

Capítulo 6

Algoritmos Heurísticos aplicados a maximizar el tiempo de Cobertura de una Red

6.1

Tiempo de vida en la Cobertura de una Red

Como ya se ha visto en los capítulos precedentes, en los problemas de monitorización de regiones u objetivos mediante Redes de sensores, el concepto de *Cobertura* entraña una *medida de la calidad del servicio que presta la red* en relación con el propósito para el que se diseña, y dicho concepto está sujeto a una amplia abanico de interpretaciones, en función del tipo de sensores empleado y de las características del escenario o zona a monitorizar. En la mayoría de los casos, la finalidad de la Red es tener cada localización (u objetivo) del espacio físico de interés dentro del rango de sensibilidad de al menos un sensor.

Por otra parte, en las aplicaciones reales, hay que tener en cuenta que en una Red de sensores inalámbrica, cada uno de los sensores de la Red es un dispositivo (de pequeña dimensión) cuya fuente de energía suele ser una batería incorporada en el propio dispositivo. Las limitaciones físicas del dispositivo implican que la cantidad de energía de cada sensor, y por tanto el tiempo de vida del mismo, es también limitada. Por añadidura, en la mayoría de las aplicaciones los sensores de una Red solo son utilizados una vez debido a que las baterías no pueden ser reemplazadas o recargadas con facilidad, ya que a menudo la red de sensores se despliega en zonas inhóspitas u hostiles. La energía es por tanto el recurso más importante de la red, y por ello en las aplicaciones de Redes de sensores (así como en las estrategias de despliegue) se debe perseguir, como uno de los requisitos más importantes, el más bajo consumo de energía posible de la red. Consecuentemente, diseñar métodos que aumenten el tiempo de vida de la Red optimizando el consumo de energía es de gran importancia en las aplicaciones reales. Secundariamente, hay que mencionar razones económicas y ambientales que refuerzan el bajo consumo de energía como un objetivo deseable.

Los nodos de una red de sensores inalámbrica tienen tres estados. En el modo *activo* un sensor puede monitorizar el entorno y comunicar con otros sensores (o

bien con la estación base). En el modo *durmiente* un sensor no puede monitorizar ni comunicar. Sin embargo, el estado de un sensor puede cambiar de *durmiente* a *activo* mediante la recepción de una señal apropiada (bien de otro sensor, bien de la estación base). Finalmente en el modo *apagado* (“off”) el nodo está completamente inactivo, bien por agotamiento de la batería, bien porque el nodo se ha apagado definitivamente. Existen varios métodos para ahorrar energía en el funcionamiento de la red, entre los que citamos:

- Organizar la red de forma que se alternen los sensores *activos* y los *durmientes*.
- Controlar la energía consumida ajustando tanto el rango de transmisión como los canales sobre los que discurre la transmisión.
- Reducir el volumen de datos transmitido y evitar la infrautilización de la capacidad de transmisión.

En lo que sigue nos centraremos en el primer método, ya que tanto el segundo como el tercero se orientan a las características técnicas de la red. Se trata de organizar óptimamente los nodos de la red entre activos y durmientes, de forma que los nodos que cubran redundantemente un objetivo puedan entrar en modo durmiente.

Concretamente, en este capítulo estudiaremos el problema de la ubicación de los nodos de sensores en un área monitorizada y su organización, tal que se consiga una cobertura completa ahorrando energía para prolongar óptimamente la vida de la red. Si todos los sensores de la red se activaran a la vez, entonces el tiempo de vida de la red sería igual al tiempo de vida de uno de sus sensores (asumiendo sensores homogéneos con la misma vida), por ejemplo, de b horas. Pero si organizamos los nodos de la red en un conjunto de subconjuntos, o *clusters*, mutuamente excluyentes, de forma que cada subconjunto sea capaz de cubrir completamente todos los objetivos y dichos subconjuntos sean disjuntos entre sí, se ahorrará energía prolongándose la vida de la red en $b \cdot k$ horas (donde k es el número de conjuntos disponibles). En efecto, esto se consigue activando sucesivamente cada uno de dichos subconjuntos, de forma que en cada instante sólo hay un subconjunto activo, mientras que los demás están durmientes,

y así hasta agotar todos los clusters disponibles. Cada uno de tales clusters se llama también “Conjunto cubridor”.

Por otra parte, y en función de los objetos a cubrir, los problemas de cobertura por WSN son de dos tipos: *Cobertura de Área* y en *Cobertura de Objetivos*. En el primer tipo, el objeto de interés a cubrir es un área, y en el segundo es un conjunto de puntos u objetivos. Sin embargo, como veremos a continuación, ambos están estrechamente relacionados: En la optimización del tiempo de vida de Cobertura de una red, la Cobertura de Área puede transformarse en Cobertura de Objetivos. Por ello, en este problema consideraremos un modelo de Cobertura de Objetivos mediante una red WSN, donde los sensores se despliegan aleatoriamente en la zona donde se sitúan los objetivos a cubrir, y envían información a una estación base. Estudiaremos el problema de alargar la vida de la red, proponiendo varios algoritmos que generen el máximo número de Conjuntos cubridores disjuntos, analizando el comportamiento de dichos algoritmos sobre distintos escenarios aleatoriamente generados, y comparando los resultados que cada uno proporciona.

6.2

Escenario del problema: Método de despliegue y Tipo de Cobertura

A) Maximización del tiempo de Cobertura: Un ejemplo ilustrativo

Antes de analizar los elementos que consideraremos en el problema, veamos un ejemplo ilustrativo de lo que se persigue. Supongamos que disponemos de un conjunto de n sensores para cubrir una colección de m objetivos. En un caso general, cada sensor cubrirá uno o más de tales objetivos, y cada objetivo estará cubierto por al menos un sensor. En nuestro ejemplo, tenemos 6 sensores: $S_1, \dots, S_6$, y 5 objetivos

$t_1, \dots, t_5$ de forma que cada uno de los sensores cubren los objetivos como sigue:
 $S_1 = \{t_1, t_2, t_3, t_4\}$, $S_2 = \{t_1, t_2, t_5\}$, $S_3 = \{t_2, t_3, t_4, t_5\}$, $S_4 = \{t_2, t_3, t_5\}$, $S_5 = \{t_1, t_3, t_5\}$ y
 $S_6 = \{t_3, t_4, t_5\}$. Este esquema de cobertura se puede también expresar mediante una
 tabla de doble entrada (Figura 6.1), que en la implementación del algoritmo será una
 matriz binaria $n \times m$, donde $a_{ij} = 1$, si el sensor S_i cubre el objetivo t_j , y $a_{ij} = 0$, en
 caso contrario.

	t_1	t_2	t_3	t_4	t_5
S_1	x	x	x	x	
S_2	x	x			x
S_3		x	x	x	x
S_4		x	x		x
S_5	x		x		x
S_6			x	x	x

	t_1	t_2	t_3	t_4	t_5
S_1	x	x	x	x	
S_4		x	x		x
S_3		x	x	x	x
S_5	x		x		x
S_2	x	x			x
S_6			x	x	x

$\left. \begin{array}{c} S_1 \\ S_4 \end{array} \right\} C_1$

$\left. \begin{array}{c} S_3 \\ S_5 \end{array} \right\} C_2$

$\left. \begin{array}{c} S_2 \\ S_6 \end{array} \right\} C_3$

Figura 6.1: Solución óptima que maximiza la vida de la red

Supongamos que todos los sensores tienen la misma energía, y por tanto el mismo tiempo de vida (hipótesis que no resta generalidad al problema, ya que la mayoría de las redes están formadas por sensores homogéneos). Para prolongar la vida de la Red cubriendo todos los objetivos, distribuimos los sensores en *clusters* o Conjuntos cubridores C_i , esto es: en conjuntos disjuntos entre sí, tales que cada uno de ellos cubra completamente todos los objetivos. Cada uno de los Conjuntos cubridores se activará sucesivamente mientras los demás permanecen en estado durmiente. Entonces, el problema de maximizar la vida de la red se traducirá en maximizar el numero de conjuntos cubridores C_i . En el ejemplo, la solución óptima (como se verá más adelante) que maximiza la vida de la red está dada por:

$$C_1 = \{S_1, S_4\}, C_2 = \{S_3, S_5\}, C_3 = \{S_2, S_6\}.$$

A continuación argumentaremos motivadamente el método de despliegue de los sensores (aleatorio) y el tipo de cobertura: Cobertura de objetivos, que consideraremos en el problema.

B) Método de despliegue: Dispersión aleatoria

La estrategia de despliegue de los sensores de la red puede ser determinística (esto es, la ubicación de los sensores está predeterminada y se conoce “a priori”), o aleatoria, dependiendo del escenario a monitorizar. Existen varias razones por las cuales un despliegue determinístico es impracticable:

- Cuando el despliegue se realiza en zonas inhóspitas, remotas, inseguras o peligrosas, o en aplicaciones militares.
- Cuando la red tiene un gran número de sensores, de forma que el despliegue determinístico conlleva un elevado coste.

Por ello el método de despliegue utilizado es aleatorio, donde los sensores son dispersados “a granel” desde un aeroplano sobre la zona en cuestión. La ausencia de una ubicación precisa de los sensores se compensa desplegando un gran número de sensores sobre la zona, lo que aumenta la probabilidad de cobertura de los objetivos. Los datos recogidos por los sensores se envían a la estación base, donde se establecen los objetivos, o campos, que cubre cada sensor.

Sin embargo, es interesante considerar el despliegue determinístico, ya que proporciona una cota inferior del número de sensores necesario para cubrir un área. Si consideramos una ubicación regular de nodos homogéneos en un área de tamaño T_{area} , siendo r el rango de sensibilidad de cada sensor (de forma que el rango de detección del sensor es un disco de radio r), el mínimo número N de sensores necesarios para cubrir dicha área está dado por la relación (Figura 6.2):

$$\frac{N r^2 \pi}{T_{area}} = \frac{2 \pi}{\sqrt{27}}$$

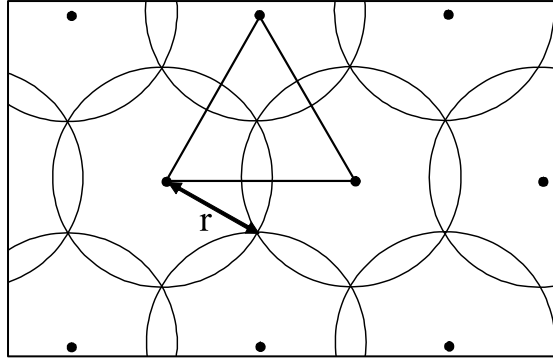


Figura 6.2: Cobertura de un área regular con el mínimo número de nodos.

Por tanto, consideraremos sensores desplegados aleatoriamente, y un modelo de control centralizado: esto es, una vez que los nodos se han desplegado, son capaces de transmitir su ubicación a una estación base. Dicha estación recogerá los datos y establecerá la organización óptima de la red para prolongar al máximo el tiempo de vida de la misma.

C) Tipo de Cobertura: Cobertura de Objetivos

Como ya se ha visto, los problemas de Cobertura presentan multitud de variantes basadas en las características del escenario, en las restricciones de la red y en el tipo de objetos a cubrir. Si el tipo de objeto a ser cubierto es un área tenemos un problema de Cobertura de Área, mientras que si es un conjunto de objetivos o puntos tenemos un problema de Cobertura de Objetivos. Sin embargo, *en los problemas de maximización de la vida de la red existe una estrecha relación entre ambos modelos*, de hecho, veremos que un problema de Cobertura de Área se puede transformar en un problema de Cobertura de Objetivos.

Supongamos tenemos n nodos (sensores) omnidireccionales $S_1, \dots, S_n$, con respectivos rangos de sensibilidad r_i ($i=1, \dots, n$), de forma cada sensor S_i puede monitorizar un disco centrado en él y de radio el rango r_i (Figura 6.3). Los rangos de

sensibilidad pueden ser distintos, y, como veremos más adelante, el área de sensibilidad del sensor podrá tener cualquier forma (no necesariamente circular).

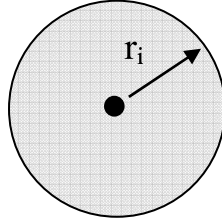


Figura 6.3: Rango r_i , y disco de sensibilidad de un sensor.

Un sensor S_i cubre un objetivo (o punto) p si y solo si dicho objetivo está en el disco de sensibilidad de S_i . Asimismo, un sensor S_i cubre un área A si y sólo si todos los puntos del área están en el disco de sensibilidad del sensor.

En los problemas de maximizar el tiempo de vida de Cobertura, el modelo de Cobertura de Área se puede transformar en el modelo de Cobertura de Objetivos mediante el concepto de “Campo” introducido por *Slijepcevic and Potkonjak* (2001), de la siguiente forma: En el modelo de la Cobertura de Área, el área a cubrir se particiona en pequeñas subregiones, llamadas *Campos*, de forma que *cada Campo está identificado únicamente por el conjunto de sensores que lo monitorizan*. Más formalmente:

Definición de Campo: Un Campo es el conjunto de todos los puntos que están cubiertos por el mismo conjunto de sensores.

Consideremos la Figura 6.4, con cuatro sensores, y donde el área de interés es el rectángulo sombreado. De acuerdo con la anterior definición, dicha área está particionada en 8 Campos: el Campo “1” es el conjunto de puntos cubiertos sólo por el sensor S_1 , el Campo “2” el conjunto de puntos cubiertos únicamente por los sensores S_1 y S_2 , etc.

Si representamos dichos Campos por los objetivos $t_1, \dots, t_8$, entonces *el problema de maximizar la vida de la red en el modelo de Cobertura de Área se puede transformar en maximizar la vida de la red en el modelo de Cobertura de Objetivos*.

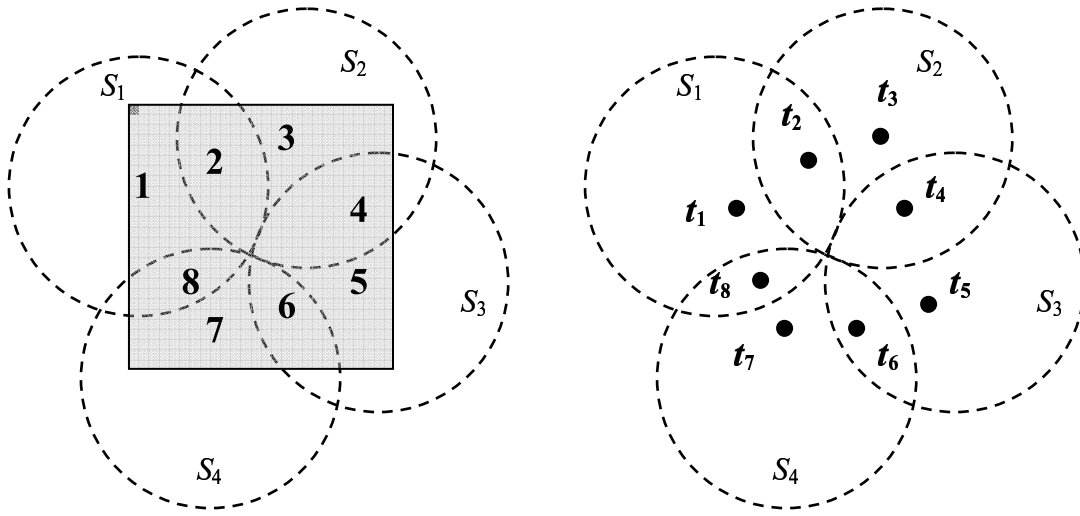


Figura 6.4: Transformación de un problema de Cobertura de Área (izqda.) en un problema de Cobertura de Objetivos o puntos (dcha.)

