

Inteligencia Artificial

Informe Final: Problema de localización de desfibriladores externos automáticos

Gabriel Carmona

11 de enero de 2021

Evaluación

Mejoras 1ra Entrega (10 %):	_____
Código Fuente (10 %):	_____
Representación (15 %):	_____
Descripción del algoritmo (20 %):	_____
Experimentos (10 %):	_____
Resultados (10 %):	_____
Conclusiones (20 %):	_____
Bibliografía (5 %):	_____
Nota Final (100):	_____

Resumen

El problema de localización de desfibriladores externos automáticos consiste en posicionar desfibriladores externos automáticos en ciertos puntos de la ciudad de forma de maximizar el recubrimiento de lugares donde hayan ocurrido paros cardíacos, este problema tiene ciertas variantes dependiendo de que se disponga para resolver al problema. Este problema al tener relación con la salud de las personas se ha vuelto importante, por lo que se ha estudiado a lo largo de la historia. Este documento plantea el origen del problema, distintas variantes por las cuales ha pasado y métodos que han utilizado para resolverlo, además de mostrar un modelo matemático con el fin de solucionarlo, junto a una representación y un algoritmo que lo utilice. Para finalmente, obtener resultados del algoritmo creado, concluir respecto al trabajo realizado e indicar que hacer a futuro.

1. Introducción

Hoy en día a nivel mundial dentro del ámbito de la salud, el poder aumentar la probabilidad de supervivencia de las personas es un foco de interés y un problema significativo. Los paros cardíacos fuera del hospital representan un problema grave: por ejemplo, en Estados Unidos en el año 2014 murieron aproximadamente 424 mil personas debido a este problema [7]. De hecho, cuando a una persona le da un paro cardíaco en Estados Unidos, según la estadística tiene un 9.6 % de probabilidades de sobrevivir o ser reanimado [10], y derechamente la mayoría de los casos de paros cardíacos fuera del hospital no logran sobrevivir debido principalmente a la falta de desfibriladores externos automáticos (DEA) cerca de la zona [8]. Entonces, tomando en cuenta esta problemática, se propone como idea poder permitir un fácil acceso a DEA en caso de cualquier emergencia fuera de un hospital, por lo cual se busca encontrar la mejor manera de

posicionar los DEA de tal forma que se cubra las zonas donde hay un mayor registro de paros cardíacos.

Para esto, en este trabajo se definirá el problema, indicando las variables y restricciones junto a las variantes más conocidas. Luego, se revisará lo que se ha realizado históricamente con el problema, incluyendo las técnicas y algoritmos que se han utilizado para resolverlo. Posterior a esto, se definirán dos modelos matemáticos de este problema, mostrando las particularidades de cada uno. Finalmente se ilustrará una representación y describirá un algoritmo que permita solucionar una variante del problema. Indicando los experimentos que se realizarán y luego mostrando los resultados obtenidos. Para finalmente, poder concluir respecto al trabajo realizado e indicar qué se podría realizar a futuro.

2. Definición del Problema

La problemática que aborda la presente investigación es la localización de dispositivos DEAs en puntos estratégico de la ciudad, fuera del hospital. Para esto, se obtiene la información de dónde ocurren paros cardíacos en una zona urbana y según esta información, y teniendo en cuenta que el radio de descubrimiento de los DEAs es de 100 metros, se busca posicionar estos de manera que se cubra la mayor cantidad de lugares donde han ocurrido paros cardíacos dentro de la zona urbana [3]. Los lugares puntuales donde se posicionan los DEAs pueden ser de dos tipos; en primer lugar, un sitio donde ya hay un DEA, y en segundo lugar, un sitio donde aún no existe un DEA, pero es posible candidato para poseer uno. Esta investigación se focalizará en el problema de localización de DEAs dentro de áreas urbanas públicas, pero se cree importante recalcar que este mismo problema se ha visto trabajado en interiores, debido a que en edificios, especialmente de carácter público, este tema es crítico [6].

El problema descrito es equivalente al problema de ubicación de cubierta máxima, en el cual se busca maximizar la cantidad de población que pueda utilizar un servicio dado [5]. De hecho, de este mismo se basa una de las dos variantes más conocidas del problema, la cual considera que los DEAs ya situados no se pueden relocalizar, por lo que se consideran puntos fijos, provocando que solo se podrían situar nuevos DEAs en puntos donde aún no se haya situado ninguno [2]. Aunque también existe una variante donde se considera que se pueden mover DEAs ya puestos a otros puntos, y al realizar esto agrega un factor de presupuesto máximo a gastar, debido a que el costo de mover o instalar un desfibrilador es distinto, por lo que sería un tema a considerar al momento de querer optimizar el problema [11]. En esta última variante, se considera que solo se pueden mover DEAs desde un punto ya fijado hacia un punto que no posee un DEA, y también se empieza a considerar el movimiento de un DEA ya situado a uno que no posea, como variable del problema. Esto debido a que es distinto un DEA relocalizado que un DEA instalado.

