: Número de restart: 1 y Semilla: 1230

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	3/0	0	44.66
402-3	390	97.01	3/0	0	80.09
500-3	336	67.20	3/0	0	134.14
500-7	499	99.80	6/0	1	279.22
708-5	706	99.72	5/0	0	421.29
708-7	706	99.72	7/0	0	433.91
708-11	707	99.86	8/0	3	1326.60

Cuadro 5: Número de restart: 2 y Semilla: 120

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	3/0	0	82.98
402-3	380	94.53	3/0	0	103.56
500-3	461	92.20	3/0	0	165.26
500-7	499	99.80	7/0	0	643.87
708-5	678	95.76	5/0	0	336.84
708-7	707	99.86	6/0	1	640.71
708-11	707	99.86	6/0	5	665.06

Cuadro 6: Número de restart: 2 y Semilla: 300

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	2/0	1	82.97
402-3	401	99.75	3/0	0	169.61
500-3	488	97.60	3/0	0	151.31
500-7	499	99.80	6/0	1	395.65
708-5	682	96.33	5/0	0	2108.54
708-7	707	99.86	7/0	0	450.15
708-11	707	99.86	9/0	2	1117.89

Cuadro 7: Número de restart: 2 y Semilla: 500

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	3/0	0	27.75
402-3	399	99.25	3/0	0	48.93
500-3	498	99.60	3/0	0	84.12
500-7	499	99.80	5/0	2	152.66
708-5	704	99.44	5/0	0	362.80
708-7	707	99.86	7/0	0	587.50
708-11	707	99.86	8/0	3	996.80

Cuadro 8: Número de restart: 2 y Semilla: 1230

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	3/0	0	42.78
402-3	390	97.01	3/0	0	75.50
500-3	336	67.20	3/0	0	134.46
500-7	499	99.80	6/0	1	393.17
708-5	706	99.72	5/0	0	581.97
708-7	706	99.72	7/0	0	392.41
708-11	707	99.86	8/0	3	981.46

Cuadro 9: Número de restart: 3 y Semilla: 120

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	3/0	0	124.94
402-3	400	99.50	3/0	0	222.30
500-3	461	92.20	3/0	0	323.08
500-7	499	99.80	5/0	2	982.96
708-5	678	95.76	5/0	0	789.27
708-7	707	99.86	6/0	1	1534.27
708-11	707	99.86	6/0	5	1390.30

Cuadro 10: Número de restart: 3 y Semilla: 300

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	2/0	1	55.07
402-3	401	99.75	3/0	0	169.27
500-3	488	97.60	3/0	0	210.88
500-7	499	99.80	6/0	1	215.01
708-5	682	96.33	5/0	0	322.78
708-7	707	99.86	7/0	0	250.89
708-11	707	99.86	6/0	5	454.91

Cuadro 11: Número de restart: 3 y Semilla: 500

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	2/0	1	35.72
402-3	401	99.75	3/0	0	48.23
500-3	499	99.80	3/0	0	80.51
500-7	499	99.80	5/0	2	217.07
708-5	704	99.44	5/0	0	318.07
708-7	707	99.86	7/0	0	458.93
708-11	707	99.86	8/0	3	937.01

Cuadro 12: Número de restart: 3 y Semilla: 1230

Resultados experimentos enfoque flexible

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	0/1	2	14.22
402-3	401	99.75	2/1	0	82.48
500-3	420	84.00	1/1	0.80	53.50
500-7	450	90.00	3/4	0.20	112.60
708-5	707	99.86	3/1	1	311.61
708-7	706	99.72	2/4	0.60	229.83
708-11	707	99.86	1/5	5	152.83

Cuadro 13: Número de restart: 1 y Semilla: 120

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	1/1	1.00	18.76
402-3	401	99.75	2/1	0.00	96.29
500-3	420	84.00	1/1	0.80	156.63
500-7	479	95.80	3/4	0.20	291.07
708-5	707	99.86	3/1	1.00	855.69
708-7	706	99.72	2/4	0.60	628.37
708-11	707	99.86	3/5	3.00	825.50

Cuadro 14: Número de restart: 1 y Semilla: 300

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	0/1	2.00	19.42
402-3	401	99.75	1/1	1.00	67.22
500-3	420	84.00	1/1	0.80	111.34
500-7	499	99.80	0/4	3.20	54.80
708-5	707	99.86	3/1	1.00	700.23
708-7	706	99.72	2/4	0.60	476.70
708-11	707	99.86	1/5	5.00	348.29

