



**Departamento de Estadística y Matemática Aplicada**

**Nuevos Modelos Probabilísticos  
de Localización de Servicios de Emergencias<sup>1</sup>**

**Fernando Borrás Rocher**

Memoria para optar al grado de Doctor por la Universidad Miguel Hernández, realizada bajo la dirección de los doctores D. Jesús T. Pastor Ciurana y D. Marc Almiñana Alemany del Área de Estadística e Investigación Operativa de la Universidad Miguel Hernández.

San Juan, a 19 de Junio de 2000

---

<sup>1</sup> Este trabajo ha sido parcialmente subvencionado por el Instituto de Cultura JUAN GIL-ALBERT a través de una ayuda a la investigación para la realización de Tesis doctorales de la convocatoria del año 1999.



D. JESÚS T. PASTOR CIURANA, Catedrático de Estadística e Investigación Operativa y D. MARC ALMIÑANA ALEMANY, Titular de Universidad de Estadística e Investigación de la Universidad Miguel Hernández,

CERTIFICAN: que la presente memoria "**Nuevos Modelos Probabilísticos de Localización de Servicios de Emergencias**" ha sido realizada bajo nuestra dirección en el Departamento de Estadística y Matemática Aplicada de la Universidad Miguel Hernández, por D. Fernando Borrás Rocher, y constituye su Tesis para optar al grado de Doctor por la Universidad Miguel Hernández.

Y para que conste, en cumplimiento de la legislación vigente y a los efectos oportunos, firmamos la presente, en Elche a veinte de Junio de dos mil.

Fdo: Jesús T. Pastor Ciurana

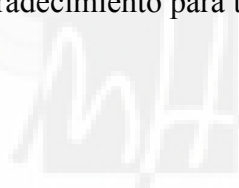
Fdo: Marc Almiñana Alemany



## AGRADECIMIENTOS

Quiero expresar todo mi agradecimiento a los Profesores D. Jesús Tadeo Pastor Ciurana y D. Marc Almiñana Alemany que a través de esta memoria me permitieron iniciarme en el complejo mundo de la investigación, hecho que habría sido imposible sin su respaldo y apoyo.

Culminar este trabajo ha necesitado de la ayuda y del aliento de diferentes personas, que me permitieron no desfallecer en su realización. Vaya desde aquí mi más profundo agradecimiento para todos ellos.

 Miguel  
Hernández



## **AGRAÏMENTS**

*Als meus pares, Fernando i Elodia, per l'esforç que han fet durant tota la meua vida, el qual em va permetre estudiar i aconseguir la llicenciatura en Ciències Matemàtiques. Perquè, com ells mateixos diuen, “els bancals es poden vendre, però el que t'hem donat no ho podràs vendre mai”.*

*A la meua dona, Lorena, que ha patit tot aquest procés i no ha pogut gaudir tot el que voldria de la meua companyia.*

*Aquesta memòria està dedicada als meus fills, Ferran i Isabel, encara que han posat molts entrebancs perquè aquesta memòria arribara a bon terme.*





## Presentación

El proyecto de Tesis Doctoral "**Nuevos Modelos Probabilísticos de Localización de Servicios de Emergencias**", que se presenta a continuación, recurre a las tres técnicas más comunes para analizar la ubicación de unidades de emergencia: modelos lineales enteros, simulación de sistemas y modelos basados en la teoría de colas.

En esta memoria se pretende aunar las tres vías, hasta ahora bastante inconexas, y realizar contribuciones en el campo del estudio de modelos de cubrimiento total probabilístico, que incluye tanto su diseño como su resolución computacional. Este tipo de modelos asume la demanda del servicio discretizada, así como la existencia de un número finito de estaciones de servicio. Además, la probabilidad de que una llamada de emergencia sea atendida dentro de un cierto intervalo de tiempo debe ser al menos tan alta como el nivel de fiabilidad fijado a priori. Más en concreto, buscamos modelos que minimicen el número de unidades de emergencia (ambulancias, coches de bomberos, patrullas policiales, coches grúas, etc.), cumpliendo con las restricciones de que el nivel de fiabilidad sea mayor que un cierto nivel prefijado para cada zona de demanda, y capaces de proporcionar buenas soluciones, con tiempos moderados, para grandes instancias de los modelos.

La conexión de las tres herramientas se efectuará verificando, a posteriori, si la configuración de vehículos obtenida, resolviendo un modelo lineal entero, alcanza el nivel de fiabilidad prefijado, usando para tal fin un simulador de sistemas de emergencia y/o la resolución de un sistema de ecuaciones no lineales, basado en los sistemas en equilibrio de los modelos de la teoría de colas.

Todos los nuevos modelos de cubrimiento total probabilístico que se presentarán, junto con los ya existentes en la literatura, resultarán ser particularizaciones del modelo de cubrimiento total generalizado. Dicho modelo pertenece al campo de la programación lineal entera y, en consecuencia, su resolución no es sencilla debido a la integralidad de las variables. En la resolución de problemas de programación entera se utilizan comúnmente esquemas de ramificación y acotación. El éxito de estos métodos depende de la rapidez con la que se encuentre una buena solución factible que permita descartar muchos nodos del árbol de ramificación. Con estas ideas desarrollaremos técnicas de obtención de buenas soluciones factibles mediante heurísticos “greedy” y aquellos basados en la relajación lagrangiana, una de las técnicas más usadas en la resolución de problemas de optimización combinatoria.

Para la consecución de los objetivos planteados ha sido necesaria la siguiente secuenciación de tareas:

- Realizar una profunda revisión bibliográfica de los modelos lineales enteros existentes en la literatura para ubicar el menor número de unidades de servicio bajo restricciones de fiabilidad.
- Revisar la literatura especializada en lo referente a los modelos basados en la teoría de colas, que permiten analizar más detalladamente el comportamiento de los sistemas de emergencia.
- Estudiar las diferentes simulaciones de sistemas de emergencia aparecidas en las revistas científicas, como último recurso para modelizar su extrema complejidad.
- Implementar computacionalmente los diferentes modelos lineales enteros para poder analizar el comportamiento de las configuraciones de vehículos obtenidas.
- Diseñar y construir un modelo, mediante simulación, que reproduzca la aleatoriedad de un sistema de emergencia desde la llegada de cada llamada, la consiguiente asignación de la unidad que la atiende de acuerdo con el estado del sistema, hasta la duración de cada servicio.

- Obtener una solución aproximada a la dada por el simulador mediante la resolución de un sistema de ecuaciones no lineales basado en un modelo de colas bajo una de las hipótesis alternativas:
  - Independencia entre las unidades de servicio. Esta hipótesis, curiosamente, no es muy restrictiva en los sistemas de emergencia poco congestionados.
  - Dependencia entre las unidades de emergencia. Esta hipótesis es importante en el estudio de sistemas con cargas de trabajo muy altas.
- Valorar las diferencias existentes entre la simulación y el sistema de ecuaciones no lineales del modelo.
- Detectar las deficiencias en los modelos existentes en la literatura revisada si las hubiera, es decir, verificar si las configuraciones de vehículos cumplen con las restricciones de fiabilidad.
- Diseñar nuevos modelos basados en cubrimiento múltiple, cuya característica es la de obtener la fiabilidad requerida a priori, determinando el número mínimo de unidades de emergencia necesarias en el entorno de cada nodo demanda. Los modelos construidos deben asegurar el cumplimiento de las restricciones de fiabilidad individuales a posteriori y ser una cota superior lo más cercana posible al número mínimo de unidades necesarias para atender todas las demandas dentro del nivel de fiabilidad prefijado, bajo la hipótesis de independencia o de dependencia entre las distintas unidades de emergencia.
- Construir nuevos modelos basados en cantidad de cubrimiento, cuya caracterización es la de obtener la fiabilidad requerida a priori, determinando la cantidad de cubrimiento que debe ser satisfecha en el entorno de cada nodo demanda y el cubrimiento aportado por la ubicación de diferentes unidades de emergencia en cada estación. Los modelos construidos deben asegurar el cumplimiento de las restricciones de fiabilidad individuales a posteriori y ser una cota superior lo más cercana posible al número mínimo de unidades necesarias para atender todas las demandas dentro del nivel de

fiabilidad prefijado, bajo la hipótesis de independencia o de dependencia entre las distintas unidades de emergencia.

- Realizar un estudio comparativo de la eficiencia de los diferentes modelos de cubrimiento total probabilístico analizados y extraer las conclusiones pertinentes.
- Construir un modelo general que incluya todos los modelos de cubrimiento total probabilístico, tanto modelos basados en el cubrimiento múltiple, obtenidos al convertir una restricción no lineal en una restricción lineal equivalente, como los basados en cantidad de cubrimiento, que se formulan mediante restricciones lineales a partir de argumentos probabilísticos.
- Diseñar un heurístico para obtener soluciones en un tiempo de ejecución razonable de los modelos construidos, basado en las siguientes etapas:
  - a) Obtención de una cota superior inicial mediante un procedimiento heurístico greedy.
  - b) Cálculo de una cota inferior inicial mediante un heurístico de ascenso dual.
  - c) Mejora de la cota inferior mediante un procedimiento subgradiente que comienza con el último vector de multiplicadores lagrangianos calculado (en la primera aplicación del procedimiento se utiliza la solución obtenida en la fase b).
  - d) Mejora de la cota superior mediante un procedimiento heurístico greedy basado en las variables duales cuando la cota inferior es mejorada en la fase c).
  - e) Criterios de parada del procedimiento iterativo. Si los criterios de parada no son satisfechos, las fases c) y d) son repetidas hasta que se verifique algún criterio de parada.
- Valorar la eficiencia de la implementación del algoritmo desarrollado comparando su solución con la solución óptima obtenida mediante un proceso de ramificación y acotación.

Los capítulos que conforman esta memoria y un resumen de su contenido son:

**Capítulo 1. Modelos de Localización de Servicios de Emergencias:** Descripción de la evolución y del estado actual de la investigación en el campo de la ubicación de servicios bajo restricciones de cubrimiento y fiabilidad, revisando los tres diseños básicos existentes: modelos lineales enteros, simulación y sistemas de colas.

**Capítulo 2. Nuevos Modelos de Cubrimiento total Probabilístico:** Descripción de las deficiencias de los modelos existentes mediante la evaluación de la fiabilidad a posteriori de la configuración de vehículos, recurriendo tanto al simulador del sistema de emergencia como a la resolución de sistemas de ecuaciones no lineales basado en modelos de colas. Formulación de nuevos modelos de cubrimiento probabilístico, basados en cubrimiento múltiple o en cantidad de cubrimiento, cuyo comportamiento sea mejor en el cumplimiento de las restricciones de fiabilidad individuales y sea una cota superior lo más cercana posible al número mínimo de unidades necesarias para atender, con las fiabilidades requeridas, las llamadas de emergencia del sistema.

**Capítulo 3. Modelo del Cubrimiento Total Generalizado:** Formulación del modelo de cubrimiento total generalizado y presentación de los diferentes problemas que unifica. Descripción del heurístico, basado en la relajación lagrangiana, presentado para su resolución y de la experiencia computacional, sobre una colección de problemas de diferente dimensión y estructura, para evaluar la eficiencia del algoritmo.

**Capítulo 4. Conclusiones.** Exposición de las conclusiones obtenidas e introducción de las futuras líneas de investigación.

En último lugar, las instancias de menor tamaño utilizadas, quedan perfectamente descritas en el apéndice A. Por otra parte, en el apéndice B se presentan algunos de los listados de los programas de ordenador, implementados en lenguaje FORTRAN y C, que se corresponden con diferentes rutinas utilizadas a lo largo de esta memoria.



# Índice

## 1. Modelos de Localización de Servicios de Emergencias.

1.1. Introducción.....	1
1.2. El problema del cubrimiento total.....	4
1.3. El problema del cubrimiento maximal.....	13
1.4. El problema del cubrimiento maximal con diferenciación de vehículos.....	19
1.5. Modelización de servicios de emergencia como un sistema de colas.....	25
1.6. Modelización de servicios de emergencia mediante simulación.....	32

## 2. Nuevos Modelos de Cubrimiento total Probabilístico.

2.1. Introducción.....	35
2.2. El simulador del Sistema de Emergencia.....	36
2.3. El sistema de ecuaciones no lineales del Sistema de Emergencia.....	40
2.4. Comparativa computacional entre el simulador y la resolución del sistema de ecuaciones no lineales.....	45
2.5. Nuevos modelos de Cubrimiento total Probabilístico basados en cubrimiento entero.	
2.5.1. Modelos existentes: el Binomial y el Queueing Probabilistic Location Set Covering Problem.....	51
2.5.2. El Poisson Probabilistic Location Set Covering Problem.....	56
2.5.3. El Revised Binomial Probabilistic Location Set Covering Problem.....	60
2.5.4. El Revised Poisson Probabilistic Location Set Covering Problem.....	68
2.6. Nuevos modelos de Cubrimiento total Probabilístico basados en cantidad de cubrimiento.	
2.6.1. Modelo existente: El Poisson Reliability location set covering problem...	72
2.6.2. El Binomial Reliability location set covering problem (BRLSCP).....	76
2.6.3. El Queueing Reliability location set covering problem (QRLSCP).....	81
2.7. Comparación global entre los modelos de cubrimiento total probabilístico.....	84

<b>3. Modelo del Cubrimiento Total Generalizado.</b>	
3.1. Introducción y formulación.....	87
3.2. Propiedades básicas del GSCP (Generalized Set Covering Problem) .....	90
3.3. Casos particulares del GSCP .....	94
3.4. Un heurístico para el GSCP.....	97
3.4.1. Obtención de la cota superior inicial .....	99
3.4.2. Obtención de la cota inferior inicial .....	104
3.4.3. Procedimiento de mejora de la cota inferior.....	108
3.4.4. Procedimiento de mejora de la cota superior.....	111
3.4.5. Criterios de parada.....	112
3.5. Resultados computacionales .....	114
<b>4. Conclusiones.....</b>	<b>131</b>
<b>Referencias.....</b>	<b>143</b>
<b>Apéndice A. Listado de redes utilizadas .....</b>	<b>159</b>
<b>Apéndice B. Listado de programas .....</b>	<b>161</b>
Simulador Sistema de Emergencias .....	162
Resolución sistema de ecuaciones no lineales Sistema de Emergencia .....	172



# Capítulo 1. Modelos de Localización de Servicios de Emergencias

## 1.1. Introducción.

Los servicios de emergencias, tales como ambulancias, bomberos o reparaciones urgentes operan en un entorno complejo debido a la variabilidad, tanto temporal como espacial, de la demanda que debe ser atendida. Durante los últimos treinta años, estos servicios se han beneficiado del desarrollo de diferentes modelos de investigación operativa aplicados a la localización de estaciones y vehículos de emergencias.

La modelización aplicada a la ubicación de servicios de emergencias se estructura a partir de una red de nodos y arcos. Los nodos pueden representar tanto a los puntos de demanda, como a las posibles ubicaciones de las estaciones de servicio. El arco que conecta cada par de nodos representa el camino más corto entre ellos. La finalidad de estos modelos es encontrar la mejor ubicación de las estaciones, así como la determinación de los diferentes tipos de vehículos estacionados en ellas. Estos modelos pueden ser clasificados en dos grandes bloques según el tipo de objetivo utilizado:

- Modelos de cubrimiento, basados en que el nodo demanda se considera cubierto si existe un servicio de emergencia a menos de una distancia máxima  $S$ , o si el nodo demanda puede ser atendido dentro de un tiempo máximo  $S$  con determinada probabilidad.
- Modelos de la p-mediana, basados en la minimización del tiempo medio de respuesta del servicio de emergencia sobre la red, donde por respuesta se entiende el tiempo

transcurrido desde que se recibe la llamada hasta que el vehículo llega al nodo demanda solicitante.

Los modelos de la p-mediana no son enteramente satisfactorios dentro del contexto de los sistemas de emergencias, dado que pueden dar soluciones en las cuales algunos puntos demanda se sitúan excesivamente alejados de la estación más cercana. El mayor auge de los modelos de cubrimiento se debe al hecho de recoger más explícitamente las regulaciones existentes en el ámbito de los sistemas de emergencias. De hecho, la regulación asociada con la *Emergency Medical Service (EMS) Act* (1973) americana indica que el número adecuado de vehículos de emergencia es aquel que permita que el 95% de las solicitudes de asistencia puedan ser atendidas en menos de 30 minutos en zonas rurales y en menos de 10 minutos en áreas urbanas. El origen de esta regulación está en un informe de la *Carnegie Commission on Higher Education* de 1970. En dicho dictamen se exigía que los 55 hospitales que iban a ser ubicados a lo largo de todo el país, debían ser localizados de forma que el 95% de la población estuviese a menos de una hora en coche.

Por otra parte, la *Insurance Services Office (ISO)* (1974) ha dictado normas para la protección contra-incendios que son similares a las dadas por la *EMS Act*, excepto que son dadas en términos de distancias y no incluyen la especificación del 95%. La respuesta estándar a un incendio determina el número de coches-cisterna y el número de coches-escalera que son necesarios tener ubicados dentro de una distancia prefijada respecto del nodo demanda, tanto para los coches-cisterna como para los coches-escalera. Los requerimientos varían directamente con el flujo de agua requerido, en galones por minuto (1 galón  $\approx$  4.5 litros), reflejando el tamaño del incendio, que generalmente es determinado por el tipo de edificio en la zona incendiada. En la tabla siguiente se presenta el número necesario de coches-cisterna y coches-escalera para dar una respuesta estándar según el tamaño del incendio. Por ejemplo, para un tamaño de incendio cuyo flujo de agua requerido es de menos de 2000 galones por minuto, se debe tener ubicados en un entorno de 4 millas, 2 coches-cisterna; y en un entorno de 2 millas, 1 coche-escalera.

**Tabla 1.1 Respuesta estándar (ISO,1974)**

Flujo de agua requerido, en galones por minuto	Coches- cisterna		Coches-escalera	
	Número	Millas	Número	Millas
Menos de 2000	2	4	1	2
2000	2	2.5	1	2
2500	2	2.5	1	2
3000	2	2.5	1	2
3500	2	2.5	1	2
4000	2	2.5	1	2
4500	2	2.5	1	2
5000	2	2	1	1.5
5500	2	2	1	1.5
6000	2	2	1	1.5
6500	2	2	1	1.5
7000	2	1.5	1	1.5
7500	2	1.5	1	1.5
8000	2	1.5	1	1.5
8500	3	1.5	2	2
9000	3	1.5	2	2
10000	3	1.5	2	2
11000	3	1.5	2	2
12000	3	1.5	2	2

Tradicionalmente, las tres vías para abordar el problema de la ubicación de las unidades de emergencia respecto al concepto del cubrimiento han sido:

- la modelización de los sistemas mediante formulaciones lineales enteras, basadas en ciertas simplificaciones de los sistemas de emergencia.
- Modelos basados en teoría de colas, que permiten analizar más detalladamente el comportamiento de los sistemas de emergencia.
- Simulación del sistema como último recurso para modelizar su extrema complejidad.

En las secciones siguientes se realizará una revisión del estado en que se encuentra la investigación en las diferentes áreas.

## 1.2. El problema del cubrimiento total.

El primer modelo en la secuencia de modelos de cubrimiento aplicado a la localización de servicios de emergencia es el Location Set Covering Problem (LSCP) (*Toregas et al.*, 1971). El objetivo del LSCP es determinar la ubicación del menor número posible de estaciones de emergencia de forma que cada punto demanda tenga una estación no más allá de una distancia o tiempo  $S$ . La formulación del problema es la siguiente

$$\text{Min } \sum_{j \in J} x_j \quad (1.1)$$

$$\text{s.a. } \sum_{j \in N_i} x_j \geq 1 \quad \forall i \in I \quad (1.2)$$

$$x_j \in \{0,1\} \quad \forall j \in J, \quad (1.3)$$

donde:

$J$  es el conjunto de potenciales ubicaciones de servicios de emergencia;

$I$  es el conjunto de nodos cuya demanda debe ser atendida;

$d_{ij}$  es la distancia que separa el nodo  $i$  del potencial puesto de servicio  $j$ ;

$S$  es el tiempo o distancia máxima permitida para que una llamada de emergencia se considere cubierta;

$N_i = \{j \in J / d_{ij} \leq S\}$  es el conjunto de ubicaciones potenciales que pueden servir al nodo  $i$  dentro del tiempo máximo o distancia permitida  $S$ ;

$$x_j = \begin{cases} 1, & \text{si una estación es ubicada en el lugar } j \\ 0, & \text{en caso contrario.} \end{cases}$$

El objetivo (1. 1) minimiza el número de estaciones necesarias de manera que, de acuerdo con (1. 2), cada nodo demanda  $i$  tenga al menos un servicio de urgencias dentro de la distancia máxima  $S$ . Finalmente (1. 3) determina el carácter binario de las variables, es decir, establece los puestos de servicio  $j$  en funcionamiento ( $x_j = 1$ ) y los inactivos ( $x_j = 0$ ).

Para resolver este problema los autores recurren a una técnica basada en la resolución del problema lineal relajado y la adición, si fuera necesario (sólo el 5% de los casos según su experiencia), de un corte adicional que elimina resultados fraccionarios, basado en que la función objetivo sea menor o igual que el valor entero por defecto de la solución del problema lineal. El aparente éxito de la adición del corte sólo es explicable por el hecho de trabajar con problemas de tamaño reducido. No obstante, es posible encontrar muchos ejemplos en los

cuales el corte no conduce a ninguna solución entera (véase *Pastor (1993)*). En 1972 y 1973, *Toregas y Reville* publican dos trabajos donde introducen técnicas de reducción para conseguir que el LSCP sea más fácil de resolver en problemas de mayor tamaño.

Desafortunadamente, la estación más cercana no siempre está disponible cuando una llamada llega al sistema. En la localización de servidores en sistemas congestionados, el LSCP puede ser usado, en un primer paso, para determinar la ubicación de las estaciones, pero no es capaz de determinar cuántos vehículos deben ser asignados a cada base. Algunas veces, un único vehículo en cada estación seleccionada puede ser bastante para cubrir las necesidades del sistema, si aseguramos que cada nodo demanda pueda ser servido por diferentes estaciones dentro del tiempo  $S$ . Por esta razón un grupo de modelos se han dirigido al problema del cubrimiento múltiple. *Berlin (1974)* y *Daskin y Stern (1981)* buscan el óptimo alternativo al LSCP con el que se obtiene el mayor número de cubrimientos adicionales sobre el conjunto de todos los nodos demanda. Matemáticamente, su formulación es:

$$\text{Max} \quad \sum_{i \in I} r_i \quad (1.4)$$

$$\text{s.a.} \quad \sum_{j \in N_i} x_j - r_i = 1 \quad \forall i \in I \quad (1.5)$$

$$\sum_{j \in J} x_j = p \quad (1.6)$$

$$x_j \in \{0,1\} \quad \forall j \in J \quad (1.7)$$

$$r_i \geq 0, \quad r_i \in Z^+ \quad \forall i \in I, \quad (1.8)$$

donde:

$r_i$  es el número de estaciones adicionales capaces de responder a una llamada en la zona  $i$  en un tiempo menor o igual a  $S$ ;

$p$  es el número de estaciones a ubicar de forma que cada nodo demanda sea cubierto al menos una vez, es decir, la solución óptima del LSCP.

El defecto de esta formulación es que maximiza el total de cubrimientos redundantes sin tener en cuenta la magnitud o el número de llamadas de cada nodo demanda. *Benedict (1983)* y *Eaton et al. (1986)* corrigen esta deficiencia y encuentran el óptimo alternativo al LSCP que maximiza la suma de cubrimientos adicionales ponderado por la frecuencia de llamadas de cada nodo. Es decir, en el modelo anterior, sustituyen (1.4) por el siguiente objetivo

$$\text{Max} \quad \sum_{i \in I} f_i r_i \quad (1.9)$$

donde:

$f_i$  es el número o frecuencia de llamadas efectuadas desde el nodo demanda  $i$ .