Aunque estas variantes permitirían resolver el problema, tienen un acercamiento poco real, ya que se considera que dentro del radio existe la misma probabilidad de que haya un paro cardíaco. Por esto, una variante en la cual se considere la probabilidad de suceso de un paro cardíaco concedería un acercamiento más cercano a la realidad. Esta probabilidad es descrita en una variante del problema donde no se pueden mover las DEAs ya situadas, generando una función que permite modelar el comportamiento de este suceso basado en la distancia entre el lugar y el DEA situado. Además, se consideran tres instancias específicas: una en la cual hay más de una persona para ayudar al sujeto, otra donde hay solo una persona, pero se encuentra en una mala posición para ayudar y una última donde se encuentra solo una persona situada en una buena posición [2]. Pero, aunque exista la variante, el hecho de ser probabilístico provoca que no haya una certeza de que ese DEA permita cubrir correctamente, pero de todas formas esto permite una mayor probabilidad de éxito que si no se considera.

3. Estado del Arte

El primer acercamiento de este problema se remonta al año 1978, con la publicación del trabajo de Church y Revelle [5]. Si bien en su publicación no se habla exactamente de situar DEAs, sí se trabaja bajo el mismo concepto que presenta la problemática. En esa instancia de habla del problema de ubicación de cubierta máxima (MCLP), en el cual se busca maximizar el cubrimiento de demanda de un servicio dado. Para ese modelo se utilizaron en un inicio algoritmos *greedy*, que permitían entregar una solución correcta y rápida, aunque también se ocupó programación lineal de tal forma que se asegure encontrar un valor óptimo. Este último enfoque tiene un problema ya que se consideran variables mayores o iguales a cero y no binarias como se da en el caso de los algoritmos *greedy*. Para solucionar esto se aplica el método *branch and bound* para la situación en la cual entregue valores fraccionarios.

A partir de este trabajo se han realizado distintas variaciones del problema, buscando mayor variedad de opciones o una solución más cercana a la realidad. Una de esas variantes se realizó utilizando ciertas variables probabilísticas, para así poder situar los DEAs de manera que la probabilidad de sobrevivir pueda ser más alta [2]. De esto surgieron tres escenarios: en primer lugar, uno donde hay múltiples personas para tomar el DEA y ayudar al sujeto, luego otros dos donde hay solo una persona, pero en un caso esta se encuentra lejos del DEA y en el otro se encuentra cerca del DEA. Estas tres situaciones, junto al modelo propuesto de MCLP, fueron testeadas utilizando un solver CPLEX presentando. Se concluyó que el modelo inicial MCLP es más rápido en ejecución que el resto de los modelos, pero al momento de analizar el cubrimiento de los datos de paros cardíacos se mostró que estos tres presentaron una gran mejora al momento de cubrir las zonas, en comparación con el modelo de MCLP [2].

Por otro parte, también se han presentado trabajos en los cuales se ha estudiado el modelo probabilístico, aplicándole algoritmos de la generación de filas y otro de generación de filas y columnas. Se observa que el algoritmo de generación de filas y columnas tiene un rendimiento que escala muy bueno a medida que la cantidad de escenarios aumenta, por lo que hace bastante eficiente en comparación al método utilizando una clásica programación lineal mixta y el algoritmo de generación de filas para poder resolver el problema [4].

Otro método que se ha utilizado para poder resolver este problema es la utilización de un conjunto de datos muy grande, que es dividido aleatoriamente en cinco partes. Se resuelve primeramente el problema para una parte de las divisiones, utilizando este conjunto a modo de entrenamiento. Luego se utiliza el resto de conjuntos para probar que tal estuvo la resolución encontrada con la data de entrenamiento. Este proceso es llamado la validación cruzada de k iteraciones, y logra demostrar que el modelo con posibilidad de mover los desfibriladores ya fijados permitiría un mejor cubrimiento de los paros cardíacos que el modelo sin la posibilidad de mover los DEAs [11].

Dentro de estas dos últimas técnicas mostradas, se incluye el uso de algoritmos genéticos, los cuales se benefician cuando en el problema no busca maximizar el cubrimiento, sino más bien minimizar el número de DEAs instalados. Para este caso se ha utilizado el algoritmo genético llamado NSGA-II, que le permite mejorar no solo tiempo sino que también resultado [1].

De esta forma, según lo explicado anteriormente se puede tener en cuenta que de los algoritmos con mejor resultado encontrados han sido el algoritmo de generación de columnas y filas, y la resolución del modelo de posición flexible resuelto con programación lineal mixta y el uso de NSGA-II.