En el mismo trabajo, *Slijepcevic and Potkonjak* (2001) proponen un método para computar todos los campos que se puede ejecutar en una fase previa, una vez que la estación base ha recogido los datos enviados por los sensores. Puesto que en éste esquema, cada campo –cada objetivo – se identifica por los sensores que lo cubren, los sensores pueden tener distinto rango de sensibilidad, e incluso el modelo se puede generalizar a sensores cuya área de sensibilidad no sea necesariamente circular, ya que para formular el problema basta con conocer los objetivos que cada sensor cubre.

Por otra parte, y como refuerzo de este planteamiento, *Cardei et al.* (2006) sostienen que incluso en un modelo más general el problema de la Cobertura de Área se puede aproximar por la Cobertura de Objetivos. En efecto, dado que este tipo de redes son de gran tamaño y densidad, cuando los sensores se despliegan aleatoriamente sobre un área a monitorizar, la cobertura de área en cuestión se puede aproximar mediante la *cobertura de los sensores desplegados* en la misma ya que, según establecen los autores, basta cubrir la localización de cada sensor para aproximar la cobertura de cada punto del área en cuestión.

Consecuentemente, y en base a todos los argumentos anteriores, ***consideraremos un modelo de Cobertura de Objetivos en el problema de maximización del tiempo de cobertura de la Red .***

6.3

Maximización del tiempo de Cobertura mediante Cubridores disjuntos: Descripción del problema

A continuación describimos las características del problema de maximizar el tiempo de cobertura, o equivalentemente la vida de la red, mediante la generación de conjuntos Cubridores disjuntos, cada uno de ellos capaz de cubrir todos los objetivos. El posterior análisis de complejidad probará el carácter NP-duro del problema, por lo que serán necesarios algoritmos heurísticos que produzcan soluciones subóptimas con razonable coste computacional.

Consideremos una red de sensores inalámbrica (WSN), de energía limitada, compuesta por un conjunto S de n sensores: $S = \{S_1, \dots, S_n\}$ de energía homogénea (esto es, del mismo tiempo de vida), desplegados aleatoriamente al objeto de cubrir una colección T de m objetivos: $T = \{t_1, \dots, t_m\}$. Como ya se ha señalado, la hipótesis de energía homogénea es asumida en la mayoría de los casos. Supondremos que cada sensor S_i tiene un rango de sensibilidad r_i (no necesariamente circular), y que dichos rangos pueden ser diferentes, ya que los algoritmos que describiremos sólo requieren saber los objetivos que cada sensor cubre. Asimismo, la Estación Base tiene las coordenadas de los sensores y de los objetivos, de modo que puede computar cuántos y cuáles objetivos cubre cada sensor, y suponemos también que cada objetivo está cubierto por al menos un sensor. El problema de la Cobertura de objetivos consiste en programar la actividad de los sensores tales que todos los objetivos estén continuamente cubiertos y se maximice el tiempo de vida de la red. El mecanismo de programación debe complementarse con las siguientes hipótesis:

- Los sensores envían sus localizaciones a la Estación Base.
- La Estación Base ejecuta el algoritmo de programación de los sensores, y transmite la planificación cuando cada nodo está activo.
- Cada sensor es capaz de autosituarse en los intervalos activo/durmiente.

Dejamos fuera del estudio los elementos técnicos para la sincronización de los nodos, la transmisión eficiente de los datos, etc. El objetivo del problema es obtener una colección $C = \{C_1, \dots, C_k\}$ de k “Conjuntos cubridores” o Clusters. Cada conjunto cubridor C_i es un subconjunto de sensores disponibles ($C_i \subseteq S$), y debe cubrir todos los objetivos de T . Además, cada sensor solo puede pertenecer a un Conjunto cubridor, o equivalentemente, los conjuntos cubridores deben ser disjuntos, esto es:

$$C_i \cap C_j = \emptyset, \quad \forall i, j: 1 \leq i, j \leq k, \text{ con } i \neq j$$

El objetivo del problema es maximizar el cardinal $|C| = k$ de la colección de cubridores, esto es, maximizar el número k de conjuntos cubridores pertenecientes a la colección C . Veamos que este es NP-duro, ya que se puede formular como un caso particular del siguiente problema de decisión:

Problema del k -Conjunto Cubridor (conjunto conteniendo a k Cubridores):

- ENTRADA: Colección C de subconjuntos de S , entero positivo k .
- CUESTIÓN: ¿Contiene C una colección de k Conjuntos cubridores disjuntos para S ? Esto es, una colección de k conjuntos $C_1, \dots, C_k$, donde $C_i \subseteq C$, y tal que cada elemento de S pertenezca al menos a un elemento de cada C_i .

Slijepcevic and Potkonjak (2001) establecen que el problema del k -Conjunto Cubridor es NP-Completo, (nótese que este problema de decisión es distinto al problema de la k -cobertura descrito en el Capítulo 4, en donde se indaga si cada objetivo está cubierto por al menos k sensores de la red). Por tanto nuestro problema, que es la versión de optimización del anterior, es NP-duro. Esto es: no existe algoritmo que resuelva, en tiempo polinomial, cualquier ejemplar del problema para cualquier tamaño de entrada del mismo.

Los problemas NP-duros se pueden resolver por procedimientos exhaustivos consistentes en la enumeración completa de todas las posibles soluciones. Este procedimiento exhaustivo sólo es posible en problemas de pequeño tamaño (como en el ejemplo de la Figura 6.1, con $n = 6$ sensores y $m = 5$ objetivos), pero es impracticable

en general. Y en el caso de las aplicaciones del problema, tengamos en cuenta que trabajamos con redes de un gran número de sensores y un gran número de objetivos a cubrir. Por tanto abordaremos el problema mediante procedimientos heurísticos que produzcan soluciones subóptimas con razonable coste computacional.

Formulación del Problema mediante Programación Entera

Antes de describir los heurísticos, veremos cómo el problema anterior se puede formular mediante un problema de Programación Entera. Para ello definimos las siguientes variables binarias:

$$x_{ip} = \begin{cases} 1, & \text{si el sensor } S_i \text{ está en el conjunto } C_p, \text{ para } i = 1, \dots, n, \text{ y } p = 1, \dots, k. \\ 0, & \text{en caso contrario} \end{cases}$$

$$a_{ij} = \begin{cases} 1, & \text{si el sensor } S_i \text{ cubre el objetivo } t_j, \text{ para } i = 1, \dots, n, \text{ y } j = 1, \dots, m \\ 0, & \text{en caso contrario} \end{cases}$$

Maximizar k

Sujeto a:

$$\sum_{p=1}^k x_{ip} \leq 1, \text{ para } i = 1, \dots, n \quad (\text{Cada sensor está como mucho en un Conjunto cubridor})$$

$$\sum_{i=1}^n x_{ip} a_{ij} \geq 1, \text{ para } p = 1, \dots, k, \text{ y } j = 1, \dots, m \quad (\text{Cada objetivo está cubierto por al menos un sensor en cada Conjunto cubridor})$$

$$x_{ip} \in \{0, 1\}, \quad a_{ij} \in \{0, 1\}, \text{ para todo } i, j, p.$$

Este tipo de formulaciones permiten otro tratamiento heurístico del problema, aplicando técnicas de relajación para transformar la anterior formulación en una de Programación Lineal. Nosotros no abordaremos estas técnicas, y nos centraremos en el diseño de heurísticos que aprovechen la estructura del problema.

6.4

Algoritmos Heurísticos para generar el máximo número de Conjuntos Cubridores disjuntos

A continuación propondremos dos algoritmos Heurísticos para resolver eficientemente el problema anteriormente descrito, esto es, para maximizar el número de Conjuntos cubridores (o Clusters) disjuntos, donde cada uno de los conjuntos cubre completamente todos los objetivos. Ambos algoritmos tienen como parámetros de entrada S (conjunto de sensores) y T (conjunto de objetivos), y producen como salida el cardinal de C (colección de Conjuntos cubridores), así como la descripción de cada Conjunto cubridor C_i . Esto es:

INPUT: $S = \{S_1, \dots, S_n\}$, (conjunto de n sensores)

$T = \{t_1, \dots, t_m\}$, (conjunto de m objetivos)

ALGORITMO: PROCEDIMIENTO PRINCIPAL

OUTPUT: $C = \{C_1, \dots, C_k\}$, (Colección de Conjuntos cubridores disjuntos)

$|C| = k$, (número de Conjuntos cubridores C_i , $i = 1, \dots, k$)

Los dos algoritmos trabajan según una estrategia constructiva, esto es, parten de una solución vacía, que iterativamente se va construyendo. Así, la construcción de cada Conjunto cubridor C_i es de la siguiente manera: inicialmente, $C_i = \emptyset$, y en cada iteración, el algoritmo selecciona *según un determinado criterio* un sensor S_u para incorporarlo al conjunto C_i . Esto es:

```

Sens_Disponibles = S
While (No STOP_CRITERIUM) do
    Procedimiento CONSTRUYE_ $C_i$ 
         $C_i = \emptyset$ , Obje_Cubiertos =  $\emptyset$ 
        While (Obje_Cubiertos  $\neq T$ , esto es,  $C_i$  no cubra  $T$ ) do
            Selecciona  $S_u \in Sens\_Disponibles$  según Criterio de Selección
             $C_i = C_i \cup S_u$ 
            Actualiza ( $Sens\_Disponibles$ , Obje_Cubiertos)
        end While
    end (Procedimiento CONSTRUYE_ $C_i$ )
end While

```

La estrategia constructiva de cada Conjunto cubridor C_i se basa en el proceso de selección (según un determinado *Criterio*) de cada nuevo sensor S_u que va a ser incorporado a la solución. En dicho Criterio de Selección estriba la diferencia entre ambos algoritmos, según explicaremos después. Sin embargo, hay elementos comunes en la estructura de los algoritmos, como son el criterio de parada (**Stop_Criterium**), y el objetivo básico de prestar especial atención a los objetivos más escasamente cubiertos, llamados **Objetivos Críticos**.

Máximo Teórico, y Objetivos Críticos:

En el criterio de parada (Stop_Criterium), el algoritmo terminará bien cuando no queden sensores disponibles ($Sens_Disponibles = \emptyset$), bien cuando los sensores disponibles no cubran todos los objetivos, o bien cuando el número de Conjuntos cubridores haya alcanzado un máximo teórico: $|C| = Max_Teor$. Este máximo teórico viene determinado por los objetivos críticos (los objetivos más escasamente cubiertos), y puede calcularse en el INPUT del problema, ya que

Máximo Teórico = Número de sensores cubriendo un Objetivo crítico

El ejemplo que sigue es mismo que el del apartado A de la sección 6.2. En él, t_1 y t_4 son ambos Objetivos críticos, ya que son los objetivos más escasamente cubiertos, ambos con nivel de cobertura 3 (cubiertos por 3 sensores). El Máximo Teórico es, por tanto 3, y constituye una **Cota Superior** de la solución del problema, esto es, del *Número de Conjuntos cubridores disjuntos que se pueden construir*.

	t_1	t_2	t_3	t_4	t_5
S_1	1	1	1	1	
S_2	1	1			1
S_3		1	1	1	1
S_4		1	1		1
S_5	1		1		1
S_6			1	1	1

Objetivos Críticos: t_1 y t_4 .

Máx_Teor = 3 (Nivel de cobertura de los Objetivos críticos).

Cota superior del cardinal de la solución = 3 (Máximo Número de Conjuntos Cubridores disjuntos)

Solución: $C = \{C_1, C_2, C_3\}$, con:

$C_1 = \{S_1, S_4\}$, $C_2 = \{S_3, S_5\}$, $C_3 = \{S_2, S_6\}$

Dicho Máximo Teórico es una *Cota Superior del cardinal de la solución*, ya que no siempre se puede alcanzar, como se puede comprobar con un ejemplo muy sencillo:

Ejemplo con 3 Sensores y 4 Objetivos:

	t_1	t_2	t_3	t_4
S_1	1	1		1
S_2	1	1	1	
S_3		1	1	1

Objetivos críticos: t_1, t_3 y t_4 , de nivel de cobertura 2

Máximo Teórico = 2 (Cota Superior de la solución)

La Cota Superior NO se alcanza:

Cualquier posible solución tiene 1 sólo Cubridor C_1

Posibles soluciones: $C_1 = \{S_1, S_2, S_3\}$, ó bien: $C_1 = \{S_1, S_2\}$, ó bien: $C_1 = \{S_1, S_3\}$, ó bien $C_1 = \{S_2, S_3\}$.

Como se puede comprobar, la solución no es única, aunque sí su cardinal. Esto es, la solución óptima puede conseguirse con distintas colecciones $C = \{C_1, \dots, C_k\}$ de Conjuntos cubridores, pero todas tienen el mismo cardinal $k = |C|$, verificando:

$$k = |C| \leq \text{Max_Teor}$$

A continuación pasamos a describir los algoritmos, que como ya hemos dicho construyen los Conjuntos cubridores recursivamente. Concretamente nos centraremos en el proceso de construcción de los Cubridores C_i , basado en el Criterio de Selección de los sensores que iterativamente se van incorporando a C_i . A pesar de que cada algoritmo aplica un Criterio de Selección distinto, ambos trabajan sobre una base común, que en cada iteración da preferencia a los sensores que cubran los objetivos pendientes más escasamente cubiertos, esto es, a los *actuales* objetivos críticos.

6.4.1 Algoritmo Heurístico H_1 (Greedy)

Un algoritmo Greedy (voraz) es un algoritmo que, en cada iteración, ejecuta los movimientos que *mayor mejora* producen (en función del objetivo que se persiga), esto es, los movimientos que maximizan una determinada función de beneficio. Las

estrategias heurísticas Greedy son ampliamente usadas por su sencillez, flexibilidad y robustez, y son utilizadas bien en solitario, bien en combinación con otras estrategias heurísticas, que trabajan sobre las soluciones iniciales obtenidas mediante dicha técnica.