Otra desventaja de estos modelos es que los cubrimientos adicionales pueden concentrarse en algunos nodo demanda, dejando otros con un único servidor. *Hogan y ReVelle* (1986) maximizan el número de nodos con un segundo servidor (primer recubrimiento) en su BACKUP COVERAGE PROBLEM 1 (BACOP1). Su modelo es similar al anterior, únicamente reemplazan la restricción (1.5) por

$$\sum_{j \in N_i} x_j - r_i \geq 1 \quad \forall i \in I, \quad (1.10)$$

donde ahora  $r_i$  es una variable binaria, con

$$r_i = \begin{cases} 1, & \text{si el número de servidores es mayor o igual a 2} \\ 0, & \text{en caso contrario.} \end{cases}$$

El siguiente paso natural dado por los investigadores consistió en incorporar en los modelos de localización la disponibilidad de los servidores. *Chapman y White* (1974) formulan la primera versión probabilística del LSCP. En su modelo, para cada zona demanda, la probabilidad de ser atendido dentro de un tiempo  $S$  es obligada a ser mayor o igual que un cierto nivel de fiabilidad prefijado  $\alpha$ , habitualmente un número próximo a 1. Su modelo puede ser formulado como:

$$\text{Min} \quad \sum_{j \in J} x_j \quad (1.11)$$

$$\text{s.a.} \quad 1 - \rho^{\sum_{j \in N_i} x_j} \geq \alpha \quad \forall i \in I \quad (1.12)$$

$$x_j \geq 0, \quad x_j \in Z^+ \quad \forall j \in J, \quad (1.13)$$

donde:

$\rho$  es la fracción de ocupación media, es decir, una estimación de la probabilidad de que un servidor esté atendiendo una llamada en el sistema;

$x_j$  es el número de vehículos a ubicar en la estación  $j$ .

La función objetivo (1.11) minimiza el número de vehículos necesarios para satisfacer las restricciones de fiabilidad impuestas. Bajo la hipótesis de que los servidores actúan independientemente, el conjunto de restricciones (1.12) impone que la probabilidad de ser

atendido dentro del tiempo de desplazamiento  $S$  para un nodo demanda es igual al complementario de no poder ser servido por los vehículos ubicados en un entorno de radio  $S$  y obliga a esta probabilidad a ser mayor que el nivel de fiabilidad prefijado  $\alpha$ . (1. 13) define las variables como enteras, permitiendo ocuparse directamente del número de vehículos necesarios en cada base. Es evidente, que cuando dicho número es 0, la base está inactiva. El modelo presentado es no lineal, aunque es linealizable tomando logaritmos. Para resolver este modelo es obvio que necesitamos una estimación a priori de  $\rho$ . Esta estimación plantea serias dificultades, ya que la fracción de ocupación media depende del número total de llamadas generadas por el sistema y del número total de vehículos ubicados. Es decir, sólo se conoce exactamente 'a posteriori', una vez resuelto el modelo planteado. *ReVelle y Hogan (1988)* presentan el Binomial Probabilistic Location Set Covering Problem (BPLSCP) un modelo que intenta eliminar la limitaciones del modelo anterior considerando que la fracción de ocupación de cada vehículo es uniforme para cada nodo demanda. Utilizando la ecuación propuesta por *Daskin (1983)*, definen la fracción de ocupación local basada en cada nodo demanda como:

$$\rho_i = \frac{\bar{t} \sum_{k \in M_i} f_k}{24 \sum_{j \in N_i} x_j} \quad \forall i \in I, \quad (1. 14)$$

donde:

$f_i$  es el número o frecuencia de llamadas del nodo demanda  $i$  (en llamadas por día);

$\bar{t}$  es la duración media de un servicio de emergencia (en horas);

$M_i = \{k \in I / d_{ki} \leq S\}$  es el conjunto de nodos demanda que están ubicados en un entorno de radio  $S$  del nodo  $i$ .

La fracción de ocupación local  $\rho_i$  definida en (1. 14) es interpretada como el cociente entre la cantidad de tiempo de servicio, medido en horas, necesario para atender los nodos demanda situados alrededor del nodo  $i$ , calculada como la cantidad de llamadas atendidas por la duración media de cada emergencia, y el número diario de horas disponible para el servicio, suponiendo que cualquier vehículo está disponible 24 horas al día. El BPLSCP es formulado como:

$$\text{Min } \sum_{j \in J} x_j \quad (1.15)$$

$$\text{s.a. } 1 - \rho_i^{\sum_{j \in N_i} x_j} \geq \alpha \quad \forall i \in I \quad (1.16)$$

$$x_j \geq 0, \quad x_j \in Z \quad \forall j \in J. \quad (1.17)$$

Las restricciones de cubrimiento probabilístico (1.16) basadas en la distribución binomial, es decir, en la independencia entre los servidores, pueden ser reescritas como

$$\left( \frac{\bar{t} \sum_{k \in M_i} f_k}{24 \sum_{j \in N_i} x_j} \right)^{\sum_{j \in N_i} x_j} \leq 1 - \alpha \quad \forall i \in I. \quad (1.18)$$

Aunque (1.18) no tiene una expresión lineal analítica equivalente, *ReVelle y Hogan* (1988) han encontrado la siguiente expresión lineal numérica equivalente:

$$\sum_{j \in N_i} x_j \geq b_i \quad \forall i \in I, \quad (1.19)$$

donde:

$$b_i = \min \left\{ n \in N / \left( \frac{\bar{t} \sum_{k \in M_i} f_k}{24n} \right)^n \leq 1 - \alpha \right\}. \quad (1.20)$$

La formulación lineal del BPLSCP queda entonces como:

$$\text{Min } \sum_{j \in J} x_j \quad (1.21)$$

$$\text{s.a. } \sum_{j \in N_i} x_j \geq b_i \quad \forall i \in I \quad (1.22)$$

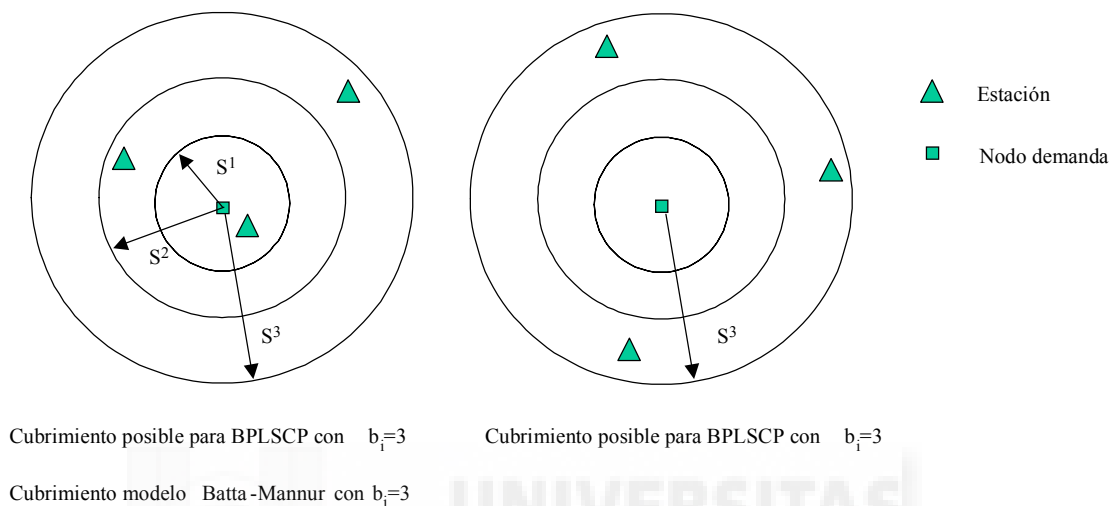
$$x_j \geq 0, \quad x_j \in Z \quad \forall j \in J. \quad (1.23)$$

*Batta y Mannur* (1990) generalizan la formulación lineal del BPLSCP presentando un modelo donde el cubrimiento es alcanzado cuando existen múltiples unidades de emergencia dispuestas de forma escalonada. Denotando por  $b_i$  la cota superior del número de unidades necesarias para responder una solicitud de servicio del nodo  $i$  y  $S^k$  el tiempo estándar para una llamada requiriendo  $k$  estaciones de servicio, se asume la siguiente relación:

$$\text{si } 1 \leq k_1 < k_2 \leq b_i, \text{ entonces } S^{k_1} < S^{k_2}.$$



En la Figura 1. 1 podemos observar el proceso de generalización efectuado respecto del BPLSCP, escalándose la ubicación de las unidades de emergencia necesarias para distintos niveles de tiempo  $S^k$ . Para  $S^3$  debemos ubicar 3 estaciones,  $b_i=3$ , las dos configuraciones presentadas serían soluciones para el BPLSCP, pero una de ellas no escala la posición de estaciones al aumentar el tiempo.



**Figura 1. 1. Diferencia entre cubrimiento BPLSCP y el cubrimiento del modelo de Batta-Mannur.**

La modelización propuesta por *Batta y Mannur* es como sigue:

$$\text{Min } \sum_{j \in J} x_j \tag{1. 24}$$

$$\text{s.a. } \sum_{j \in N_i^k} x_j \geq k \quad \forall i \in I, \quad k = 1, 2, \dots, b_i \tag{1. 25}$$

$$x_j \geq 0, \quad x_j \in Z \quad \forall j \in J, \tag{1. 26}$$

donde:

$N_i^k = \{j \in J / d_{ij} \leq S^k\}$  es el conjunto de ubicaciones potenciales que pueden servir al nodo  $i$  dentro del tiempo máximo  $S^k$ . Obsérvese que este modelo requiere la definición de una red de estaciones de servicio más densa que el BPLSCP.

Recientemente, *Ball y Lin* (1993) han formulado una nueva versión probabilística del LSCP. Las hipótesis de su modelo son:

- Las llamadas de emergencia se distribuyen según una distribución de Poisson.
- $T$  es una cota superior para el tiempo de servicio.

El modelo necesita las siguientes definiciones:

- R es un subconjunto de ubicaciones potenciales de estaciones en la red.
- $D(R)$  es el número de llamadas que llegan a  $\bigcup_{j \in R} B_j$  durante el intervalo  $(t_0-T, t_0)$ , donde  $B_j = \{i \in I / d_{ji} \leq S\}$  es el conjunto de nodos demanda que pueden ser atendidos por la estación j.
- $NA(i) = \bigcap_{R \subseteq N_i} \{D(R) \geq \sum_{j \in R} x_j\}$  es el suceso aleatorio de que el nodo i no sea atendido en el tiempo  $t_0$  dentro de S, es decir, que el número de llamadas que llegan en el intervalo anterior de longitud T sea mayor que el número de vehículos ubicados.

La cadena de desigualdades utilizada para la modelización se basa en establecer una cota superior de la probabilidad de no ser atendido para cada nodo demanda y restringir dicha cota a ser menor que  $1-\alpha$ .

$$P\left(\bigcap_{R \subseteq N_i} \{D(R) \geq \sum_{j \in R} x_j\}\right) \leq \prod_{j \in N_i} P(D(j) \geq x_j) \leq 1-\alpha \quad \forall i \in I, \quad (1.27)$$

donde:

$D(j)$  es el número de llamadas que llegan a  $B_j$  durante el intervalo  $(t_0-T, t_0)$  y que sigue una distribución de Poisson con media  $\lambda_j = \frac{T}{24} \left(\sum_{i \in B_j} f_i\right)$  llamadas por hora.

La primera desigualdad se obtiene al utilizar que las llamadas generadas son basadas en una distribución de probabilidad con la propiedad “nuevo mejor que usado”, i.e., una variable aleatoria no negativa X cumpliendo que

$$P(X \geq u+v \mid X \geq u) \leq P(X \geq v) \quad u > 0, v > 0.$$

Redefiniendo  $x_j$ , el número de vehículos a ubicar en la estación j, con las variables

$$x_{jk} = \begin{cases} 1, & \text{si } k \text{ vehículos son ubicados en la estación } j \\ 0, & \text{en caso contrario,} \end{cases}$$

y tomando L como el número máximo de vehículos que se pueden ubicar en cada estación podemos reescribir (1.27) como

$$\prod_{j \in N_i} \prod_{k=1}^L P(D(j) \geq k)^{x_{jk}} \leq 1-\alpha \quad \forall i \in I \quad (1.28)$$

$$\sum_{k=1}^L x_{jk} \leq 1 \quad \forall j \in J. \quad (1.29)$$

La restricción (1. 28) impone la condición de que la probabilidad de que una llamada de emergencia desde un nodo demanda cualquiera no sea atendida debe ser inferior a la cota  $1-\alpha$ . Las restricciones (1. 29) son necesarias para convertir la variable entera  $x_j$  en  $L$  variables binarias y afirma que en cada estación o no se ubica ningún vehículo o se ubica un número  $k$  de vehículos. Tomando logaritmos en ambos lados y cambiando los signos, (1. 28) puede ser transformada en una restricción lineal. La formulación completa del Poisson Reliability Location Set Covering Problem (PRLSCP) de *Ball y Lin* (1993) es

$$\text{Min } \sum_{j \in J} \sum_{k=1}^L kx_{jk} \quad (1. 30)$$

$$\text{s.a. } \sum_{j \in N_i} \sum_{k=1}^L -\log(P(D(j) \geq k))x_{jk} \geq -\log(1-\alpha) \quad \forall i \in I \quad (1. 31)$$

$$\sum_{k=1}^L x_{jk} \leq 1 \quad \forall j \in J \quad (1. 32)$$

$$x_{jk} \in \{0,1\} \quad \forall j \in J, \quad k = 1,2,\dots,L. \quad (1. 33)$$

La última versión probabilística del LSCP, debida a *Marianov y ReVelle* (1994), se conoce como el Queueing Probabilistic Location Set Covering Problem (QPLSCP). En esta versión se modeliza el comportamiento en cada entorno de un nodo demanda como una cola con llegadas distribuidas Poisson, tiempo de servicio exponencial y pérdida de llamadas cuando el sistema está saturado.

La probabilidad de que todos los vehículos disponibles en el entorno de radio  $S$  de un nodo demanda estén ocupados puede ser calculada usando la función de probabilidad de una Poisson truncada de media  $\lambda_i/\mu_i$ :

$$P\left(\begin{array}{l} \text{nodo } i \text{ no sea atendido} \\ \text{dentro del tiempo } S \end{array}\right) = \frac{\frac{1}{k!} \left(\frac{\lambda_i}{\mu_i}\right)^k}{1 + \left(\frac{\lambda_i}{\mu_i}\right) + \frac{1}{2!} \left(\frac{\lambda_i}{\mu_i}\right)^2 + \dots + \frac{1}{k!} \left(\frac{\lambda_i}{\mu_i}\right)^k} \quad \forall i \in I, \quad (1. 34)$$

donde:

$k$  es el número de unidades de emergencia que pueden atender al nodo demanda  $i$  en no más del tiempo estándar  $S$ ;

$\lambda_i$  es la media de llamadas de emergencia, por día, en el entorno de tiempo  $S$  del nodo

$i$ , que puede ser calculada como  $\sum_{k \in M_i} f_k \quad \forall i \in I$ ;

$1/\mu_i$  es el tiempo medio de servicio necesario para atender una llamada, en días, que puede ser calculado como  $\frac{\bar{t}}{24}$ .

Las restricciones de cubrimiento probabilístico basadas en la distribución Poisson truncada, pueden ser escritas como

$$\frac{\frac{1}{\left(\sum_{j \in N_i} x_j\right)!} \left(\frac{\lambda_i}{\mu_i}\right)^{\sum_{j \in N_i} x_j}}{1 + \left(\frac{\lambda_i}{\mu_i}\right) + \frac{1}{2!} \left(\frac{\lambda_i}{\mu_i}\right)^2 + \dots + \frac{1}{\left(\sum_{j \in N_i} x_j\right)!} \left(\frac{\lambda_i}{\mu_i}\right)^{\sum_{j \in N_i} x_j}} \leq 1 - \alpha \quad \forall i \in I. \quad (1.35)$$

Aunque (1.35) no tiene una expresión lineal analítica equivalente, *Marianov y ReVelle* (1994) han encontrado la siguiente expresión lineal numérica equivalente:

$$\sum_{j \in N_i} x_j \geq b_i \quad \forall i \in I, \quad (1.36)$$

donde:

$$b_i = \min \left\{ n \in \mathbb{N} / \frac{\frac{1}{n!} \left(\frac{\lambda_i}{\mu_i}\right)^n}{1 + \left(\frac{\lambda_i}{\mu_i}\right) + \frac{1}{2!} \left(\frac{\lambda_i}{\mu_i}\right)^2 + \dots + \frac{1}{n!} \left(\frac{\lambda_i}{\mu_i}\right)^n} \leq 1 - \alpha \right\}. \quad (1.37)$$

La formulación lineal del QPLSCP es análoga a la del BPLSCP, (1.21), (1.22) y (1.23), excepto en el cálculo de  $b_i$  que utiliza (1.37) en lugar de (1.20).

Los modelos anteriores presentaban la característica de la obligatoriedad de que cada nodo demanda fuese cubierto determinística o probabilísticamente. En la siguiente sección presentaremos modelos donde se optimiza el cubrimiento, de manera determinista o probabilística para un número de vehículos fijado a priori.

### 1.3. El problema del cubrimiento maximal.

El LSCP requiere que cada nodo sea cubierto al menos una vez, circunstancia que muchas veces no puede ser satisfecha, dado que el número de estaciones que deben ser ubicadas para verificar la restricción es excesivo. Además, en muchas ocasiones, se ubican estaciones que cubren nodos periféricos con muy poca demanda. Para evitar estos inconvenientes, *Church y Reville* en 1974, formulan el Maximal Covering Location Problem (MCLP) que maximiza el número de llamadas que son atendidas por al menos una estación cuando se dispone de  $p$  estaciones, donde  $p$  es menor o igual que el valor óptimo del LSCP. En un artículo de ese mismo año, *White y Case* formulan un caso particular del MCLP, donde se maximiza el número de nodos cubiertos con  $p$  estaciones. El MCLP se modeliza como

$$\text{Max } \sum_{i \in I} f_i y_i \quad (1.38)$$

$$\text{s.a. } \sum_{j \in N_i} x_j \geq y_i \quad \forall i \in I \quad (1.39)$$

$$\sum_{j \in J} x_j = p \quad (1.40)$$

$$x_j, y_i \in \{0,1\} \quad \forall j \in J, \forall i \in I, \quad (1.41)$$

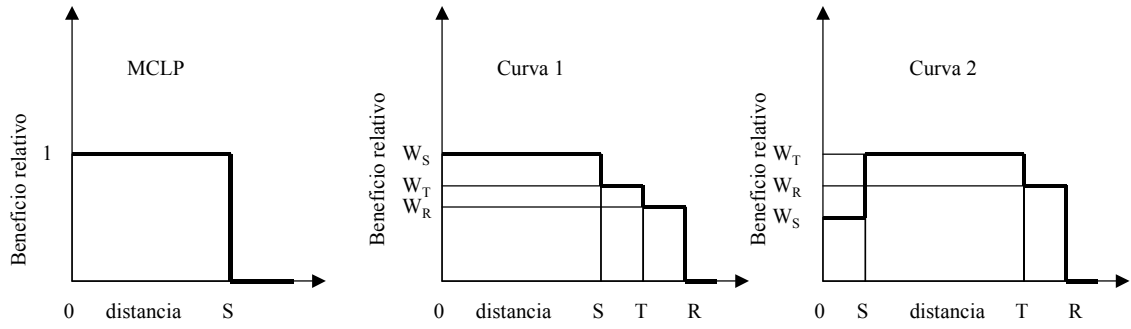
donde las variables y parámetros no definidos todavía son:

$p$  es el número de estaciones a ubicar;

$$y_i = \begin{cases} 1, & \text{si el nodo } i \text{ es cubierto} \\ 0, & \text{en caso contrario.} \end{cases}$$

El objetivo (1.38) maximiza el número de llamadas que son atendidas por una estación ubicada a menos de la distancia  $S$  prefijada. Las desigualdades (1.39) exigen que un nodo  $i$  no puede ser cubierto mientras no se haya posicionado al menos una estación en un lugar que diste menos de  $S$  unidades de distancia. Con la igualdad (1.40) se asegura que el número de estaciones a ubicar sea  $p$ . Por último, (1.41) explicita el carácter binario de las variables.

En 1983, *Church y Roberts* formulan el Weighted Benefit Maximal Covering Location problem (WBMCLP), donde critican la curva de beneficio relativo del MCLP, debido a que nodos cubiertos en el rango de 0 a  $S$ , presentan un beneficio relativo del 100% y en cambio, nodos alejados una distancia  $S+\epsilon$  no son considerados cubiertos y el beneficio, por tanto, es cero.



**Figura 1. 2 Diferentes curvas de beneficio en función de la distancia.**

La curva del beneficio relativo del MCLP y otras curvas se presentan en la Figura 1. 2. La curva 1 representa una función escalonada que decrece con la distancia y que permite evaluar el cubrimiento alcanzado de una manera más eficiente. La curva 2 presenta una evaluación más realista de los servicios de emergencia, considerando como negativo estar demasiado cerca de ellos por las incomodidades que produce, pero valorando adecuadamente su cercanía. La modelización del WBMCLP es como sigue

$$\text{Max} \quad W_S \sum_{i \in I} f_i y_i^{0-S} + W_T \sum_{i \in I} f_i y_i^{S-T} + W_R \sum_{i \in I} f_i y_i^{T-R} \quad (1.42)$$

$$\text{s.a.} \quad \sum_{j \in N_i} x_j \geq y_i^{0-S} \quad \forall i \in I \quad (1.43)$$

$$\sum_{j \in N_i^{S-T}} x_j \geq y_i^{S-T} \quad \forall i \in I \quad (1.44)$$

$$\sum_{j \in N_i^{T-R}} x_j \geq y_i^{T-R} \quad \forall i \in I \quad (1.45)$$

$$y_i^{0-S} + y_i^{S-T} + y_i^{T-R} \leq 1 \quad \forall i \in I \quad (1.46)$$

$$x_j + y_i^{S-T} \leq 1 \quad \forall i \in I, j \in N_i \quad (1.47)$$

$$x_j + y_i^{T-R} \leq 1 \quad \forall i \in I, j \in N_i^{S-T} \quad (1.48)$$

$$\sum_{j \in J} x_j = p \quad (1.49)$$

$$x_j, y_i^{0-S}, y_i^{S-T}, y_i^{T-R} \in \{0,1\} \quad \forall j \in J, \forall i \in I, \quad (1.50)$$

donde:

$N_i^{S-T} = \{j \in J / S < d_{ij} \leq T\}$  es el conjunto de ubicaciones potenciales que pueden servir al nodo  $i$  entre los tiempos máximos  $S$  y  $T$ ;

$N_i^{T-R} = \{j \in J / T < d_{ij} \leq R\}$  es el conjunto de ubicaciones potenciales que pueden servir al nodo  $i$  entre los tiempos máximos  $T$  y  $R$ ;

$$y_i^{0-S}, y_i^{S-T}, y_i^{T-R} = \begin{cases} 1, & \text{si el nodo } i \text{ es cubierto, respectivamente, por al menos una estación,} \\ & \text{que no dista más de } S, \text{ entre } S \text{ y } T \text{ o entre } T \text{ y } R \text{ unidades de distancia.} \\ 0, & \text{en caso contrario.} \end{cases}$$

La función objetivo (1. 42) permite establecer mediante los pesos  $W_S$ ,  $W_T$  y  $W_R$ , diferentes curvas de beneficio relativo. Las desigualdades (1. 43), (1. 44) y (1. 45) condicionan que un nodo  $i$  no puede ser cubierto, mientras no se haya posicionado al menos una estación en un lugar que diste menos de  $S$ , entre  $S$  y  $T$  o entre  $T$  y  $R$  unidades de distancia. Por su parte (1. 46) asegura que el cubrimiento de un nodo  $i$  será bien de  $[0,S]$ , bien de  $]S,T]$  o de  $]T,R]$ . Las restricciones (1. 47) y (1. 48) son necesarias para poder asegurar que cada nodo  $i$  es cubierto por la estación más próxima. De hecho, el primer bloque de desigualdades impide contar el cubrimiento de  $S$  a  $T$  si hay una estación ubicada dentro de  $S$  y el segundo bloque descarta el cubrimiento de un nodo de  $T$  a  $R$  si hay una estación ubicada entre  $S$  y  $T$ . Con la igualdad (1. 49) se asegura que el número de estaciones a ubicar sea exactamente  $p$ . Finalmente, (1. 50) indica el carácter binario de las variables.

El MCLP no tiene en cuenta si un nodo demanda es cubierto más de una vez. En 1982, *Storbeck* formula el Goal Location Covering Problem (GLCP), basado en la programación por metas, cuyo objetivo es maximizar el número de llamadas que tiene una estación ubicada a menos de la distancia  $S$  prefijada y, adicionalmente, maximizar el cubrimiento múltiple. Su modelo es como sigue:

$$\text{Max } W \sum_{i \in I} f_i y_i + \sum_{i \in I} z_i \tag{1. 51}$$

$$\text{s.a. } \sum_{j \in N_i} x_j = y_i + z_i \quad \forall i \in I \tag{1. 52}$$

$$\sum_{j \in J} x_j = p \tag{1. 53}$$

$$x_j, y_i \in \{0,1\} \quad \forall j \in J, \forall i \in I \tag{1. 54}$$

$$z_i \geq 0 \quad \forall i \in I, \tag{1. 55}$$

donde:

$z_i$  es el número de cubrimientos adicionales del nodo  $i$ , y

$W$  representa un peso suficientemente grande que propicia que en la función objetivo (1. 51) primero se maximice el cubrimiento  $y$ , en segundo lugar, se optimice el cubrimiento múltiple.