4. Modelo Matemático

4.1. Modelo basado en MCLP

Se definirá el modelo en base al MCLP mencionado anteriormente. Para esto, inicialmente se tienen que definir los siguientes parámetros del problema, puesto que son datos que vienen

entregados para el desarrollo del problema.

$$\begin{aligned}
I_e &: \text{conjunto de lugares donde hay un DEA instalado} \\
I_c &: \text{conjunto de lugares candidatos para instalar un nuevo DEA} \\
I &: I_e \cup I_c \text{ conjunto de tamaño } m \\
J &: \text{conjunto de tamaño } n \text{ lugares donde ha ocurrido un paro cardíaco} \\
a_{ij} &= \begin{cases} 1 & \text{si es que el paro cardíaco en el lugar } j \text{ es cubierto por el radio del } i \\ 0 & \text{si no} \end{cases}
\end{aligned}$$

Ahora, se pueden definir las siguientes variables del problema que serán utilizadas para poder desarrollar la función objetivo más adelante, y las restricciones a las que se va a ver enfrentado el problema.

$$\begin{aligned}
x_j &= \begin{cases} 1 & \text{si es que el paro cardíaco en el lugar } j \text{ esta cubierto} \\ 0 & \text{si no} \end{cases} \\
y_i &= \begin{cases} 1 & \text{si un DEA esta instalado en el lugar } i \\ 0 & \text{si no} \end{cases} \\
z_{ij} &= \begin{cases} 1 & \text{si un DEA instalado en el lugar } i \text{ es usado para cubrir el paro cardíaco en el lugar } j \\ 0 & \text{si no} \end{cases}
\end{aligned}$$

$$\text{maximizar } f = \sum_{j \in J} x_j \quad (1)$$

La función objetivo lo que busca es maximizar la cantidad de paros cardiacos cubiertos según su posición j .

Ahora se pasarán a formular las siguientes restricciones para poder representar de mejor forma el problema.

$$\sum_{i \in I_c} y_i \leq N \quad (2)$$

$$y_i = 1, \forall i \in I_e \quad (3)$$

$$z_{ij} \leq a_{ij} \cdot y_i, \forall i \in I, j \in J \quad (4)$$

$$x_j = \sum_{i \in I} z_{ij}, \forall j \in J \quad (5)$$

$$x_j, y_i, z_{ij} \in \{0, 1\}, \forall i \in I, j \in J \quad (6)$$

La restricción (2) define que solo se puedan instalar una cantidad máxima de n DEAs, esto es debido a que como la cantidad de lugares donde ha ocurrido un paro cardíaco son n , entonces tener más de n DEAs sería contraproducente ya que podría suceder que al instalar una cantidad gigantesca de DEAs se resuelva el problema. La restricción (3) sirve para fijar todos los puntos que tienen instalado un DEA en el lugar i , por ende es imposible que valga 0. La restricción (4) permite asegurar que un DEA puede ser usado para cubrir el paro cardíaco en el lugar j solo si es que ese lugar se encuentra cubierto por el radio desde i . La restricción (5) permite fijar la variable x_j en 1 cuando el paro cardíaco en j esté cubierto por algún DEA en la posición i , y 0 en cualquier otro caso. Y la restricción (6) representa el dominio de las tres variables definidas, que corresponden a variables binarias [2] [5].

4.2. Modelo que permite mover DEAs instalados

En esta sección, se planteará un modelo basado igualmente en MCLP, pero con ciertas modificaciones de tal forma que permita tener las posibilidad de mover los DEA de posición a otro puesto. Para esto se considera un costo asociado a mover un DEA desde una posición a otra y un costo asociado a instalar un nuevo DEA con un presupuesto máximo. Por esto, a los parámetros pasados se la dos nuevos.

- I_e : conjunto de lugares donde hay un DEA instalado
- I_c : conjunto de lugares candidatos para instalar un nuevo DEA
- I : $I_e \cup I_c$ conjunto de tamaño m
- J : conjunto de tamaño n lugares donde ha ocurrido un paro cardíaco
- CE : costo unitario de mover un DEA de una posición en I_e a otra en I_c
- CN : costo unitario de instalar un DEA en una posición en I_c
- B : presupuesto total para la instalación o cambio de posición de los DEAs
- $a_{ij} = \begin{cases} 1 & \text{si es que el paro cardíaco en el lugar } j \text{ es cubierto por el radio del } i \\ 0 & \text{si no} \end{cases}$

También, al tener esta nueva instancia donde un DEA puede moverse de lugar, se definirá una variable extra con respecto al modelo anterior de forma de poder representar ese movimiento. Además, se modificará la variable y_i para así diferenciar correctamente entre un DEA trasladado y otro nuevo instalado.

$$x_j = \begin{cases} 1 & \text{si es que el paro cardíaco en el lugar } j \text{ esta cubierto} \\ 0 & \text{si no} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{si se instala un nuevo DEA en el lugar } i \text{ o bien ya hay un DEA instalado} \\ 0 & \text{si no} \end{cases}$$

$$z_{ij} = \begin{cases} 1 & \text{si un DEA instalado en el lugar } i \text{ es usado para cubrir el paro cardíaco en el lugar } j \\ 0 & \text{si no} \end{cases}$$

$$v_{ik} = \begin{cases} 1 & \text{si un DEA en la ubicación } i \text{ se mueve a la ubicación } k \\ 0 & \text{si no} \end{cases}$$