En nuestro caso, aplicaremos una estrategia Greedy para el procedimiento CONSTRUYE_ C_i , que construye iterativamente cada Conjunto cubridor C_i seleccionando, de entre los sensores actualmente disponibles que cubran los actuales objetivos críticos, aquél sensor que mayor *contribución* realice a la cobertura de objetivos. Básicamente, el procedimiento CONSTRUYE_ C_i opera como sigue: inicialmente tenemos $C_i = \emptyset$, el conjunto de sensores disponibles: “Sens_Disponibles” (con los sensores que no se han utilizado en los cubridores previos $C_1, \dots, C_{i-1}$), y el conjunto “Obje_Cubiertos” con los objetivos que cubre C_i (inicialmente, $Obje_Cubiertos = \emptyset$). Asimismo, “ $T- Obje_Cubiertos$ ” contiene los objetivos no cubiertos por C_i .

En cada iteración del procedimiento CONSTRUYE_ C_i aplicamos una estrategia Greedy para seleccionar el sensor S_u de mayor contribución que cubra algún objetivo escasamente cubierto. Esto es:

- Calculamos $t_{criticos} \subseteq T- Obje_Cubiertos$, esto es, calculamos la colección de *actuales* objetivos críticos (seleccionados entre los objetivos todavía no cubiertos por C_i).
- Computamos $S_{candidatos} \subseteq Sens_Disponibles$, tal que “ $S_{candidatos}$ ” contiene a todos aquellos sensores disponibles cubriendo algún objetivo crítico del conjunto “ $t_{criticos}$ ”.
- Seleccionamos aquél sensor $S_u \in S_{candidatos}$ que maximice una determinada “*Función de Contribución*” $f(S_u)$.

Cuando estén cubiertos todos los objetivos (esto es, cuando $Obje_Cubiertos = T$), el nuevo cubridor C_i estará construido.

La “**Función de Contribución**” la hemos definido para seleccionar aquél sensor que “cubra el mayor número de objetivos no cubiertos”. Esto es, definiendo:

$$f(S_u) = \{\text{Número de objetivos que cubre } S_u \text{ y que todavía no están cubiertos por } C_i\}$$

O, equivalentemente:

$$f(S_u) = \text{Cardinal de } (\{\text{Objetivos cubiertos por } S_u\} \cap \{T - \text{Obje_cubiertos}\})$$

Si hay varios sensores con el mismo valor de la Función de Contribución $f(S_u)$, se elige uno cualquiera de ellos indistintamente, por ejemplo el primero.

Una vez seleccionado el sensor S_u , se añade al cubridor C_i y se actualizan los conjuntos “Sens_Disponibles” y “Obje_Cubiertos” de la siguiente forma:

$$\text{Sens_Disponibles} = \text{Sens_Disponibles} - S_u$$

$$\text{Obje_Cubiertos} = \text{Obje_Cubiertos} \cup \{\text{Objetivos cubiertos por } S_u\}$$

En la siguiente página se describe el pseudocódigo del algoritmo.

Complejidad

Para evaluar su complejidad, observamos que *cada iteración* de “CONSTRUYE_ C_i ” se ejecuta en tiempo $O(mn)$, ya que en cada iteración del procedimiento, la complejidad de obtener la colección de objetivos críticos (m en el peor caso) y el subsiguiente conjunto de sensores $S_{\text{candidatos}}$ es en total $O(mn)$. Por otra parte, en el peor caso, hay $O(n)$ sensores candidatos, y para cada sensor candidato, la evaluación de la función de Contribución requiere $O(m)$. Luego la complejidad de cada Cubridor C_i es de $O(mn^2)$, por tanto la complejidad del algoritmo es $O(kmn^2)$, donde k es el número final de conjuntos Cubridores (recordemos que $k \leq \text{Max_Teor}$).

H₁: Algoritmo Heurístico Greedy

```

INPUT   $S = \{S_1, \dots, S_n\}$  ( $n$  sensores),  $T = \{t_1, \dots, t_m\}$  ( $m$  objetivos)
        (Matriz binaria  $(a_{ij})_{n \times m}$ , con  $a_{ij} = 1$  si y sólo si  $S_i$  cubre  $t_j$ .)

 $i = 0$ 
 $Sens\_Disponibles = S$ 
While ( $Sens\_Disponibles$  cubra completamente  $T$ ) do
    Procedimiento CONSTRUYE_ $C_i$ 
         $i = i + 1$ 
         $C_i = \emptyset$ ,  $Obj\_Cubiertos = \emptyset$ 

        While ( $Obj\_Cubiertos \neq T$ , esto es,  $C_i$  no cubra  $T$ ) do
            Calcula el conjunto  $t_{criticos} \subseteq T - Obj\_Cubiertos$ 
                (*  $t_{criticos}$  : conjunto de actuales objetivos críticos *)

            Computa  $S_{candidatos} \subseteq Sens\_Disponibles$ 
                (*  $S_{candidatos}$  : sensores cubriendo algún objetivo de  $t_{criticos}$  *)

            Selecciona  $S_u \in S_{candidatos}$  tal que
                
$$f(S_u) = \max \{f(S_w) : S_w \in S_{candidatos}\}$$


             $C_i = C_i \cup S_u$ 

            Actualiza ( $Sens\_Disponibles$ ,  $Obj\_Cubiertos$ ):
                 $Sens\_Disponibles = Sens\_Disponibles - S_u$ 
                 $Obj\_Cubiertos =$ 
                     $= Obj\_Cubiertos \cup \{\text{Objetivos cubiertos por } S_u\}$ 

        end While
    end (Procedimiento CONSTRUYE_ $C_i$ )
end While
RETURN  $i$  (número de Conjuntos Cubridores), y la colección de Cubridores:
         $C_1, C_2, \dots, C_i$ .
    
```

Aplicación del Algoritmo H₁: Un Ejemplo.

Veamos cómo trabaja el algoritmo utilizando el ejemplo anterior:

- Construcción de C_1

	t_1	t_2	t_3	t_4	t_5
S_1	1	1	1	1	
S_2	1	1			1
S_3		1	1	1	1
S_4		1	1		1
S_5	1		1		1
S_6			1	1	1

$$C_1 = \{S_1, S_2\}$$

$$t_{critico} = \{t_1, t_4\}$$

$$S_{candidatos} = \{S_1, S_2, S_3, S_5, S_6\}$$

$$f(S_i) \rightarrow \{4, 3, 4, 3, 3\}$$

$$\text{Elige} \rightarrow S_1$$

$$t_{critico} = \{t_5\}$$

$$S_{candidatos} = \{S_2, S_3, S_4, S_5, S_6\}$$

$$f(S_i) \rightarrow \{1, 1, 1, 1, 1\}$$

$$\text{Elige} \rightarrow S_2$$

- Construcción de C_2

	t_1	t_2	t_3	t_4	t_5
S_3		1	1	1	1
S_4		1	1		1
S_5	1		1		1
S_6			1	1	1

$$C_2 = \{S_3, S_5\}$$

$$t_{critico} = \{t_1\}$$

$$S_{candidatos} = \{S_5\}$$

$$f(S_i) \rightarrow 3$$

$$\text{Elige} \rightarrow S_5$$

$$t_{critico} = \{t_2, t_4\}$$

$$S_{candidatos} = \{S_3, S_4, S_6\}$$

$$f(S_i) \rightarrow \{2, 1, 1\}$$

$$\text{Elige} \rightarrow S_3$$

- STOP: No se puede formar un tercer Cubridor, ya que los sensores disponibles que quedan (S_4, S_6) no cubren todos los objetivos (de hecho, no cubren t_1).

Solución Final del Algoritmo: $C_1 = \{S_1, S_2\}$, $C_2 = \{S_3, S_5\}$

Vemos que la cota superior no se alcanza, ya que la Solución óptima obtenida por el método exhaustivo, es de 3 Cubridores: $C_1 = \{S_1, S_4\}$, $C_2 = \{S_3, S_5\}$, $C_3 = \{S_2, S_6\}$

6.4.2 Algoritmo Heurístico H_2 (con Penalización)

Como ya explicamos antes, este segundo algoritmo se diferencia del anterior en el procedimiento “CONSTRUYE_ C_i ” de construcción de los Cubridores, y la *diferencia estriba en la función de Selección* $f(S_u)$ mediante la cual se elige a los sensores que recursivamente se van incorporando a la construcción de C_i . Básicamente, proponemos un **refinamiento** de dicha función de Selección, al objeto de potenciar las siguientes pautas en la selección del sensor:

- Cubrir objetivos escasamente cubiertos (objetivos críticos).
- Cubrir un elevado número de objetivos no cubiertos.
- No cubrir objetivos redundantemente.

Para lograr la primera pauta se mantendrá el cálculo del conjunto de objetivos críticos: $t_{criticos}$, y de los sensores candidatos: $S_{candidatos}$, como en el algoritmo anterior, esto es: en cada iteración del procedimiento, $t_{criticos}$ es la colección de *actuales* objetivos críticos (seleccionados entre los objetivos todavía no cubiertos), y $S_{candidatos}$ el conjunto sensores cubriendo alguno de dichos objetivos críticos. Y para lograr las otras dos metas se propone una nueva función de selección que:

- *Incentive* la cobertura de objetivos no cubiertos incrementando el valor de $f(S_u)$ por cada objetivo no cubierto que S_u cubra.
- *Penalice* la cobertura redundante de objetivos disminuyendo el valor de $f(S_u)$ por cada objetivo ya cubierto que S_u cubra.

La Función de Selección se computa para cada uno de los sensores candidatos del conjunto $S_{candidatos}$, y como en el método anterior, el algoritmo selecciona el sensor S_u que maximice dicha Función de Selección. A continuación hacemos una descripción detallada de la misma.

Descripción de la Función de Selección

Considérese el conjunto de sensores candidatos: “ $S_{\text{candidatos}}$ ” obtenido en una iteración del procedimiento de construcción del Cubridor C_i .

- Inicializamos la Función de Selección, asignando a cada sensor $S_u \in S_{\text{candidatos}}$ el valor $f(S_u) = 0$.
- A continuación, para cada $S_u \in S_{\text{candidatos}}$ el algoritmo examina cada uno de los objetivos cubiertos por S_u , y por cada objetivo el algoritmo va modificando el valor de $f(S_u)$ sumando y/o restando determinadas cantidades que cuantifican el “impacto” que para la cobertura del sistema supone que dicho sensor se cubra por el candidato S_u .

Más concretamente, supongamos que S_u cubre los objetivos $t_u^{(1)}, \dots, t_u^{(h)}$. El algoritmo examina cada uno de dichos objetivos $t_u^{(j)}$, $j = 1, \dots, h$, y con cada objetivo $t_u^{(j)}$ la función se actualiza como sigue:

$$f(S_u) = \begin{cases} f(S_u) + M_1 - N_1, & \text{si } t_u^{(j)} \text{ no está cubierto por } C_i, \\ f(S_u) - M_2 + N_2, & \text{en caso contrario} \end{cases}, \quad \text{para } j = 1, \dots, h$$

Esto es, si S_u cubre un objetivo $t_u^{(j)}$ todavía no cubierto por el Cubridor C_i , el valor de $f(S_u)$ se modifica con una bonificación M_1 y una penalización N_1 , donde:

- $M_1 \rightarrow$ Número de objetivos no cubiertos que cubre S_u (se bonifica la cobertura de objetivos no cubiertos)
- $N_1 \rightarrow$ Número de sensores disponibles capaces de cubrir $t_u^{(j)}$ (y que incurrirán en redundancia si S_u es elegido).

Esto es, se potencia la elección de sensores que cubran un elevado número de objetivos no cubiertos, penalizándose el nivel de redundancia que dicha elección causará al sistema.

Por el contrario, si S_u cubre un objetivo $t_u^{(j)}$ ya cubierto por el cubridor C_i (lo que significa que si S_u es elegido cubrirá el objetivo redundantemente), el valor de $f(S_u)$ se modifica con una penalización M_2 y una bonificación N_2 , donde:

- $M_2 \rightarrow$ Número de objetivos ya cubiertos por C_i que también cubre S_u (penaliza el nivel de redundancia que aporta S_u si finalmente es elegido).
- $N_2 \rightarrow$ Número de sensores disponibles que también cubren $t_u^{(j)}$ (la anterior penalización se atenúa con la redundancia que introducirían los demás sensores redundantes que pudieran ser elegidos en lugar de S_u).

Esto es, aunque se penaliza la redundancia que presenta el candidato S_u , en la elección final dicha penalización se sitúa frente a la redundancia que también presentan otros potenciales candidatos.

Una vez calculados los valores de $f(S_u)$, para todo $S_u \in S_{candidatos}$, se selecciona el sensor S_u de valor máximo, esto es: $f(S_u) = \max \{f(S_w) : S_w \in S_{candidatos}\}$. En el caso de que varios sensores produzcan el mismo máximo, el algoritmo elige el primero de ellos en el orden en que aparece en los datos de entrada.

Complejidad

Al igual que en anterior algoritmo, en cada iteración del procedimiento CONSTRUYE_ C_i , la obtención de los objetivos críticos y el correspondiente conjunto de sensores candidatos $S_{candidatos}$ requiere $O(mn)$. Y para cada uno de los sensores candidatos, la evaluación de la Función de Selección tiene de complejidad $O(mn)$ (ciclo For), luego el cómputo de la Función de Selección para todos los sensores consume un tiempo de $O(mn^2)$. Luego la complejidad de cada Cubridor es $O(mn^3)$, consecuentemente, la complejidad total del algoritmo es $O(kmn^3)$, donde k es el número final de Cubridores.

A continuación se describe el pseudocódigo del algoritmo.

H₂: Algoritmo Heurístico con Penalización

```

INPUT   $S = \{S_1, \dots, S_n\}$  ( $n$  sensores),  $T = \{t_1, \dots, t_m\}$  ( $m$  objetivos)

 $i = 0$ ,  $Sens\_Disponibles = S$ 

While ( $Sens\_Disponibles$  cubra completamente  $T$ ) do
    Procedimiento CONSTRUYE_ $C_i$ 
         $i = i + 1$ 
         $C_i = \emptyset$ ,  $Obje\_Cubiertos = \emptyset$ 
        While ( $Obje\_Cubiertos \neq T$ ) do
            Calcula el conjunto  $t_{criticos} \subseteq T - Obje\_Cubiertos$ 
            Computa  $S_{candidatos} \subseteq Sens\_Disponibles$ 
            For (cada  $S_u \in S_{candidatos}$ ) do
                Procedimiento COMPUTA_ $f(S_u)$ 
                 $f(S_u) = 0$ 
                For (cada  $t_u^{(j)}$  cubierto por  $S_u$ ) do
                    If ( $t_u^{(j)}$  no está cubierto) then
                         $f(S_u) = f(S_u) + M_1 - N_1$ 
                    else  $f(S_u) = f(S_u) - M_2 + N_2$ 
                    end If
                end For
            end (Procedimiento COMPUTA_ $f(S_u)$ )
        end For
        Selecciona  $S_u$  tal que:  $f(S_u) = \max \{f(S_w) : S_w \in S_{candidatos}\}$ 
         $C_i = C_i \cup S_u$ 
        Actualiza ( $Sens\_Disponibles$ ,  $Obje\_Cubiertos$ ):
    end While
end (Procedimiento CONSTRUYE_ $C_i$ )
end While

RETURN  $i$  (número de Conjuntos Cubridores), y los Cubridores  $C_1, C_2, \dots, C_i$ .
    
```

Aplicación del Algoritmo H₂: Un Ejemplo.