Cuadro 15: Número de restart: 1 y Semilla: 500

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	1/1	2.00	31.44
402-3	401	99.75	2/1	0.00	89.39
500-3	420	84.00	1/1	0.80	89.31
500-7	499	99.80	0/4	3.20	42.97
708-5	707	99.86	3/1	1.00	486.92
708-7	706	99.72	2/4	0.60	371.60
708-11	707	99.86	3/5	3.00	493.64

Cuadro 16: Número de restart: 1 y Semilla: 1230

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	0/1	2.00	75.94
402-3	401	99.75	2/1	0.00	273.61
500-3	420	84.00	1/1	0.80	348.05
500-7	499	99.80	1/4	2.20	470.34
708-5	707	99.86	3/1	1.00	1760.39
708-7	707	99.86	0/4	2.60	932.89
708-11	707	99.86	1/5	5.00	1411.56

Cuadro 17: Número de restart: 2 y Semilla: 120

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	1/1	1.00	64.01
402-3	401	99.75	2/1	0.00	170.00
500-3	455	91.00	1/1	0.80	186.97
500-7	479	95.80	3/4	0.20	474.03
708-5	707	99.86	3/1	1.00	1299.93
708-7	706	99.72	2/4	0.60	964.86
708-11	707	99.86	3/5	3.00	1605.84

Cuadro 18: Número de restart: 2 y Semilla: 300

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	0/1	2.00	80.48
402-3	401	99.75	1/1	1.00	265.05
500-3	420	84.00	1/1	0.80	316.84
500-7	499	99.80	0/4	3.20	155.60
708-5	707	99.86	3/1	1.00	1694.44
708-7	706	99.72	2/4	0.60	4256.12
708-11	707	99.86	1/5	5.00	853.25

Cuadro 19: Número de restart: 2 y Semilla: 500

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	1/1	1.00	55.98
402-3	401	99.75	2/1	0.00	150.72
500-3	420	84.00	1/1	0.80	187.47
500-7	499	99.80	0/4	3.20	179.57
708-5	707	99.86	3/1	1.00	973.15
708-7	707	99.86	2/4	0.60	753.49
708-11	707	99.86	2/5	4.00	1643.76

Cuadro 20: Número de restart: 2 y Semilla: 1230

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	0/1	2.00	88.74
402-3	401	99.75	2/1	0.00	270.99
500-3	420	84.00	1/1	0.80	277.97
500-7	499	99.80	1/4	2.20	305.13
708-5	707	99.86	3/1	1.00	985.30
708-7	707	99.86	0/4	2.60	585.59
708-11	707	99.86	1/5	5.00	2019.39

Cuadro 21: Número de restart: 3 y Semilla: 120

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	1/1	1.00	141.58
402-3	401	99.75	2/1	0.00	408.10
500-3	455	91.00	1/1	0.80	489.41
500-7	499	99.80	0/4	3.20	891.09
708-5	707	99.86	3/1	1.00	2681.63
708-7	707	99.86	0/4	2.60	1275.37
708-11	707	99.86	3/5	3.00	1150.17

Cuadro 22: Número de restart: 3 y Semilla: 300

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	0/1	2.00	44.58
402-3	401	99.75	1/1	1.00	129.10
500-3	420	84.00	1/1	0.80	161.87
500-7	499	99.80	0/4	3.20	162.65
708-5	707	99.86	3/1	1.00	927.64
708-7	706	99.72	2/4	0.60	688.85
708-11	707	99.86	1/5	5.00	775.83

Cuadro 23: Número de restart: 3 y Semilla: 500

Inst.	#eventos cubiertos	% cobertura	#AEDs agregado/movido	Ppto. sobrante	Tpo. ejecución [s]
324-3	323	99.69	1/1	1.00	53.78
402-3	401	99.75	2/1	0.00	152.01
500-3	420	84.00	1/1	0.80	166.66
500-7	499	99.80	0/4	3.20	195.60
708-5	707	99.86	3/1	1.00	11813.08
708-7	707	99.86	0/4	2.60	739.04
708-11	707	99.86	1/5	5.00	4399.35

Cuadro 24: Número de restart: 3 y Semilla: 1230

Comparando enfoque flexible vs enfoque fijo (trabajo actual vs estado del arte)