Las restricciones (1. 52) determinan que el número de servidores del nodo  $i$  es igual al primer cubrimiento más el cubrimiento múltiple adicional. Con la igualdad (1. 53) se asegura que el número de estaciones a ubicar sea  $p$ . Por último, (1. 54) y (1. 55) determinan el carácter de las variables.

Si bien el GLCP constituye el primer modelo de cubrimiento que aúna dos aspectos hasta entonces inconexos, también es cierto que presenta cierta falta de flexibilidad. De hecho, busca la solución con cubrimiento maximal que, subsidiariamente, ofrezca el mayor cubrimiento múltiple. Para salvar este inconveniente, *Storbeck y Vohra* (1988) han diseñado el Natural Slack Covering Problem (NSCP), que recurriendo a la programación multiobjetivo, permite el balanceo entre cubrimiento maximal y múltiple.

$$\text{Max} \quad \sum_{i \in I} f_i y_i \quad (1. 56)$$

$$\text{Max} \quad \sum_{i \in I} f_i z_i \quad (1. 57)$$

$$\text{s.a.} \quad \sum_{j \in N_i} x_j = y_i + z_i \quad \forall i \in I \quad (1. 58)$$

$$\sum_{j \in J} x_j = p \quad (1. 59)$$

$$z_i - k_i y_i \leq 0 \quad \forall i \in I \quad (1. 60)$$

$$x_j, y_i \in \{0,1\} \quad \forall j \in J, \forall i \in I \quad (1. 61)$$

$$z_i \geq 0 \quad \forall i \in I, \quad (1. 62)$$

donde:

$k_i$  es el número entero que limita la cobertura adicional en cada punto demanda.

Para homogeneizar la formulación (1. 56) maximiza la demanda de llamadas cubiertas una vez y (1. 57) maximiza la suma ponderada por la cantidad de llamadas de cada nodo de los cubrimientos adicionales en cada punto demanda. Las restricciones (1. 58) determinan que el número de cubridores del nodo  $i$  es igual al primer cubrimiento más el cubrimiento múltiple adicional. Con la igualdad (1. 59) se asegura que el número de estaciones a ubicar sea  $p$ . Las restricciones (1. 60) impiden el cubrimiento adicional en el caso de que no haya primer



cubrimiento y si lo hay aseguran que el nodo demanda no será cubierto adicionalmente más de  $k_i$  veces. Por último, (1. 61) y (1. 62) explicitan el carácter de las variables.

Desafortunadamente, la estación más cercana no siempre está disponible cuando una llamada llega al sistema, y por tanto, el cubrimiento maximal alcanzado por una configuración de unidades de emergencia no permite una evaluación realista del sistema. Reconociendo estas deficiencias, *Daskin* (1983) formula el Maximum Expected Covering Location Problem, denotado por MEXCLP, donde maximiza el cubrimiento esperado por el sistema de emergencia. Dicho cubrimiento esperado de cada nodo se modeliza suponiendo que las unidades de emergencia actúan independientemente y suponiendo conocida la fracción de ocupación media  $\rho$  del sistema. Matemáticamente:

$$\text{Max} \quad \sum_{i \in I} f_i \left( \sum_{k=1}^p (1-\rho)\rho^{k-1} y_{ik} \right) \quad (1. 63)$$

$$\text{s.a.} \quad \sum_{j \in N_i} x_j = \sum_{k=1}^p y_{ik} \quad \forall i \in I \quad (1. 64)$$

$$\sum_{j \in J} x_j = p \quad (1. 65)$$

$$x_j \in \{0,1,2,\dots,p\} \quad \forall j \in J \quad (1. 66)$$

$$y_{ik} \in \{0,1\} \quad \forall i \in I \quad k = 1,2,\dots,p, \quad (1. 67)$$

donde:

$x_j$  es el número de vehículos ubicado en cada estación  $j$ ;

$y_{ik} = \begin{cases} 1, & \text{si el nodo } i \text{ es cubierto por al menos } k \text{ unidades de emergencia;} \\ 0, & \text{en caso contrario;} \end{cases}$

$\rho = \frac{\bar{t} \sum_{i \in I} f_i}{24p}$  es la estimación de la fracción de ocupación media del sistema.

La función objetivo (1. 63) representa el número esperado de llamadas servidas, siendo el término dentro del paréntesis la probabilidad de que una llamada del nodo  $i$  sea atendida por el sistema, dependiendo de la cantidad de servidores disponibles dentro del tiempo estándar  $S$ . Las restricciones (1. 64) imponen que un nodo  $i$  es cubierto por al menos  $k$  unidades de emergencia,  $y_{ik}=1$  si, y sólo si,  $k$  unidades de emergencia son ubicadas en el entorno de radio  $S$  del nodo demanda  $i$ . La igualdad (1. 65) determina el número de vehículos a ubicar en el sistema. (1. 66) y (1. 67) determinan el carácter entero de las variables del modelo.

Reconociendo el hecho de que el MEXCLP consume mucho tiempo computacional en la obtención de la solución óptima, *Daskin* en el mismo trabajo, desarrolla un heurístico para su resolución. Posteriormente, *Saydam y NcKnew* (1985) reformulan el MEXCLP como un problema de cubrimiento esperado maximal no lineal, que permite obtener la solución óptima rápidamente y realizar extensiones para asegurar cierto nivel de cubrimiento para cada nodo demanda.

*Batta et al.* (1989) constatan la inexactitud del MEXCLP debida a la simplificación asumida por el modelo, la independencia entre los servidores. Su versión, denotada por Adjusted-MEXCLP (AMEXCLP), simplemente sustituye (1. 63) por

$$\text{Max} \sum_{i \in I} f_i \left( \sum_{k=1}^p Q(p, \rho, k-1)(1-\rho)\rho^{k-1} y_{ik} \right) \quad (1. 68)$$

donde los  $Q(p, \rho, k)$  son los factores  $Q$  de *Larson* (1975) que se presentarán a continuación en la sección 1.5 dedicada a la modelización de los sistemas de emergencia mediante colas.

Más recientemente, *Repede y Bernardo* (1994) extienden el MEXCLP, denotado por TIMEXCLP, al objeto de incluir diferentes fracciones de ocupación media, debido a su variabilidad a lo largo de un día.

*ReVelle y Hogan* (1989a) extienden el uso de las restricciones de cubrimiento probabilístico a la situación donde el cubrimiento con fiabilidad  $\alpha$  es deseado para el máximo de número de nodos demanda desde donde se producen más llamadas. Este nuevo modelo es llamado Maximum Availability Location Problem (MALP), basado en el BPLSCP, y su formulación es

$$\text{Max} \sum_{i \in I} f_i y_{i, b_i} \quad (1. 69)$$

$$\text{s.a.} \sum_{j \in N_i} x_j \geq \sum_{k=1}^{b_i} y_{ik} \quad \forall i \in I \quad (1. 70)$$

$$\sum_{j \in J} x_j = p \quad (1. 71)$$

$$y_{ik} \leq y_{i, k-1} \quad \forall i \in I \quad k = 2, \dots, b_i \quad (1. 72)$$

$$x_j \in \{0, 1, 2, \dots, p\} \quad \forall j \in J \quad (1. 73)$$

$$y_{ik} \in \{0, 1\} \quad \forall i \in I \quad k = 1, 2, \dots, b_i, \quad (1. 74)$$

donde:

$$b_i = \min \left\{ n \in \mathbb{N} / \left( \frac{\bar{t} \sum_{k \in M_i} f_k}{24n} \right)^n \leq 1 - \alpha \right\};$$

$$y_{i,b_i} = \begin{cases} 1, & \text{si el nodo } i \text{ es cubierto por al menos } b_i \text{ unidades de emergencia.} \\ 0, & \text{en caso contrario.} \end{cases}$$

La función objetivo (1. 69) representa el número de llamadas atendidas con fiabilidad  $\alpha$ , aunque en el modelo se obtiene el número de llamadas de cada nodo que son cubiertas al menos  $b_i$  veces. Las restricciones (1. 70) afirman que el nodo  $i$  es cubierto  $b_i$  veces sólo si hay al menos  $b_i$  unidades de emergencia estacionadas dentro del entorno de radio  $S$ . El conjunto de restricciones (1. 72) asegura que el nodo no será cubierto  $k$  veces si no es cubierto  $k-1$  veces.

Más recientemente, *Marianov y ReVelle* (1996) han formulado el Queueing Maximal Availability Location Problem (QMALP), incorporando al MALP el comportamiento en cada entorno de un nodo demanda de una cola con llegadas distribuidas Poisson, tiempo de servicio exponencial y pérdida de llamadas cuando el sistema está saturado.

#### 1.4. El problema del cubrimiento maximal con diferenciación de vehículos.

Una nueva serie de modelos han sido formulados para intentar reflejar con mayor realismo el comportamiento de un sistema de emergencias. El primer paso fue la generalización del cubrimiento maximal a la protección de incendios realizada por *Schilling et al.* (1979). Su modelo FLEET (Facility Location, Equipment Emplacement Technique model) localiza un número limitado de coches-cisterna y coches con escalera, así como las estaciones que los albergan. El objetivo de este modelo es maximizar el número de llamadas atendidas, simultáneamente, por un coche-cisterna dentro de la distancia máxima prefijada para este tipo de vehículos,  $S^E$ , y por un coche con escalera dentro de la distancia máxima  $S^T$ . Matemáticamente, puede ser expresado como:

$$\text{Max} \quad \sum_{i \in I} f_i y_i \quad (1.75)$$

$$\text{s.a.} \quad \sum_{j \in N_i^E} x_j^E \geq y_i \quad \forall i \in I \quad (1.76)$$

$$\sum_{j \in N_i^T} x_j^T \geq y_i \quad \forall i \in I \quad (1.77)$$

$$x_j^E \leq x_j^S \quad \forall j \in J \quad (1.78)$$

$$x_j^T \leq x_j^S \quad \forall j \in J \quad (1.79)$$

$$\sum_{j \in J} x_j^E = p^E \quad (1.80)$$

$$\sum_{j \in J} x_j^T = p^T \quad (1.81)$$

$$\sum_{j \in J} x_j^S = p^S \quad (1.82)$$

$$x_j^E, x_j^T, x_j^S, y_i \in \{0,1\} \quad \forall j \in J, i \in I, \quad (1.83)$$

donde:

$S^E$  es la distancia máxima permitida para los coches-cisterna;

$N_i^E = \{j \in J / d_{ij} \leq S^E\}$  es el conjunto de ubicaciones potenciales que pueden servir al nodo  $i$  dentro del tiempo máximo o distancia permitida para los coches-cisterna;

$S^T$  es la distancia máxima permitida para los coches con escaleras extensibles;

$N_i^T = \{j \in J / d_{ij} \leq S^T\}$  es el conjunto de ubicaciones potenciales que pueden servir al nodo  $i$  dentro del tiempo máximo o distancia permitida para los coches con escaleras extensibles;

$p^S, p^E, p^T$  son, respectivamente, el número de estaciones a ubicar, la cantidad de coches-cisterna y la de coches con escaleras extensibles;

$$x_j^S, x_j^E, x_j^T = \begin{cases} 1, & \text{si en el lugar } j \text{ son ubicados, respectivamente, una estación, un} \\ & \text{coche - cisterna o un coche - escalera} \\ 0, & \text{en caso contrario.} \end{cases}$$

Las restricciones (1.76) y (1.77) determinan que un nodo se considera cubierto si al menos un coche-cisterna y al menos un coche-escalera están, respectivamente, dentro de las distancias  $S^E$  y  $S^T$ . El conjunto de restricciones (1.78) y (1.79) permiten ubicar los distintos tipos de vehículos sólo donde se ha localizado una estación de bomberos. Las igualdades (1.

80), (1. 81) y (1. 82) determinan, respectivamente, el número de coches-cisterna, coches-escalera y estaciones a ubicar. Por último, (1. 83) determina el carácter binario de las variables del modelo.

La extensión probabilística del modelo FLEET fue propuesta por *ReVelle y Marianov* en 1991. En este modelo se maximiza el número de llamadas que tiene un coche-cisterna, disponible dentro de la distancia máxima prefijada para este tipo de vehículos,  $S^E$ , con fiabilidad  $\alpha$  y un coche con escalera extensible, disponible dentro de la distancia máxima  $S^T$  con fiabilidad  $\alpha$ , suponiendo independencia entre ambos tipos de servidores y suponiendo disponibilidad conjunta. La formulación completa del modelo es la siguiente:

$$\text{Max } \sum_{i \in I} f_i y_i \quad (1. 84)$$

$$\text{s.a. } \sum_{j \in N_i^E} x_j^E \geq \sum_{k=1}^{e_i-1} y_{ik}^E + y_i \quad \forall i \in I \quad (1. 85)$$

$$\sum_{j \in N_i^T} x_j^T \geq \sum_{k=1}^{t_i-1} y_{ik}^T + y_i \quad \forall i \in I \quad (1. 86)$$

$$y_i \leq y_{i(e_i-1)}^E \quad \forall i \in I \quad (1. 87)$$

$$y_i \leq y_{i(t_i-1)}^T \quad \forall i \in I \quad (1. 88)$$

$$y_{ik}^E \leq y_{i(k-1)}^E \quad \forall i \in I, \quad k = 2, 3, \dots, e_i - 1 \quad (1. 89)$$

$$y_{ik}^T \leq y_{i(k-1)}^T \quad \forall i \in I, \quad k = 2, 3, \dots, t_i - 1 \quad (1. 90)$$

$$x_j^E \leq x_j^S \quad \forall j \in J \quad (1. 91)$$

$$x_j^T \leq x_j^S \quad \forall j \in J \quad (1. 92)$$

$$\sum_{j \in J} x_j^E = p^E \quad (1. 93)$$

$$\sum_{j \in J} x_j^T = p^T \quad (1. 94)$$

$$\sum_{j \in J} x_j^S = p^S \quad (1. 95)$$

$$x_j^E, x_j^T, x_j^S \in \{0,1\} \quad \forall j \in J \quad (1. 96)$$

$$y_i, y_{ik}^E, y_{ik}^T \in \{0,1\} \quad \forall i \in I, \quad k = 1, 2, \dots, e_i \text{ ó } t_i, \quad (1. 97)$$

donde:

$$y_{ik}^E, y_{ik}^T = \begin{cases} 1, & \text{si el nodo demanda } i \text{ es cubierto al menos por } k \\ & \text{coches - cisterna, o por } k \text{ coches - escalera, respectivamente.} \\ 0, & \text{en caso contrario.} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{si el nodo demanda } i \text{ es cubierto por un coche - cisterna y} \\ & \text{por un coche - escalera, ambos con fiabilidad } \alpha. \\ 0, & \text{en caso contrario.} \end{cases}$$

$e_i$  = número de coches-cisterna que deben ser localizados dentro de la distancia  $S^E$  del nodo  $i$  para asegurar que es cubierto por un coche-cisterna con fiabilidad  $\alpha$ .

$t_i$  = número de coches-escalera que deben ser localizados dentro de la distancia  $S^T$  del nodo  $i$  para asegurar que es cubierto por un coche-escalera con fiabilidad  $\alpha$ .

El conjunto de restricciones (1. 85) indica que el número de veces que cada nodo  $i$  es cubierto por coches-cisterna, es menor o igual que el número de coches-cisterna que se localizan en el entorno de radio  $S^E$  de  $i$ . El conjunto de restricciones (1. 86) afirma que el número de veces que cada nodo  $i$  es cubierto por coches-escalera, es menor o igual que el número de coches-escalera que se localizan en el entorno de radio  $S^E$  de  $i$ . La variable  $y_i$  no puede tomar el valor 1, mientras que no se ubiquen en el entorno de radio  $S^E$ ,  $e_i$  coches-cisterna y en el entorno de radio  $S^T$ ,  $t_i$  coches-escalera, gracias al conjunto de restricciones (1. 87), (1. 88) (1. 89) y (1. 90). Las restricciones (1. 91), (1. 92), (1. 93), (1. 94) y (1. 95) son análogas a las del modelo FLEET. Por último, (1. 96) y (1. 97) determinan el carácter binario de las variables del modelo.

*Marianov y ReVelle* (1991) presentan el Standard Response Fire Protection Siting Problem (SRFPSP) que localiza un número limitado de coches-cisterna y coches-escalera, así como las estaciones que los albergan, con el objetivo de maximizar el número de llamadas que tienen al menos tres coches-cisterna dentro de la distancia máxima  $S^E$  y al menos dos coches-escalera dentro de la distancia máxima  $S^T$ , que es la respuesta considerada estándar por la *Insurance Services Office* para un incendio, con un volumen de flujo de agua requerido de al menos 8500 galones por minuto. El modelo matemático es el siguiente:

$$\text{Max } \sum_{i \in I} f_i y_i^{E3T2} \quad (1.98)$$

$$\text{s.a. } \sum_{j \in N_i^E} x_j^E \geq y_i^{E1} + y_i^{E2} + y_i^{E3T2} \quad \forall i \in I \quad (1.99)$$

$$\sum_{j \in N_i^T} x_j^T \geq y_i^{T1} + y_i^{E3T2} \quad \forall i \in I \quad (1.100)$$

$$y_i^{E2} \leq y_i^{E1} \quad \forall i \in I \quad (1.101)$$

$$y_i^{E3T2} \leq y_i^{E2} \quad \forall i \in I \quad (1.102)$$

$$y_i^{E3T2} \leq y_i^{T1} \quad \forall i \in I \quad (1.103)$$

$$x_j^E \leq x_j^S \quad \forall j \in J \quad (1.104)$$

$$x_j^T \leq x_j^S \quad \forall j \in J \quad (1.105)$$

$$\sum_{j \in J} x_j^E = p^E \quad (1.106)$$

$$\sum_{j \in J} x_j^T = p^T \quad (1.107)$$

$$\sum_{j \in J} x_j^S = p^S \quad (1.108)$$

$$x_j^E, x_j^T, x_j^S \in \{0,1\} \quad \forall j \in J \quad (1.109)$$

$$y_i^{E1}, y_i^{E2}, y_i^{T1}, y_i^{E3T2} \in \{0,1\} \quad \forall i \in I \quad (1.110)$$

donde:

$$y_i^{E1}, y_i^{E2}, y_i^{T1}, y_i^{E3T2} = \begin{cases} 1, & \text{si el nodo demanda } i \text{ es cubierto al menos por un} \\ & \text{coche - cisterna, por dos coches - cisterna, por un coche - escalera o} \\ & \text{por tres coches - cisterna y dos coches - escaleras, respectivamente.} \\ 0, & \text{en caso contrario.} \end{cases}$$

Las restricciones (1. 101) y (1. 102) impiden el cubrimiento por el segundo coche-cisterna hasta que no esté cubierto por el primero y que el tercer cubrimiento no puede alcanzarse mientras no se alcance el segundo. Estas dos restricciones en combinación con (1. 99) indican que el primer cubrimiento requiere uno o más coches-cisterna, el segundo cubrimiento requiere dos o más y el tercero sólo es alcanzado por la presencia de 3 o más coches-cisterna. La restricción (1. 103) en combinación con (1. 100) indican que el primer cubrimiento requiere uno o más coches-escalera y el segundo cubrimiento requiere dos o más. Las restricciones (1. 104), (1. 105), (1. 106), (1. 107) y (1. 108) son análogas a aquellas en el modelo

FLEET. Por último, (1. 109) y (1. 110) determinan el carácter binario de las variables del modelo.

La extensión probabilística del modelo SRFSP fue propuesta por *Marianov y ReVelle* en 1992. En este modelo se maximiza el número de llamadas que tienen tres coches-cisterna disponibles dentro de la distancia máxima prefijada para este tipo de vehículos,  $S^E$ , con fiabilidad  $\alpha$  y dos coches con escalera extensible disponibles dentro de la distancia máxima  $S^T$  con fiabilidad  $\alpha$ , suponiendo independencia entre ambos tipos de servidores.

Recientemente, *ReVelle y Snyder* (1995) presentan un modelo determinista que integra los servicios de emergencias médicas con los departamentos de bomberos, permitiendo la ubicación de ambulancias en las estaciones de bomberos.

Como colofón a las secciones 1.2, 1.3, 1.4, donde se han revisado los diferentes modelos de cubrimiento aplicados a la localización de servicios de emergencia podemos citar diferentes revisiones del tema como las realizadas por *ReVelle* (1989), *Schilling et Al.* (1993) y *Marianov y ReVelle* (1995).

En la siguiente sección presentaremos un nuevo enfoque que permitirá describir con mayor verosimilitud el comportamiento de los sistemas de emergencia dada su extrema complejidad, es decir, se analizarán modelos descriptivos que permitirán obtener medidas muy precisas de la actuación de diferentes configuraciones espaciales de los vehículos.



### 1.5. Modelización de servicios de emergencia como un sistema de colas.

A partir de hipótesis apropiadas respecto a la llegada de llamadas y de tiempos de servicios, se ha analizado, dentro de la literatura especializada, un sistema de emergencia con  $N$  vehículos como un proceso de Markov con un número finito de estados continuos en el tiempo. Recordemos que con la notación de *Kendall (1953)*, una cola se representa con una cadena de símbolos  $A/B/X/Y/Z$ , donde

$A$  indica la distribución del tiempo entre llegadas, usando

$M$  para distribución exponencial

$E_k$  para distribución Erlang tipo  $k$  ( $k=1,2,\dots$ )

$G$  para distribución general

$B$  es la distribución de probabilidad para el tiempo de servicio, usando

$M$  para una exponencial

$E_k$  para una distribución Erlang tipo  $k$  ( $k=1,2,\dots$ )

$G$  para una distribución general

$X$  es el número de servidores,

$Y$  es la capacidad del sistema, y

$Z$  es la forma en que los usuarios son atendidos, usando

FIFO para First In, First Out (el primero en llegar es el primero en ser atendido)

LIFO para Last In, First Out (el último en llegar es el primero en ser atendido)

SIRO para Service In Random Order (atendidos aleatoriamente).

En este contexto, *Larson (1974)*, presenta el Hypercube Queueing Model (HQM), una generalización de una cola  $M/M/N/N/FIFO$ , que permite la identificación de los vehículos que están ocupados. El modelo supone que los  $I$  nodos demanda generan llamadas de emergencia independientes siguiendo una distribución de Poisson con media  $f_i$  (número de llamadas por unidad de tiempo). Por tanto, la media de llamadas por unidad de tiempo en el total del sistema es:

$$\lambda = \sum_{i \in I} f_i \quad (1.111)$$

La media del tiempo del servicio es  $1/\mu$ , independiente de la identidad del servidor, de la localización del demandante y de la historia del sistema. Cada servidor tiene dos posibles

estados (libre u ocupado), por tanto el sistema tiene  $2^N$  estados, que se pueden indexar lexicográficamente de 0 a  $2^N-1$ . Definiendo:

$$e(k, j) = \begin{cases} 1, & \text{si el servidor } j \text{ está libre cuando el sistema está en el estado } k. \\ 0, & \text{si el servidor } j \text{ está ocupado cuando el sistema está en el estado } k. \end{cases}$$

para  $k=0,1, \dots, 2^N-1$  y  $j=1,2, \dots, N$ , podemos representar cada estado  $k$  como:

$$k = e(k, N) 2^{N-1} + e(k, N-1) 2^{N-2} + \dots + e(k, j) 2^{j-1} + \dots + e(k, 2) 2 + e(k, 1) 2^0,$$

y la secuencia ordenada de dígitos binarios  $(e(k, N), e(k, N-1), \dots, e(k, 2), e(k, 1))$  nos permite conocer que servidores están libres u ocupados en el estado  $k$ . Si al producirse una llamada del nodo  $i$  se le asigna por criterios de proximidad geográfica un vehículo  $j$  libre a dicha emergencia estando el sistema en el estado  $k$ , entonces denotaremos a esta asignación como  $a(k, i) = j$ .

La probabilidad en el equilibrio de que el sistema esté en el estado  $k$ ,  $P(k)$ , se obtiene igualando el flujo probabilístico hacia diferentes estados del sistema partiendo del estado  $k$  al flujo probabilístico que llega de diferentes estados del sistema hacia el estado  $k$ :

$$P(k) \left[ \lambda \delta + \sum_{j: e(k, j)=1} \mu \right] = \sum_{j: e(k, j)=1} P(k - 2^{j-1}) \sum_{i: a(k-2^{j-1}, i)=j} f_i + \sum_{j: e(k, j)=0} P(k + 2^{j-1}) \mu \quad \forall k = 0, 1, \dots, 2^N - 1, \quad (1. 112)$$

donde:

$$\delta = \begin{cases} 1, & \text{si al menos un servidor está libre.} \\ 0, & \text{si todos los servidores están ocupados.} \end{cases}$$

El lado izquierdo de las ecuaciones de equilibrio (1. 112) representa los estados a los que podemos llegar si se producen nuevas llamadas o si terminan su servicio los vehículos ocupados en el estado  $k$ . El primer término del lado derecho de (1. 112) presenta todos los estados que tienen un vehículo menos que el estado  $k$  y que asignándoles un servicio de algún nodo  $i$  llegarían al estado  $k$ . El segundo sumando presenta todos los estados que tienen un vehículo ocupado más que el estado  $k$  y que completando el servicio de emergencia llegarían al estado  $k$ .