Aunque se agregue esta variante de poder mover dispositivos de un lado a otro, de todas formas la función objetivo será la misma que en el caso anterior, que corresponde a cubrir la mayor cantidad de lugares en donde haya ocurrido un paro cardíaco.

$$\text{maximizar } f = \sum_{j \in J} x_j \tag{7}$$

Ahora, debido a esta variante se agregaran ciertas restricciones para poder cumplir con lo explicitado con la variante actual.

$$z_{ij} \leq a_{ij} \cdot y_i + a_{ij} \cdot \sum_{k \in I} v_{ki}, \forall i \in I, j \in J \quad (8)$$

$$y_i = 1 - \sum_{k \in I_e} v_{ik}, \forall i \in I_e \quad (9)$$

$$y_i = 1 - \sum_{k \in I_e} v_{ki}, \forall i \in I_e \quad (10)$$

$$v_{ik} = 0, \forall i \in I, k \in I_e \quad (11)$$

$$x_j = \sum_{i \in I} z_{ij}, \forall j \in J \quad (12)$$

$$B \geq \sum_{i \in I_e} \sum_{k \in I_e} v_{ik} \cdot CE + y_i \cdot (1 - v_{ik}) \cdot CN \quad (13)$$

$$x_j, y_i, z_{ij} \in \{0, 1\}, \forall i \in I, j \in J \quad (14)$$

La restricción (8) permite asegurar que un DEA instalado en i o movido de k hacia i puede ser usado para cubrir el paro cardíaco en el lugar j solo si es que ese lugar se encuentra cubierto por el radio desde i . La restricción (9) permite que, si un DEA que estaba instalado de antes se movió a algún lugar donde no había DEA, entonces y_i va a valer 0. Y, en el caso de que no se haya movido entonces quiere decir que sigue en i , por lo que va a valer 1. La restricción (10) permite decir que en los lugares nuevos donde no había DEA se instale o venga de algún otro lado, pero no ambas a la vez. La restricción (11) asegura que solo se pueden mover DEAs desde un lugar en I_e hacia I_c . La restricción (12) posibilita que la variable x_j valga 1 si está cubierta por algún DEA. La restricción (13) es la que limita el gasto de mover o instalar DEAs al presupuesto máximo dado, donde si es que se movió un DEA desde i hasta k , entonces se considera el costo unitario de mover un DEA, y si es que se instalo un nuevo DEA se considera el costo unitario de instalarlo [11].

5. Representación

Para este caso, se va a resolver el problema representado por el modelo que permite mover DEAs instalados previamente. Además se considerarán como lugares candidatos para instalar nuevos DEAs, los mismos lugares donde han ocurridos paros cardíacos (OHCA).

Con el fin de resolver el problema se decidió representar las soluciones a través de un arreglo de números, donde cada posición es un DEA y el valor que se almacena corresponde a la posición en que se instalará ese DEA, este valor irá de 0 hasta la cantidad de posiciones posibles. Donde, 0 corresponde a la coordenada número 0, 1 corresponde a la coordenada número 1 y así. El tamaño de este arreglo será basado en cuantos DEA puede instalar a lo más con el presupuesto dado.

índice	0	1	2	3
valor	4	1	7	3

Tabla 1: Ejemplo de representación sin DEA instalado previamente con presupuesto 4, donde el coste de instalar un DEA es 1.

En el ejemplo de la tabla 1 se puede observar un arreglo de tamaño 4 puesto que el presupuesto es 4, también que el DEA 0 se instaló en la posición 4, el DEA 1 se instaló en la posición 1, el DEA 2 se instaló en la posición 7 y el DEA 3 se instaló en la posición 3. Ahora bien, en el caso de que ya hayan DEA instalados, el arreglo no necesitará modificar su estructura

para representarlos, no obstante, el total de DEA será igual a la suma de los DEA previamente instalados más la cantidad de DEA que se puedan instalar con el presupuesto. Como se observa en la tabla 2 donde los primeros dos DEA serían los ya instalados y el resto correspondería a los DEA que se podrían instalar si es que se gasta todo el presupuesto en instalar DEA

índice	0	1	2	3	4	5
valor	4	1	7	3	10	2

Tabla 2: Ejemplo de representación con 2 DEA instalados previamente con presupuesto 4, donde el coste de instalar un DEA es 1.

Esta representación es apropiada para resolver el problema puesto que permite mostrar las soluciones del problema indicando donde se instalarán los DEAs, además que representar de esta forma permite beneficiarse del hecho de que generalmente la cantidad de puntos donde se pueden instalar DEAs es mayor a la cantidad de DEAs que se pueden instalar debido a que el presupuesto es un recurso limitado. Entonces, si es que la representación fuera, un arreglo donde cada índice corresponde a una posición y el valor indicaría si es que voy a poner un DEA en el punto o no, dará un tamaño de espacio de búsqueda de 2^n , donde n es la cantidad de puntos donde se pueden instalar DEA. Pero con la representación ofrecida el tamaño de espacio de búsqueda se reduce a $n^{\text{instalados} + \text{presupuesto} / \text{coste_de_instalar}}$, donde si bien n es mayor a 2, la potencia a la cual se esta elevando es inferior debido a esa característica de recurso limitado dado anteriormente, siendo así una representación que beneficia a la resolución del problema. Para este problema se decide instanciar las variables con el orden siguiente: primero los DEAs ya instalados y después los nuevos DEAs nuevos que se pueden instalar. Esto es debido a que los DEAs ya instalados ya consideran como un valor inicial, entonces primero se intentará dejar los DEAs ya instalados donde estaban y luego se intentaran mover.