Al comparar los resultados generados por el algoritmo al utilizar el enfoque fijo (datos que pueden ser comparables al estado del arte) vs los resultados obtenidos por el algoritmo usando el enfoque flexible, se puede observar que en la mayoría de los casos el porcentaje de cobertura del enfoque flexible es igual o mayor al del enfoque fijo esto queda reflejado con mayor fuerza en las instancias 500-3 con número de restart 1 y semilla 120 donde el enfoque fijo obtuvo 67.20 % de cobertura y el enfoque flexible un 84.00 %, 500-3 con número de restart 1 y semilla 1230 obteniendo 72.20 y 84.00 % de cobertura, 708-5 con número de restart 1 y semilla 1230 obteniendo 53.81 y 99.86 %, 500-3 con número de restart 2 y semilla 120 obteniendo 67.20 y 84.00 % y 500-3 con número de restart 3 y semilla 120 obteniendo 67.20 y 84.00 %. A pesar de existir grandes mejoras en la cobertura del enfoque flexible hay casos en las que el enfoque fijo cubre mejor los eventos OHCA tales como: instancia 500-3 con número de restart 1 y semilla 300, donde el enfoque fijo obtuvo una cobertura de 92.20 % y el enfoque flexible cubrió 84.00 %, 500-3 con número de restart 2 y semilla 500 obteniendo 97.60 y 84.00 % de cobertura, 500-3 con número de restart 2 y semilla 1230 obteniendo 99.60 y 84.00 %, 500-3 con número de restart 3 y semilla 500 obteniendo 97.60 y 84.00 % y 500-3 con número de restart 3 y semilla 1230 obteniendo 99.80 y 84.00 %. En general, el presente algoritmo tarda más tiempo en resolver las instancias del problema donde se utiliza el enfoque flexible que en aquellas donde se ocupa el enfoque fijo, y en algunos casos se obtiene la misma cobertura a cambio de diferencias de tiempos considerables, esto queda reflejado en casos como: instancia 708-11 con número de restart 1 y semilla 300 obteniendo un porcentaje de cobertura de 99.86 % y tiempos de ejecución 161.08 y 825.50[s] respectivamente y 708-11 con número de restart 3 y semilla 1230 obteniendo un porcentaje de cobertura de 99.86 % y tiempos de ejecución 937.01 y 4399.35[s] respectivamente. También cabe mencionar que existen casos en los que es necesario realizar más de un restart en el caso de enfoque fijo para obtener un porcentaje de cobertura cercano al del enfoque flexible, como es el caso de la instancia 708-5 con número de semilla 1230, donde el enfoque fijo utilizó un número de restart 2 y el enfoque flexible utilizó un número de restart 1 obteniendo un porcentaje de cobertura de 99.44 y 99.86 % respectivamente.

Conclusiones

Luego de analizar los resultados obtenidos por el enfoque fijo y flexible, se puede concluir:

- El enfoque flexible tarda mucho más tiempo en entregar una solución de calidad en comparación con el enfoque fijo, pero el porcentaje de cobertura del primero es mucho mejor que la del segundo.
- Existen casos en el que el enfoque fijo va a poder encontrar una mejor solución que el enfoque flexible.
- En general el enfoque flexible va a poder encontrar una solución mejor que la del enfoque fijo realizando solo un restart, en cambio el enfoque fijo deberá realizar al menos dos restart para tener soluciones tan buenas como las del enfoque flexible.