Añadiendo la condición

$$\sum_{k=0}^{2^N-1} P(k) = 1, \quad (1. 113)$$

una de las ecuaciones del equilibrio es redundante y puede ser eliminada. Teóricamente, la solución de este conjunto de ecuaciones requiere la inversión de una única matriz, pero el tamaño de la matriz (medido en número total de elementos) es igual a  $2^{2N}$ . Para instancias con  $N=10$  la matriz contiene 1048576 elementos, hecho que limita seriamente la aplicabilidad del modelo. Reconociendo estas limitaciones, *Larson* (1975) presenta su **Approximated Hypercube Queueing Model** (AHQM). La aproximación está basada en la selección aleatoria sin reposición de los servidores hasta encontrar el primero libre que es asignado a la llamada entrante. Dados

$O_j$ =suceso que el  $j$ -ésimo vehículo esté ocupado,

$L_j$ =suceso que el  $j$ -ésimo vehículo esté libre,

$P(O_1O_2 \dots O_jL_{j+1})$ = probabilidad de que el primer servidor libre sea el  $j$ -ésimo+1 servidor seleccionado,

$S_k$ =estado del sistema, sin identificar los vehículos ocupados, con exactamente  $k$  servidores ocupados,

$P_k$ =Probabilidad de equilibrio del estado  $S_k$ ,

y usando el teorema de la probabilidad total, se puede escribir

$$P(O_1O_2 \dots O_jL_{j+1}) = \sum_{k=0}^N P(O_1O_2 \dots O_jL_{j+1} | S_k) P_k. \quad (1.114)$$

Además, por el teorema de las probabilidades compuestas

$$P(O_1O_2 \dots O_jL_{j+1} | S_k) = P(L_{j+1} | O_1O_2 \dots O_jS_k) P(O_j | P(O_1O_2 \dots O_{j-1}S_k)) \dots P(O_1 | S_k). \quad (1.115)$$

Considerando que  $P(O_1 | S_k)$  es la probabilidad que el primer servidor seleccionado al azar esté ocupado, condicionado a que haya exactamente  $k$  servidores ocupados, es evidente que su valor es  $k/N$ . Extendiendo el razonamiento a las otras unidades ocupadas tenemos que

$$P(O_j | O_1O_2 \dots O_{j-1}S_k) = \frac{k - (j-1)}{N - (j-1)} \quad j = 1, 2, \dots, k+1. \quad (1.116)$$

Similarmente

$$P(L_{j+1} | O_1O_2 \dots O_jS_k) = \frac{N - k}{N - j} \quad j = 0, 1, \dots, k. \quad (1.117)$$

Combinando (1.116) y (1.117) en (1.115) y sustituyendo en (1.114), encontramos que la probabilidad de que el primer servidor libre sea el  $j$ -ésimo+1 servidor seleccionado es igual a

$$P(O_1O_2 \dots O_jL_{j+1}) = \sum_{k=j}^{N-1} \left( \frac{k}{N} \frac{k-1}{N-1} \dots \frac{k-(j-1)}{N-(j-1)} \frac{(N-k)}{(N-j)} \right) P_k. \quad (1.118)$$

Por otra parte, dada la cola M/M/N/N/FIFO, se sabe que

$$P_k = P(S_k) = \frac{N^k \rho^k}{\sum_{r=0}^N \frac{N^r \rho^r}{r!}} \quad k = 0, 1, \dots, N, \quad (1.119)$$

siendo  $\rho = \frac{\lambda}{N\mu}$  el factor de utilización para sistemas de colas con capacidad infinita.

Utilizando la fracción media de tiempo que cada servidor está ocupado

$$r = \frac{1}{N} \sum_{j=0}^N j P_j = \rho(1 - P_N), \quad (1.120)$$

podemos escribir (1.118) como

$$P(O_1 O_2 \dots O_j L_{j+1}) = Q(N, \rho, j) r^j (1 - r), \quad (1.121)$$

donde:

$$Q(N, \rho, j) = \sum_{k=j}^{N-1} \frac{(N-k) (N^k) (\rho^{k-j}) P_0 (N-j-1)!}{(k-j)! (1 - P_N)^j N! (1 - \rho(1 - P_N))} \quad j = 0, 1, \dots, N-1. \quad (1.122)$$

Teniendo en cuenta la expresión (1.121), la probabilidad de que el primer servidor libre sea el  $j$ -ésimo+1 servidor seleccionado es calculada como si todos los vehículos actuaran independientemente con una fracción de ocupación media  $r$  corregida por los factores  $Q$ . Con esta aproximación se puede desarrollar un procedimiento iterativo para calcular las cargas de trabajo de cada servidor. Sea  $r_j$  la fracción de tiempo que el vehículo  $j$  está ocupado sirviendo llamadas, para  $j = 1, 2, \dots, N$ . Una estimación de la carga de trabajo para cada vehículo puede ser calculada mediante la siguiente ecuación

$$r_j = \sum_{i \in I} f_i P_{ji} \frac{1}{\mu} \quad j = 1, 2, \dots, N, \quad (1.123)$$

donde:

$P_{ji}$  es la probabilidad que el vehículo  $j$  sea asignado a una llamada de la zona  $i$ .

El producto  $f_i P_{ji}$  es la fracción de llamadas por unidad de tiempo del nodo  $i$  que es asignado al vehículo  $j$  y  $1/\mu$  es el tiempo de servicio esperado para dicha asignación. Sumando para todos los generadores de llamadas se obtendrá una estimación de la carga de trabajo. La probabilidad  $P_{ji}$  que el vehículo  $j$  sea asignado a una llamada de la zona  $i$  para la cual es el  $k$ -ésimo preferido, puede ser aproximada por:

$$P_{ji} = Q(N, \rho, k-1) (1 - r_j) \prod_{\ell=1}^{k-1} r_{a_{i\ell}}, \quad (1.124)$$

donde el subíndice asociado con el  $\ell$ -ésimo vehículo preferido para el nodo  $i$  es denotado por  $a_{i\ell}$ , donde  $j=a_{ik}$ .

Sustituyendo (1. 124) en (1. 123) obtenemos el siguiente sistema de  $N$  ecuaciones no lineales:

$$r_j = (1-r_j) \sum_{i \in I} \sum_{k: a_{ik}=j} f_i \frac{1}{\mu} Q(N, \rho, k-1) \prod_{\ell=1}^{k-1} r_{a_{i\ell}} \quad j = 1, 2, \dots, N. \quad (1. 125)$$

El procedimiento usado para su resolución es un método iterativo de punto fijo donde

a) la solución inicial es  $r_j^c = r = \rho(1-P_N)$  para  $j=1, 2, \dots, N$ ;

b) las cargas de trabajo de cada vehículo pueden ser aproximadas por

$$r_j^{c+1} = \frac{V_j^c}{1 + V_j^c} \quad j = 1, 2, \dots, N, \quad (1. 126)$$

donde

$$V_j^c = \sum_{i \in I} \sum_{k: a_{ik}=j} f_i \frac{1}{\mu} Q(N, \rho, k-1) \prod_{\ell=1}^{k-1} r_{a_{i\ell}}^c \quad j = 1, 2, \dots, N. \quad (1. 127)$$

c) Proceso de normalización de carga de trabajo

$$r_j^{c+1} \leftarrow \frac{N\rho(1-P_N)}{\sum_{k=1}^N r_k^{c+1}} r_j^{c+1} \quad j = 1, 2, \dots, N. \quad (1. 128)$$

d) (Criterio de convergencia). Si no se cumple

$$\sum_{j=1}^N |r_j^{c+1} - r_j^c| \leq \epsilon, \quad (1. 129)$$

ir al paso b. En otro caso, terminar.

*Jarvis* (1985) extiende el AHQM al caso de distribuciones del tiempo de servicio específicas, permitiendo dependencias respecto al servidor y al nodo demanda. Suponiendo  $\tau_{ji}$  el tiempo esperado de servicio para el vehículo  $j$  y el demandante  $i$  se plantea la ecuación homóloga de (1. 123)

$$r_j = \sum_{i \in I} f_i P_{ji} \tau_{ji} \quad j = 1, 2, \dots, p \quad (1. 130)$$

El procedimiento iterativo utilizado para calcular las cargas de trabajo es análogo al descrito por *Larson* (1975) excepto que  $\rho$ ,  $P_N$  y  $P_0$  cambian de iteración a iteración dado que  $\tau$  es variable. Ahora, las etapas son las siguientes:

a) La solución inicial que se toma asume que cada nodo demanda es asignado al primer servidor preferido y no hay interacción entre vehículos

$$r_j^c = \sum_{i: a_{i1}=j} f_i \tau_{ji} \quad j=1,2,\dots,N, \quad (1.131)$$

$$\tau = \sum_{i \in I} \left( \frac{f_i}{\lambda} \right) \tau_{a_{i1}, i}. \quad (1.132)$$

b) las cargas de trabajo de cada vehículo pueden ser aproximadas tomando  $\rho = \lambda \tau / N$  por

$$r_j^{c+1} = \frac{V_j^c}{1 + V_j^c} \quad j=1,2,\dots,N, \quad (1.133)$$

donde

$$V_j^c = \sum_{i \in I} \sum_{k: a_{ik}=j} f_i \tau_{ji} Q(N, \rho, k-1) \prod_{\ell=1}^{k-1} r_{a_{i\ell}}^c \quad j=1,2,\dots,N. \quad (1.134)$$

c) (Criterio de convergencia). Si no se cumple:

$$\sum_{j=1}^N |r_j^{c+1} - r_j^c| \leq \epsilon, \quad (1.135)$$

ir al paso d.

d) Aproximación al tiempo medio de servicio general calculando:

$$\tau = \sum_{i \in I} \left( \frac{f_i}{\lambda} \right) \sum_{j=1}^N \tau_{ji} \frac{P_{ji}}{1 - P_N}, \quad (1.136)$$

$$P_N = 1 - \frac{\sum_{j=1}^N r_j^{c+1}}{N\rho}. \quad (1.137)$$

e ir al paso b. En otro caso, terminar.

*Burwell, Jarvis y McKnew* (1993) extienden el AHQM con distribuciones del tiempo de servicio específicas al caso con múltiples servidores ubicados en una misma estación y todos igualmente preferibles para ser asignados a una llamada procedente del nodo  $i$ . Sea  $x_j$  el número de vehículos estacionados en una misma base  $j$ . Suponiendo que son los  $k$ -ésimos preferidos para atender las llamadas de la zona  $i$ , y que todos los vehículos de la misma estación tienen la misma carga de trabajo podemos estimar la probabilidad  $P_{ji}$  de que algún vehículo de la base  $j$  sea asignado a una llamada de la zona  $i$  por:

$$P_{ji} = (1 - r_j) \left( \sum_{s=0}^{x_j-1} Q(N, \rho, \sum_{\ell=1}^{k-1} x_{a_{i\ell}} + s) (r_j)^s \right) \prod_{\ell=1}^{k-1} (r_{a_{i\ell}})^{x_{a_{i\ell}}}, \quad (1.138)$$

donde

$r_j$  es una estimación de la carga de trabajo para cada vehículo de la base  $j$ ;  
 el subíndice asociado con la  $\ell$ -ésima base preferida para el nodo  $i$  es denotado por  $a_{i\ell}$ ,  
 siendo en particular,  $j=a_{ik}$ .

El procedimiento iterativo utilizado para calcular las cargas de trabajo es análogo al descrito por *Jarvis* (1985), sustituyendo  $P_{ji}$  por la expresada en la igualdad (1. 138).

Una vez obtenidos los niveles de carga de trabajo de cada vehículo, se pueden calcular diferentes parámetros para evaluar el comportamiento de la ubicación actual de los vehículos. Considerando  $N_i$  el conjunto de estaciones ubicadas que pueden servir al nodo  $i$  dentro del tiempo máximo  $S$  podemos calcular diferentes parámetros como desequilibrio entre cargas de trabajo para las distintas estaciones o tiempo medio de servicio, aunque estamos interesados en parámetros que midan el cubrimiento probabilístico como la

- Probabilidad de que una llamada cualquiera del nodo  $i$  sea atendida dentro del tiempo  $S$ ,

$$\sum_{j \in N_i} P_{ji} \tag{1. 139}$$

- Probabilidad de que cualquier llamada del sistema sea atendida dentro del tiempo  $S$ ,

$$\frac{\sum_{i \in I} f_i \left( \sum_{j \in N_i} P_{ji} \right)}{\sum_{i \in I} f_i} \tag{1. 140}$$

- Probabilidad de ser servido dentro de  $S$  correspondiente al peor nodo demanda atendido,

$$\text{Min}_{i \in I} \sum_{j \in N_i} P_{ji} \tag{1. 141}$$

## 1.6. Modelización de servicios de emergencia mediante simulación.

Otra vía alternativa a la modelización de un sistema de emergencia mediante colas, aunque más costosa computacionalmente, es el uso de la simulación. Diferentes autores han empleado la simulación para ayudar a diseñar servicios de emergencia mejores como:

- *Savas* (1969) usa la simulación en la ciudad de Nueva York para mostrar que una mejora substancial en el tiempo medio de respuesta, periodo entre la recepción de la llamada en la estación y la llegada de la ambulancia, puede ser alcanzada dispersando alguna de las ambulancias ubicadas en el Kings County Hospital a un garaje satélite más cercano a las zonas con mayor demanda.
- *Swoveland et al.* (1973) utilizan simulación combinada con un proceso de ramificación y acotación para determinar las posiciones óptimas de las ambulancias de la ciudad de Vancouver (Canada).
- *Fitzsimmons* (1973) desarrolló un modelo para predecir el tiempo de respuesta para una configuración determinada, así como un proceso de búsqueda de la mejor configuración de vehículos y presentó una aplicación para el servicio de ambulancias de la ciudad de Los Angeles.
- *Berlin y Liebman* (1974), en una primera fase, utilizaron el LSCP para determinar la posición de las estaciones en el distrito de Columbia, seguido de una simulación para asignar vehículos a las estaciones elegidas.
- *Chaiken* (1978), presenta diferentes aplicaciones del FIRESIM( Simulation Model of Fire Department Operations) y del PATROLSIM (Simulation Model of Police Patrol Operations), aunque el propio autor reconoce su nula utilización debido a su elevado coste.
- *Uyeno y Seeberg* (1984) han diseñado un sistema de ubicación de ambulancias, usando simulación, para la British Columbia Provincial Ambulance Service para

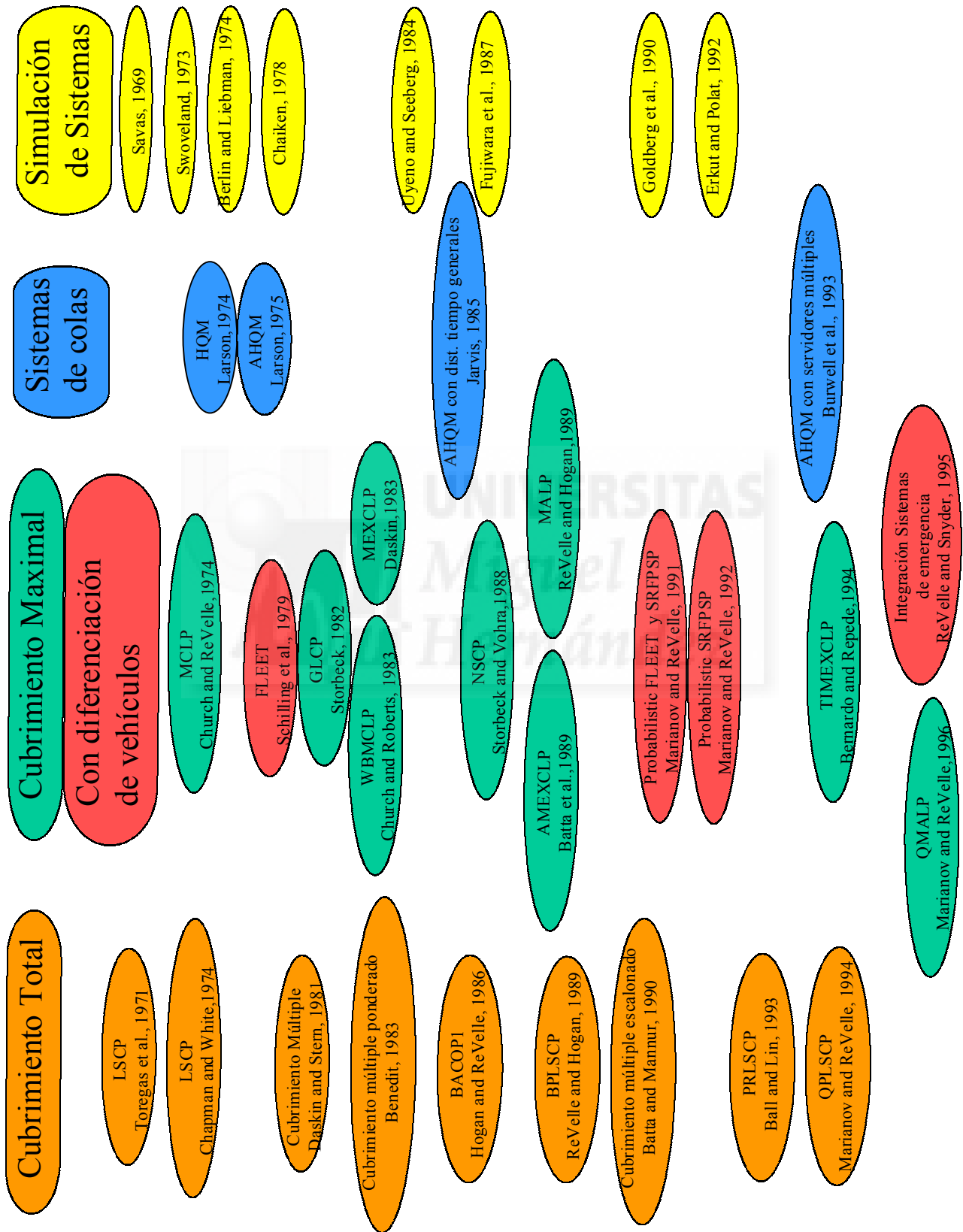


ayudar a los gestores locales a mejorar la respuesta a las llamadas de emergencia con una aplicación en la capital regional del distrito, sita en Victoria (Canada).

- *Fujiwara et al.* (1987) analizaron varias configuraciones obtenidas mediante MEXCLP gracias a un modelo de simulación y obtuvieron una nueva distribución espacial de los vehículos de emergencia en Bangkok con los que se obtenía las fiabilidades del sistema estudiado pero con un menor número de vehículos.
- *Goldberg et al.* (1990) desarrollaron un modelo de simulación capaz de ayudar al departamento de bomberos de Tucson (Arizona) a determinar las estaciones que se abren o cierran, así como la asignación de vehículos a ellas, incorporando una modelización de tiempos de viaje y un proceso de validación.
- *Erkut y Polat* (1992) han analizado el comportamiento mediante simulación del servicio de bomberos de la ciudad de Estambul (Turquía) evaluando el comportamiento con diferentes configuraciones de estaciones de bomberos abiertas y la velocidad de las unidades de emergencia según el día de la semana y la hora del día.
- *Repede y Bernardo* (1994) consiguieron reducir un 36% el tiempo de respuesta de las unidades de emergencia en Louisville (Kentucky) utilizando un sistema de decisión que aunaba una extensión del MEXCLP de *Daskin* y un modelo de simulación. Dicha mejora incorporaba una variación temporal en la estimación de la carga de trabajo global de las unidades de emergencia, que permitía determinar las posiciones iniciales de los vehículos en cada turno y se utilizaba como input en el proceso de simulación posterior.

Todos los modelos de simulación aportan gran cantidad de outputs que permiten evaluar adecuadamente un sistema de emergencia, pero raramente son usados por el alto coste computacional y el coste de obtención de los datos iniciales.

**Grafo temporal de la revisión bibliográfica**



## **Capítulo 2. Nuevos Modelos de Cubrimiento total Probabilístico**

### **2.1. Introducción**

Tras la revisión bibliográfica realizada, se observan claramente tres caminos para analizar la ubicación de unidades de emergencia: modelos lineales enteros, que implican una simplificación de los sistemas de emergencia, modelos basados en la teoría de colas, que permiten analizar más detalladamente el comportamiento de los sistemas de emergencia y, simulación del sistema, como último recurso para modelizar su extrema complejidad. En este capítulo aunaremos los tres caminos, hasta ahora bastante inconexos, en el estudio de los modelos de cubrimiento total probabilístico existentes. Esto es, modelos que minimizan el número de unidades de emergencia (ambulancias, coches de bomberos, grúas de avería, etc.), cumpliendo con la restricción de que el nivel de fiabilidad (probabilidad de que una llamada urgente sea atendida dentro de cierto periodo) sea mayor que cierto nivel prefijado para cada zona de demanda. Toda zona de demanda queda pues atendida o cubierta. La verificación de ese nivel de fiabilidad prefijado para una configuración determinada se realizará mediante un simulador de sistemas de emergencias, que reproduzca la aleatoriedad en la llegada de llamadas, la asignación de la unidad que atiende cada llamada según el estado del sistema y la duración del servicio. Dado el excesivo coste computacional de la simulación, recurriremos a otra herramienta, la resolución de un sistema de ecuaciones no lineales, que nos permita obtener estimaciones de los parámetros cercanas a las obtenidas con la simulación, si bien con un coste computacional muchísimo menor. Por último, propondremos nuevos modelos cuyo comportamiento sea mejor, tanto en lo que se refiere al cumplimiento de las restricciones de fiabilidad individuales, como a que el número de unidades de emergencia ubicadas por el

La formulación no lineal del Poisson Probabilistic Location Set Covering Problem (PPLSCP) es

$$\text{Min } \sum_{j \in J} x_j \quad (2.25)$$

$$\text{s.a. } \sum_{k=0}^{x_j-1} e^{-\lambda_i/\mu_i} \frac{1}{k!} \left( \frac{\lambda_i}{\mu_i} \right)^k \geq \alpha \quad \forall i \in I \quad (2.26)$$

$$x_j \geq 0, \quad x_j \in \mathbb{N} \quad \forall j \in J. \quad (2.27)$$

La función objetivo (2. 25) representa el número de vehículos necesarios para satisfacer las restricciones de fiabilidad impuestas. El conjunto de restricciones (2. 26) impone que la probabilidad de ser atendido dentro del tiempo S para un nodo demanda sea mayor o igual que el nivel de fiabilidad prefijado  $\alpha$ . El conjunto (2. 27) define las variables como enteras, permitiendo tratar separadamente los vehículos y las bases que los albergan.

La restricción (2. 26) no tiene una expresión lineal analítica equivalente, pero obviamente es una función estrictamente creciente respecto del número de vehículos ubicados en el entorno de  $i$  de radio S, por tanto podemos encontrar el mínimo número de vehículos necesario para verificar la fiabilidad impuesta con la expresión

$$b_i = \min \left\{ n \in \mathbb{N} / \sum_{k=0}^{n-1} e^{-\lambda_i/\mu_i} \frac{1}{k!} \left( \frac{\lambda_i}{\mu_i} \right)^k \geq \alpha \right\}, \quad (2.28)$$

que permite sustituir la restricción (2. 26) por

$$\sum_{j \in N_i} x_j \geq b_i \quad \forall i \in I, \quad (2.29)$$

obteniéndose la misma formulación lineal entera que el BPLSCP y el QPLSCP, excepto en el cálculo del número de servidores  $b_i$  necesarios para cada nodo demanda que utiliza la expresión (2. 28).

En las tablas siguientes se presentan los resultados de la experiencia computacional. En cada tabla se ha calculado el número de instancias que no verifican el nivel de fiabilidad prefijado y el porcentaje que suponen respecto a las presentadas en cada tabla.