6. Descripción del algoritmo

El algoritmo que se implementó, utilizando la representación antes mencionada, utiliza las técnicas de Forward Checking y CBJ. Donde la primera técnica se encargará de reducir el dominio de las variables que faltan por instanciar cuando alguna fue instanciada y el segundo para poder realizar un retorno guiado a base de los conflictos de restricciones cuando este no pueda instanciar más.

En el algoritmo 1, la data entregada tiene la siguiente descripción:

- *dondeInstalarOptimo*: corresponde a las posiciones de los DEA donde está el óptimo
- *maxPuntosCubiertos*: la cantidad máxima de puntos cubiertos dado el posicionamiento óptimo
- *costeMaximo*: coste total de instalar o mover dado las posiciones del óptimo
- *posiblesPosiciones*: arreglo que contiene todas las posiciones posibles de cada DEA
- *presupuesto*: presupuesto que se tiene para la instancia dada
- *costeDeMoverDEA*: coste de mover un DEA ya instalado
- *costeDeInstalarDEA*: coste de instalar un DEA nuevo

Además, se puede observar estructuras reconocibles donde de la línea 2 a 5 se suma la nuevos puntos cubiertos y se aplica Forward Checking, implementado según el algoritmo 2, donde esta técnica saca del conjunto de posibles posiciones de un DEA toda posición ya utilizada. Luego

Algorithm 1: Algoritmo que permite solucionar DRP

Result: Encontrar las posiciones que permita cubrir óptimamente los puntos OHCA

Data: *dondeInstalarOptimo*, *maxPuntosCubiertos*, *costeMaximo*,
posiblesPosiciones, *presupuesto*, *costeDeMoverDEA*, *costeDeInstalarDEA*

```
1 Procedure DRP(DEAAInstanciar, DEAInstanciado, coste, puntosCubiertos,  
   dondeInstalar, conf)  
2   if DEAAInstanciar != 0 then  
3     puntosCubiertos ← puntos que son cubiertos al instanciar DEAInstanciado en  
       dondeInstalar[DEAInstanciado]  
4     posiblesPosiciones ← actualizar dominio con FC(DEAInstanciado,  
       posiblesPosiciones)  
5   end  
6   if presupuesto < coste then  
7     conf ← actualizar conflictos de DEAInstanciado debido a restricción fallida  
       CBJ(DEAInstanciado)  
8     return DEAInstanciado  
9   end  
10  if maxPuntosCubiertos < puntosCubiertos then  
11    dondeInstalarOptimo ← dondeInstalar  
12    maxPuntosCubiertos ← puntosCubiertos  
13    costeMaximo ← coste  
14  end  
15  if DEAAInstanciar == largo(dondeInstalar) then  
16    return -1  
17  end  
18  if DEAAInstanciar ya estaba instanciado then  
19    dondeInstalar[DEAAInstanciar] ← valor donde ya esta instalado  
20    DRP(DEAAInstanciar + 1, DEAAInstanciar, coste, puntosCubiertos,  
       dondeInstalar, conf)  
21  end  
22  foreach posicion ∈ posiblesPosiciones[DEAAInstanciar] do  
23    dondeInstalar[DEAAInstanciar] ← posicion  
24    if DEAAInstanciar ya estaba instalado then  
25      DEAConConflicto ← DRP(DEAAInstanciar + 1, DEAAInstanciar,  
        coste + costeDeMoverDEA, puntosCubiertos, dondeInstalar, conf)  
26    else  
27      DEAConConflicto ← DRP(DEAAInstanciar + 1, DEAAInstanciar,  
        coste + costeDeInstalarDEA, puntosCubiertos, dondeInstalar, conf)  
28    end  
29    if DEAConConflicto no es DEAAInstanciar && DEAAInstancias esta en  
       conf[DEAConConflicto] then  
30      limpiar conf[DEAConConflicto]  
31    else  
32      return DEAConConflicto  
33    end  
34  end  
35 end
```

de línea 6 a 9 se observa la revisión se la restricción dada por el presupuesto, donde al ser un generador de conflicto este agrega toda variable instancia anteriormente al conjunto de conflicto

del DEA instanciado en este punto. Esto es debido a que el presupuesto es una restricción que afecta a todas las variables instanciadas juntas. Entonces al tener relación con toda variable instanciada anteriormente, se agregan estas al conjunto de conflictos. Ahora en la línea 10 a 14, esta corresponde a cuando se encuentra posiciones de tal forma que cubran más puntos cubiertos que la solución actual. Luego le siguen las líneas 15 a 17, las cuales sirven en el caso que se hayan instanciados todas los DEAs posibles y no se haya pasado del presupuesto.