Al igual que antes ilustraremos el funcionamiento del algoritmo con el mismo ejemplo utilizado en el algoritmo anterior, y cuya descripción viene dada por la tabla:

	t_1	t_2	t_3	t_4	t_5
S_1	1	1	1	1	
S_2	1	1			1
S_3		1	1	1	1
S_4		1	1		1
S_5	1		1		1
S_6			1	1	1

- **Construcción de C_1 :** Comenzamos con $Sens_Disponibles = S$

Primera Iteración: $t_{critico} = \{t_1, t_4\}$, $S_{candidatos} = \{S_1, S_2, S_3, S_5, S_6\}$.

(nótese que al principio todos los objetivos son “No Cubiertos”).

S_1	t_1	t_2	t_3	t_4
M_1	4	4	4	4
N_1	2	3	4	2
$M_1 - N_1$	2	1	0	2

S_2	t_1	t_2	t_5
M_1	3	3	3
N_1	2	3	4
$M_1 - N_1$	1	0	-1

$$f(S_1) = 2 + 1 + 0 + 2 = 5$$

$$f(S_2) = 0$$

De igual forma obtenemos $f(S_3) = 3$, $f(S_5) = -1$, $f(S_6) = -1$.

Luego Selecciona $S_1 \rightarrow C_1 = \{S_1\}$.

Segunda Iteración: $Obje_Cubiertos = \{t_1, t_2, t_3, t_4\}$, $t_{critico} = \{t_5\}$,

$Sens_Disponibles = \{S_2, S_3, S_4, S_5, S_6\}$, $S_{candidatos} = \{S_2, S_3, S_4, S_5, S_6\}$.

S_2	t_1	t_2	t_5
M_1			1
N_1			4
M_2	2	2	
N_2	1	2	
$(M_1 - N_1) + (-M_2 + N_2)$	-1	0	-3

$$f(S_2) = -4$$

S_3	t_2	t_3	t_4	t_5
M_1				1
N_1				4
M_2	3	3	3	
N_2	2	3	1	
$(M_1 - N_1) + (-M_2 + N_2)$	-1	0	-2	-3

$$f(S_3) = -6$$

Igualmente, obtenemos: $f(S_4) = -2$, $f(S_5) = -3$, $f(S_6) = -3$

Selecciona $S_4 \rightarrow C_1 = \{S_1, S_4\}$.

Como $Obje_Cubiertos = T$, se procede a construir el siguiente Cubridor.

- **Construcción de C_2** , siendo $Sens_Disponibles = \{S_2, S_3, S_5, S_6\}$

	t_1	t_2	t_3	t_4	t_5
S_2	1	1			1
S_3		1	1	1	1
S_5	1		1		1
S_6			1	1	1

Primera Iteración: $t_{critico} = \{t_1, t_2, t_4\}$, $S_{candidatos} = \{S_2, S_3, S_5, S_6\}$.

(ahora todos los Objetivos son No Cubiertos).

S_2	t_1	t_2	t_5
M_1	3	3	3
N_1	1	1	3
$M_1 - N_1$	2	2	0

$$f(S_2) = 4$$

S_3	t_2	t_3	t_4	t_5
M_1	4	4	4	4
N_1	1	2	1	3
$M_1 - N_1$	3	2	3	1

$$f(S_3) = 9$$

Análogamente se obtiene: $f(S_5) = 3$, $f(S_6) = 3$

Luego Selecciona $S_3 \rightarrow C_2 = \{S_3\}$.

Segunda Iteración: $Obje_Cubiertos = \{t_2, t_3, t_4, t_5\}$, $t_{critico} = \{t_1\}$,

$$Sens_Disponibles = \{S_2, S_5, S_6\}, \quad S_{candidatos} = \{S_2, S_5\}.$$

S_2	t_1	t_2	t_5
M_1	1		
N_1	1		
M_2		2	2
N_2		0	2
$(M_1 - N_1) + (-M_2 + N_2)$	0	-2	0

$$f(S_2) = -2$$

S_5	t_1	t_3	t_5
M_1	1		
N_1	1		
M_2		2	2
N_2		1	2
$(M_1 - N_1) + (-M_2 + N_2)$	0	-1	0

$$f(S_5) = -1$$

Por tanto, Selecciona $S_5 \rightarrow C_2 = \{S_3, S_5\}$

Como $Obje_Cubiertos = T$, se procede a construir el siguiente Cubridor.

- **Construcción de C_3** , siendo $Sens_Disponibles = \{S_2, S_6\}$

No se detalla la construcción de C_3 ya que, claramente, $C_3 = \{S_2, S_6\}$.

6.4.3 Análisis Comparativo: Experiencia Computacional

Como se puede observar, la aplicación del Algoritmo con Penalización ha producido una solución óptima, ya que se ha alcanzado la Cota superior ($\text{Max_Teor} = 3 = \text{Número de Cubridores}$), lo que supone una mejora sobre los resultados del Algoritmo Greedy aplicado al mismo ejemplo (donde se obtuvieron 2 Cubridores). Asimismo, un primer análisis comparativo sobre los siguientes ejemplos parece confirmar este mejor comportamiento del segundo heurístico respecto del primero. Esto se puede comprobar en los ejemplos que se muestran a continuación, donde como ya se ha detallado, las matrices binarias $(a_{ij})_{n \times m}$ (Matrices de Cobertura) representan los datos de entrada en términos de “Cobertura de los sensores”, esto es,

$$(a_{ij})_{n \times m}, \text{ con } a_{ij} = 1 \text{ si y sólo si } S_i \text{ cubre } t_j$$

Los siguientes tres ejemplos consideran $n = 10$ sensores y $m = 17$ objetivos, y las matrices de cobertura de entrada han sido aleatoriamente generadas. Las soluciones obtenidas muestran el mejor comportamiento del segundo algoritmo en la mayoría de los casos, aunque también parecen sugerir que cuando la cobertura de los objetivos es muy elevada, ambos algoritmos producen soluciones muy similares.

Ejemplo 1:

Input: Matriz de Cobertura: ($n = 10$, $m = 17$)

1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1
1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	0	1	1	0	1	1	1	0	0	1	1	1	1	1	0	1
1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
1	0	1	0	1	1	0	1	1	1	1	1	1	0	1	1	1

Máximo Teórico = 6,

Heurístico H₁: Número de Cubridores = 4

$$C_1 = \{S_1, S_4\}, C_2 = \{S_9, S_3\}, C_3 = \{S_5, S_2\}, C_4 = \{S_7, S_6\}$$

Heurístico H₂: Número de Cubridores = 5

$$\text{Cubridores: } C_1 = \{S_4, S_6\}, C_2 = \{S_9, S_{10}\}, C_3 = \{S_1, S_8\}, C_4 = \{S_3, S_5\}, C_5 = \{S_2, S_7\}$$

Ejemplo 2:

Matriz de Cobertura: ($n = 10, m = 17$)

1	0	1	1	0	0	1	0	0	1	0	1	1	0	1	0	0
1	1	0	1	1	1	0	1	1	0	1	1	0	1	1	1	1
1	1	0	1	0	1	1	1	1	1	1	1	0	0	1	1	1
1	1	0	0	0	0	0	0	1	0	1	1	1	1	1	1	1
1	1	1	0	1	1	0	0	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	0	0	0	0	1	1	1	0	1	0	1
1	1	0	0	0	1	0	0	1	1	1	1	1	1	0	1	0
1	0	0	0	0	1	0	0	1	1	1	1	1	0	0	1	1
1	0	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1
1	1	0	0	0	1	1	1	1	1	1	0	0	1	1	1	0

Máximo Teórico = 3,

Heurístico H₁: Número de Cubridores = 2

$$\text{Cubridores: } C_1 = \{S_2, S_3, S_5\}, C_2 = \{S_6, S_{10}\}$$

Heurístico H₂: Número de Cubridores = 3

$$\text{Cubridores: } C_1 = \{S_2, S_1\}, C_2 = \{S_3, S_5\}, C_3 = \{S_6, S_{10}\}$$

Ejemplo 3:

Matriz de Cobertura: ($n = 10, m = 17$)

1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Máximo Teórico = 9,

Heurístico H₁: Número de Cubridores = 7,

$$C_1 = \{S_3\}, C_2 = \{S_4\}, C_3 = \{S_5\}, C_4 = \{S_6\}, C_5 = \{S_{10}\}, C_6 = \{S_2, S_7\}, C_7 = \{S_1, S_8\}$$

Heurístico H₂: Número de Cubridores = 7,

$$C_1 = \{S_3\}, C_2 = \{S_4\}, C_3 = \{S_5\}, C_4 = \{S_6\}, C_5 = \{S_{10}\}, C_6 = \{S_2, S_8\}, C_7 = \{S_1, S_7\}$$

Como se puede ver, en este último ejemplo ambos algoritmos han alcanzado el mismo número de cubridores (7).

Esta dinámica también se mantiene si invertimos la relación anterior entre el número de sensores y el de objetivos: en el siguiente ejemplo consideraremos que el número de sensores es mayor que el de objetivos: concretamente, $n = 20$ sensores y $m = 15$ objetivos:

Ejemplo 4:

Matriz de Cobertura: ($n = 20$, $m = 15$)

1	1	1	1	0	1	0	1	0	1	1	1	1	0	1
1	1	1	1	0	1	1	1	1	1	1	0	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0	1
1	0	0	1	1	0	0	1	1	0	1	1	1	1	1
1	1	0	1	0	1	0	1	0	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	0	1	1	1	1	1
1	1	1	1	1	1	0	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	0	1	1	0	1	1	1	0	1
1	1	0	1	1	1	0	1	1	1	1	0	1	1	1
0	1	1	1	0	0	1	1	0	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	0	1	0	1
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1
0	0	0	1	1	1	1	1	0	1	1	0	1	0	1
1	0	1	1	1	0	1	0	0	0	1	0	1	0	1
1	0	1	1	1	1	0	1	0	1	0	1	1	0	1

Máximo Teórico = 11,

Heurístico H₁: Número de Cubridores = 7,

$$C_1 = \{S_3, S_{11}\}, \quad C_2 = \{S_7, S_{12}\}, \quad C_3 = \{S_{17}, S_2\}, \quad C_4 = \{S_8, S_{10}\}, \quad C_5 = \{S_{14}, S_4\}, \\ C_6 = \{S_1, S_6, S_{16}\}, \quad C_7 = \{S_9, S_{15}\}$$

Heurístico H₂: Número de Cubridores = 9,

$$C_1 = \{S_{11}, S_{13}\}, \quad C_2 = \{S_3, S_{19}\}, \quad C_3 = \{S_7, S_5\}, \quad C_4 = \{S_{12}, S_{15}\}, \quad C_5 = \{S_{17}, S_9\}, \\ C_6 = \{S_8, S_6\}, \quad C_7 = \{S_2, S_{20}\}, \quad C_8 = \{S_{10}, S_4\}, \quad C_9 = \{S_4, S_{16}, S_1\}$$

Para analizar si esta tendencia se mantiene en términos estadísticos realizaremos un estudio comparativo del comportamiento de ambos algoritmos evaluando las soluciones que producen sobre un conjunto de Matrices de Cobertura aleatoriamente generadas, para poder disponer de una base experimental que avale la anterior conjetura.

Experiencia Computacional: Resultados.

Para realizar la comparación, hemos considerado distintos escenarios, y sobre cada uno de ellos se han ejecutado ambos algoritmos.

Los parámetros de entrada de cada escenario son: n (número de sensores) y m (número de objetivos). Sobre ellos el algoritmo genera aleatoriamente una matriz binaria: la Matriz de Cobertura, con la única restricción de que no exista una columna de ceros (para garantizar que cada objetivo esté cubierto por al menos un sensor), esto es:

$$(a_{ij})_{n \times m}, \text{ tal que } a_{ij} = \begin{cases} 1, & \text{si } S_i \text{ cubre a } t_j \\ 0, & \text{caso contrario} \end{cases}, \text{ tal que } \sum_{i=1}^n a_{ij} \geq 1, \text{ para cada } j = 1, \dots, m$$

A) Primer escenario: $n = 50$ (fijo), y m variando de 10 a 100 con paso de 10.

En este escenario el objetivo era estudiar el comportamiento de los algoritmos cuando el número de objetivos variaba en un cierto entorno del número de Sensores. Concretamente, seleccionando los m objetivos en un intervalo centrado en $n = 50$.

Por ello, cada caso se caracteriza por sus parámetros de entrada (n, m) , donde $n = 50$ Sensores y $m = 10, 20, 30, \dots, 100$ objetivos. Para cada pareja $(50, m)$ hemos ejecutado los dos algoritmos 20 veces (generándose cada vez una nueva matriz). Sobre los 20 resultados finales, hemos calculado las *Medias del número de Cubridores* producidos por cada algoritmo, que son las que figuran en la Tabla 6.1.

<i>Número de Objetivos</i>										
$(n = 50)$	10	20	30	40	50	60	70	80	90	100
Medias H_1	26,9	20,4	15,5	18,3	14,2	11,7	11,9	18,5	10,7	9,2
Medias H_2	27,6	21,3	16,1	19,2	16,6	13,1	14,6	20,3	12,0	11,9
Medias Max_Teor	32,0	29,8	23,2	27,9	26,3	21,2	24,1	31,7	20,8	19,1

Tabla 6.1: Número Medio de Cubridores, y valor medio del Máximo Teórico en 20 casos para cada par (n, m) , con $n = 50$ sensores y $m = 10, 20, \dots, 100$ objetivos

B) Segundo escenario: Evolución de los resultados para un gran incremento del número de objetivos sobre el número de sensores.

Aquí intentamos observar el comportamiento de los resultados cuando el número de objetivos sufre un gran aumento sobre el número de sensores. Por ello, nuevamente hemos considerado $n = 50$ sensores, y $m = 200, 300, 400, 500, 800, 1000$ objetivos a la hora de generar las matrices. Al igual que antes, para cada par de entrada (n, m) se han generado 20 casos, ejecutándose los algoritmos sobre cada caso, obteniéndose de cada uno de ellos una lista de 20 soluciones en cada caso. En la Tabla 6.2 figuran las medias del Número de Cubridores obtenidos en cada caso, así como el valor medio del Máximo Teórico.