Comparando resultados del enfoque fijo y flexible con diferentes parámetros

Al comparar los diferentes resultados de los experimentos con enfoque fijo y flexible, es posible ver que la elección de determinadas semillas (utilizadas para la construcción de las soluciones iniciales) y el número de restart tienen gran relevancia en el desempeño del algoritmo y la calidad de las soluciones encontradas por el, esto se debe a que determinadas semillas crean soluciones iniciales muy buenas que permiten que el algoritmo encuentre rápidamente una solución de calidad en cambio la elección de "malas" semillas hacen que el algoritmo realice gran cantidad de iteraciones para encontrar una solución de tanta calidad como las encontradas usando "buenas" semillas, esto queda en evidencia en casos tales como resolver la instancia 708-11 de enfoque flexible utilizando número de restart 1 y semillas 120 y 500, para este caso se puede ver que la elección de semillas (que crearán las soluciones iniciales) tiene gran relevancia, ya que eligiendo la semilla 120 el algoritmo tarda 152.83 [s] en encontrar una solución que cubre el 99.86 % de los eventos OHCA con costo 5 en cambio al elegir la semilla 500 el algoritmo tarda 348.29[s] (más del doble de tiempo que con la semilla 120) en encontrar una solución con el mismo porcentaje de cobertura y costo, también cabe destacar que la elección de la semilla no solo afecta el tiempo de ejecución y porcentaje de cobertura, ya que también afecta el costo de la solución, esto queda en evidencia en los casos 708-11 con número de restart 1 y semillas 300 y 1230, ya que el algoritmo encuentra soluciones de la misma calidad con respecto al porcentaje de cobertura pero la calidad del costo de ambos es peor en comparación con el resultado obtenido por la semilla 120 y 500 (el presupuesto sobrante con las semillas 120 y 500 es de 5.00 en cambio con 300 y 1230 es 3.00). Al variar el número de restart se obtiene un aumento en el tiempo de computación del algoritmo debido a que debe realizar una mayor cantidad de operaciones, pero en general los resultados obtenidos luego de realizar los experimentos muestran que aumentar la cantidad de restart no afecta demasiado la calidad del porcentaje de cobertura, más bien mejora el presupuesto sobrante, esto queda en evidencia en la resolución de la instancia 500-7 de enfoque fijo con número de restart 1, 2 y 3 y semilla 500, se puede apreciar que usando número de restart 1 se obtiene un presupuesto sobrante de 0.00, con restart 2 el presupuesto sobrante es de 1.00 y con número de restart 3 el presupuesto sobrante es de 1.00, por lo que se puede esperar que al aumentar el número de restart del algoritmo se encuentren soluciones con porcentaje de cobertura con variaciones no muy grandes pero que cuesten menos, otro punto importante es que la cantidad de restart puede afectar en gran medida el tiempo de ejecución del algoritmo sin afectar la calidad de la cobertura de la solución encontrada, esto queda plasmado en la instancia 708-5 de enfoque flexible con número de restart 1, 2 y 3 y semilla 1230, donde al usar un restart el tiempo de ejecución fue de 486.92[s], con 2 restart 973.15[s] y con 3 restart 11813.08[s].

Conclusiones

Luego de comparar algunos de los resultados de las instancias procesadas por el algoritmo utilizado en este trabajo se puede concluir lo siguiente:

- La elección de las semillas utilizadas en este algoritmo en particular es relevante, debido a que afecta directamente a la generación de soluciones candidatas iniciales, lo que provoca que el algoritmo encuentre soluciones de calidad en tiempos grandes o "pequeños".
- La cantidad de restart utilizadas en este algoritmo afecta la calidad de la cobertura de la solución, el presupuesto sobrante y el tiempo de ejecución. En general, el impacto mayor radica en el tiempo de ejecución del algoritmo y el presupuesto sobrante, y en determinados casos el porcentaje de cobertura, esto se debe a que en general al utilizar solo un restart el algoritmo es capaz de encontrar una solución con porcentaje de cobertura mayor

al 99 % provocando que los demás restart puedan encontrar soluciones más baratas, pero para lograr lo anterior es necesario invertir una mayor cantidad de tiempo en ejecución. En aquellos casos en el que el porcentaje de cobertura con solo un restart es inferior al 90 % al aumentarlo mejorará dicho porcentaje.

- En general el presente algoritmo puede encontrar soluciones con un porcentaje de cobertura de eventos OHCA sobre el 99 % usando un solo restart y usando más de un restart va a tender a encontrar soluciones igual de buenas (con respecto a la cobertura) pero más baratas. En casos en donde al usar un restart el porcentaje de cobertura de la solución sea bajo, al aumentar el número de restart se tenderá a encontrar soluciones con mayor cobertura.

9. Conclusiones

Las técnicas utilizadas por Chan et al. (2013)., Sun et al. (2016)., Chan et al. (2016). y Chan et al. (2017). lograron resolver instancias del problema utilizando datos de la ciudad de Toronto, Tierney et al. (2018). lograron resolver una instancia del problema utilizando datos de la localidad de Ticino y finalmente Nazarian (2018). logró resolver una instancia del problema utilizando datos de North Holland y Twente regiones de los países bajos.

Las diferencias entre ellas radican en la forma en que abordan el problema y las características que estos presentan, por ejemplo: Sun et al. (2016). se dan cuenta que algunos de los AEDs no están disponibles en determinados momentos del día por lo que desarrollan un modelo matemático que considera el acceso temporal además del espacial, Chan et al. (2016). abordaron el problema como uno de tipo MCLP y generaron modelos para tres casos considerados realistas, Chan et al. (2017). consideraron en su modelo la incertidumbre en la distribución espacial de los paros cardíacos.