Finalmente, comienza donde se instanciaran las variables, primero se revisa el caso especial donde el DEA a instanciar ya ha estado instalado, entonces se revisará el caso que se deje en el mismo lugar, lo cual no cuesta nada. Después, por cada posible posición del DEA se instanciar, donde al coste se le sumará el coste de mover, si es que este DEA ya estaba instalado, o donde el coste sería sumará el coste de instalara un nuevo DEA. Y si es que el DEA a instanciar recibe un DEA con conflicto que no es él, quiere decir que hubo un DEA que tuvo conflicto y no pudo instanciar más, por lo tanto se revisa si es que el DEA a instanciar está dentro del conjunto, ya que si es está dentro quiere decir que es la variable más reciente instanciada que tuvo conflicto con ese DEA, entonces se queda ahí y se limpia el conjunto. Si no se sigue retornando hasta que llegue a la variable correspondiente.

Algorithm 2: Forward Checking implementado

Result: Encontrar las posiciones que permita cubrir óptimamente los puntos OHCA

Data: *posiblesPosiciones*, *DEAQuefaltanInstanciar*

```

1 Procedure FC(DEAAInstanciado)
2   foreach DEA  $\in$  DEAQuefaltanInstanciar do
3     foreach posicion  $\in$  posiblesPosiciones do
4       if en posicion ya esta siendo ocupada then
5         | sacar posicion de posiblesPosiciones
6       end
7     end
8   end
9 end

```

El orden de instansación corresponderá a primero instanciar variables que correspondan DEAs previamente instalados y luego instanciar variables que correspondan a nuevos DEAs. Esto se realiza debido a que los DEAs previamente instalados es como si ya tuvieran un valor inicial por el cual se puede iniciar, entonces se comenzaría a instanciar la solución con estos valores. Luego, hay dos momentos donde se revisan restricciones: cuando se aplica Forward Checking y al revisar el presupuesto. Esto es porque en el Forward Checking se reduce el dominio de las variables a instanciar, ya que se quitan toda posición utilizada por algún DEA instanciado previamente, porque no pueden haber dos DEA en la misma posición. El presupuesto se revisa como restricción que detiene el backtracking, ya que es un factor que limita la instanciación de DEA y donde se aplica el retorno guiado con la técnica CBJ.

7. Experimentos

Para los experimentos, se decidió dos grupos de instancias: uno que no contenga DEAs instalados previamente y con un presupuesto; otro que contenga DEAs instalados previamente y con un presupuesto igualmente. Además los puntos dentro del primer grupo son los mismo que dentro del segundo grupo, solo que el segundo tiene algunos puntos con DEAs instalados previamente. Esto es para poder comparar el comportamiento de la respuesta dentro del mismo conjunto de puntos, pero con la singularidad de poseer DEAs instalados previamente.

Utilizando las instancias descritas en la tabla 3 y 4, se busca evaluar el tiempo que se

Nombre	Cantidad Coordenadas	Presupuesto	Radio de Cobertura
SJC324-3	324	3	800
SJC402-3	402	3	800
SJC500-3	500	3	800
SJC500-7	500	7	800
SJC708-5	708	5	800
SJC708-7	708	7	800
SJC708-11	708	11	800

Tabla 3: Característica de las instancias utilizadas sin DEA instalado previamente

Nombre	Cantidad Coordenadas	Presupuesto	Radio de Cobertura	Instalados Previamente
SJC324-3	324	2.2	800	1
SJC402-3	402	2.2	800	1
SJC500-3	500	2	800	1
SJC500-7	500	4	800	4
SJC708-5	708	4.2	800	1
SJC708-7	708	3.4	800	4
SJC708-11	708	7	800	5

Tabla 4: Característica de las instancias utilizadas con DEAs instalados previamente

demora en encontrar la solución máxima, en el caso que se demore un largo tiempo se tiene un tiempo de ejecución máxima de 6 horas, donde se tomará la mejor solución hasta el momento. Y de la solución se observará la cantidad de puntos cubiertos por la solución, cuantos DEAs fueron instalados, cuantos DEAs instalados previamente fueron reposicionados y el presupuesto restante.

Todas las instancias descritas tienen como los casos de prueba utilizados en el trabajo de Max-Covering de Lorena y Pereira [9]. Todas las instancias se ejecutaron dentro de una instancia de Amazon EC2 z1d con un procesador Intel(R) Xeon(R) Platinum 8151 CPU @3.40GHz. Y el cálculo de tiempo de ejecución se obtiene utilizando el `time` sistema, donde el tiempo del algoritmo será `user + sys`.