**Tabla 2.10. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del PPLSCP para las instancias realizadas sobre la red de 55 nodos y distancia máxima S=1.5 y 3.**

	S	ALFA	PPLSCP			S	ALFA	PPLSCP		
			Nº de veh.	MLR -I	MLR-D			Nº de veh.	MLR -I	MLR -D
red 55.1	1.5	0.800	6	0.895	0.891	3	0.800	2	0.904	0.877
		0.825	7	0.897	0.894			2	0.904	0.877
		0.850	7	0.920	0.917			3	0.914	0.896
		0.875	7	0.846	0.847			3	0.943	0.927
		0.900	7	0.797	0.799			3	0.933	0.917
		0.925	8	0.872	0.873			3	0.937	0.921
		0.950	9	0.961	0.961			3	0.937	0.921
		0.975	10	0.993	0.993			4	0.993	0.985
		0.990	11	0.995	0.995			4	0.996	0.990
red 55.2	1.5	0.800	8	0.698	0.699	3	0.800	4	0.847	0.828
		0.825	9	0.897	0.897			4	0.933	0.903
		0.850	9	0.948	0.946			4	0.929	0.899
		0.875	9	0.949	0.946			4	0.930	0.900
		0.900	10	0.911	0.910			4	0.947	0.916
		0.925	10	0.890	0.892			5	0.963	0.942
		0.950	11	0.896	0.898			5	0.989	0.970
		0.975	13	0.979	0.977			5	0.992	0.975
		0.990	14	0.995	0.995			6	0.999	0.993
red 55.3	1.5	0.800	12	0.708	0.705	3	0.800	7	0.839	0.812
		0.825	12	0.721	0.718			7	0.848	0.819
		0.850	13	0.717	0.718			7	0.863	0.829
		0.875	15	0.877	0.867			8	0.946	0.907
		0.900	15	0.900	0.890			8	0.936	0.899
		0.925	15	0.950	0.946			8	0.946	0.908
		0.950	17	0.966	0.956			9	0.987	0.957
		0.975	19	0.977	0.976			10	0.994	0.976
		0.990	22	0.995	0.994			11	0.999	0.994
red 55.4	1.5	0.800	17	0.820	0.804	3	0.800	12	0.888	0.840
		0.825	18	0.865	0.840			12	0.950	0.898
		0.850	20	0.891	0.876			13	0.976	0.927
		0.875	20	0.871	0.870			13	0.990	0.948
		0.900	21	0.901	0.883			13	0.937	0.882
		0.925	22	0.963	0.951			14	0.993	0.958
		0.950	23	0.966	0.962			15	0.985	0.949
		0.975	26	0.984	0.983			16	1.000	0.990
		0.990	29	0.990	0.988			17	1.000	0.995
RESUMEN										
Nº de instancias que no verifican fiabilidad prefijada			10	14	Nº de instancias que no verifican fiabilidad prefijada			1	10	
% de instancias que no verifican fiabilidad prefijada			27.77	38.88	% de instancias que no verifican fiabilidad prefijada			2.77	27.77	



**Tabla 2.11. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del PPLSCP para las instancias realizadas sobre la red de 79 nodos y distancia máxima S=1.5 y 3.**

	S	ALFA	PPLSCP			S	ALFA	PPLSCP		
			Nº de veh.	MLR- I	MLR- D			Nº de veh.	MLR- I	MLR- D
red 79.1	1.5	0.800	11	0.817	0.818	3	0.800	6	0.874	0.861
		0.825	11	0.849	0.849			6	0.899	0.881
		0.850	13	0.864	0.863			7	0.918	0.905
		0.875	13	0.906	0.906			7	0.917	0.906
		0.900	13	0.978	0.976			7	0.924	0.914
		0.925	13	0.978	0.976			7	0.949	0.941
		0.950	13	0.978	0.976			8	0.983	0.980
		0.975	17	0.989	0.988			10	0.998	0.997
		0.990	19	0.993	0.992			10	0.996	0.995
red 79.2	1.5	0.800	13	0.934	0.928	3	0.800	7	0.833	0.815
		0.825	13	0.938	0.932			8	0.931	0.906
		0.850	13	0.938	0.932			8	0.935	0.910
		0.875	14	0.958	0.955			8	0.938	0.918
		0.900	14	0.960	0.956			9	0.965	0.949
		0.925	15	0.960	0.957			9	0.940	0.931
		0.950	16	0.963	0.960			9	0.927	0.917
		0.975	20	0.995	0.994			12	0.996	0.993
		0.990	21	0.997	0.996			12	0.994	0.992
red 79.3	1.5	0.800	19	0.806	0.793	3	0.800	13	0.882	0.852
		0.825	19	0.902	0.885			13	0.913	0.876
		0.850	20	0.840	0.831			13	0.900	0.864
		0.875	21	0.879	0.873			14	0.889	0.869
		0.900	22	0.951	0.944			15	0.888	0.869
		0.925	23	0.950	0.943			16	0.956	0.928
		0.950	25	0.982	0.980			17	0.992	0.978
		0.975	28	0.993	0.991			19	0.979	0.970
		0.990	32	0.997	0.996			20	0.999	0.997
red 79.4	1.5	0.800	26	0.859	0.834	3	0.800	19	0.783	0.754
		0.825	26	0.872	0.852			19	0.793	0.759
		0.850	27	0.873	0.857			19	0.853	0.812
		0.875	28	0.915	0.906			20	0.874	0.835
		0.900	30	0.949	0.943			21	0.866	0.828
		0.925	32	0.966	0.956			22	0.962	0.943
		0.950	33	0.972	0.964			24	0.959	0.908
		0.975	37	0.993	0.990			27	0.994	0.989
		0.990	42	0.997	0.996			29	0.997	0.994
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	1	3	Nº de instancias que no verifican fiabilidad prefijada		6	10		
		% de instancias que no verifican fiabilidad prefijada	2.77	8.33	% de instancias que no verifican fiabilidad prefijada		16.66	27.77		

La tabla comparativa entre los tres modelos de cubrimiento total probabilístico permite afirmar que nuestro modelo obtiene en más ocasiones el nivel de fiabilidad prefijado, aunque todavía presenta unos altos niveles de no cumplimiento de las restricciones impuestas.

**Tabla 2.12. Número de instancias y porcentaje global de instancias que no verifican el nivel de fiabilidad prefijado para diferentes modelos de cubrimiento total probabilístico**

	BPLSCP		QPLSCP		PPLSCP	
	MLR -I	MLR -D	MLR -I	MLR -D	MLR -I	MLR -D
Red de 55 y S=1.5	24	24	19	22	10	14
Red de 55 y S=3	19	28	10	25	1	10
Red de 79 y S=1.5	11	15	9	10	1	3
Red de 79 y S=3	15	23	14	19	6	10
<b>Total</b>	69	90	52	76	18	37
<b>%</b>	47.9	62.5	36.1	52.8	12.5	25.69

Por esta razón, en las subsecciones siguientes nos centraremos en obtener modelos que verifiquen las restricciones de fiabilidad impuestas, calculadas con los procedimientos iterativos de punto fijo anteriormente descritos.

### 2.5.3. El Revised Binomial Probabilistic Location Set Covering Problem.

La revisión del BPLSCP, que denotaremos por RBPLSCP, está basada en una nueva definición de la fracción de ocupación local específica del nodo demanda  $i$ , que necesita para su implementación del conocimiento de la ubicación de una configuración inicial de vehículos,  $x_j^0 \quad \forall j \in J$ . El valor  $\rho_i$  es calculado como:

$$\rho_i = \frac{\bar{t} \sum_{j \in J^0 \cap N_i} \sum_{i \in I} f_i P_{ji}^0}{24 \sum_{j \in N_i} x_j} \quad \forall i \in I. \tag{2.30}$$

En esta expresión de ocupación, se considera en el numerador las estaciones  $j$  de la configuración a priori que pueden servir el nodo  $i$  dentro del tiempo máximo  $S$  y la parte de demanda de cada nodo que es servida por ella,  $P_{ji}^0$ , obtenida resolviendo el conjunto de restricciones (2.5) y en el denominador, el número total de horas disponible para el servicio.



El procedimiento presentado está dividido en 3 pasos:

**Paso 1. Determinar la solución inicial.**

- Obtener  $x_j^0 \quad \forall j \in J$ , la solución óptima del BPLSCP, siendo

$$b_i^0 = \min \left\{ n \in \mathbb{N} / \left( \frac{\bar{t} \sum_{k \in M_i} f_k}{24n} \right)^n \leq 1 - \alpha \right\}. \quad (2.31)$$

- Calcular la fiabilidad local mínima,  $MLR^0$  y las  $P_{ji}^0$ .
- Si  $MLR^0 \geq \alpha$ , ir al paso 3; en otro caso ir al paso 2.

**Paso 2. Actualizar la configuración de vehículos existentes.**

- Calcular  $b_i^* = \min \left\{ n \in \mathbb{N} / \left( \frac{\bar{t} \sum_{j \in J^0 \cap N_i} \sum_{i \in I} f_i P_{ji}^0}{24n} \right)^n \leq 1 - \alpha \right\} \quad \forall i \in I$ .
- Obtener  $x_j^* \quad \forall j \in J$ , la solución óptima del BPLSCP con los nuevos  $b_i^* \quad \forall i \in I$ , definidos.
- Obtener resolviendo el sistema (2.5) la fiabilidad local mínima,  $MLR^*$  y las  $P_{ji}^*$ .
- Si  $MLR^* \geq \alpha$ , ir a paso 3; en otro caso actualizar  $x_j^0 \leftarrow x_j^* \quad \forall j \in J$ ,  $P_{ji}^0 \leftarrow P_{ji}^* \quad \forall j \in J \quad \forall i \in I$ ,  $MLR^0 \leftarrow MLR^*$  e ir al paso 2.

Paso 3. Stop.

El hecho diferencial de nuestro RBPLSCP es que garantiza que la fiabilidad local mínima de la configuración de vehículos obtenida es mayor o igual que el nivel de fiabilidad  $\alpha$  exigido, bajo la hipótesis de independencia. Para la demostración de la afirmación anterior es necesario formular y demostrar dos teoremas previos.

**Teorema 2.1** Si  $u_1, u_2, x_1$  y  $x_2$  son reales positivos, entonces

$$\left( \frac{u_1}{x_1} \right)^{x_1} \left( \frac{u_2}{x_2} \right)^{x_2} \leq \left( \frac{u_1 + u_2}{x_1 + x_2} \right)^{x_1 + x_2}. \quad (2.32)$$

Demostración: la desigualdad que se desea probar es equivalente a

$$\frac{(x_1 + x_2)^{x_1+x_2}}{x_1^{x_1} x_2^{x_2}} \leq \frac{(u_1 + u_2)^{x_1+x_2}}{u_1^{x_1} u_2^{x_2}}. \quad (2.33)$$

Tomando logaritmos se llega a la siguiente expresión equivalente

$$(x_1 + x_2) \log(x_1 + x_2) - (x_1 \log x_1 + x_2 \log x_2) \leq (x_1 + x_2) \log(u_1 + u_2) - (x_1 \log u_1 + x_2 \log u_2). \quad (2.34)$$

Consideremos las siguientes funciones auxiliares:

$$\begin{aligned} h(a, b) &= (x_1 + x_2) \log(a + b) \quad \forall (a, b) \in \mathfrak{R}^+ \times \mathfrak{R}^+, \\ g(a, b) &= x_1 \log a + x_2 \log b \quad \forall (a, b) \in \mathfrak{R}^+ \times \mathfrak{R}^+, \end{aligned} \quad (2.35)$$

y definamos  $f(a, b) = h(a, b) - g(a, b)$ . Claramente,  $f(a, b)$  es una función continua y diferenciable.

La desigualdad que se desea probar es

$$f(x_1, x_2) - f(u_1, u_2) \leq 0. \quad (2.36)$$

Si  $(u_1, u_2) = (x_1, x_2)$  el resultado es inmediato. Supongamos que  $(u_1, u_2) \neq (x_1, x_2)$ . Aplicando el teorema del valor medio a la función  $f$ , obtenemos un número real  $\theta$ ,  $0 < \theta < 1$ , y un punto  $(s, t) = \theta(u_1, u_2) + (1-\theta)(x_1, x_2)$  que verifica

$$f(x_1, x_2) - f(u_1, u_2) = \nabla f(s, t)(x_1 - u_1, x_2 - u_2). \quad (2.37)$$

Dado que

$$\nabla f(s, t)(x_1 - u_1, x_2 - u_2) = \frac{\partial f(s, t)}{\partial a}(x_1 - u_1) + \frac{\partial f(s, t)}{\partial b}(x_2 - u_2), \quad (2.38)$$

y como

$$\begin{aligned} \frac{\partial f(s, t)}{\partial a} &= \frac{x_1 + x_2}{s + t} - \frac{x_1}{s} \\ \frac{\partial f(s, t)}{\partial b} &= \frac{x_1 + x_2}{s + t} - \frac{x_2}{t}, \end{aligned} \quad (2.39)$$

la desigualdad que se quiere probar es equivalente a

$$\left( \frac{x_1 + x_2}{s + t} - \frac{x_1}{s} \right) (x_1 - u_1) + \left( \frac{x_1 + x_2}{s + t} - \frac{x_2}{t} \right) (x_2 - u_2) \leq 0. \quad (2.40)$$

Tomando el denominador común de las fracciones y eliminando éste, se consigue

$$t(x_2 s - x_1 t)(x_1 - u_1) + s(x_1 t - x_2 s)(x_2 - u_2) \leq 0, \quad (2.41)$$

dato que  $s > 0$  y  $t > 0$ . Extrayendo  $(x_2s - x_1t)$  como factor común, se obtiene

$$(x_2s - x_1t)(tx_1 - sx_2 - tu_1 + su_2) = -(x_2s - x_1t)^2 + (x_2s - x_1t)(su_2 - tu_1) \leq 0. \quad (2.42)$$

Como el primer sumando de la última desigualdad es siempre negativo, será suficiente probar que

$$(x_2s - x_1t)(su_2 - tu_1) \leq 0. \quad (2.43)$$

Esta desigualdad es una consecuencia directa de la definición del punto  $(s,t)$ , ya que por medio de un proceso de sustitución se obtiene

$$(x_2s - x_1t)(su_2 - tu_1) = -\theta(1-\theta)(u_1x_2 - u_2x_1)^2. \quad (2.44)$$

Obviamente, esta última expresión es menor o igual que cero, verificándose que la igualdad se cumple si

$$\frac{u_1}{x_1} = \frac{u_2}{x_2}. \quad (2.45)$$

**Teorema 2.2** Si  $u_j$ , y  $x_j$  son reales positivos para  $j=1,2,\dots,n$ , entonces

$$\prod_{j=1}^n \left( \frac{u_j}{x_j} \right)^{x_j} \leq \left( \frac{\sum_{j=1}^n u_j}{\sum_{j=1}^n x_j} \right)^{\sum_{j=1}^n x_j} \quad (2.46)$$

Demostración: Usando el método de inducción para  $n$ . Para  $n=1$ , ambos lados de la desigualdad son el mismo y trivialmente se verifica. Supongamos que la desigualdad es válida para  $n-1$

$$\prod_{j=1}^{n-1} \left( \frac{u_j}{x_j} \right)^{x_j} \leq \left( \frac{\sum_{j=1}^{n-1} u_j}{\sum_{j=1}^{n-1} x_j} \right)^{\sum_{j=1}^{n-1} x_j}, \quad (2.47)$$

y demostraremos que se verifica para  $n$ . Multiplicando en ambos lados de la desigualdad anterior por  $(u_n / x_n)^{x_n}$ , se obtiene para  $n-1$

$$\prod_{j=1}^n \left( \frac{u_j}{x_j} \right)^{x_j} \leq \left( \frac{u_n}{x_n} \right)^{x_n} \left( \frac{\sum_{j=1}^{n-1} u_j}{\sum_{j=1}^{n-1} x_j} \right)^{\sum_{j=1}^{n-1} x_j}. \quad (2.48)$$

Finalmente, aplicando el teorema 2.1 al lado derecho de la expresión se obtiene que la desigualdad es válida para  $n$ .

Veamos a continuación la demostración del teorema siguiente:

**Teorema 2.3** Nuestro modelo RBPLSCP garantiza que la fiabilidad local mínima de la configuración de vehículos obtenida es mayor o igual que el nivel de fiabilidad  $\alpha$  exigido

Demostración: Supongamos, por reducción absurdo, que el procedimiento no garantiza la fiabilidad local mínima exigida. Esto significa que el RBPLSCP construye una secuencia de configuraciones de vehículos con fiabilidades locales mínimas monótona creciente y cuyo límite es una configuración,  $x_j^0 \quad \forall j \in J$ , con un valor de  $MLR^0 < \alpha$ . Por hipótesis existe algún  $i$  verificando que la probabilidad de no ser atendido es mayor que  $1-\alpha$ .

$$\prod_{j \in N_i \cap J^0} (r_j^0)^{x_j^0} = \prod_{j \in N_i \cap J^0} \left( \frac{\bar{t} \sum_{i \in I} f_i P_{ji}^0}{24 x_j^0} \right)^{x_j^0} > 1 - \alpha. \quad (2.49)$$

Por teorema 2.2 se obtiene que

$$\prod_{j \in N_i \cap J^0} \left( \frac{\bar{t} \sum_{i \in I} f_i P_{ji}^0}{24 x_j^0} \right)^{x_j^0} \leq \left( \frac{\bar{t} \sum_{j \in N_i \cap J^0} \sum_{i \in I} f_i P_{ji}^0}{24 \sum_{j \in N_i \cap J^0} x_j^0} \right)^{\sum_{j \in N_i \cap J^0} x_j^0}. \quad (2.50)$$

Como  $x_j^0 \quad \forall j \in J$  es el límite de la sucesión, también cumplirá las restricciones del modelo resuelto en el paso 2, y por tanto por la nueva definición de los  $b_i$  se cumple

$$\left( \frac{\bar{t} \sum_{j \in N_i \cap J^0} \sum_{i \in I} f_i P_{ji}^0}{24 \sum_{j \in N_i \cap J^0} x_j^0} \right)^{\sum_{j \in N_i \cap J^0} x_j^0} \leq 1 - \alpha \quad \forall i \in I, \quad (2.51)$$

que contradice la hipótesis establecida.

Otra posibilidad es que construya una secuencia de configuraciones de vehículos con fiabilidades locales mínimas estrictamente creciente, que obviamente es un absurdo, por que la única manera de que la  $MLR$  crezca es aumentando en cada iteración el valor de  $b_i$  de algún

nodo demanda y por tanto el límite de esta sucesión, que está acotada superiormente por 1, sería 1 y verificaría el nivel de fiabilidad exigido. n

Notar que la incorporación en el paso 2 de nuestro procedimiento del RBPLSCP del cálculo de  $MLR^*$  y de la posterior verificación de la desigualdad,  $MLR^* \geq \alpha$ , ha sido incorporado para agilizar la obtención de la solución. Dado que, basándonos en la demostración del teorema, la regla de parada que se hubiese podido utilizar podría haberse formulado como “detener el proceso cuando las configuraciones de vehículos entre dos iteraciones consecutivas sean la misma”. Obviamente, esta regla de parada obligaría a calcular al menos una instancia más del BPLSCP que el procedimiento descrito.

Nuestro procedimiento iterativo es relativamente rápido, según se desprende de la siguiente tabla, donde se presenta la distribución de frecuencias del número de problemas lineales enteros, como el del BPLSCP, que son necesarios resolver para asegurar que la fiabilidad local mínima,  $MLR$ , sea mayor que el nivel de fiabilidad.

**Tabla 2.13. Distribución de frecuencia del número de iteraciones realizadas ( problemas lineales enteros resueltos) para la muestra de 144 instancias.**

Nº de iteraciones	Frecuencia absoluta	Porcentaje
1	75	52.08
2	57	39.58
3	10	6.94
4	2	1.40
total	144	100.00

Como se puede observar se obtiene la fiabilidad requerida en la primera iteración en el 52.08% de las veces, el 39.58% requiere resolver un problema adicional para verificar la fiabilidad fijada, el 6.94% necesitan resolver dos problemas adicionales y solamente en 2 instancias se han necesitado 4 iteraciones para satisfacer la restricción impuesta. Estos datos dan una evidencia estadística de que el esfuerzo computacional adicional es mínimo.

En las tablas siguientes se presentan los resultados de la experiencia computacional. En cada tabla se ha calculado el número de instancias que no verifican el nivel de fiabilidad prefijado y el porcentaje que suponen respecto a las presentadas en cada tabla.

**Tabla 2.14 Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del RBPLSCP para las instancias realizadas sobre la red de 55 nodos y distancia máxima S =1.5 y 3.**

	S	ALFA	RBPLSCP			S	ALFA	RBPLSCP		
			Nº de veh.	MLR -I	MLR-D			Nº de veh.	MLR -I	MLR -D
red 55.1	1.5	0.800	7	0.897	0.894	3	0.800	2	0.904	0.877
		0.825	7	0.920	0.917			2	0.904	0.877
		0.850	7	0.920	0.917			2	0.904	0.877
		0.875	7	0.917	0.916			2	0.904	0.877
		0.900	8	0.900	0.900			3	0.939	0.924
		0.925	8	0.957	0.957			3	0.933	0.917
		0.950	9	0.956	0.955			3	0.988	0.973
		0.975	10	0.995	0.995			3	0.988	0.973
		0.990	11	0.992	0.992			4	0.992	0.984
		red 55.2	1.5	0.800	8			0.887	0.885	3
0.825	9			0.884	0.883	3	0.891	0.849		
0.850	9			0.891	0.892	3	0.891	0.849		
0.875	9			0.949	0.946	4	0.976	0.943		
0.900	10			0.948	0.945	4	0.929	0.899		
0.925	10			0.954	0.947	4	0.934	0.904		
0.950	11			0.957	0.950	4	0.976	0.943		
0.975	13			0.978	0.976	5	0.986	0.966		
0.990	14			0.997	0.996	5	0.997	0.982		
red 55.3	1.5			0.800	12	0.839	0.820	3	0.800	
		0.825	13	0.861	0.845	6	0.917			0.859
		0.850	13	0.876	0.862	6	0.871			0.828
		0.875	14	0.934	0.916	7	0.884			0.845
		0.900	14	0.929	0.917	7	0.907			0.866
		0.925	14	0.928	0.907	7	0.953			0.903
		0.950	15	0.951	0.939	7	0.954			0.904
		0.975	18	0.977	0.970	8	0.993			0.959
		0.990	19	0.992	0.991	8	0.993			0.959
		red 55.4	1.5	0.800	18	0.842	0.804			3
0.825	18			0.872	0.832	10	0.876	0.832		
0.850	19			0.859	0.821	10	0.876	0.832		
0.875	20			0.936	0.917	11	0.937	0.882		
0.900	20			0.943	0.914	11	0.936	0.880		
0.925	21			0.932	0.903	11	0.942	0.884		
0.950	21			0.951	0.934	12	0.954	0.904		
0.975	24			0.984	0.974	12	0.986	0.932		
0.990	26			0.990	0.986	13	0.994	0.953		
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	0	8	Nº de instancias que no verifican fiabilidad prefijada		0	22		
		% de instancias que no verifican fiabilidad prefijada	0	22.22	% de instancias que no verifican fiabilidad prefijada		0	61.11		

**Tabla 2.15. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del RBPLSCP para las instancias realizadas sobre las red de 79 nodos y distancia máxima  $S=1.5$  y 3.**

	S	ALFA	RBPLSCP			S	ALFA	RBPLSCP		
			Nº de veh.	MLR -I	MLR- D			Nº de veh.	MLR- I	MLR- D
red 79.1	1.5	0.800	11	0.849	0.849	3	0.800	6	0.892	0.875
		0.825	12	0.880	0.880			6	0.899	0.882
		0.850	13	0.875	0.874			6	0.892	0.876
		0.875	13	0.978	0.976			7	0.933	0.920
		0.900	13	0.978	0.976			7	0.920	0.908
		0.925	13	0.978	0.976			7	0.960	0.954
		0.950	13	0.978	0.976			7	0.952	0.945
		0.975	14	0.981	0.980			8	0.981	0.978
		0.990	17	0.991	0.990	0.990	10	0.992	0.988	
red 79.2	1.5	0.800	13	0.935	0.930	3	0.800	7	0.834	0.814
		0.825	13	0.938	0.932			7	0.861	0.844
		0.850	13	0.938	0.932			8	0.948	0.925
		0.875	13	0.938	0.932			8	0.947	0.925
		0.900	13	0.926	0.920			8	0.932	0.906
		0.925	14	0.933	0.928			8	0.941	0.917
		0.950	15	0.960	0.957			9	0.968	0.951
		0.975	18	0.978	0.977			11	0.994	0.988
		0.990	19	0.995	0.994	0.990	11	0.993	0.987	
red 79.3	1.5	0.800	18	0.863	0.852	3	0.800	12	0.846	0.807
		0.825	19	0.838	0.828			12	0.857	0.824
		0.850	19	0.864	0.854			13	0.896	0.839
		0.875	19	0.901	0.883			13	0.909	0.871
		0.900	20	0.924	0.911			14	0.946	0.900
		0.925	21	0.946	0.937			15	0.978	0.944
		0.950	22	0.960	0.953			15	0.959	0.935
		0.975	25	0.978	0.975			16	0.979	0.946
		0.990	27	0.994	0.992	0.990	18	0.995	0.980	
red 79.4	1.5	0.800	23	0.824	0.797	3	0.800	18	0.833	0.796
		0.825	24	0.835	0.809			18	0.843	0.809
		0.850	24	0.856	0.837			18	0.891	0.850
		0.875	26	0.880	0.860			19	0.911	0.867
		0.900	26	0.921	0.896			19	0.926	0.888
		0.925	27	0.926	0.900			20	0.934	0.903
		0.950	29	0.953	0.942			20	0.963	0.929
		0.975	32	0.977	0.972			22	0.979	0.943
		0.990	35	0.993	0.990	0.990	24	0.993	0.986	
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	0	8	Nº de instancias que no verifican fiabilidad prefijada			0	18	
		% de instancias que no verifican fiabilidad prefijada	0	22.22	% de instancias que no verifican fiabilidad prefijada			0	50	

Destacar que el RBPLSCP garantiza la fiabilidad local mínima bajo la hipótesis, propia del modelo, de independencia. Para asegurarla bajo la hipótesis de dependencia necesitamos, en consecuencia, presentar un nuevo modelo de cubrimiento total probabilístico.