<i>Número de Objetivos</i>						
$(n = 50)$	200	300	400	500	800	1000
Medias H_1	8,8	10,1	10,4	10,8	10,7	10,4
Medias H_2	12,1	12,3	13,8	12,2	13,5	12,6
Medias Max_Teor	22,7	23,8	27,8	24,9	27,6	23,4

Tabla 6.2: Número Medio de Cubridores, y valor medio del Máximo Teórico para cada par (n, m) , con $n = 50$ sensores, y $m = 200, 300, 400, 500, 800, 1000$ objetivos

C) Tercer escenario: Evolución de los resultados para un gran incremento del número de sensores sobre el número de objetivos.

Aquí vamos a considerar la relación inversa entre sensores y algoritmos. Concretamente, para un número fijo de $m = 50$ objetivos, estudiaremos cómo se comportan los algoritmos cuando el número de sensores se incrementa, y toma los valores $n = 50, 100, 200, 400, 500, 800, 1000$. La generación de las matrices, y el número de 20 ejemplares considerados de cada caso siguen la misma dinámica que en los escenarios anteriores. En la Tabla 6.3 se muestran los resultados.

<i>Número de Sensores</i>								
$(m = 50)$	50	100	200	300	400	500	800	1000
Medias H_1	14,2	34,0	71,6	114,8	175,2	181,3	287,6	224,4
Medias H_2	16,6	34,4	73,8	116,9	178,8	185,2	289,7	234,4
Medias Max_Teor	26,3	41,6	88,1	140,8	211,2	226,4	353,5	312,0

Tabla 6.3: Número Medio de Cubridores, y valor medio del Máximo Teórico para cada par (n, m) , con $n = 50, 100, 200, 300, 400, 500, 800, 1000$ sensores, y $m = 50$ objetivos

Las gráficas de las anteriores simulaciones se muestran a continuación:

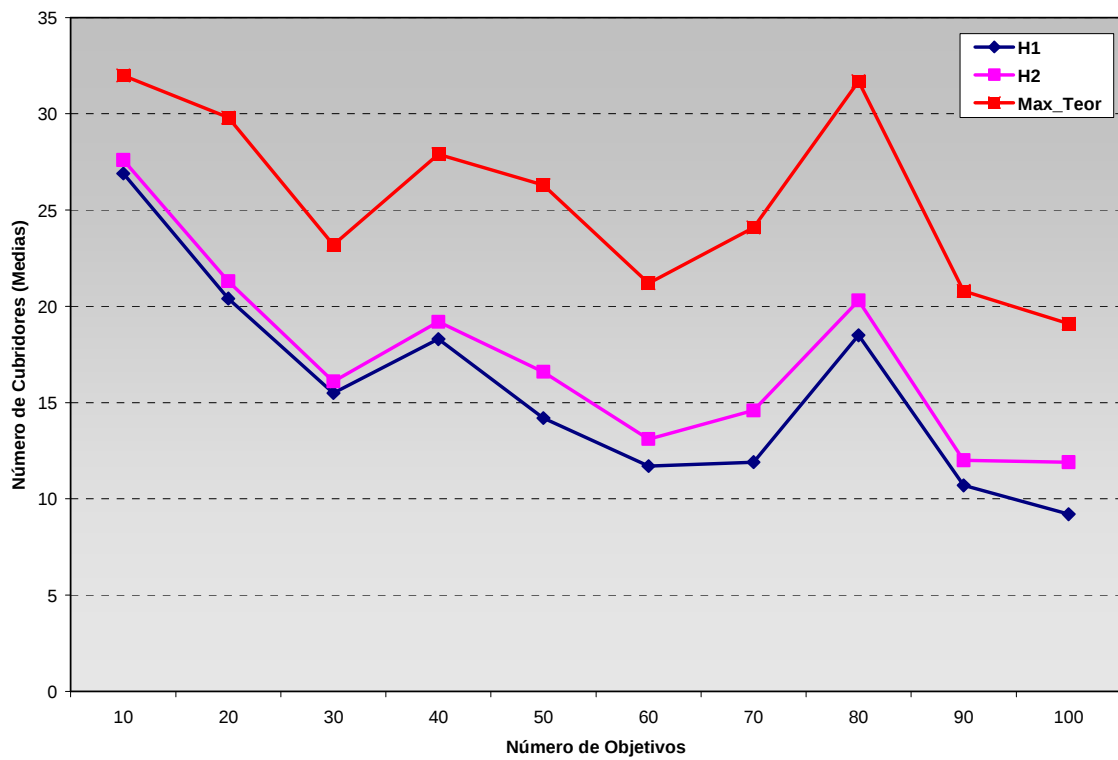


Figura 6.5: Escenario 1, con $n = 50$ sensores y $m = 10, 20, \dots, 100$ objetivos

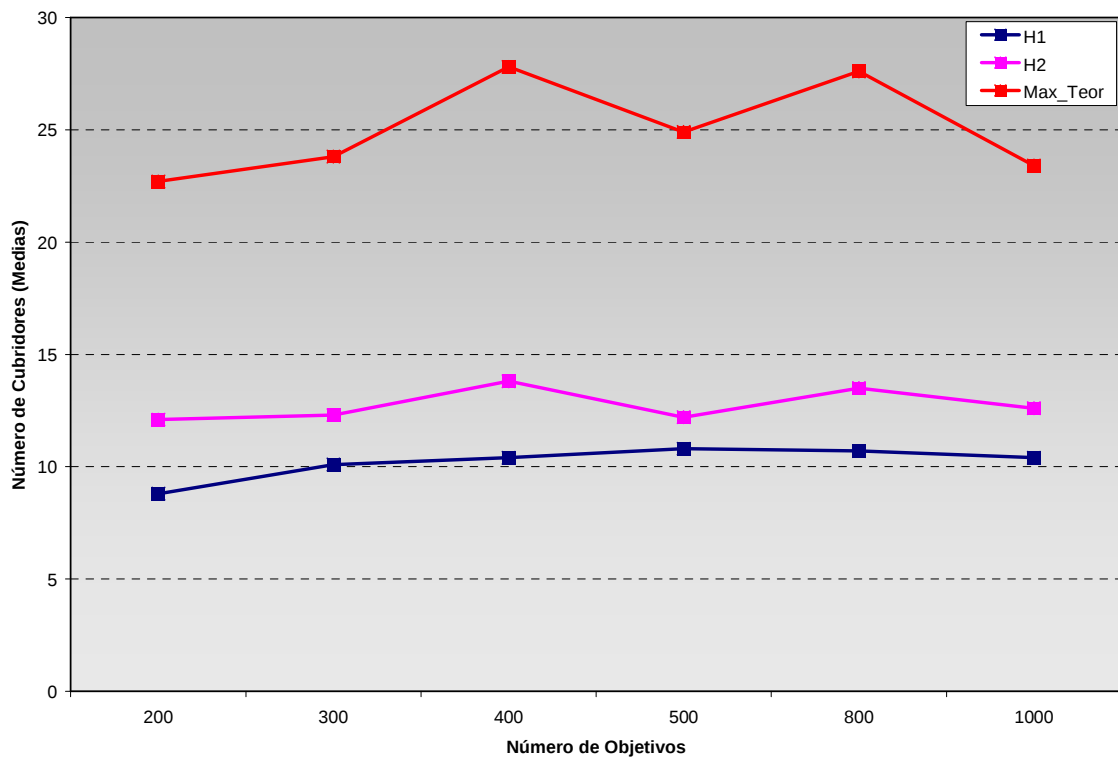


Figura 6.6: Escenario 2, con $n = 50$ sensores y $m = 200, 300, \dots, 1000$ objetivos

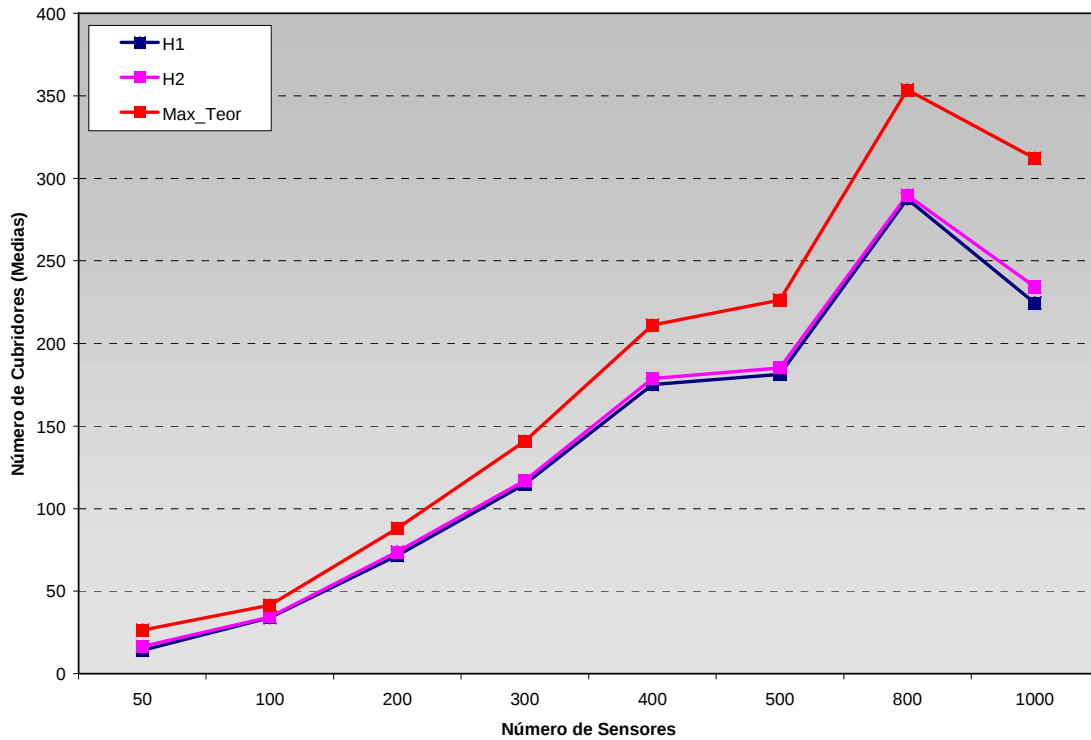


Figura 6.7: Escenario 3, con $m = 50$ objetivos y $n = 50, 100, 200, \dots, 1000$ sensores

De los datos recogidos en la experiencia computacional y del examen de las correspondientes gráficas, se puede deducir que:

- El segundo algoritmo Heurístico H_2 exhibe un mejor comportamiento que el primer Heurístico H_1 en todos los escenarios considerados y para todos los parámetros de entrada, ya que, en términos estadísticos, las soluciones del segundo algoritmo presentan una mejor aproximación al óptimo que las soluciones del primer algoritmo.
- Esta mayor eficiencia de H_2 destaca en todas las situaciones, especialmente ante un aumento considerable del número de objetivos respecto del número de Sensores (Escenario 2), y la eficiencia disminuye cuando el número de objetivos es muy pequeño respecto al número de sensores, donde las soluciones proporcionadas por ambos algoritmos tienen a ser muy parecidas (Escenario 1).

- El incremento del número de Objetivos (Escenario 2) parece repercutir en un aumento de la mejora del segundo algoritmo H_2 respecto del algoritmo Greedy H_1 . Pero también se observa que la diferencia entre ambos heurísticos parece estabilizarse para valores muy elevados del número de objetivos, como se visualiza en la gráfica de dicho escenario.
- Por el contrario, cuando analizamos el efecto que sobre los resultados tiene un incremento excesivo del número de sensores (Escenario 3), la correspondiente gráfica muestra un comportamiento muy similar de ambos heurísticos, lo que permite deducir que la diferencia en la efectividad de ambos heurísticos disminuye ante un gran aumento del número de sensores respecto del número de objetivos.
- Por último, se observa que el “Máximo Teórico” se aleja de los resultados al aumentar el número de objetivos y también al aumentar el número de sensores (Escenario 3). Si bien dicho máximo es una cota superior y puede no ser alcanzable, este último hecho indica que redes con una elevada densidad de sensores pueden resultar poco efectivas (además de costosas), ya que el aumento de la distancia al Máximo Teórico puede también suponer un aumento del número de sensores infrautilizados en la red. Por ello la eficiencia óptima no está necesariamente asociada con un aumento ilimitado del número de sensores, sino que requiere un análisis previo del problema para decidir la mejor estrategia asociada a cada escenario de despliegue.

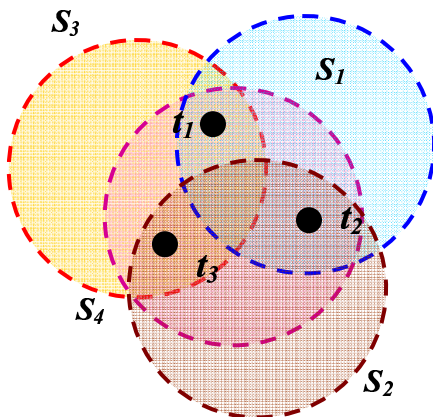
6.5

Maximización del tiempo de Cobertura mediante Cubridores no disjuntos

Hasta ahora hemos abordado el problema de prolongar la vida de la red ahorrando energía mediante la distribución de los sensores en conjuntos Cubridores **disjuntos**, esto es, en conjuntos de sensores excluyentes entre sí capaces de monitorizar completamente todos los objetivos del a cubrir. En dicho esquema el funcionamiento de los sensores es exhaustivo, esto es: en cada conjunto Cubridor los sensores funcionan ininterrumpidamente hasta consumir todo su tiempo de vida.

Sin embargo, un planteamiento alternativo consiste en limitar el tiempo de funcionamiento de cada Cubridor, o equivalentemente, permitir que los conjuntos Cubridores sean **no disjuntos**. Si en cada Cubridor los sensores sólo funcionan durante un porcentaje q de su tiempo total, entonces un mismo sensor puede participar en varios conjuntos Cubridores (ya que en cada uno de ellos sólo gasta una fracción q de su energía total), y esto puede redundar en alargar la vida de la red más que en el esquema de Cubridores disjuntos. Ilustraremos esta idea con un ejemplo.