8. Resultados

Para los resultados mostrados en la tabla 5 y 6 se muestra la siguiente información:

- # Coords Cub: cantidad de coordenadas cubiertas donde han habido paros cardiacos
- % Coords Cub: porcentaje de coordenadas cubiertas donde han habido paros cardiacos
- \$ Restante: presupuesto que sobra con la solución entregada
- # DEA I: cantidad de DEA nuevos instalados
- # DEA M: cantidad de DEA instalados previamente movidos
- % Cov: porcentaje de cobertura en el experimento de Lorena y Pereira
- # Instalaciones: cantidad de instalaciones realizadas en el experimento de Lorena y Pereira

Comparando la tabla 5 y la tabla 6, se puede observar que los porcentajes de las 3 primeras instancias tuvieron valores coherentes consigo mismo, respecto a que la instancia SJC324-3 tuvo más porcentaje que SJC402-3, y SJC402-3 tuvo más porcentaje que SJC500-3. Los valores no

Nombre	% Cov	# Instalaciones	Tiempo Ejecución
SJC324-3	95.5 %	3	5.33 s
SJC402-3	91.9 %	3	11.09 s
SJC500-3	79.8 %	3	16.42 s
SJC500-7	99.9 %	7	85.58 s
SJC708-5	88.8 %	5	54.65 s
SJC708-7	95.7 %	7	74.65 s
SJC708-11	100.0 %	11	207.51 s

Tabla 5: Resultados de las instancias según el experimento realizado por Lorena y Pereira [9]

Nombre	# Coords Cub	% Coords Cub	\$ Restante	# DEA I	Tiempo Ejecución
SJC324-3	306	94.4 %	0	3	2 m 52.174 s
SJC402-3	353	87.8 %	0	3	5 m 37.132 s
SJC500-3	339	67.8 %	0	3	9 m 46.651 s
SJC500-7	405	90.0 %	0	7	6 horas
SJC708-5	488	68.9 %	0	5	6 horas
SJC708-7	527	74.4 %	0	7	6 horas
SJC708-11	521	73.6 %	0	11	6 horas

Tabla 6: Resultados de las instancias sin DEA instalado previamente

son iguales puesto en los experimentos de la tabla 5 Lorena y pereira consideraron demandas para cada punto, en cambio en la tabla 6 para ser acorde al problema cada punto era considerado con demanda 1. Ahora bien, se puede obtener también que los tiempos obtenidos en la tabla 5 son inferiores con creces respecto a los tiempos obtenidos en la tabla 6, hasta ahí casos en la tabla 6 donde las ultimas 4 instancias se tomaron un tiempo grande y no alcanzaron a terminar. Esto último se debe principalmente a la técnica aplicada, ya que en la tabla 5 se ocupo una técnica heurística para resolver el problema, en cambio en la tabla 6, que fueron los resultados obtenidos con el algoritmo descrito en la sección anterior, el cual corresponde a una técnica de búsqueda completa.

Ahora bien, con los resultados obtenidos en la tabla 7, comparandoló con la tabla 6, se puede observar ciertas situaciones interesantes. Primero, para las dos primeras instancias el DEA instalado previamente no estaba en una coordenada considerada en el óptimo, entonces lo que realiza el algoritmo es moverlo, ya que el presupuesto lo permite, obteniendo así el mismo porcentaje de cobertura en ambas situaciones. Lo mismo sucede para la segunda instancia.

Nombre	# Coords Cub	% Coords Cub	\$ Restante	# DEA I	# DEA M	Tiempo Ejecución
SJC324-3	306	94.4 %	0	2	1	2 m 47.662 s
SJC402-3	353	87.8 %	0	2	1	5 m 35.682 s
SJC500-3	308	61.6 %	0	2	0	10 m 15.591 s
SJC500-7	472	94.4 %	0	4	0	6 horas
SJC708-5	492	69.4 %	0.2	4	0	6 horas
SJC708-7	499	70.5 %	0.2	3	1	6 horas
SJC708-11	592	83.6 %	0	7	0	6 horas

Tabla 7: Resultados de las instancias con DEAs instalados previamente

Segundo, en el caso del que presupuesto sea un limitante se observa lo que sucede en la instancia SJC500-3 debido a que al tener un presupuesto inferior a la instancia de la tabla 6, este opta por dejar el DEA instalado donde estaba y instalar dos DEA nuevos, provocando

así que el porcentaje de cobertura haya disminuido, puesto que solo podían instalar dos DEA nuevos y el DEA instalado previamente no se puede mover.

Además, se puede observar que en las últimas cuatro instancias de la tabla 6 y 7, las cuales son instancias que no alcanzaron a llegar al óptimo, solo en dos de esas en la tabla 7 entregó un resultado peor que el indicado en la tabla 6, y estas dos tienen un presupuesto restante de 0,2, por lo que si el tiempo de ejecución hubiese sido mayor, eventualmente hubiera reposicionado un DEA en otro punto de tal forma de aumentar su porcentaje de cubierta. En los otros dos casos donde la tabla 7 superó en porcentaje de cobertura que la tabla 6, se puede observar que se quede a que ya habían DEA instalados y esos entregados un porcentaje no menor, entonces eso hace que dentro del mismo tiempo la instancia de la tabla 7 pueda observar más formas de instalar nuevos DEAs manteniendo los ya instalados en su lugar.