En general la mayoría de las técnicas estudiadas presentan la limitación que al momento de determinar las ubicaciones de los AEDs no tienen en cuenta el acceso temporal a él esto provoca que algunos de los OHCAs no puedan ser cubiertos, debido a que algunos de los edificios donde se decide instalar los AEDs se encontrarán cerrados en algún momento del día. Además en el caso de Tierney et al. (2018). utilizaron una determinada definición de área urbana y rural (se definió como municipio urbano aquel que tiene una población superior a 7000 individuos y una densidad de población superior a 200 por kilómetro cuadrado) para aplicarlas en su modelo, por lo que esta puede variar dependiendo del sitio geográfico donde se aplique, también cabe destacar que como mencionan Tierney et al. (2018). el modelo encuentra la ubicación optima de los AEDs pero esto no se traduce en un uso optimo de ellos, esto puede ser que aplique en otros estudios, en el caso de Nazarian (2018). su estudio utilizó solamente un subconjunto de AEDs registrados del área de estudio pese a que también existen AEDs no registrados (algo similar sucede en otros estudios), aunque como mencina es poco probable que una persona encuentre un AED no registrado, también menciona que el modelo puede encontrar ubicaciones para los AEDs que no sean factibles.

Entre las estrategias o técnicas más prometedoras se encuentran:

- Aquellas que consideran el acceso temporal al AED, ya que es razonable considerar que no todos los edificios están abiertos en el mismo rango de tiempo por lo tanto determinados AEDs estarán o no disponibles en esos rangos.
- Las que agregan y relocalizan los AEDs existentes, esto resulta interesante debido a que puede ser que la ubicación de los AEDs previamente instalados no optimicen la cobertura de los OHCAs por algún cambio en su comportamiento por lo que una combinación de redistribución con agregación de AEDs puede resultar en un aumento considerable de la cobertura de OHCAs.

Debido a que el aspecto temporal tiene gran importancia en las ubicaciones de los AEDs y que el enfoque flexible resulta útil para aumentar la cobertura de OHCAs sin necesidad de comprar una gran cantidad de AEDs, para el futuro es de esperar que se empleen métodos que resuelvan el Defibrillator Relocation Problem considerando aspectos espacio - temporal y ubicación flexible para obtener la ubicación de los AEDs que maximice la cobertura de los OHCAs de una determinada área geográfica.

Con respecto al análisis de los resultados obtenidos se puede apreciar que en general los problemas con enfoque flexible toman una cantidad de tiempo mayor en ser resueltos por el

algoritmo pero el porcentaje de cobertura obtenidos por ellos en la mayoría de los casos es igual o mayor al de los problemas con enfoque fijo llegando en la mayoría de los casos a cubrir un porcentaje mayor al 99 % de los eventos OHCA, esto demuestra que en general es más complejo redistribuir y agregar AEDs que solo agregarlos.

En general desarrollar un algoritmo basado en Hill Climbing con mejor mejora que puede moverse en regiones factibles e infactibles puede ser una buena opción para enfrentarse al problema, ya que el porcentaje más bajo de cobertura que se obtuvo resolviendo problemas con enfoque fijo fue de 53.81 % (instancia 708-5 con número de restart 1 y semilla 1230) y para los problemas con enfoque flexible fue de 84.00 % (instancias 500-3 para todas las combinaciones de número de restart y semillas, exceptuando número de restart 2 y 3 con semilla 300), además logró obtener soluciones con un porcentaje de cobertura mayor a 99 % en varias ocasiones para los problemas basados en enfoque flexible. También es de destacar que la elección de semillas y número de restart afectan en gran medida al desempeño del algoritmo utilizado, ya que elegir "malos" valores para las semillas provocan la creación de soluciones iniciales que necesitan de bastante computo para transformarlas en soluciones de calidad, además en el caso del número de restart quedó en evidencia que en el caso del enfoque flexible aumentarlo no mejora mucho el porcentaje de cobertura de las soluciones, más que nada ayuda a aumentar el presupuesto restante, por lo que al elegir una buena semilla bastará con ejecutar el algoritmo con un solo restart para encontrar soluciones con gran porcentaje de cobertura.