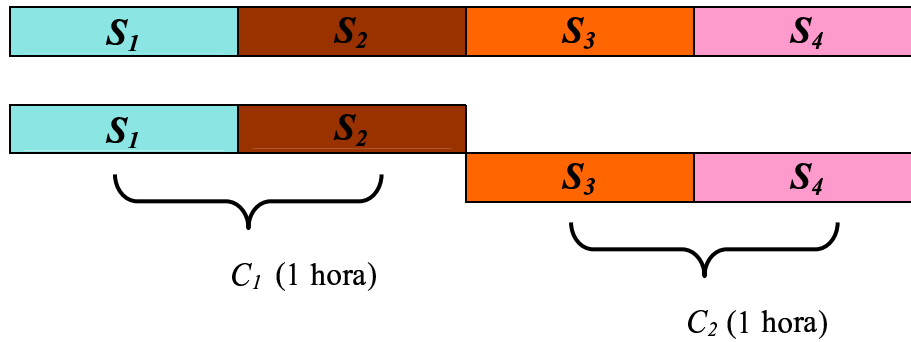
Ejemplo: Supongamos un conjunto de 4 sensores: $S = \{S_1, S_2, S_3, S_4\}$ que cubren 3 objetivos $T = \{t_1, t_2, t_3\}$ de la siguiente forma:



	t_1	t_2	t_3
S_1	1	1	
S_2		1	1
S_3	1		1
S_4	1	1	1

Supongamos que todos los sensores tienen energía homogénea, esto es, funcionan ininterrumpidamente durante un tiempo de b horas. Sin pérdida de generalidad podemos asumir $b = 1$ hora (en el sentido de considerar b como la unidad de tiempo total de funcionamiento del sensor).

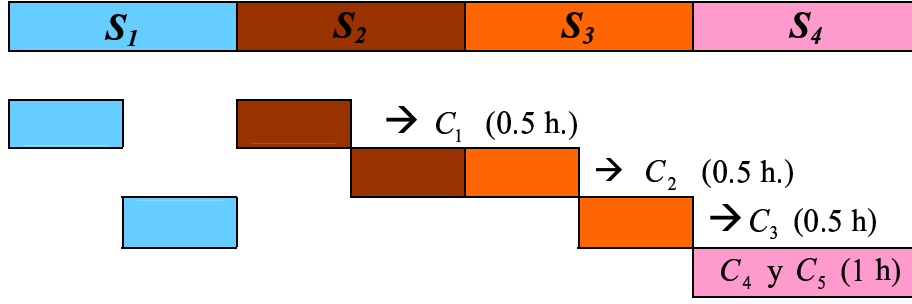
- **Cubridores disjuntos:** en este esquema, cada Cubridor funciona ininterrumpidamente durante todo el tiempo posible, esto es, durante $b = 1$ hora. Obtendríamos un máximo de 2 conjuntos Cubridores: $C_1 = \{S_1, S_2\}$, $C_2 = \{S_3, S_4\}$ (aunque esta combinación no es única), trabajando durante 2 horas.



Tiempo de Cobertura total: 2 horas

- **Cubridores No disjuntos:** Supongamos ahora que permitimos que cada sensor forme parte de más de un conjunto Cubridor, y que los Cubridores se activen por períodos de tiempo inferiores a la vida del sensor. Con estas premisas se podrán construir Cubridores no disjuntos, que con una adecuada elección de los intervalos de tiempo, pueden prolongar el tiempo de vida de la red. En el ejemplo anterior, esto podría lograrse con los siguientes Cubridores:
 $C_1 = \{S_1, S_2\}$, $C_2 = \{S_2, S_3\}$, y $C_3 = \{S_1, S_3\}$, cada uno por 0.5 horas
 $C_4 = \{S_4\}$, $C_5 = \{S_4\}$, durante 0.5 horas cada uno (en realidad supone activar el sensor S_4 durante 1 hora, como si hubiera un solo Cubridor C_4 en dicho tiempo).

El tiempo de Cobertura total conseguido de esta manera es de 2.5 horas.



Tiempo de Cobertura total: 2.5 horas

Vemos en este ejemplo cómo el permitir Cubridores no disjuntos ha prolongado el tiempo de Cobertura de la red un 25% más que el tiempo obtenido para los Cubridores disjuntos.

Claramente, si un sensor S_j participa en h conjuntos Cubridores $C_1, \dots, C_h$, de tiempos respectivos de vida $c_1, \dots, c_h$, la suma de tales tiempos no puede superar el tiempo total de vida del sensor, esto es:

$$\sum_{i=1}^h c_i \leq b_j = (\text{tiempo de vida del Sensor } S_j)$$

(que en principio estamos igual para todos los sensores, esto es, $b_j = b$, aunque más adelante se podrá generalizar fácilmente).

El problema de optimizar el tiempo de Cobertura de la red permitiendo Cubridores no disjuntos es una generalización del anterior problema, por tanto sigue siendo NP-duro y requiere procedimientos heurísticos para proporcionar soluciones aproximadas.

El heurístico que proponemos parte del Algoritmo Heurístico H_1 (Greedy) ya descrito en el apartado anterior, en el cual introduciremos una modificación en el cálculo de los objetivos críticos: t_{criticos} , y en la consiguiente Función de Selección que incorpore, en la elección del sensor $S_u \in S_{\text{candidatos}}$, los siguientes criterios

- El número de objetivos no cubiertos.
- El tiempo residual de vida de cada sensor.

6.5.1 Algoritmo Heurístico $H(q)$

Veremos primero el caso de tiempo de vida homogéneo para todos los sensores, esto es, inicialmente, $b_j = b$, para todo sensor S_j . Sin pérdida de generalidad podemos suponer $b=1$, (ya que los resultados finales se rehacen multiplicándolos por b), pero al objeto de facilitar la generalización a tiempos distintos, mantendremos el valor “ b ” en la explicación del heurístico. Además, consideraremos los siguientes elementos:

- Tasa de participación: Introduciremos un parámetro q : *tasa de participación*, con $0 < q \leq 1$, que representará la fracción de tiempo que un sensor “gasta” en un Cubridor, de forma que el tiempo en el que el sensor S_j participa en un Cubridor está dado por: qb . Esto es, si por ejemplo $b = 5$ horas, la tasa $q = 0.2$ indica que, cada vez que el sensor S_j participa en un conjunto Cubridor, consume un 20% ($1/5$) de su tiempo, en este caso, $qb = 1$ hora, o equivalentemente, que S_j puede participar en un máximo de 5 Cubridores. Nótese que el caso $q = 1$ significa que cada sensor puede participar como mucho en un Cubridor, y por tanto los Cubridores que se formarán serán disjuntos. En otras palabras, la formación de Cubridores disjuntos se puede considerar como un caso particular de éste, para $q = 1$.
- Tiempo residual: Cada vez que sensor S_j participa en un Cubridor, gasta parte de su tiempo, y le queda un *tiempo residual* r_j . Inicialmente, $r_j = b$, para todos los sensores. Cuando el sensor S_j se incorpora a un Cubridor, dicho tiempo residual disminuye, y hay que actualizarlo de la forma: $r_j = r_j - qb$.
- Actualización del conjunto $S_{disponibles}$ (sensores disponibles): Cuando el tiempo residual de un sensor es nulo (sensor agotado), o bien es inferior a su tiempo de participación, se elimina de los sensores disponibles. Esto es, cuando $r_j < qb$.

Al igual que en el algoritmo H_1 , también este algoritmo trabaja según una estrategia constructiva Greedy, que va construyendo los Cubridores recursivamente,

hasta que no quedan sensores disponibles cubriendo los objetivos. Básicamente, y siguiendo dicho esquema:

Algoritmo Heurístico H(q) para Cubridores no disjuntos

```

INPUT   $S = \{S_1, \dots, S_n\}$  ( $n$  sensores),  $T = \{t_1, \dots, t_m\}$  ( $m$  objetivos),
        (Matriz de Cobertura  $(a_{ij})_{n \times m}$ ).

Inicializar tiempo residual  $r_j = b$ ,  $j = 1, \dots, n$ 

Fijar  $q \in (0, 1]$ 

Sens_Disponibles =  $S$ 

While (Sens_Disponibles cubra completamente  $T$ ) do
    Procedimiento CONSTRUYE_  $C_i$ 

         $C_i = \emptyset$ ,  $Obje\_Cubiertos = \emptyset$ 

        While ( $Obje\_Cubiertos \neq T$ , esto es,  $C_i$  no cubra  $T$ ) do

            Calcula los conjuntos  $t_{criticos}$ ,  $S_{candidatos}$ 

            Selecciona  $S_u \in S_{candidatos}$  según Criterio de Selección

             $C_i = C_i \cup S_u$ 

            Actualiza ( $Obje\_Cubiertos$ )

        end While

    end (Procedimiento CONSTRUYE_  $C_i$ )

    For ( todos los sensores  $S_j \in C_i$ ) do

        Actualiza tiempo residual:  $r_j = r_j - qb$ 

        If ( $r_j < qb$ , esto es,  $S_j$  incapaz de participar) then

            Actualiza Sens_Disponibles, eliminando  $S_j$ 

        end If

    end For

end While
    
```

A continuación explicamos más detalladamente el procedimiento CONSTRUYE_ C_i , concretamente nos centramos en la elección de los objetivos críticos, del conjunto de Sensores candidatos y la descripción de la Función de Selección.

- Calculamos $t_{criticos} \subseteq T-Obje_Cubiertos$ como sigue: seleccionamos, de entre los objetivos todavía no cubiertos por C_i , aquellos objetivos:
 - Más escasamente cubiertos en número de Sensores.
 - Más escasamente cubiertos en tiempo residual: presentan el menor valor de la suma de los tiempos residuales de los sensores que lo cubren
- Computamos $S_{candidatos} \subseteq Sens_Disponibles$ seleccionando como candidatos aquellos sensores disponibles cubriendo algún objetivo crítico del conjunto " $t_{criticos}$ ".
- Como Función de Selección establecemos la siguiente: Para cada sensor $S_j \in S_{candidatos}$, calculamos
$$f(S_u) = M_1 + R_j, \text{ donde:}$$
 - $M_1 \rightarrow$ Número de Objetivos no cubiertos que cubre S_j .
 - $R_j \rightarrow$ Número de Cubridores en los que todavía puede participar S_j .

Hay que tener en cuenta que, a partir del tiempo residual r_j de S_j , y teniendo en cuenta que cada Cubridor "consume" un tiempo de qb , se puede calcular R_j

como: $R_j = \left\lfloor \frac{r_j}{qb} \right\rfloor$ (parte entera por defecto de dicho cociente).

- Por último, el Criterio de Selección elige aquél sensor candidato que maximiza la Función de Selección.

Con todo lo anterior, el algoritmo queda como sigue:

Algoritmo Heurístico H(q) para Cubridores no disjuntos

```

INPUT      Matriz de Cobertura  $(a_{ij})_{n \times m}$ .
Inicializar  $r_j = b, (j = 1, \dots, n), q \in (0, 1], Sens\_Disponibles = S$ 

While ( $Sens\_Disponibles$  cubra completamente  $T$ ) do
    Procedimiento CONSTRUYE_  $C_i$ 
         $C_i = \emptyset, Obje\_Cubiertos = \emptyset$ 
        While ( $Obje\_Cubiertos \neq T$ , esto es,  $C_i$  no cubra  $T$ ) do
            Calcula el conjunto  $t_{criticos} \subseteq T - Obje\_Cubiertos$ 
                (objetivos menos cubiertos en número de sensores
                 y en suma de tiempos residuales)
            Calcula  $S_{candidatos} \subseteq S_{disponibles}$  (sensores cubriendo
                algún objetivo crítico)
            For (cada  $S_j \in S_{candidatos}$ ) do
                 $f(S_j) = M_1 + R_j$ 
            end For
            Selecciona  $S_u \in S_{candidatos}$  tal que
                 $f(S_u) = \max \{f(S_w) : S_w \in S_{candidatos}\}$ 
             $C_i = C_i \cup S_u$ 
            Actualiza ( $Obje\_Cubiertos$ )
        end While
    end (Procedimiento CONSTRUYE_  $C_i$ )
    For ( todos los sensores  $S_j \in C_i$ ) do
        Actualiza tiempo residual:  $r_j = r_j - qb$ 
        If ( $r_j < qb$ , esto es,  $S_j$  incapaz de participar) then
            Actualiza  $Sens\_Disponibles$ , eliminando  $S_j$ 
        end If
    end For
end While
    
```

Complejidad

Si la complejidad del algoritmo H_1 (Greedy) para Cubridores disjuntos era $O(k m n^2)$ (siendo k el número de Cubridores disjuntos), la complejidad de este algoritmo se incrementará en función de la tasa q . Así, y considerando $b=1$ como el tiempo unitario de vida de todos los sensores, cada sensor puede (potencialmente) ser “elegido” para intervenir en $\left\lfloor \frac{1}{q} \right\rfloor$ Cubridores no disjuntos, o equivalentemente, el procedimiento “CONSTRUYE_ C_i ” dispone de $\left\lfloor \frac{n}{q} \right\rfloor$ potenciales “sensores a tiempo parcial” para elegir, lo que incrementa la complejidad de cada Cubridor C_i a $O\left(\left\lfloor \frac{n}{q} \right\rfloor m n\right)$. Por tanto, la complejidad final de el anterior Heurístico $H(q)$ viene dada por $O\left(k \left\lfloor \frac{n}{q} \right\rfloor m n\right)$ ($O\left(k \left\lfloor \frac{nb}{q} \right\rfloor m n\right)$ para un valor general de b), donde k es el número final de Cubridores construidos. Esto supone que para valores muy pequeños de q (como $q = 10^{-2}$, o inferiores) se incrementará sensiblemente el tiempo de computación.

Ejemplo:

A continuación ponemos un ejemplo con 8 Sensores y 6 Objetivos, dado por la Matriz de Cobertura:

1	1	1	0	0	1
1	1	1	0	0	1
1	1	1	1	0	1
1	1	1	1	0	1
0	1	1	0	1	1
1	1	0	1	0	1
1	1	0	1	1	0
1	1	1	0	1	1

Y suponemos que todos los sensores tienen el mismo tiempo de vida $b=1$. Veamos los tiempos de cobertura que se obtienen al aplicar el Método de Obtención de Cubridores disjuntos mediante el Algoritmo H_2 (de Penalización de la Redundancia,

que era el que mejores resultados producía), y el Método de obtención de Cubridores no disjuntos según el anterior algoritmo, para distintos valores de la tasa q .

- **Cubridores Disjuntos: Algoritmo H₂ (Penalización).**

Número de Cubridores disjuntos: 2 ($C_1 = \{S_7, S_5\}$, $C_2 = \{S_8, S_6\}$).

Tiempo de Cobertura: 2.

- **Cubridores No Disjuntos: Algoritmo H(q).**

$$q = 0.1 \begin{cases} \text{Tiempo de Cobertura} = 3 \\ \text{Número de Cubridores} = 30 \end{cases}$$

Los Cubridores son (“ i ” representa al sensor S_i):

$\{8,3\}, \{5,4\}, \{7,1\}, \{8,6\}, \{5,3\}, \{7,2\}, \{8,4\}, \{5,6\}, \{7,1\}, \{8,3\}, \{5,4\}, \{7,2\}, \{8,6\}, \{5,3\},$
 $\{7,1\}, \{8,4\}, \{5,6\}, \{7,2\}, \{8,3\}, \{5,4\}, \{7,1\}, \{8,6\}, \{5,3\}, \{7,2\}, \{8,4\}, \{5,6\}, \{7,1\}, \{8,3\},$
 $\{5,4\}, \{7,2\}$

$$q = 0.5 \begin{cases} \text{Tiempo de Cobertura} = 3 \\ \text{Número de Cubridores} = 6 \end{cases}$$

Los Cubridores son: $\{8,3\}, \{5,4\}, \{7,1\}, \{8,6\}, \{5,3\}, \{7,2\}$.