### 2.5.4. El Revised Poisson Probabilistic Location Set Covering Problem.

La revisión del PPLSCP, que denotaremos por RPPLSCP, está basada en la misma definición de la fracción de ocupación local específica del nodo demanda  $i$  que la utilizada en el RBPLSCP, con la única diferencia que para obtener la parte de demanda de cada nodo que es servida por ella,  $P_{ji}^0$ , es necesario resolver el conjunto de restricciones (2.13) y (2.14) en vez del conjunto (2.5), necesitando, de nuevo, el conocimiento de la ubicación de una configuración inicial de vehículos  $x_j^0 \quad \forall j \in J$ .

El procedimiento presentado está dividido en 3 pasos:

Paso 1. Determinar la solución inicial.

- Obtener  $x_j^0 \quad \forall j \in J$ , la solución óptima del PPLSCP, siendo

$$b_i = \min \left\{ n \in \mathbb{N} / \sum_{k=0}^{n-1} e^{-\lambda_i / \mu_i} \frac{1}{k!} \left( \frac{\lambda_i}{\mu_i} \right)^k > \alpha \right\}. \quad (2.52)$$

- Calcular resolviendo el sistema (2.13) y (2.14) la fiabilidad local mínima,  $MLR^0$  y las  $P_{ji}^0$ .
- Si  $MLR^0 \geq \alpha$ , ir a paso 3; en otro caso ir al paso 2.

Paso 2. Actualizar la configuración de vehículos existentes:

- Calcular  $b_i^* = \min \left\{ n \in \mathbb{N} / \sum_{k=0}^{n-1} e^{-\lambda_i^0 / \mu_i^0} \frac{1}{k!} \left( \frac{\lambda_i^0}{\mu_i^0} \right)^k > \alpha \right\}$ ,

donde  $\frac{\lambda_i^0}{\mu_i^0} = \frac{\bar{t} \sum_{j \in J^0 \cap N_i} \sum_{i \in I} f_i P_{ji}^0}{24} \quad \forall i \in I$ .



- Obtener  $x_j^* \quad \forall j \in J$ , la solución óptima del PPLSCP con los nuevos  $b_i^* \quad \forall i \in I$ , definidos.
- Obtener resolviendo el sistema (2.13) y (2.14) la fiabilidad local mínima,  $MLR^*$  y las  $P_{ji}^*$ .
- Si  $MLR^* \geq \alpha$ , ir a paso 3; en otro caso actualizar  $x_j^0 \leftarrow x_j^* \quad \forall j \in J$ ,  $P_{ji}^0 \leftarrow P_{ji}^* \quad \forall j \in J \quad \forall i \in I$ ,  $MLR^0 \leftarrow MLR^*$  e ir al paso 2.

Paso 3. Stop.

Nuestro procedimiento iterativo es relativamente rápido, según se desprende de la siguiente tabla, donde se presenta la distribución de frecuencias del número de problemas lineales enteros, como el del PPLSCP, que son necesarios resolver para asegurar que la fiabilidad local mínima,  $MLR$ , sea mayor que el nivel de fiabilidad.

**Tabla 2.13. Distribución de frecuencia del número de iteraciones realizadas ( problemas lineales enteros resueltos) para la muestra de 144 instancias**

Nº de iteraciones	Frecuencia absoluta	Porcentaje
1	107	74.305
2	36	25.000
3	0	0.000
4	0	0.000
5	1	0.695
total	144	100.000

Como se puede observar se obtiene la fiabilidad requerida en la primera iteración en el 74,305% de las veces, el 25% requiere resolver un problema adicional para verificar la fiabilidad fijada y solamente en una instancia necesitamos resolver 5 iteraciones para satisfacer la restricción impuesta. Estos datos dan una evidencia estadística que el esfuerzo computacional adicional es pequeño.

En las tablas siguientes se presentan los resultados de la experiencia computacional. En cada tabla se ha calculado el número de instancias que no verifican el nivel de fiabilidad prefijado y el porcentaje que suponen respecto a las presentadas en cada tabla.

**Tabla 2.16 Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del RPPLSCP para las instancias realizadas sobre las red de 55 nodos y distancia máxima S =1.5 y 3.**

	S	ALFA	RPPLSCP			S	ALFA	RPPLSCP		
			Nº de veh.	MLR -I	MLR-D			Nº de veh.	MLR -I	MLR -D
red 55.1	1.5	0.800	6	0.895	0.891	3	0.800	2	0.904	0.877
		0.825	7	0.897	0.894			2	0.904	0.877
		0.850	7	0.920	0.917			3	0.914	0.896
		0.875	7	0.886	0.886			3	0.943	0.927
		0.900	8	0.917	0.917			3	0.933	0.917
		0.925	9	0.974	0.970			3	0.988	0.973
		0.950	9	0.961	0.961			3	0.988	0.973
		0.975	10	0.993	0.993			4	0.993	0.985
		0.990	11	0.995	0.995			4	0.999	0.995
		red 55.2	1.5	0.800	8			0.818	0.819	3
0.825	9			0.897	0.897	4	0.933	0.903		
0.850	9			0.948	0.946	4	0.929	0.899		
0.875	9			0.949	0.946	4	0.930	0.900		
0.900	10			0.911	0.910	4	0.947	0.916		
0.925	11			0.963	0.957	5	0.963	0.942		
0.950	12			0.972	0.969	5	0.989	0.970		
0.975	13			0.979	0.977	5	0.997	0.982		
0.990	14			0.995	0.995	6	0.999	0.993		
red 55.3	1.5			0.800	13	0.836	0.822	3	0.800	
		0.825	14	0.878	0.864	7	0.912			0.869
		0.850	14	0.894	0.887	7	0.955			0.905
		0.875	15	0.951	0.947	8	0.946			0.907
		0.900	16	0.955	0.950	8	0.993			0.959
		0.925	15	0.950	0.946	8	0.988			0.950
		0.950	17	0.966	0.956	9	0.987			0.957
		0.975	19	0.977	0.976	10	0.994			0.976
		0.990	22	0.995	0.994	11	0.999			0.994
		red 55.4	1.5	0.800	17	0.820	0.804			3
0.825	18			0.865	0.840	12	0.950	0.898		
0.850	20			0.891	0.876	13	0.976	0.927		
0.875	22			0.960	0.947	13	0.990	0.948		
0.900	23			0.951	0.942	13	0.978	0.931		
0.925	22			0.963	0.951	14	0.993	0.958		
0.950	23			0.966	0.962	14	0.995	0.965		
0.975	26			0.984	0.983	16	1.000	0.990		
0.990	32			0.998	0.997	17	1.000	0.995		
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	0	0	Nº de instancias que no verifican fiabilidad prefijada		0	0		
		% de instancias que no verifican fiabilidad prefijada	0	0	% de instancias que no verifican fiabilidad prefijada		0	0		

**Tabla 2.17. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del RPPLSCP para las instancias realizadas sobre las red de 79 nodos y distancia máxima  $S = 1.5$  y 3.**

	S	ALFA	RPPLSCP			S	ALFA	RPPLSCP		
			Nº de veh.	MLR- I	MLR- D			Nº de veh.	MLR- I	MLR- D
red 79.1	1.5	0.800	11	0.817	0.818	3	0.800	6	0.874	0.861
		0.825	11	0.849	0.849		0.825	6	0.899	0.881
		0.850	13	0.864	0.863		0.850	7	0.918	0.905
		0.875	13	0.906	0.906		0.875	7	0.917	0.906
		0.900	13	0.978	0.976		0.900	7	0.924	0.914
		0.925	13	0.978	0.976		0.925	7	0.949	0.941
		0.950	13	0.978	0.976		0.950	8	0.983	0.980
		0.975	17	0.989	0.988		0.975	10	0.998	0.997
		0.990	19	0.993	0.992		0.990	10	0.996	0.995
red 79.2	1.5	0.800	13	0.934	0.928	3	0.800	7	0.833	0.815
		0.825	13	0.938	0.932		0.825	8	0.931	0.906
		0.850	13	0.938	0.932		0.850	8	0.935	0.910
		0.875	14	0.958	0.955		0.875	8	0.938	0.918
		0.900	14	0.960	0.956		0.900	9	0.965	0.949
		0.925	15	0.960	0.957		0.925	9	0.940	0.931
		0.950	16	0.963	0.960		0.950	11	0.988	0.983
		0.975	20	0.995	0.994		0.975	12	0.996	0.993
		0.990	21	0.997	0.996		0.990	12	0.994	0.992
red 79.3	1.5	0.800	19	0.843	0.834	3	0.800	13	0.882	0.852
		0.825	19	0.902	0.885		0.825	13	0.913	0.876
		0.850	21	0.952	0.943		0.850	13	0.900	0.864
		0.875	21	0.937	0.927		0.875	15	0.928	0.902
		0.900	22	0.951	0.944		0.900	15	0.962	0.943
		0.925	23	0.950	0.943		0.925	16	0.956	0.928
		0.950	25	0.982	0.980		0.950	17	0.992	0.978
		0.975	28	0.993	0.991		0.975	20	0.997	0.996
		0.990	32	0.997	0.996		0.990	20	0.999	0.997
red 79.4	1.5	0.800	26	0.859	0.834	3	0.800	19	0.911	0.862
		0.825	26	0.872	0.852		0.825	20	0.907	0.866
		0.850	27	0.873	0.857		0.850	21	0.950	0.915
		0.875	28	0.915	0.906		0.875	22	0.967	0.949
		0.900	30	0.949	0.943		0.900	22	0.965	0.951
		0.925	32	0.966	0.956		0.925	22	0.962	0.943
		0.950	33	0.972	0.964		0.950	26	0.984	0.954
		0.975	37	0.993	0.990		0.975	27	0.994	0.989
		0.990	42	0.997	0.996		0.990	29	0.997	0.994
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	0	0			Nº de instancias que no verifican fiabilidad prefijada	0	0	
		% de instancias que no verifican fiabilidad prefijada	0	0			% de instancias que no verifican fiabilidad prefijada	0	0	

La experiencia computacional demuestra que el RPPLSCP es un modelo que garantiza la obtención de una fiabilidad local mínima por encima del nivel de fiabilidad prefijado, tanto suponiendo independencia como dependencia. En la última sección de este capítulo compararemos las eficiencias de las soluciones obtenidas con todos los modelos de cubrimiento probabilísticos presentados basados en cubrimiento múltiple junto con los modelos que se presentarán en la próxima sección.

## 2.6. Nuevos modelos de Cubrimiento total Probabilístico basados en cantidad de cubrimiento.

En esta sección dedicaremos nuestra atención a otros modelos cuya estructura conceptual es distinta de la presentada en la sección anterior. La diferencia estriba en que construyen restricciones lineales a partir de argumentos probabilísticos, sin recurrir a construir un modelo de cubrimiento múltiple, al convertir una restricción no lineal en una restricción lineal equivalente.

### 2.6.1. Modelo existente: El Poisson Reliability Location Set Covering Problem.

El primer modelo presentado no basado en cubrimiento múltiple, y que nos ha permitido una nueva secuencia de modelos basado en su estructura conceptual es el modelo expuesto por *Ball y Lin* (1993), llamado Poisson Reliability Location Set Covering Problem (PRLSCP), presentado en el capítulo 1, cuya formulación lineal es:

$$\text{Min } \sum_{j \in J} \sum_{k=1}^L kx_{jk} \quad (2.53)$$

$$\text{s.a. } \sum_{j \in N_i} \sum_{k=1}^L -\log(P(D(j) \geq k))x_{jk} \geq -\log(1 - \alpha) \quad \forall i \in I \quad (2.54)$$

$$\sum_{k=1}^L x_{jk} \leq 1 \quad \forall j \in J \quad (2.55)$$

$$x_{jk} \in \{0,1\} \quad \forall j \in J, \quad k = 1,2,\dots,L, \quad (2.56)$$

donde

$B_j = \{i \in I / d_{ji} \leq S\}$  es el conjunto de nodos demanda que pueden ser atendidos por la estación  $j$ .

$D(j)$  es el número de llamadas que llega a  $B_j$  durante el intervalo  $T$  y que sigue una distribución de Poisson con media  $\lambda_j = \frac{T}{24} \left( \sum_{i \in B_j} f_i \right)$  llamadas por hora.

$$x_{jk} = \begin{cases} 1, & \text{si } k \text{ vehículos son ubicados en la estación } j \\ 0, & \text{en caso contrario.} \end{cases}$$

$L$  es el número máximo de vehículos que se pueden ubicar en cada estación.

La cadena de desigualdades utilizada para la modelización se basa en establecer una cota superior de la probabilidad de no ser atendido para cada nodo demanda y restringir dicha cota a ser menor que  $1-\alpha$  y posteriormente tomando logaritmos en ambos lados y cambiando los signos, se obtiene el conjunto de restricciones lineales (2. 54).

Obviamente, podemos calcular los coeficientes de las restricciones (2. 54) con la distribución de Poisson como:

$$P(D(j) \geq k) = 1 - \sum_{s=0}^{k-1} e^{-\lambda_j} \frac{\lambda_j^s}{s!}.$$

En las tablas siguientes se presentan los resultados de la experiencia computacional. En cada tabla se ha calculado el número de instancias que no verifican el nivel de fiabilidad prefijado y el porcentaje que suponen respecto a las presentadas en cada tabla.

**Tabla 2.18. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del PRLSCP para las instancias realizadas sobre la red de 55 nodos y distancia máxima S=1.5 y 3.**

	S	ALFA	PRLSCP			S	ALFA	PRLSCP		
			Nº de veh.	MLR- I	MLR- D			Nº de veh.	MLR- I	MLR- D
red 55.1	1.5	0.800	6	0.837	0.834	3	0.800	2	0.904	0.877
		0.825	6	0.878	0.875			2	0.904	0.877
		0.850	7	0.907	0.904			3	0.988	0.973
		0.875	7	0.893	0.890			3	0.988	0.973
		0.900	7	0.937	0.934			3	0.988	0.973
		0.925	8	0.974	0.970			3	0.988	0.973
		0.950	9	0.982	0.981			3	0.988	0.973
		0.975	10	0.993	0.992			4	0.999	0.995
		0.990	11	0.993	0.992			4	0.999	0.995
red 55.2	1.5	0.800	8	0.883	0.882	3	0.800	4	0.976	0.943
		0.825	8	0.906	0.893			4	0.976	0.943
		0.850	9	0.943	0.935			4	0.976	0.943
		0.875	9	0.936	0.935			4	0.976	0.943
		0.900	10	0.904	0.904			4	0.976	0.943
		0.925	11	0.938	0.936			5	0.997	0.982
		0.950	12	0.960	0.959			5	0.997	0.982
		0.975	14	0.997	0.996			5	0.997	0.982
		0.990	15	0.998	0.998			6	1.000	0.995
red 55.3	1.5	0.800	13	0.860	0.852	3	0.800	7	0.972	0.920
		0.825	13	0.860	0.852			7	0.972	0.920
		0.850	14	0.863	0.856			7	0.972	0.920
		0.875	16	0.969	0.962			8	0.993	0.959
		0.900	17	0.976	0.974			8	0.993	0.959
		0.925	17	0.986	0.986			8	0.993	0.959
		0.950	19	0.988	0.987			9	0.999	0.981
		0.975	21	0.995	0.994			10	1.000	0.992
		0.990	24	0.999	0.999			11	1.000	0.997
red 55.4	1.5	0.800	20	0.923	0.915	3	0.800	12	0.986	0.932
		0.825	21	0.925	0.919			12	0.986	0.932
		0.850	23	0.938	0.921			13	0.996	0.957
		0.875	23	0.928	0.924			13	0.996	0.957
		0.900	25	0.979	0.976			13	0.996	0.957
		0.925	25	0.979	0.976			14	0.999	0.974
		0.950	27	0.993	0.992			15	1.000	0.986
		0.975	30	0.993	0.992			16	1.000	0.992
		0.990	33	1.000	0.999			17	1.000	0.996
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	0	0	Nº de instancias que no verifican fiabilidad prefijada			0	0	
		% de instancias que no verifican fiabilidad prefijada	0	0	% de instancias que no verifican fiabilidad prefijada			0	0	

**Tabla 2.19. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del PRLSCP para las instancias realizadas sobre la red de 79 nodos y distancia máxima S=1.5 y 3.**

	S	ALFA	PRLSCP			S	ALFA	PRLSCP		
			Nº de veh.	MLR- I	MLR- D			Nº de veh.	MLR- I	MLR- D
red 79.1	1.5	0.800	10	0.837	0.836	3	0.800	6	0.877	0.860
		0.825	11	0.845	0.844		0.825	6	0.917	0.904
		0.850	12	0.889	0.888		0.850	6	0.903	0.887
		0.875	13	0.883	0.883		0.875	7	0.966	0.958
		0.900	13	0.979	0.977		0.900	7	0.949	0.939
		0.925	13	0.980	0.978		0.925	7	0.965	0.956
		0.950	14	0.978	0.977		0.950	8	0.981	0.978
		0.975	17	0.989	0.989		0.975	10	0.997	0.996
		0.990	19	0.997	0.997		0.990	11	0.998	0.998
red 79.2	1.5	0.800	13	0.936	0.931	3	0.800	7	0.885	0.863
		0.825	14	0.953	0.949		0.825	7	0.887	0.868
		0.850	14	0.932	0.927		0.850	8	0.953	0.935
		0.875	14	0.937	0.933		0.875	9	0.980	0.967
		0.900	14	0.939	0.934		0.900	9	0.980	0.969
		0.925	15	0.969	0.967		0.925	9	0.977	0.967
		0.950	17	0.975	0.974		0.950	10	0.978	0.974
		0.975	20	0.995	0.994		0.975	12	0.998	0.995
		0.990	22	0.998	0.998		0.990	13	0.999	0.998
red 79.3	1.5	0.800	20	0.890	0.882	3	0.800	14	0.962	0.922
		0.825	21	0.901	0.894		0.825	14	0.937	0.895
		0.850	21	0.948	0.944		0.850	15	0.978	0.954
		0.875	22	0.947	0.943		0.875	15	0.978	0.958
		0.900	23	0.964	0.958		0.900	15	0.966	0.958
		0.925	25	0.976	0.973		0.925	17	0.994	0.983
		0.950	28	0.989	0.986		0.950	18	0.997	0.988
		0.975	30	0.996	0.995		0.975	19	0.998	0.995
		0.990	35	0.999	0.999		0.990	22	1.000	0.999
red 79.4	1.5	0.800	28	0.935	0.928	3	0.800	21	0.978	0.943
		0.825	29	0.967	0.954		0.825	21	0.973	0.935
		0.850	30	0.967	0.955		0.850	21	0.974	0.942
		0.875	31	0.958	0.949		0.875	21	0.974	0.942
		0.900	33	0.976	0.971		0.900	23	0.975	0.962
		0.925	34	0.980	0.976		0.925	24	0.996	0.977
		0.950	37	0.993	0.992		0.950	25	0.996	0.987
		0.975	40	0.997	0.996		0.975	28	0.999	0.997
		0.990	47	1.000	0.999		0.990	30	1.000	0.999
RESUMEN										
	Nº de instancias que no verifican fiabilidad prefijada		0	0	Nº de instancias que no verifican fiabilidad prefijada		0	0		
	% de instancias que no verifican fiabilidad prefijada		0	0	% de instancias que no verifican fiabilidad prefijada		0	0		

La experiencia computacional muestra que el PRLSCP es un modelo que garantiza la obtención de una fiabilidad local mínima por encima del nivel de fiabilidad prefijado, tanto suponiendo independencia como dependencia. Este resultado, a priori sorprende, dado que consigue en la batería de problemas utilizada satisfacer todas las restricciones impuestas con la evaluación a posteriori. En la última sección de este capítulo se mostrará que consigue estos resultados gracias a que ubica muchos más vehículos de los necesarios. En otras palabras, la cota superior establecida de la probabilidad de no ser atendido para cada nodo demanda en muchas ocasiones es excesiva para garantizar la fiabilidad impuesta

### 2.6.2. El Binomial Reliability location set covering problem (BRLSCP).

En la presentación del BPLSCP, *ReVelle y Hogan (1988)* presentan una modelización de la probabilidad de que un nodo demanda no pueda ser atendido, conceptualmente atractiva según sus palabras, pero con dificultades computacionales a su entender insalvables.

Esa modelización, basada en una hipótesis de independencia, utiliza la fracción de ocupación de un vehículo ubicado en la estación  $j$ ,  $r_j$ , desconocida a priori, y enuncia que la probabilidad de que un vehículo esté disponible para atender una llamada procedente del nodo  $i$  dentro del tiempo máximo  $S$  sea mayor que el nivel de fiabilidad prefijado  $\alpha$  como

$$1 - \prod_{j \in N_i} r_j^{x_j} \geq \alpha. \quad (2.57)$$

Redefiniendo  $x_j$ , el número de vehículos a ubicar en la estación  $j$ , con las variables

$$x_{jk} = \begin{cases} 1, & \text{si } k \text{ vehículos son ubicados en la estación } j \\ 0, & \text{en caso contrario,} \end{cases}$$

y tomando  $L$  como el número máximo de vehículos que se pueden ubicar en cada estación, podemos reescribir (2.57) como

$$\prod_{j \in N_i} \prod_{k=1}^L (r_{jk})^{x_{jk}} \leq 1 - \alpha \quad \forall i \in I \quad (2.58)$$

$$\sum_{k=1}^L x_{jk} \leq 1 \quad \forall j \in J, \quad (2.59)$$



donde

$r_{jk}$  es la fracción de ocupación de un vehículo ubicado en la estación  $j$  cuando  $k$  vehículos son ubicados en la estación  $j$ .

Las restricciones (2. 59) son necesarias para convertir la variable entera  $x_j$  en  $L$  variables binarias y afirma que en cada estación o no se ubica ningún vehículo o se ubican un número  $k$  de vehículos. Tomando logaritmos en ambos lados y cambiando los signos, (2. 58) puede ser transformada en una restricción lineal y podemos formular el Binomial Reliability Location Set Covering Problem (BRLSCP) como

$$\text{Min } \sum_{j \in J} \sum_{k=1}^L kx_{jk} \tag{2. 60}$$

$$\text{s.a. } \sum_{j \in N_i} \sum_{k=1}^L -\log((r_{jk})^k)x_{jk} \geq -\log(1 - \alpha) \quad \forall i \in I \tag{2. 61}$$

$$\sum_{k=1}^L x_{jk} \leq 1 \quad \forall j \in J \tag{2. 62}$$

$$x_{jk} \in \{0,1\} \quad \forall j \in J, \quad k = 1,2,\dots,L. \tag{2. 63}$$

La formulación presentada es análoga al PRLSCP, excepto en el cálculo de los coeficientes de las restricciones (2. 61). El único problema que presenta es la determinación de las fracciones de ocupación de cada vehículo, dependiendo del número  $k$  de unidades de emergencia ubicado en cada estación  $j$ . Basada en la definición de carga de trabajo de un vehículo, i.e., el tiempo esperado en el que la estación está ocupada durante el día dividido por el tiempo disponible de trabajo de todos las unidades de emergencia ubicadas en la estación, se propone la fracción de ocupación de cada vehículo en cada estación como:

$$r_j = \frac{\bar{t} \sum_{i \in B_j} f_i}{24x_j}, \tag{2. 64}$$

donde se ha supuesto que las únicas llamadas que aportan carga de trabajo a los vehículos de la estación  $j$  es el conjunto de nodos demanda situados a no más del tiempo máximo  $S$  de la estación. Basada en esta definición, podemos definir las fracciones de ocupación de un vehículo, dependiendo del número  $k$  de unidades de emergencia ubicado en cada estación  $j$  como

$$r_{jk} = \frac{\bar{t} \sum_{i \in B_j} f_i}{24k}. \tag{2. 65}$$

Dado que las fracciones de ocupación no pueden ser mayores que 1, para cada estación  $j$  se determina el mínimo número de unidades de emergencia  $k_{0j}$  que debería ser ubicado en cada estación para poder satisfacer toda la demanda requerida. Obtenidos estos valores podemos reformular el conjunto de restricciones (2. 61) y (2. 62) como

$$\sum_{j \in N_i} \sum_{k=k_{0j}}^L -\log((r_{jk})^k) x_{jk} \geq -\log(1-\alpha) \quad \forall i \in I \quad (2. 66)$$

$$\sum_{k=k_{0j}}^L x_{jk} \leq 1 \quad \forall j \in J, \quad (2. 67)$$

donde  $k_{0j} = \min\{k / k \in \{1,2,\dots,L\} \wedge r_{jk} \leq 1\}$ .