En este estudio, no se detectaron fallas una vez terminado el algoritmo y testeado, algunas mejoras que se le podrían hacer al presente algoritmo pueden ser: cambiar la representación de la solución (puede ser que exista otra representación que maneje mejor las restricciones expuestas por el problema, reduciendo el tamaño de almacenamiento de la solución), cambiar el movimiento utilizado (en lugar de hacer un flip bit, se podría realizar algún otro tipo de movimiento o quizás aplicar el mismo movimiento a otra representación cuyo tamaño de almacenamiento de la solución sea menor) y también se podría aplicar paralelismo al momento de tener un número de restart mayor a uno (se podrían tener varios hilos mejorando diferentes soluciones iniciales de forma paralela con el fin de reducir el tiempo de espera percibido por el usuario), entre el trabajo futuro que se podría realizar esta la posibilidad de agregar al algoritmo utilizado en este trabajo la restricción de la componente temporal de la disponibilidad de los AEDs, ya que el presente algoritmo solo trabaja con componentes espaciales y la componente temporal al igual que la temporal son de gran importancia, ya que nos indica en que instantes un AED deja de estar disponible por lo que esto puede afectar las ubicaciones en donde serán finalmente ubicados.

10. Bibliografía

Referencias

- [1] Perkins G., Handley A., Koster R., Castrén M., Smyth M., Olasveengen T., Monsieurs K., Raffay V., Gräsner J., Wenzel V., Ristagno G. y Soar J. «European Resuscitation Council Guidelines for Resuscitation 2015: Section 2. Adult basic life support and automated external defibrillation». En: *Resuscitation* 95 (2015). Recuperado el 13 de Octubre del 2020, págs. 81-99. URL: <https://www.sciencedirect.com/science/article/pii/S0300957215003275>.
- [2] Schmid V. y Doerner K. «Ambulance location and relocation problems with time-dependent travel times». En: *European Journal of Operational Research* 207.3 (2010). Recuperado el 11 de Octubre del 2020, págs. 1293-1303. URL: <https://www.sciencedirect.com/science/article/pii/S0377221710004728>.
- [3] Chan T., Li H., Lebovic G., Tang S., Chan J., Cheng H., Morrison L. y Brooks S. «Identifying Locations for Public Access Defibrillators Using Mathematical Optimization». En: *Circulation* 127.17 (2013). Recuperado el 05 de Octubre del 2020, págs. 1801-1809. URL: <https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.113.001953>.
- [4] Sun C., Demirtas D., Brooks S., Morrison L. y Chan T. «overcoming Spatial and Temporal Barriers to Public Access Defibrillators Via Optimization». En: *JACC JOURNAL OF THE AMERICAN COLLEGE OF CARDIOLOGY* 68.8 (2016). Recuperado el 07 de Octubre del 2020, págs. 836-845. URL: <https://www.onlinejacc.org/content/68/8/836>.
- [5] Chan T. y Demirtas D. and Kwon R. «Optimizing the Deployment of Public Access Defibrillators». En: *MANAGEMENT SCIENCE* 62.12 (2016). Recuperado el 07 de Octubre del 2020, págs. 3393-3672. URL: <https://pubsonline.informs.org/doi/10.1287/mnsc.2015.2312>.
- [6] Chan T., Shen Z. y Siddiq A. «Robust Defibrillator Deployment Under Cardiac Arrest Location Uncertainty via Row-and-Column Generation». En: *OPERATIONS RESEACH* 66.2 (2017). Recuperado el 07 de Octubre del 2020, págs. 301-596. URL: <https://pubsonline.informs.org/doi/abs/10.1287/opre.2017.1660?journalCode=opre>.
- [7] Tierney N., Reinhold H., Mira A., Weiser M., Burkart R., Benvenuti C. y Auricchio A. «Novel relocation methods for automatic external defibrillator improve out-of-hospital cardiac arrest coverage under limited resources». En: *Resuscitation* 125 (2018). Recuperado el 05 de Octubre del 2020, págs. 83-89. URL: <https://www.sciencedirect.com/science/article/pii/S0300957218300650>.
- [8] Nazarian A. «Optimizing the Deployment of Automated External Defibrillators by a Data-Driven Algorithmic Approach». Recuperado el 10 de Octubre del 2020. Master dissertation. Enschede, the Netherlands: UNIVERSITY OF TWENTE, jun. de 2018. URL: http://essay.utwente.nl/74410/1/Nazarian_MA_BMS.pdf.
- [9] Gallardo S. *DRP - Inteligencia Artificial 2020-2*. Sebastián Gallardo. Recuperado el 09 de Octubre del 2020. Sep. de 2020. URL: <https://www.youtube.com/watch?v=D9E8giCevR0&feature=youtu.be>.