Vemos que el factor $q = 0.5$ produce el mismo tiempo de cobertura que el obtenido para $q = 0.1$, pero con menos computación. Por tanto el factor q requiere de una adecuada elección valorando conjuntamente la mejora que introduce (respecto al Método de Cubridores disjuntos) frente al tiempo al incremento del tiempo de computación que conlleva. A ello dedicaremos la siguiente sección.

6.6

Análisis comparativo de ambos métodos

Aquí compararemos el comportamiento del Método de *Cubridores No Disjuntos* frente al Método de *Cubridores Disjuntos*. Para éste último, hemos trabajado con el algoritmo Heurístico H_2 de *Penalización* ya que, como se vio en su momento, era el que mejores resultados producía.

Como antes hemos considerado matrices de Cobertura generadas aleatoriamente y, sobre cada una de ellas, hemos ejecutado el Heurístico H_2 (Penalización), y el anterior Heurístico $H(q)$ (Cubridores no disjuntos) considerando distintos valores del factor q , concretamente, hemos considerado: $q = 0.01, 0.1, 0.3, 0.5$. Para un número fijo de objetivos: $m = 15$, hemos considerado un número creciente de sensores, de 10 a 60 sensores variando de 10 en 10. Y, como antes, para cada par de valores (n, m) generado 20 casos al objeto de hallar las medias de los tiempos de cobertura obtenidos. La tabla siguiente muestra dichos resultados, y éstos se reflejan en la gráfica posterior.

($m=15$ objet.)		valores de q			
n (nº de Sens.)	H_2 (Penaliz.)	0.01	0.1	0.3	0.5
10	5.4	5.8	5.86	5.13	5.75
20	9.1	9.89	9.84	9.22	9.35
30	13,4	16,1	15,21	13,62	14,87
40	18.6	20.39	19.8	17.68	19.35
50	24.4	27.83	26.76	22.52	25.8
60	30.5	33.59	32.36	29.2	32.35

Tabla 6.4: Tiempos de cobertura medios producidos por ambos métodos, para un número fijo de objetivos

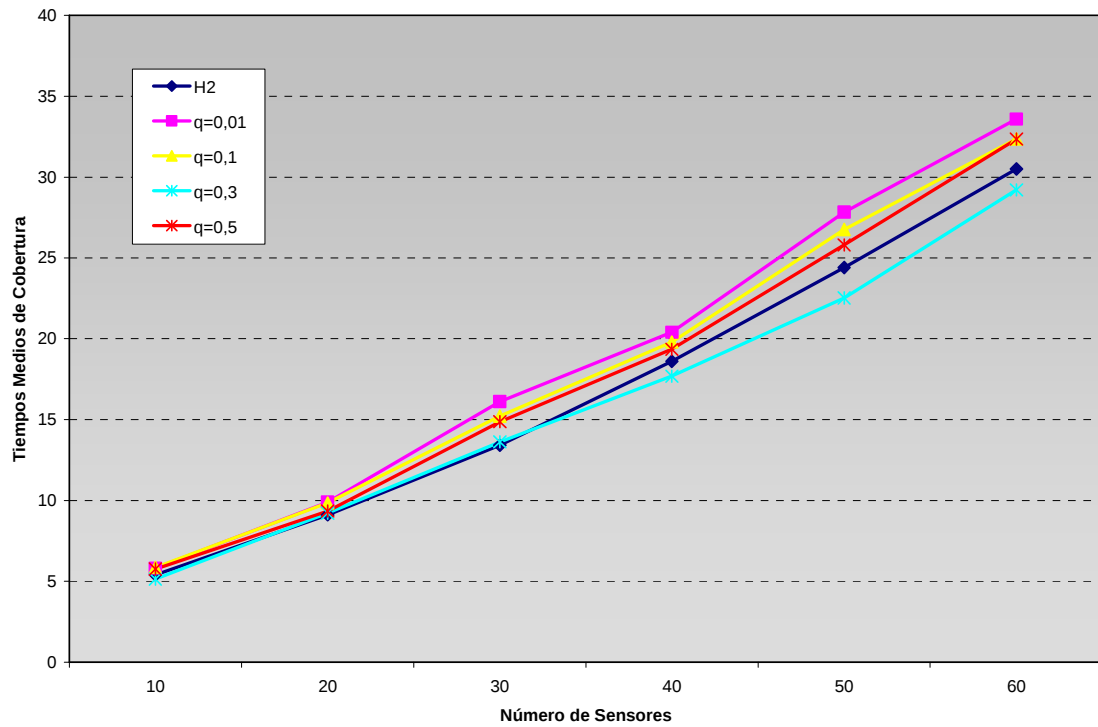


Figura 6.8: Gráfica para un número fijo de objetivos y creciente de sensores

En la siguiente tabla repetimos la experiencia computacional invirtiendo la relación de sensores y objetivos: consideramos ahora un número fijo de $n = 15$ sensores, y un número creciente de objetivos, desde 10 a 60, variando de 10 en 10. Los resultados se visualizan en la siguiente gráfica.

($n=15$ objet.)		valores de q			
m (nº de objet.)	H_2 (Penaliz.)	0.01	0.1	0.3	0.5
10	7.3	7.72	8.11	7.03	7.75
20	6.6	7.43	7.03	6.52	6.8
30	4.9	5.71	5.66	4.93	5.1
40	4.8	5.43	5.29	4.49	4.95
50	7.1	7.6	7.45	6.86	7.25
60	6.5	7.02	6.93	6.52	6.65

Tabla 6.5: Tiempos de cobertura medios producidos por ambos métodos para un número fijo de sensores.

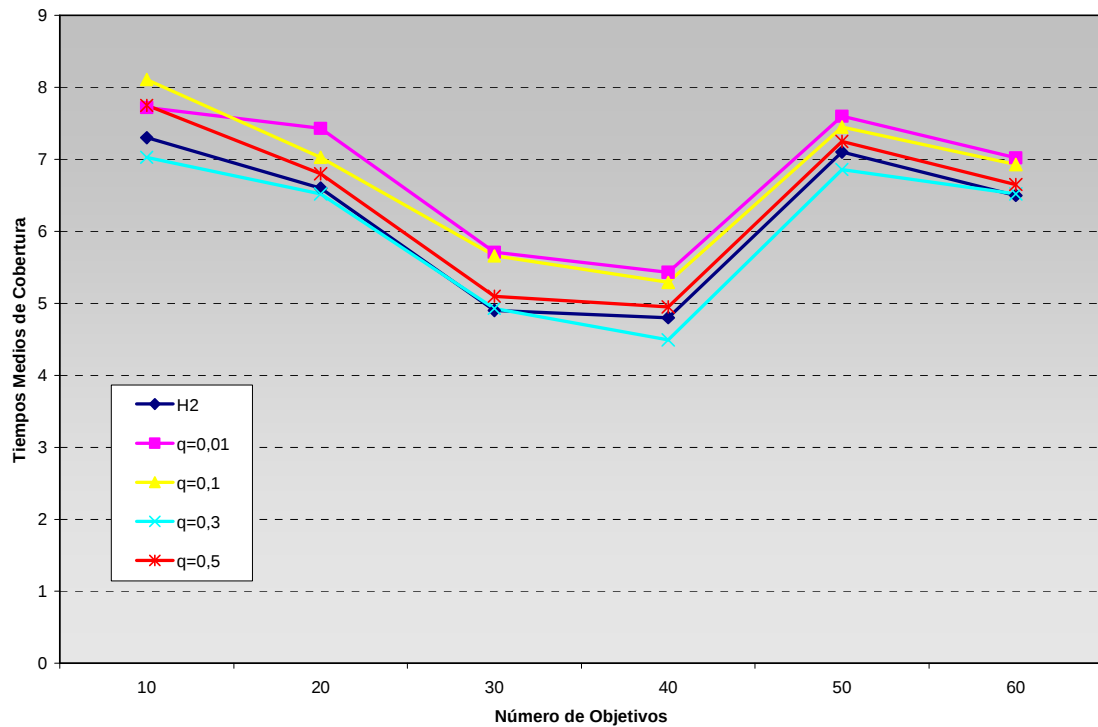


Figura 6.9: Gráfica para un número fijo de sensores pero creciente de objetivos

Del examen de todos los resultados obtenidos, podemos establecer las siguientes conclusiones respecto al comportamiento de ambos métodos, en términos estadísticos:

- El Algoritmo $H(q)$ (Cubridores no disjuntos) produce mejores resultados que el Algoritmo H_2 (Penalización) en las dos simulaciones descritas, pero sólo para determinadas valores de q .
- En las gráficas podemos observar que los mejores resultados se producen para valores pequeños de q : concretamente para $q = 0.01$ y $q = 0.1$. En las dos gráficas se observa cómo las líneas correspondientes a dichos valores quedan siempre por encima de las demás.
- También se observa que los valores intermedios de q (como $q = 0.3$), produjeron peores resultados que todos los demás, de hecho el comportamiento para tales valores fue peor que el del algoritmo H_2 (Cubridores disjuntos).
- El tiempo de Cobertura mediante Cubridores no disjuntos aumenta al disminuir el valor de q , lo que aconseja elegir valores pequeños del parámetro en este método, aunque esta decisión hay que calibrarla con el aumento en tiempo de computación que dicha elección conlleva.

6.7

Generalización a sensores no homogéneos

El anterior algoritmo $H(q)$ para Cubridores no disjuntos es fácilmente generalizable al caso de sensores de energía no homogénea, o en otras palabras, sensores con diferente tiempo de vida. Esto es:

- Partimos ahora de que cada sensor S_j dispone de un tiempo de vida b_j , $j = 1, \dots, n$ (en lugar de “ b ”).
- Sustituimos el anterior tiempo de participación “ qb ” de cada sensor en un Cubridor por un tiempo fijo “ r ”, tal que $r \leq \min\{b_j, j = 1, \dots, n\}$ (así garantizamos que todos los sensores pueden ser utilizados al menos una vez). Para ello, basta con seleccionar $b = \min\{b_j, j = 1, \dots, n\}$, y hacer $r = qb$, manteniendo la tasa de participación $q \in (0, 1]$.

Con estas modificaciones podemos aplicar el anterior algoritmo para prolongar el tiempo de cobertura de la red calculando el máximo número de Cubridores no disjuntos, de forma que cada sensor tenga un tiempo de participación “ r ” en los Cubridores en los que intervenga o, equivalentemente, que cada Cubridor se active por un tiempo r . También es evidente que el algoritmo anterior $H(q)$ (para un tiempo de vida b homogéneo) se puede considerar un caso particular de éste, cuando $b_j = b$, para todos los sensores, y cuando $q = \frac{r}{b}$, (ó equivalentemente, $qb = r$).

Sensores no homogéneos: Un Ejemplo.

A continuación mostramos el funcionamiento de esta generalización sobre el ejemplo visualizado en la Figura 6.10, en donde consideramos un escenario con 6 sensores de energía no homogénea actuando sobre 5 objetivos. Asimismo, y como ya se señaló al principio de este capítulo, los sensores pueden tener distinto rango de

sensibilidad, e incluso el área de sensibilidad de cada sensor puede no ser circular, ya que para establecer la matriz de cobertura basta conocer los objetivos cubiertos por cada sensor.

Matriz de Cobertura:

1	1	0	0	0
0	1	1	1	1
0	0	1	1	0
0	0	0	1	1
1	1	0	0	1
1	1	1	0	1

Tiempos de vida $b_j, j = 1, \dots, 6$, de cada sensor (puede representar horas, o cualquier otra unidad de tiempo):

	S_1	S_2	S_3	S_4	S_5	S_6
Energía (horas)	(1.5)	(1)	(2.5)	(1.5)	(2)	(1)

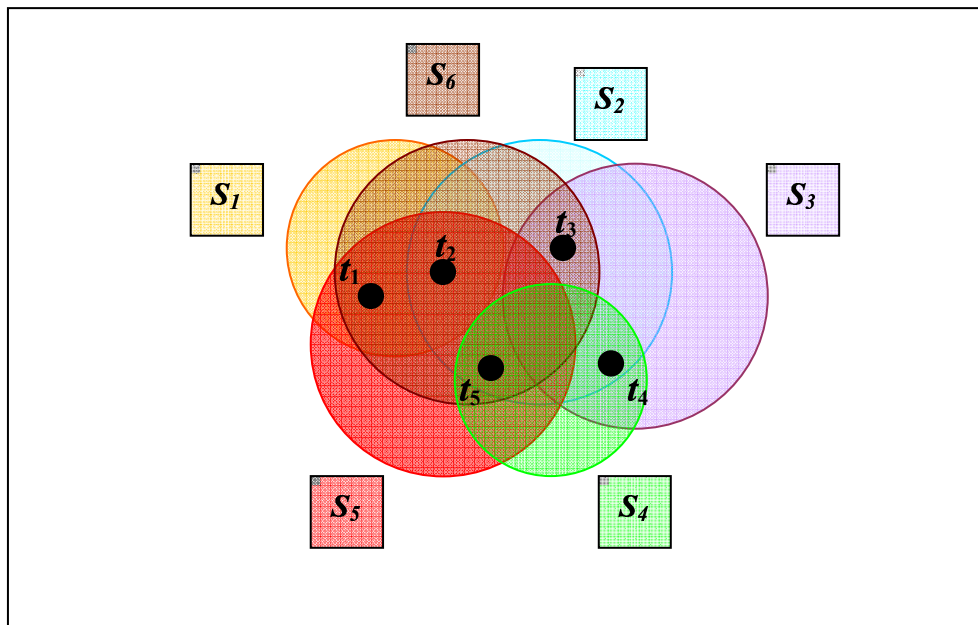


Figura 6.10: Ejemplo con 6 sensores y 5 objetivos

Para facilitar la lectura del proceso, la matriz de cobertura se visualiza mediante la siguiente tabla. El número entre paréntesis es el tiempo de vida de cada sensor. Claramente, $b = \min\{b_j, j = 1, \dots, 6\} = 1$.

	t_1	t_2	t_3	t_4	t_5
(1.5) S_1	1	1			
(1) S_2		1	1	1	1
(2.5) S_3			1	1	
(1.5) S_4				1	1
(2) S_5	1	1			1
(1) S_6	1	1	1		1

El resultado final depende obviamente de la tasa de participación q que se fije, o equivalentemente, del tiempo de vida r de cada Cubridor.