Con objeto de presentar una formulación que no esté condicionada por el número máximo de vehículos,  $L$ , podemos calcular el valor  $L$  que verifique

$$L = \text{Max}_{j \in J} \left\{ \min \left\{ k / -\log(r_{jk}^k) > -\log(1-\alpha) \right\} \right\}$$

Es decir, se obtiene para cada estación  $j$ , el valor de  $k$  mínimo de manera que pueda ser satisfecha la cantidad prefijada ubicando todos los vehículos en la estación  $j$ . A continuación se calcula el máximo de estas cantidades para cada posible ubicación.

Intuitivamente, esta versión basada en que la distribución de vehículos disponible se ajusta a una distribución Binomial debe obtener mejores resultados que el BPLSCP de *ReVelle* y *Hogan*, ya que está basada en fracciones de ocupación centradas en los nodos servicio, en vez de en los puntos demanda.

En las tablas siguientes se presentan los resultados de la experiencia computacional, en cada tabla se ha calculado el número de instancias que no verifican el nivel de fiabilidad prefijado y el porcentaje que suponen respecto a las presentadas en cada tabla.

**Tabla 2.20. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del BRLSCP para las instancias realizadas sobre la red de 55 nodos y distancia máxima S=1.5 y 3.**

	S	ALFA	BRLSCP			S	ALFA	BRLSCP		
			Nº de veh.	MLR- I	MLR- D			Nº de veh.	MLR- I	MLR- D
red 55.1	1.5	0.800	6	0.880	0.877	3	0.800	2	0.904	0.877
		0.825	7	0.906	0.902		0.825	2	0.904	0.877
		0.850	7	0.891	0.887		0.850	2	0.904	0.877
		0.875	7	0.891	0.887		0.875	2	0.904	0.877
		0.900	7	0.937	0.934		0.900	3	0.988	0.973
		0.925	8	0.939	0.937		0.925	3	0.988	0.973
		0.950	8	0.973	0.969		0.950	3	0.988	0.973
		0.975	10	0.992	0.991		0.975	3	0.988	0.973
		0.990	10	0.995	0.995		0.990	4	0.999	0.995
red 55.2	1.5	0.800	8	0.906	0.893	3	0.800	3	0.891	0.849
		0.825	8	0.911	0.898		0.825	3	0.891	0.849
		0.850	8	0.911	0.898		0.850	4	0.976	0.943
		0.875	9	0.943	0.935		0.875	4	0.976	0.943
		0.900	10	0.954	0.948		0.900	4	0.976	0.943
		0.925	10	0.937	0.935		0.925	4	0.976	0.943
		0.950	11	0.972	0.968		0.950	4	0.976	0.943
		0.975	13	0.994	0.992		0.975	5	0.997	0.982
		0.990	14	0.995	0.994		0.990	5	0.997	0.982
red 55.3	1.5	0.800	12	0.849	0.839	3	0.800	6	0.917	0.859
		0.825	13	0.862	0.854		0.825	6	0.917	0.859
		0.850	13	0.835	0.825		0.850	6	0.917	0.859
		0.875	14	0.941	0.921		0.875	7	0.972	0.920
		0.900	15	0.964	0.952		0.900	7	0.972	0.920
		0.925	15	0.964	0.952		0.925	7	0.972	0.920
		0.950	17	0.957	0.952		0.950	7	0.972	0.920
		0.975	17	0.985	0.981		0.975	8	0.993	0.959
		0.990	19	0.994	0.993		0.990	8	0.993	0.959
red 55.4	1.5	0.800	19	0.857	0.828	3	0.800	11	0.960	0.897
		0.825	19	0.925	0.909		0.825	11	0.960	0.897
		0.850	20	0.925	0.919		0.850	11	0.960	0.897
		0.875	20	0.925	0.919		0.875	11	0.960	0.897
		0.900	21	0.925	0.919		0.900	11	0.960	0.897
		0.925	22	0.977	0.960		0.925	11	0.960	0.897
		0.950	24	0.979	0.975		0.950	12	0.986	0.932
		0.975	24	0.988	0.975		0.975	12	0.986	0.932
		0.990	28	0.993	0.992		0.990	13	0.996	0.957
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	1	2	Nº de instancias que no verifican fiabilidad prefijada		0	12		
		% de instancias que no verifican fiabilidad prefijada	2.77	5.55	% de instancias que no verifican fiabilidad prefijada		0	33.33		

**Tabla 2.21. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del BRLSCP para las instancias realizadas sobre la red de 79 nodos y distancia máxima S=1.5 y 3.**

	S	ALFA	BRLSCP			S	ALFA	BRLSCP		
			Nº de veh.	MLR- I	MLR- D			Nº de veh.	MLR- I	MLR-D
red 79.1	1.5	0.800	11	0.845	0.844	3	0.800	6	0.878	0.862
		0.825	11	0.858	0.857			6	0.897	0.880
		0.850	12	0.886	0.886			6	0.895	0.879
		0.875	13	0.898	0.897			6	0.903	0.886
		0.900	13	0.979	0.977			7	0.961	0.951
		0.925	14	0.978	0.976			7	0.959	0.949
		0.950	14	0.977	0.976			7	0.970	0.963
		0.975	14	0.983	0.981			8	0.981	0.975
		0.990	18	0.996	0.996			10	0.993	0.991
red 79.2	1.5	0.800	13	0.830	0.829	3	0.800	7	0.889	0.864
		0.825	14	0.952	0.948			7	0.881	0.856
		0.850	14	0.941	0.936			7	0.882	0.857
		0.875	14	0.951	0.948			8	0.937	0.913
		0.900	14	0.947	0.943			8	0.931	0.911
		0.925	14	0.939	0.934			8	0.948	0.926
		0.950	15	0.969	0.967			9	0.978	0.968
		0.975	18	0.979	0.977			10	0.983	0.969
		0.990	20	0.992	0.992			11	0.994	0.988
red 79.3	1.5	0.800	18	0.856	0.842	3	0.800	12	0.871	0.824
		0.825	19	0.867	0.856			12	0.878	0.835
		0.850	20	0.910	0.893			13	0.919	0.866
		0.875	20	0.910	0.893			13	0.919	0.867
		0.900	21	0.946	0.942			13	0.919	0.867
		0.925	22	0.954	0.944			14	0.962	0.923
		0.950	23	0.964	0.958			15	0.965	0.946
		0.975	25	0.978	0.975			16	0.983	0.949
		0.990	28	0.994	0.992			18	0.996	0.987
red 79.4	1.5	0.800	25	0.862	0.849	3	0.800	17	0.826	0.790
		0.825	25	0.877	0.841			17	0.826	0.790
		0.850	26	0.893	0.862			19	0.920	0.853
		0.875	28	0.944	0.929			19	0.933	0.866
		0.900	29	0.963	0.954			19	0.921	0.853
		0.925	30	0.962	0.954			20	0.934	0.889
		0.950	30	0.968	0.953			20	0.963	0.903
		0.975	34	0.980	0.976			22	0.982	0.940
		0.990	37	0.992	0.990			24	0.996	0.977
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	0	2	Nº de instancias que no verifican fiabilidad prefijada			0	17	
		% de instancias que no verifican fiabilidad prefijada	0	5.55	% de instancias que no verifican fiabilidad prefijada			0	47.22	

La experiencia computacional muestra que el BRLSCP es un modelo que garantiza la obtención de una fiabilidad local mínima por encima del nivel de fiabilidad prefijado suponiendo independencia, excepto en un caso. Este resultado no sorprende debido a las hipótesis del modelo. Obviamente, cuanto mayor es la carga de trabajo soportada por los vehículos más veces incumple las restricciones de fiabilidad impuestas bajo la hipótesis de dependencia. En la última sección de este capítulo se mostrará como conseguir resultados similares con un menor número de vehículos.

### 2.6.3. El Queueing Reliability location set covering problem (QRLSCP).

En el último modelo de la secuencia de cubrimientos probabilísticos basados en cantidad de cubrimiento, se modeliza el comportamiento en cada entorno de cada estación como un sistema de colas con llegadas distribuidas Poisson, tiempos de servicio exponenciales y pérdida de llamadas en el caso de sistema saturado. Este nuevo modelo lo denominaremos Queueing Reliability Location Set Covering Problem (QRLSCP) y constituye una nueva versión (basada en los puntos servicios) del QPLSCP de *ReVelle* y *Marianov* (1994), que aún formulando las mismas hipótesis está basado en los puntos demanda. La formulación obtenida es un refinamiento del PRLSCP de *Ball* y *Lin*(1993), dado que incorpora en dicho modelo la hipótesis de pérdida de llamadas en el caso de sistema saturado. Obviamente, podemos calcular los coeficientes de las restricciones (2. 54) con la distribución de Poisson truncada como:

$$P(D(j) \geq k) = P(D(j) = k) = \frac{e^{-\lambda_j} \frac{\lambda_j^k}{k!}}{\sum_{s=0}^k e^{-\lambda_j} \frac{\lambda_j^s}{s!}} \quad \text{donde} \quad \lambda_j = \frac{T}{24} \left( \sum_{i \in B_j} f_i \right)$$

ya que la probabilidad de que una estación no pueda atender una llamada proveniente de algún nodo demanda es igual a la probabilidad de que dicha estación tenga todos sus vehículos ocupados.

En las tablas siguientes se presentan los resultados de la experiencia computacional. En cada tabla se ha calculado el número de instancias que no verifican el nivel de fiabilidad prefijado y el porcentaje que suponen respecto a las presentadas en cada tabla.

**Tabla 2.22. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del QRLSCP para las instancias realizadas sobre la red de 55 nodos y distancia máxima S=1.5 y 3.**

	S	ALFA	QRLSCP			S	ALFA	QRLSCP		
			Nº de veh.	MLR -I	MLR- D			Nº de veh.	MLR -I	MLR- D
red 55.1	1.5	0.800	6	0.837	0.834	3	0.800	2	0.904	0.877
		0.825	6	0.880	0.877		0.825	2	0.904	0.877
		0.850	7	0.891	0.887		0.850	2	0.904	0.877
		0.875	7	0.882	0.880		0.875	2	0.904	0.877
		0.900	7	0.915	0.912		0.900	3	0.988	0.973
		0.925	8	0.939	0.937		0.925	3	0.988	0.973
		0.950	9	0.953	0.952		0.950	3	0.988	0.973
		0.975	10	0.993	0.992		0.975	4	0.999	0.995
		0.990	11	0.995	0.995		0.990	4	0.999	0.995
red 55.2	1.5	0.800	8	0.821	0.815	3	0.800	3	0.891	0.849
		0.825	8	0.857	0.855		0.825	3	0.891	0.849
		0.850	8	0.906	0.893		0.850	4	0.976	0.943
		0.875	9	0.943	0.935		0.875	4	0.976	0.943
		0.900	9	0.936	0.935		0.900	4	0.976	0.943
		0.925	10	0.959	0.958		0.925	4	0.976	0.943
		0.950	12	0.960	0.959		0.950	5	0.997	0.982
		0.975	14	0.988	0.987		0.975	5	0.997	0.982
		0.990	15	0.998	0.998		0.990	6	1.000	0.995
red 55.3	1.5	0.800	12	0.850	0.839	3	0.800	6	0.917	0.859
		0.825	12	0.851	0.840		0.825	6	0.917	0.859
		0.850	13	0.860	0.852		0.850	6	0.917	0.859
		0.875	14	0.946	0.930		0.875	7	0.972	0.920
		0.900	15	0.969	0.959		0.900	7	0.972	0.920
		0.925	17	0.975	0.973		0.925	8	0.993	0.959
		0.950	17	0.985	0.981		0.950	8	0.993	0.959
		0.975	20	0.995	0.994		0.975	9	0.999	0.981
		0.990	23	0.995	0.994		0.990	10	1.000	0.992
red 55.4	1.5	0.800	17	0.876	0.846	3	0.800	10	0.914	0.853
		0.825	18	0.844	0.803		0.825	10	0.914	0.853
		0.850	19	0.925	0.909		0.850	10	0.914	0.853
		0.875	21	0.925	0.919		0.875	11	0.960	0.897
		0.900	21	0.925	0.919		0.900	12	0.986	0.932
		0.925	24	0.984	0.979		0.925	12	0.986	0.932
		0.950	25	0.979	0.976		0.950	13	0.996	0.957
		0.975	29	0.993	0.992		0.975	15	1.000	0.986
		0.990	33	0.998	0.998		0.990	16	1.000	0.992
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	0	1	Nº de instancias que no verifican fiabilidad prefijada		0	0		
		% de instancias que no verifican fiabilidad prefijada	0	2.77	% de instancias que no verifican fiabilidad prefijada		0	0		

**Tabla 2.23. Número de unidades de emergencia y fiabilidades locales mínimas alcanzadas por las soluciones del QRLSCP para las instancias realizadas sobre la red de 79 nodos y distancia máxima S=1.5 y 3.**

	S	ALFA	QRLSCP			S	ALFA	QRLSCP		
			Nº de veh.	MLR- I	MLR- D			Nº de veh.	MLR- I	MLR-D
red 79.1	1.5	0.800	10	0.812	0.812	3	0.800	6	0.864	0.847
		0.825	10	0.837	0.836			6	0.874	0.857
		0.850	11	0.859	0.858			6	0.917	0.904
		0.875	13	0.884	0.884			6	0.903	0.886
		0.900	13	0.977	0.975			7	0.964	0.955
		0.925	13	0.979	0.977			7	0.958	0.948
		0.950	14	0.980	0.979			8	0.981	0.973
		0.975	17	0.989	0.988			10	0.998	0.997
		0.990	19	0.999	0.999			10	0.997	0.995
red 79.2	1.5	0.800	13	0.841	0.840	3	0.800	7	0.887	0.868
		0.825	13	0.931	0.925			7	0.904	0.881
		0.850	13	0.942	0.937			7	0.885	0.863
		0.875	14	0.953	0.949			8	0.938	0.913
		0.900	14	0.959	0.956			8	0.938	0.915
		0.925	15	0.961	0.959			9	0.973	0.958
		0.950	16	0.974	0.972			10	0.978	0.975
		0.975	20	0.993	0.993			12	0.998	0.996
		0.990	22	0.998	0.998			13	0.999	0.998
red 79.3	1.5	0.800	18	0.858	0.846	3	0.800	11	0.803	0.774
		0.825	18	0.840	0.827			12	0.879	0.846
		0.850	20	0.894	0.888			13	0.933	0.880
		0.875	21	0.912	0.899			13	0.919	0.865
		0.900	21	0.946	0.942			14	0.962	0.922
		0.925	23	0.964	0.958			15	0.978	0.958
		0.950	25	0.978	0.975			17	0.991	0.979
		0.975	30	0.997	0.996			19	0.996	0.994
		0.990	33	0.998	0.998			22	1.000	0.999
red 79.4	1.5	0.800	23	0.831	0.795	3	0.800	16	0.738	0.718
		0.825	25	0.884	0.864			17	0.832	0.795
		0.850	26	0.888	0.877			18	0.891	0.822
		0.875	28	0.948	0.941			19	0.922	0.853
		0.900	30	0.957	0.946			20	0.966	0.909
		0.925	32	0.974	0.962			21	0.974	0.942
		0.950	35	0.987	0.985			24	0.996	0.976
		0.975	40	0.997	0.995			26	0.997	0.992
		0.990	45	0.999	0.999			29	1.000	0.998
RESUMEN										
		Nº de instancias que no verifican fiabilidad prefijada	0	1	Nº de instancias que no verifican fiabilidad prefijada		1	6		
		% de instancias que no verifican fiabilidad prefijada	0	2.77	% de instancias que no verifican fiabilidad prefijada		2.77	16.66		

La experiencia computacional muestra que el QRLSCP es un modelo que casi en todas las instancias evaluadas garantiza la obtención de una fiabilidad local mínima por encima del nivel de fiabilidad prefijado, tanto suponiendo independencia o dependencia. Notar que obtiene unos resultados mucho más ajustados que el PRLSCP, aunque en alguna ocasión se viole alguna restricción. En la última sección de este capítulo se analizará el comportamiento del modelo frente a los diferentes modelos de cubrimiento probabilísticos analizados.

## 2.7. Comparación global entre los modelos de cubrimiento total probabilístico.

En esta última sección dedicamos nuestra atención a comparar los diferentes modelos entre sí. Para establecer alguna medida de la eficiencia, en primer lugar, se ha obtenido para cada instancia estudiada la configuración con menor número de vehículos, entre todos los modelos presentados, que verifique las restricciones de fiabilidad impuestas. A continuación, se ha determinado el número de instancias para cada modelo que consigue satisfacer la fiabilidad requerida con un número igual al menor obtenido. Los resultados computacionales se presentan en las tablas siguientes, bajo la hipótesis de independencia o dependencia. En las dos primeras tablas se producen, con frecuencia, solapamientos (veasé la última columna en cada tabla).

**Tabla 2.22. Número de instancias y porcentaje global de instancias que verifican el nivel de fiabilidad prefijado con el menor número de vehículos conocido para diferentes modelos de cubrimiento total probabilístico suponiendo independencia.**

	Red de 55 y S=1.5	Red de 55 y S=3	Red de 79 y S=1.5	Red de 79 y S=3	total	%
BPLSCP	5	16	22	19	62	43.056
QPLSCP	9	14	11	12	46	31.944
PPLSCP	8	8	0	8	31	21.528
RBPLSCP	25	35	31	28	119	82.639
RPPLSCP	10	9	7	8	34	23.611
PRLSCP	10	9	5	7	31	21.528
BRLSCP	20	23	11	19	73	50.694
QRLSCP	22	32	8	34	96	66.667



Bajo la hipótesis de independencia, nuestro RBPLSCP con un 82.64%, nuestro QRLSCP con un 66.67% y nuestro BRLSCP con un 50.69% ocupan los tres primeros puestos del ranking, con bastante distancia respecto a los modelos existentes en la literatura especializada hasta la fecha. Notar, por ejemplo, que el modelo PRLSCP que en todas las instancias resueltas obtenía la fiabilidad impuesta, solo en un 21.5% de las instancias consigue una configuración con el menor número de vehículos conocido.

**Tabla 2.23. Número de instancias y porcentaje global de instancias que verifican el nivel de fiabilidad prefijado con el menor número de vehículos conocido para diferentes modelos de cubrimiento total probabilístico suponiendo dependencia.**

	Red de 55 y S=1.5	Red de 55 y S=3	Red de 79 y S=1.5	Red de 79 y S=3	total	%
BPLSCP	5	7	19	12	43	29.861
QPLSCP	10	11	11	13	45	31.250
PPLSCP	12	9	8	11	40	27.778
RBPLSCP	18	13	24	16	71	49.306
RPPLSCP	14	14	8	13	49	34.028
PRLSCP	10	14	5	8	37	25.694
BRLSCP	20	36	14	21	91	63.194
QRLSCP	25	21	15	19	80	55.556

Bajo la hipótesis de dependencia, nuestro BRLSCP con un 63,19%, nuestro QRLSCP con un 55.56%, nuestro RBPLSCP con un 49.31% y el RPPLSCP con un 34.03% ocupan los cuatro primeros puestos del ranking.

En segundo lugar, se ha determinado en que instancias un único modelo obtenía el menor número de vehículos, verificando las restricciones de fiabilidad impuestas, para determinar el grado de mejores soluciones aportadas en exclusiva por cada modelo. Los resultados computacionales se presentan en las tablas siguientes, bajo las dos hipótesis de independencia o dependencia.

**Tabla 2.24. Número de instancias y porcentaje global de instancias que verifican el nivel de fiabilidad prefijado siendo la mejor y la única solución conocida para diferentes modelos de cubrimiento total probabilístico suponiendo independencia.**

	Red de 55 y S=1.5	Red de 55 y S=3	Red de 79 y S=1.5	Red de 79 y S=3	total	%
BPLSCP	0	0	0	0	0	0.000
QPLSCP	0	0	1	0	1	0.694
PPLSCP	0	0	0	0	0	0.000
RBPLSCP	6	1	8	0	15	10.417
RPPLSCP	0	0	0	0	0	0.000
PRLSCP	0	0	0	0	0	0.000
BRLSCP	1	1	3	1	6	4.167
QRLSCP	3	0	0	4	7	4.861

Bajo la hipótesis de independencia, nuestro RBPLSCP con un 10.41%, nuestro QRLSCP con un 4.86% y nuestro BRLSCP con un 4.16% ocupan los tres primeros puestos del ranking, aportando ellos tres en el conjunto de instancias resueltas casi un 20% de mejores soluciones.

**Tabla 2.25. Número de instancias y porcentaje global de instancias que verifican el nivel de fiabilidad prefijado siendo la mejor y la única solución conocida para diferentes modelos de cubrimiento total probabilístico suponiendo dependencia**

	Red de 55 y S=1.5	Red de 55 y S=3	Red de 79 y S=1.5	Red de 79 y S=3	total	%
BPLSCP	0	0	0	0	0	0.000
QPLSCP	1	0	1	3	5	3.472
PPLSCP	0	0	0	0	0	0.000
RBPLSCP	1	0	4	1	6	4.167
RPPLSCP	0	0	0	2	2	1.389
PRLSCP	0	0	0	1	1	0.694
BRLSCP	2	8	3	4	17	11.806
QRLSCP	4	0	4	2	10	6.944

Bajo la hipótesis de dependencia, nuestro BRLSCP con un 11.8%, nuestro QRLSCP con un 6.9%, nuestro RBPLSCP con un 4.16% y nuestro RPPLSCP con un 1.38% aportan conjuntamente aproximadamente un 24% de mejores soluciones.

A la vista de los resultados presentados, tanto por lo que se refiere a configuraciones con menos vehículos cumpliendo la fiabilidad, como a la obtención de la mejor solución conocida, podemos concluir que nuestros modelos presentan un mejor comportamiento que el resto de los modelos existentes hasta la fecha en la literatura.

## Capítulo 3. Modelo del cubrimiento total generalizado

### 3.1. Introducción y formulación.

Todos los modelos de cubrimiento total probabilísticos presentados en el capítulo anterior pueden ser expresados mediante un modelo general que llamaremos Generalized Set Covering Problem (GSCP). Para comprender el proceso de generalización llevado a cabo, se comenzará con la presentación de la formulación del Set Covering Problem (SCP) (*Hakimi, 1963,1964*):

$$\text{Min } \mathbf{w}'\mathbf{x} \quad (3.1)$$

$$\text{s.a. } \mathbf{Ax} \geq \mathbf{e} \quad (3.2)$$

$$\mathbf{x} \in \{0,1\}^n, \quad (3.3)$$

donde  $\mathbf{A} = [a_{ij}]$  es una matriz denominada matriz del cubrimiento de dimensión  $m \times n$  cuyos elementos  $a_{ij}$  valen 1 ó 0, dependiendo de que la fila o punto demanda  $i$  esté cubierta o no por la columna o punto servicio  $j$ ,  $\mathbf{e} \in \mathcal{R}^m$  es el vector formado por  $m$  unos,  $\mathbf{w} \in \mathcal{R}_+^n$  es el vector de costes y  $\mathbf{x}$  es el vector de las variables a determinar, siendo

$$x_j = \begin{cases} 1, & \text{si la columna } j \text{ está en la solución} \\ 0, & \text{en caso contrario} \end{cases} \quad j = 1, 2, \dots, n.$$

El problema consiste en cubrir las  $m$  filas de  $\mathbf{A}$  mediante un subconjunto de las  $n$  columnas de manera que el coste sea mínimo. La inecuación (3. 2) nos asegura que cada fila estará cubierta por al menos una columna, mientras que (3. 3) explicita el carácter binario de las variables.