Por último, los tiempos de la tabla 7 en los casos que llegaron a terminar son superiores a lo de la tabla 6. Esto se debe por lo explicado en la representación, que indica que tener DEAs instalados previamente, se agregan estos como extras haciendo que dentro de la misma instancia el tamaño del espacio de búsqueda tenga para el caso con DEAs, ya instalados sea un poco mayor, provocando que el tiempo de ejecución tenga la diferencia mostrada.

9. Conclusiones

Para concluir, las técnicas expuestas resuelven el mismo problema, pero con ciertas diferencias puntuales, donde generalmente difieren en tomar en cuenta otros aspectos al momento de tener que relocalizar DEAs. Estas diferencias pueden ser no solo buscar el mayor cubrimiento de zonas, sino que también minimizar la cantidad de dispositivos instalados, o también considerar el radio como una función de probabilidad que permita indicar los resultados más cercanos a la realidad. Aunque existan diferencias entre los modelos, todos tienen la limitación de los datos de entrada, debido a que si bien la cantidad de datos puede ser grande y decidir a base de estos, este proceso es bastante azaroso. Esto debido a que en verdad no se puede asegurar que los siguientes paros cardíacos irán ocurriendo nuevamente en esos mismos lugares, ya que pueden cambiar. Siguiendo con esto último, una técnica que permitiría disminuir con creces esta probabilidad, es realizar la validación cruzada de k iteraciones. Disminuiría la probabilidad ya que esta técnica permite, después de fijar los DEAs, verificar cómo se comportarían las localizaciones encontradas si se le pasan nuevos datos, de manera de poder discernir si está correcta la solución o no.

Ahora bien, con la propuesta realizada para solucionar el problema dado, se puede observar que las ventajas son que permite encontrar la respuesta óptima exacta, ya que es una técnica de búsqueda completa y que se aprovecha correctamente de los datos de manera de buscar el óptimo por donde sería más problema, como se pudo observar en las instancias analizadas de la tabla 6 y 7, donde al tenerlo representado de esa forma permitía llegar en el mismo lapso de tiempo a soluciones mejores.

Por otra parte, las desventajas observadas son que toma tiempo para resolver un sistema pequeño, que la técnica de CBJ no es adecuada para el problema, debido a que el grafo de restricciones es completo, ya que el presupuesto es una restricción donde todas las variables se afectan entre todas, por lo que no se aprovecha la técnica haciendo que el retorno guiado no sea útil y que agregue complejidad extra al problema.

Para el futuro, habría que buscar una técnica que permita mejorar los tiempos obtenidos y pueda acercarse a los valores óptimos teóricos, por lo que se debería buscar alguna técnica heurística que permita resolver este problema. Además analizar los datos de tal forma de no considerar solamente posiciones donde han habido paros cardíacos, como se realizó en este caso, para ello habría que tener en consideración la cantidad de paros cardíacos y sus posiciones, de manera de analizar estos datos y obtener un valor probabilístico que permita decidir con mejor precisión donde posicionar un DEA.

Referencias

- [1] Byran Bonnet et al. Optimal placement of public-access aeds in urban environments. *Computers & Industrial Engineering*, 90:269–280, 2015.
- [2] Timothy C.Y. Chan, Derya Demirtas, and Roy H. Kwon. Optimizing the deployment of public access defibrillators. *Management Science*, 2016.
- [3] Timothy C.Y. Chan et al. Identifying locations for public access defibrillators using mathematical optimization. *Circulation*, pages 1800–1809, 2013.
- [4] Timothy C.Y. Chan, Zuo-Jun Max Shen, and Auyon Siddiq. Robust defibrillator deployment under cardiac arrest location uncertainty via row-and-column generation. *Operations Research*, pages 1–22, 2017.
- [5] Richard Church and Charles Reville. The maximal covering location problem. *Papers Regional Sci Assoc*, 32:101–120, 1974.
- [6] Thi Hong Diep Dao. Spatio-temporal location modeling in a 3d indoor environment: the case of aeds as emergency medical devices. *International Journal of Geographical Information Science*, 26:469–494, 2012.
- [7] Alan S. Go et al. Heart disease and stroke statistics. *Circulation*, 129:e206, 2014.
- [8] Jan-Throsten Gräsner et al. Eureka one-27nations, one europe, one registry a prospective one month analysis of out-of-hospital cardiac arrest outcomes in 27 countries in europe. *Resuscitation*, 105:188–195, 2016.
- [9] Luiz A. N. Lorena and Marcos A. Pereira. A langragean/surrogate heuristic for the maximal covering location problema using hillsman’s edition. 2015.
- [10] Alexander Nürnberger et al. Out of hospital cardiac arrest in vienna: Incidence and outcome. *Resuscitation*, 84:42–47, 2013.
- [11] Nicholas John Tierney et al. Novel relocation methods for automatic external defibrillator improve out-of-hospital cardiac arrest coverage under limited resources. *Resuscitation*, 125:83–89, 2018.