En esta simulación, tomaremos $q = 1$.

Por tanto $r = qb = 1$, lo que significa que cada sensor puede participar en cada Cubridor durante 1 hora.

Al principio, $Sens_Disponibles = S$, $Obje_Cubiertos = \emptyset$,

Tiempos residuales: $r_j = b_j, j = 1, \dots, 6$

Tiempo de participación $r = 1$

- Construcción de C_i

r_j		t_1	t_2	t_3	t_4	t_5
(1.5)	S_1	1	1			
(1)	S_2		1	1	1	1
(2.5)	S_3			1	1	
(1.5)	S_4				1	1
(2)	S_5	1	1			1
(1)	S_6	1	1	1		1

$$t_{critico} = \{t_1, t_3, t_4\}$$

$$S_{candidatos} = \{S_1, S_2, S_3, S_5, S_6\}$$

	S_1	S_2	S_3	S_4	S_5	S_6
M_1	2	4	2	2	3	4
$R_j = \frac{r_j}{r}$	1.5	1	2.5	1.5	2	1
$M_1 + R_j$	3.5	5	4.5	3.5	5	5

Como $f(S_j) = M_1 + R_j$, tenemos $f(S_j) \rightarrow \{3.5, 5, 4.5, 3.5, 5, 5\}$. $\rightarrow$ Elige S_2
(ya que al coincidir varios máximos, el algoritmo elige el primero en el orden de entrada).

$$C_1 = \{S_2\}$$

$$Obje_Cubiertos = \{t_2, t_3, t_4, t_5\},$$

$$t_{critico} = \{t_1\}, S_{candidatos} = \{S_1, S_5, S_6\}$$

$$f(S_j) \rightarrow \{2.5, 3, 2\}.$$

$$\text{Elige} \rightarrow S_5$$

$$C_1 = \{S_2, S_5\}.$$

$$\text{Como } Obje_Cubiertos = T = \{t_1, t_2, t_3, t_4, t_5\}, \text{ entonces}$$

End (Construcción de C_1)

ACTUALIZA Tiempo residual: $r_j = r_j - r, \forall S_j \in C_1 \rightarrow r_2 = 0, r_5 = 1$

$Sens_Disponibles = \{S_1, S_3, S_4, S_5, S_6\}$ (se ha eliminado S_2)

$Obje_Cubiertos = \emptyset$,

• Construcción de C_2

r_j		t_1	t_2	t_3	t_4	t_5
(1.5)	S_1	1	1			
(2.5)	S_3			1	1	
(1.5)	S_4				1	1
(1)	S_5	1	1			1
(1)	S_6	1	1	1		1

$$t_{critico} = \{t_1, t_2, t_3, t_4\}$$

$$S_{candidatos} = \{S_1, S_3, S_4, S_5, S_6\}$$

	S_1	S_3	S_4	S_5	S_6
M_1	2	2	2	3	4
R_j	1.5	2.5	1.5	1	1
$f(S_j)$	3.5	4.5	3.5	4	5

$$\text{Elige} \rightarrow S_6$$

$$C_2 = \{S_6\},$$

$$Obje_Cubiertos = \{t_1, t_2, t_3, t_5\}$$

$$t_{critico} = \{t_4\}, S_{candidatos} = \{S_3, S_4\}$$

$$\text{Elige} \rightarrow S_3$$

	S_3	S_4
M_1	1	1
R_j	2.5	1.5
$f(S_j)$	3.5	2.5

$$C_2 = \{S_3, S_6\},$$

$$\text{Como } Obje_Cubiertos = T = \{t_1, t_2, t_3, t_4, t_5\}, \text{ entonces}$$

End (Construcción de C_2)

$$\text{ACTUALIZA Tiempo residual: } r_j = r_j - r, \forall S_j \in C_2 \rightarrow r_3 = 1.5, r_6 = 0$$

$$Sens_Disponibles = \{S_1, S_3, S_4, S_5\} \text{ (se ha eliminado } S_6)$$

$$Obje_Cubiertos = \emptyset,$$

- Construcción de C_3

r_j		t_1	t_2	t_3	t_4	t_5
(1.5)	S_1	1	1			
(1.5)	S_3			1	1	
(1.5)	S_4				1	1
(1)	S_5	1	1			1

$$t_{critico} = \{t_3\}$$

$$S_{candidatos} = \{S_3\}$$

$$\text{Luego: Elige} \rightarrow S_3$$

$$C_3 = \{S_3\}$$

$$Obje_Cubiertos = \{t_3, t_4\}$$

$$t_{critico} = \{t_1, t_2, t_5\}, S_{candidatos} = \{S_1, S_4, S_5\}$$

$$\text{Elige} \rightarrow S_5$$

$$C_3 = \{S_3, S_5\}$$

$$\text{Como } Obje_Cubiertos = T = \{t_1, t_2, t_3, t_4, t_5\}, \text{ entonces}$$

	S_1	S_4	S_5
M_1	2	1	3
R_j	1.5	1.5	1
$f(S_j)$	3.5	2.5	4

End (Construcción de C_3)

ACTUALIZA Tiempo residual: $r_j = r_j - r, \forall S_j \in C_3 \rightarrow r_3 = 0.5, r_5 = 0$

$Sens_Disponibles = \{S_1, S_4\}$ (se han eliminado S_3 y S_5).

Como $Sens_Disponibles$ no cubren T , entonces:

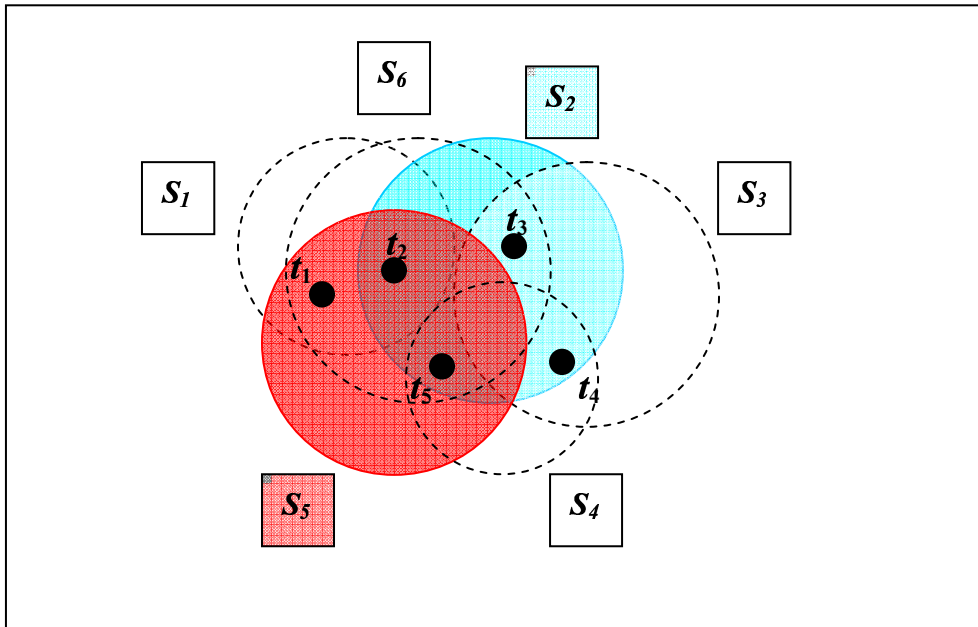
End Algoritmo.

Solución Final para un tiempo de participación $r = 1$ hora:

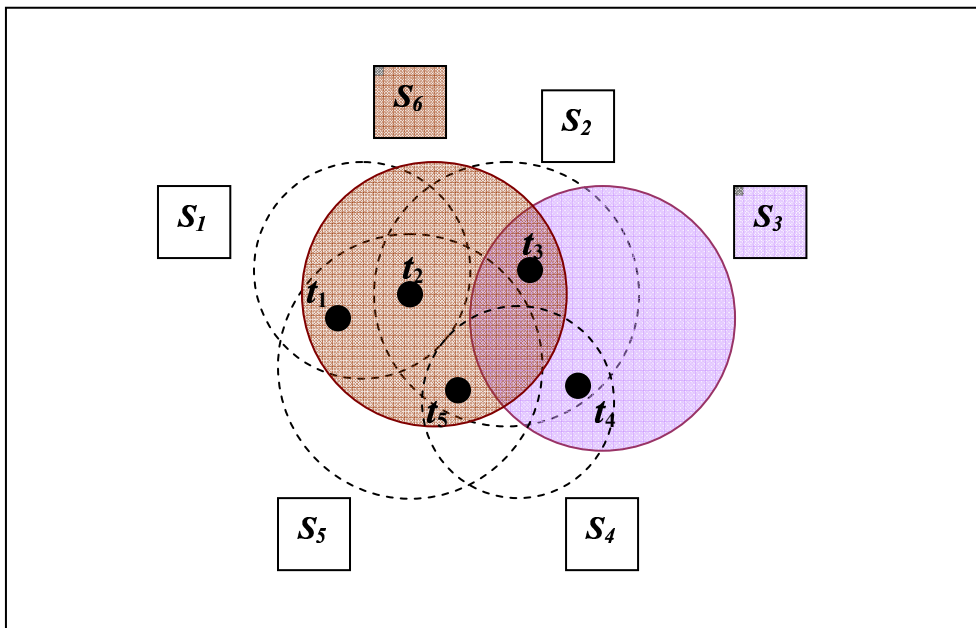
$C_1 = \{S_2, S_5\}, C_2 = \{S_3, S_6\}, C_3 = \{S_3, S_5\}$. Tiempo de Cobertura: 3 horas.

Las siguientes gráficas visualizan la cobertura de cada Conjunto Cubridor:

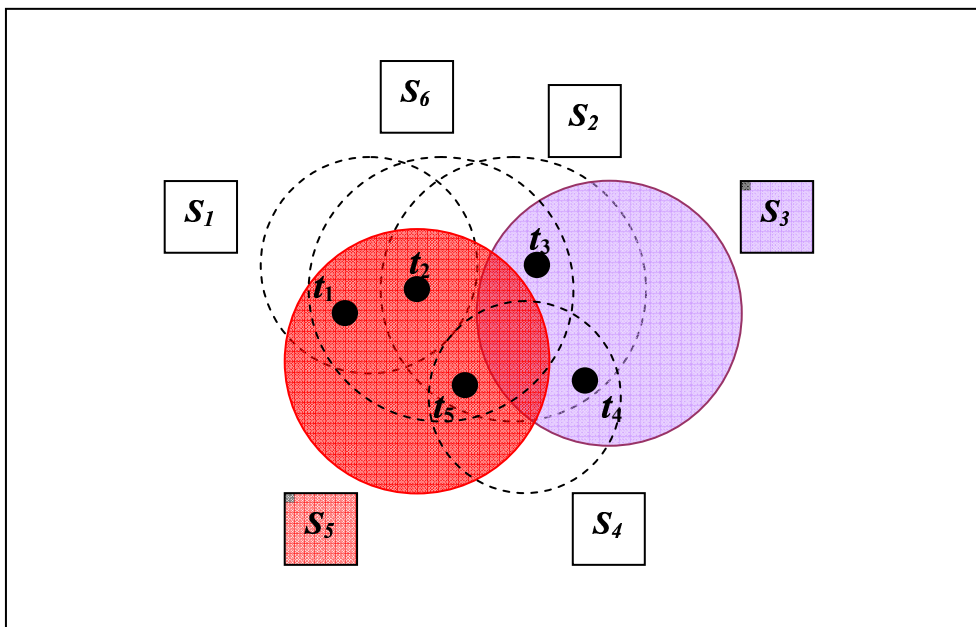
- $C_1 = \{S_2, S_5\}$



▪ $C_2 = \{S_3, S_6\}$



▪ $C_3 = \{S_3, S_5\}$



La siguiente tabla muestra, sobre el mismo ejemplo, los resultados (en tiempo de cobertura) que se obtienen al variar la tasa de participación q . Como se puede comprobar, tales resultados se ajustan a las conclusiones establecidas en la experiencia computacional de la sección anterior, en donde se comprobó que, al disminuir el valor de q aumentaba el tiempo de cobertura:

Tasa q	1	0.5	0.4	0.3	0.2	0.1	0.01	0.001
Nº de Cubridores	3	8	9	12	21	44	446	4497
Tiempo de Cobertura	3	4	3.6	3.6	4.2	4.4	4.46	4.497

El detalle de algunos de estos resultados se ofrece a continuación:

$$q = 1 \begin{cases} \text{Tiempo de Cobertura} = 3 \\ \text{Número de Cubridores} = 3 \end{cases}, \quad \text{Cubridores: } \{2,5\}, \{6,3\}, \{5,3\}$$

$$q = 0.5 \begin{cases} \text{Tiempo de Cobertura} = 4 \\ \text{Número de Cubridores} = 8 \end{cases}$$

Cubridores: $\{3, 5\}, \{2, 1\}, \{3, 5\}, \{6, 3\}, \{2, 1\}, \{4, 5, 3\}, \{6, 4\}, \{5, 3\}$

$$q = 0.3 \begin{cases} \text{Tiempo de Cobertura} = 3.6 \\ \text{Número de Cubridores} = 12 \end{cases}$$

Cubridores: $\{3, 5\}, \{3, 5\}, \{3, 1, 4\}, \{2, 1\}, \{3, 5\}, \{6, 3\}, \{2, 1\}, \{4, 5, 3\}, \{6, 4\},$
 $\{2, 5\}, \{6, 3\}, \{5, 3\}$

$$q = 0.1 \begin{cases} \text{Tiempo de Cobertura} = 4.4 \\ \text{Número de Cubridores} = 44 \end{cases}$$

Cubridores: $\{3,5\}, \{3,5\}, \{3,5\}, \{3,5\}, \{3,5\}, \{3,1,4\}, \{3,5\}, \{3,1,4\}, \{3,5\}, \{3,5\}, \{3,1,4\},$
 $\{3,5\}, \{2,1\}, \{3,5\}, \{6,3\}, \{2,1\}, \{4,6\}, \{2,1\}, \{3,5\}, \{4,6\}, \{2,1\}, \{3,5\}, \{4,6\},$
 $\{2,1\}, \{3,5\}, \{4,6\}, \{2,1\}, \{3,5\}, \{4,6\}, \{2,1\}, \{3,5\}, \{4,6\}, \{2,1\}, \{3,5\}, \{4,6\},$
 $\{2,1\}, \{3,5\}, \{4,6\}, \{2,1\}, \{3,5\}, \{4,6\}, \{3,5\}, \{4,1,3\}, \{4,1,3\}$

$$q = 0.01 \begin{cases} \text{Tiempo de Cobertura} = 4.46 \\ \text{Número de Cubridores} = 446 \end{cases}$$

Cubridores:

[illegible]