La siguiente extensión del SCP incorpora, en primer lugar, la posibilidad de determinar el número de facilidades a ubicar en un punto servicio. En segundo lugar, incluye un concepto de cantidad de cubrimiento, es decir, el cubrimiento que se permite no es el clásico “cubierto” o “no cubierto” si no que se fija una cantidad de cubrimiento que debe ser satisfecho para cada punto demanda para ser considerado cubierto. La ubicación de varias facilidades en el mismo punto servicio aporta una cantidad de cubrimiento determinado que puede no satisfacer las necesidades de los puntos cubiertos por él. El problema consiste en cubrir las necesidades de cubrimiento de los  $m$  puntos demanda mediante un número de facilidades de manera que el coste sea mínimo. La formulación usual del Generalized Set Covering Problem (GSCP) es

$$\text{Min } \mathbf{w}^t \mathbf{x} \quad (3.4)$$

$$\text{s.a. } \mathbf{D}\mathbf{x} \geq \mathbf{c} \quad (3.5)$$

$$\mathbf{x} \in \mathbb{Z}_+^n, \quad (3.6)$$

donde  $\mathbf{D} = [d_{ij}]$  es una matriz denominada matriz cantidad de cubrimiento de dimensión  $m \times n$  cuyos elementos  $d_{ij}$  indican la cantidad de cubrimiento que aportan si la fila o punto demanda  $i$  está cubierta por la columna o punto servicio  $j$ , tomando el valor cero en caso de no cubrir al punto demanda  $i$ . Por otra parte,  $\mathbf{c} \in \mathbb{R}^m$  es el vector que determina la cantidad de cubrimiento a satisfacer para cada punto demanda,  $\mathbf{w} \in \mathbb{R}_+^n$  es el vector de costes y  $\mathbf{x}$  es el vector del número de facilidades a ubicar en cada punto servicio  $j$ .

La extensión que se propone para el GSCP incorpora la posibilidad de que la cantidad de cubrimiento puede ser modelizada de una manera más eficiente y permita reflejar la importancia de ubicar diferentes unidades en el punto servicio  $j$ , ya que en el GSCP si la cantidad de cubrimiento es  $d_{ij}$  y asignamos  $x_j$  unidades, la cantidad cubierta es de  $d_{ij}x_j$ . La nueva forma general del Generalized Set Covering Problem (GSCP):

$$\text{Min } \mathbf{w}^t \mathbf{x} \quad (3.7)$$

$$\text{s.a. } \mathbf{A}\mathbf{B}\mathbf{x} \geq \mathbf{c} \quad (3.8)$$

$$\mathbf{G}\mathbf{x} \leq \mathbf{e} \quad (3.9)$$

$$\mathbf{x} \in \{0,1\}^p, \quad (3.10)$$

donde:

$\mathbf{A} = [a_{ij}]$  es una matriz denominada matriz del cubrimiento de dimensión  $m \times n$  cuyos elementos  $a_{ij}$  valen 1 ó 0, dependiendo de que la fila o punto demanda  $i$  esté cubierta o no por la columna o punto servicio  $j$ ,

$\mathbf{B} = [b_{ij}]$  es una matriz denominada matriz cantidad de cubrimiento de dimensión  $n \times p$ , donde  $p = nL$ , siendo  $L$  el número máximo de facilidades que pueden ubicarse en cualquier punto servicio  $j$ .

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1L} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & & & & b_{21} & b_{22} & \dots & b_{2L} & \mathbf{0} & \vdots \\ \vdots & & & & \vdots & & \ddots & & \mathbf{0} \\ \mathbf{0} & & & & \dots & & \mathbf{0} & b_{n1} & b_{n2} & \dots & b_{nL} \end{bmatrix}$$

Los elementos  $b_{jk}$  para  $j=1,2,\dots,n$  y  $k=1,2,\dots,L$  son mayores o iguales que cero, interpretándose estos elementos como la cantidad de cubrimiento que aporta la ubicación de  $k$  facilidades en el punto servicio  $j$ . Obviamente,  $b_{jk} < b_{j,k+1}$  para  $k=1,2,\dots,L-1$ .

$\mathbf{c} \in \mathcal{R}_+^m$  es el vector que determina la cantidad de cubrimiento a satisfacer para cada punto demanda.

$\mathbf{G} = [g_{ij}]$  es denominada matriz unidad generalizada de dimensión  $n \times p$ .

$$\mathbf{G} = \begin{bmatrix} 1 & 1 \dots 1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & & 1 & 1 \dots 1 & \mathbf{0} & \vdots \\ \vdots & & \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & & \dots & & \mathbf{0} & 1 & 1 \dots 1 \end{bmatrix}$$

$\mathbf{e} \in \mathcal{R}^n$  es el vector formado por  $n$  unos.

$\mathbf{w} \in \mathcal{R}_+^p$  es el vector de costes, cuya representación es

$$\mathbf{w}^t = (w_{11}, w_{12}, \dots, w_{1L}, w_{21}, w_{22}, \dots, w_{2L}, \dots, w_{n1}, w_{n2}, \dots, w_{nL}),$$

siendo  $w_{jk}$  =coste de emplazar  $k$  facilidades en el puesto de servicio  $j$  para  $j=1,2,\dots,n$  y  $k=1,2,\dots,L$ . Obviamente,  $w_{jk} < w_{j,k+1}$  para  $k=1,2,\dots,L-1$ .

$\mathbf{x} \in \{0,1\}^p$  es el vector de variables, cuya representación es

$$\mathbf{x}^t = (x_{11}, x_{12}, \dots, x_{1L}, x_{21}, x_{22}, \dots, x_{2L}, \dots, x_{n1}, x_{n2}, \dots, x_{nL}),$$

siendo

$$x_{jk} = \begin{cases} 1, & \text{si } k \text{ facilidades son instaladas en el puesto servicio } j \\ 0, & \text{en caso contrario} \end{cases}$$

para  $j=1,2,\dots,n$  y  $k=1,2,\dots,L$ .

El problema consiste en cubrir las  $m$  filas de  $\mathbf{A}$  mediante la cantidad de cubrimiento definido por la matriz  $\mathbf{B}$ , verificando las necesidades definidas por el vector  $\mathbf{c}$  de forma que el coste sea mínimo. La inequación (3. 8) nos asegura que cada fila obtendrá el cubrimiento deseado, mientras que (3. 9) asegura que o no se ubica nada o se instalan un número  $k$  de facilidades,  $k \leq L$  y (3. 10) determina el carácter binario de las variables.

La formulación del GSCP en formato estándar es

$$\text{Min} \quad \sum_{j=1}^n \sum_{k=1}^L w_{jk} x_{jk} \quad (3. 11)$$

$$\text{s.a.} \quad \sum_{j=1}^n \sum_{k=1}^L a_{ij} b_{jk} x_{jk} \geq c_i \quad i = 1,2,\dots,m \quad (3. 12)$$

$$\sum_{k=1}^L x_{jk} \leq 1 \quad j = 1,2,\dots,n \quad (3. 13)$$

$$x_{jk} \in \{0,1\} \quad j = 1,2,\dots,n; \quad k = 1,2,\dots,L. \quad (3. 14)$$

### 3.2. Propiedades básicas del GSCP

En esta sección se presentarán algunas propiedades elementales del GSCP que permitirán la reducción del problema original por eliminación de variables o de restricciones. Estas propiedades se inspiran en las correspondientes al SCP dadas por *Balinski* (1965), *Lemke et al.* (1971) y *Garfinkel y Nemhauser* (1972).

**Propiedad 3.1.** El problema es irresoluble si, y sólo si,

$$\exists i_0 \quad / \quad \sum_{j=1}^n a_{i_0j} b_{jL} < c_{i_0}$$

Obviamente, es trivial que si la cantidad máxima de cubrimiento que puede ubicarse para satisfacer la demanda del punto demanda  $i_0$  es menor que la cantidad requerida el problema no tiene ninguna solución factible.

**Definición 3.1.** Un punto demanda diremos que es un nodo simple cuando sólo existe un único punto-servicio que pueda aportar cubrimiento, o, equivalentemente, cuando la correspondiente fila de  $\mathbf{A}$  es un vector de la base canónica.

**Propiedad 3.2.** Si la fila  $i$  de  $\mathbf{A}$  es un nodo simple cubierto por el punto servicio  $j_0$ , entonces necesariamente la ubicación  $j_0$  deberá formar parte de cualquier solución. Basándonos en la restricción

$$\sum_{k=1}^L x_{j_0k} \leq 1, \quad (3.15)$$

podemos añadir las restricciones más fuertes de la forma

$$\sum_{k=1}^{k_0-1} x_{j_0k} = 0 \quad (3.16)$$

$$\sum_{k=k_0}^L x_{j_0k} = 1, \quad (3.17)$$

donde  $k_0 = \min\{r \in \{1, 2, \dots, L\} / b_{j_0r} \geq c_i\}$ . Dado que  $k_0$  asegura las necesidades de cubrimiento del nodo  $i$ , el número de unidades ubicadas en  $j_0$  deberá ser mayor o igual que  $k_0$ .

**Definición 3.2.** Una columna  $\mathbf{a}_r$  está dominada por otra columna  $\mathbf{a}_s$ , si se verifica

- $a_{is} \geq a_{ir}$  para todo punto demanda  $i$  (definición de dominancia en el SCP)
- $b_{sk} \geq b_{rk}$  para  $k=1, 2, \dots, L$
- $w_{sk} \leq w_{rk}$  para  $k=1, 2, \dots, L$  (definición de dominancia en el SCP)

Es decir,  $\mathbf{a}_s$  domina  $\mathbf{a}_r$ , porque el conjunto de filas cubiertas por la columna  $r$  está dentro del cubierto por la columna  $s$ , cumpliéndose que la cantidad de cubrimiento que aporta facilidades ubicadas en  $s$  es mayor o igual que la que aporta si son ubicadas en  $r$  y consiguiéndose con un coste menor o igual que con la columna  $r$ .

**Propiedad 3.3.** Si el punto servicio  $\mathbf{a}_r$  está dominado por el punto servicio  $\mathbf{a}_s$ , entonces el punto servicio  $\mathbf{a}_r$  puede eliminarse, ya que se puede conseguir lo mismo ubicando facilidades en el punto servicio  $s$ .

**Definición 3.3.** Una fila  $i$  está dominada por la fila  $t$  si se verifica

- $a_{ij} \geq a_{tj}$  para todo punto servicio  $j$  (definición de dominancia en el SCP)
- $c_t \geq c_i$

Es decir, el punto demanda  $t$  domina al punto demanda  $i$ , si toda columna que cubre a la fila  $t$  cubre también a la fila  $i$  y cumpliéndose que la cantidad de cubrimiento demandado por el punto demanda  $t$  es mayor o igual que el solicitado por  $i$ .

**Propiedad 3.4.** Si el punto demanda  $i$  está dominado por el punto demanda  $t$ , entonces se puede eliminar la fila  $i$  del problema, ya que el cumplimiento de la cantidad de cubrimiento a satisfacer para el punto demanda  $t$ , automáticamente satisface las necesidades de cubrimiento del punto demanda  $i$ .

Las propiedades 3.2, 3.3 y 3.4 son una generalización de las técnicas de reducción usadas en el SCP. A continuación se visualizará la utilización de estas propiedades sobre un ejemplo.

**Ejemplo 3.1.** Dadas las siguientes matrices que definen el GSCP con  $m=7$ ,  $n=6$  y  $L=3$

$$\begin{array}{l}
 \text{fila 1} \\
 \text{fila 2} \\
 \text{fila 3} \\
 \text{fila 4} \\
 \text{fila 5} \\
 \text{fila 6} \\
 \text{fila 7}
 \end{array}
 \mathbf{A} = \begin{pmatrix}
 1 & 1 & 0 & 1 & 0 & 0 \\
 1 & 1 & 0 & 0 & 1 & 0 \\
 0 & 0 & 1 & 0 & 1 & 1 \\
 1 & 1 & 0 & 1 & 1 & 0 \\
 0 & 0 & 1 & 1 & 1 & 0 \\
 0 & 0 & 1 & 0 & 0 & 1 \\
 0 & 0 & 0 & 1 & 0 & 0
 \end{pmatrix}
 \quad
 \mathbf{c} = \begin{pmatrix}
 12 \\
 14 \\
 15 \\
 13 \\
 12 \\
 18 \\
 10
 \end{pmatrix}$$

$$\begin{array}{l}
 \text{col 1} \\
 \text{col 2} \\
 \text{col 3} \\
 \text{col 4} \\
 \text{col 5} \\
 \text{col 6}
 \end{array}
 \mathbf{B} = \begin{pmatrix}
 9 & 11 & 13 & & & \\
 & & & 8 & 10 & 12 \\
 & & & & & & 7 & 13 & 14 \\
 & & & & & & & & & 8 & 11 & 13 \\
 & & & & & & & & & & & & 6 & 8 & 9 \\
 & & & & & & & & & & & & & & & 5 & 6 & 13
 \end{pmatrix}$$

$$\mathbf{w}^t = (6 \ 7 \ 8 \ 7 \ 8 \ 9 \ 10 \ 12 \ 14 \ 9 \ 13 \ 16 \ 14 \ 15 \ 17 \ 4 \ 5 \ 6).$$

- La fila 4 está dominada en el sentido del SCP por la fila 1, pero  $c_1=12$  y  $c_4=13$ , luego no satisface las condiciones de la dominancia entre filas y no podemos eliminar la fila 4.



- La fila 4 está dominada en el sentido del SCP por la fila 2, verificándose que  $c_2=14$  y  $c_4=13$ , luego se satisfacen las condiciones de la dominancia entre filas y podemos eliminar la fila 4.
- La fila 3 está dominada en el sentido del SCP por la fila 6, verificándose que  $c_6=18$  y  $c_3=15$ , luego se satisfacen las condiciones de la dominancia entre filas y podemos eliminar la fila 3.
- Las filas 1 y 5 están dominadas por la fila 7, pero  $c_7$  es menor que  $c_1$  y  $c_5$ , luego no satisface las condiciones de la dominancia entre filas y no podemos eliminar ninguna de las dos filas.
- La fila 7 es un nodo simple, ya que la correspondiente fila de  $\mathbf{A}$  es un vector de la base canónica. Por tanto podemos añadir a la formulación del GSCP las restricciones

$$\begin{aligned}x_{41} &= 0 \\x_{42} + x_{43} &= 1,\end{aligned}$$

dado que el punto servicio 4 es el único que puede satisfacer su demanda y el valor de  $k_0$  es igual a 2, puesto que  $b_{41}=8 < c_7=10$  y  $b_{42}=11 > c_7=10$ .

- La columna 2 está dominada por la columna 1, ya que el conjunto de filas cubiertas por la columna 2 está dentro del cubierto por la columna 1, cumpliéndose que la cantidad de cubrimiento aportado por facilidades en el puesto de servicio 1 es mayor o igual que la que aporta si son ubicadas en el 2 y consiguiéndose con un coste menor o igual. Por tanto, la columna 2 puede ser eliminada.
- La columna 6 no está dominada por la columna 3, ya que el conjunto de filas cubiertas por la columna 3 está dentro del cubierto por la columna 6, cumpliéndose que la cantidad de cubrimiento aportado por facilidades en el puesto de servicio 3 es mayor o igual que la que aporta si son ubicadas en el 6, pero no podemos eliminar la columna 6 dado que no se consigue con un coste menor o igual.

Las propiedades 3.2, 3.3 y 3.4 permiten reducir el tamaño del problema original y por otra parte, fijar ciertas variables a nivel 0. Las variables fijadas a nivel 0 coinciden con las asociadas con los puntos de servicio  $j$  eliminados y con las asociadas a una columna  $j$  que sirve a un nodo simple y no garantizan la cantidad de cubrimiento del nodo simple. El nuevo problema reducido de GSCP que se resuelve tiene  $m=5$ ,  $n=5$  y  $L=3$ .



Considerando  $L=1$ ,  $\mathbf{B}=\mathbf{I}_n$ ,  $\mathbf{G}=\mathbf{I}_n$ ,  $\mathbf{c}=\mathbf{e}_m$  y  $\mathbf{w}=\mathbf{e}_n$  obtenemos la formulación del LSCP una vez eliminadas el conjunto de restricciones (3.9).

- MSCP (Multicovering Problem). El problema del multicovering es una generalización del SCP, donde se determina el número de veces  $b_i$  que cada nodo demanda debe ser cubierto. La literatura revisada incluye una aplicación a un problema de radio-comunicaciones en Alemania de *Van Slyke* (1981), así como diferentes heurísticos como el propuesto por *Hall y Hochbaum* (1986) o los propuestos por *Gonsalvez et al* (1987).

Considerando  $L=1$ ,  $\mathbf{B}=\mathbf{I}_n$ ,  $\mathbf{G}=\mathbf{I}_n$ ,  $\mathbf{c}^t=(b_1, b_2, \dots, b_m)$  obtenemos la formulación del MSCP con el conjunto de restricciones (3.9) redundantes.

- BPLSCP (Binomial Location Set Covering Problem) y QPLSCP (Queueing Probabilistic Locations set Covering Problem) y PPLSCP (Poisson Location Set Covering Problem) descritos en el capítulo anterior.

*ReVelle y Hogan* (1988) presentan el BPLSCP, un modelo que minimiza el número de vehículos de emergencia con la restricción de tener para cada nodo demanda una probabilidad de ser atendido dentro del tiempo  $S$  mayor que el nivel de fiabilidad prefijado. Estas restricciones de cubrimiento probabilístico, bajo la hipótesis de una distribución binomial para el número de vehículos ocupados, se utilizan para obtener el número de veces  $b_i$  que cada nodo demanda debe ser cubierto, permitiendo la ubicación de más de un vehículo en la misma estación.

*Marianov y ReVelle* (1994) presentan el QPLSCP, con la misma formulación general que el BPLSCP pero con la hipótesis alternativa relativa a la distribución del número de vehículos ocupados que ahora pasa a ser una Poisson truncada, que permite obtener el número de veces  $b_i$  que cada nodo demanda debe ser cubierto, permitiendo la ubicación de más de un vehículo en la misma estación.

En el capítulo anterior se formuló, por vez primera, el PPLSCP, con la misma formulación general que el BPLSCP pero con la hipótesis alternativa de una distribución de Poisson para la distribución de vehículos ocupados. Todos ellos son casos particulares del modelo general con las siguientes elecciones:

$$L = \underset{i=1,2,\dots,m}{Max} \{b_i\}, p=nL, \mathbf{c}^t=(b_1,b_2,\dots,b_m) \text{ y}$$

$$\mathbf{B} = \begin{bmatrix} 1 & 2 \dots L & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & 1 & 2 \dots L & \mathbf{0} & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \dots & \mathbf{0} & 1 & 2 \dots L \end{bmatrix}$$

$$\mathbf{w}^t = (1,2, \dots, L, 1,2, \dots, L, \dots, 1,2, \dots, L).$$

- PRLSCP (Poisson Reliability Location Set Covering Problem). *Ball y Lin* (1993) han formulado el PRLSCP, con la hipótesis que las llamadas de emergencia se distribuyen según una Poisson y que T es una cota superior para el tiempo de servicio.

$$\text{Considerando } \lambda_j = \frac{T}{24} \left( \sum_{i \in B_j} f_i \right), \text{ tenemos que los } b_{jk} = -\log \left( 1 - \sum_{s=0}^{k-1} e^{-\lambda_j} \frac{\lambda_j^s}{s!} \right),$$

$$L = \underset{j=1,2,\dots,m}{Max} \left\{ \min \{ k / b_{jk} > -\log(1-\alpha) \} \right\}, \mathbf{c}^t = (-\log(1-\alpha), -\log(1-\alpha), \dots, -\log(1-\alpha)) \text{ y}$$

$$\mathbf{w}^t = (1,2, \dots, L, 1,2, \dots, L, \dots, 1,2, \dots, L).$$

- QRLSCP (Queueing Reliability Location Set Covering Problem). Formulada en el capítulo anterior, con las hipótesis que las llamadas de emergencia se distribuyen según una Poisson, T es una cota superior para el tiempo de servicio y que el número de vehículos ocupados sigue una distribución de Poisson truncada.

$$\text{Considerando } \lambda_j = \frac{T}{24} \left( \sum_{i \in B_j} f_i \right), \text{ tenemos que los } b_{jk} = -\log \left( \frac{e^{-\lambda_j} \frac{\lambda_j^k}{k!}}{\sum_{s=0}^k e^{-\lambda_j} \frac{\lambda_j^s}{s!}} \right),$$

$$L = \underset{j=1,2,\dots,m}{Max} \left\{ \min \{ k / b_{jk} > -\log(1-\alpha) \} \right\}, \mathbf{c}^t = (-\log(1-\alpha), -\log(1-\alpha), \dots, -\log(1-\alpha)) \text{ y}$$

$$\mathbf{w}^t = (1,2, \dots, L, 1,2, \dots, L, \dots, 1,2, \dots, L).$$

- BRLSCP (Binomial Reliability Location Set Covering Problem). Formulado en el capítulo anterior, con las hipótesis T es el tiempo medio de servicio y que el número de vehículos ocupados sigue una distribución Binomial.

Considerando  $\lambda_j = \frac{T}{24} \left( \sum_{i \in B_j} f_i \right)$ , tenemos que los  $b_{jk} = -\log \left( \left( \frac{\lambda_j}{k} \right)^k \right)$ ,

$L = \underset{j=1,2,\dots,m}{Max} \left\{ \min \{ k / b_{jk} > -\log(1-\alpha) \} \right\}$ ,  $\mathbf{c}^t = (-\log(1-\alpha), -\log(1-\alpha), \dots, -\log(1-\alpha))$  y

$\mathbf{w}^t = (1, 2, \dots, L, 1, 2, \dots, L, \dots, 1, 2, \dots, L)$ .

Dado que el SCP es un caso particular del GSCP, y conociendo, según demostró *Karp* (1972), que el SCP es un problema NP-completo, se puede inferir la NP-completitud del GSCP. Es decir, es muy improbable que exista un algoritmo que requiera un tiempo de resolución acotado por una función polinómica del tamaño del problema para el GSCP, y por tanto los tiempos de cálculo crecerán exponencialmente con el tamaño del problema.

El GSCP pertenece al campo de la programación entera, cuya resolución no es sencilla debido a la integralidad de las variables. En la resolución de problemas de programación entera se utilizan comúnmente esquemas de Ramificación y Acotación (Branch & Bound) o relacionados. El éxito de estos métodos depende de la rapidez con la que se encuentre una buena solución factible que permita eliminar muchos nodos del árbol de ramificación. Los algoritmos heurísticos o aproximativos son aquellos que, al ser aplicados sobre problemas de modelos de difícil solución, permiten encontrar buenas soluciones factibles en tiempos razonables. Por tanto, puesto que no se garantiza la optimalidad, la rapidez con que se localicen dichas soluciones, y lo próximo que se encuentre su valor del valor óptimo, serán las dos medidas utilizadas para el estudio de la bondad de estos procedimientos.

### 3.4. Un Heurístico para el GSCP

Muchos algoritmos han sido propuestos en la literatura para la resolución exacta del Set Covering Problem (veasé *Balas y Ho* (1980), *Beasley* (1987), *Fisher y Kedia* (1990), *Beasley y Jörnsten* (1992), *Nobili y Sassano* (1992), y *Balas y Carrera* (1996)) que permiten resolver instancias de no más de unos pocos cientos de filas y unos pocos miles de columnas. Para grandes instancias del SCP, los procedimientos heurísticos son necesarios.

Los algoritmos *greedy* son muy rápidos en la práctica, pero no dan una alta calidad en las soluciones, como es señalado por *Balas y Ho* (1980) o *Balas y Carrera* (1996). *Almiñana y Pastor* (1995) han desarrollado dos nuevos heurísticos *greedy* para el LSCP, denominados FMC y CMA, que se diferencian en la selección del conjunto de variables candidatas a entrar en la solución parcial. En el FMC se consideran aquellas que cubren a las filas menos cubiertas y en el CMA se tiene en cuenta las variables que cubren a un mayor número de filas, conteniendo como elementos más destacados un paso de búsqueda de nodos simples y otra regla de desempate basada en maximizar las concomitancias hacia delante de las variables candidatas.

Los heurísticos más efectivos para el SCP, están basados en la relajación lagrangiana con optimización subgradiente, siguiendo los trabajos seminales de *Balas y Ho* (1980), y las mejoras de *Beasley* (1990), *Fisher y Kedia* (1990), *Ceria et al.* (1995), *Balas y Carrera* (1996), *Pezzella y Faggioli* (1997) y *Caprara et al.* (1999). *Lorena y Lopes* (1994) proponen una aproximación análoga basada en la relajación subrogada. *Wedelin* (1995) propone un heurístico general para problemas enteros con matriz de restricciones 0-1. *Beasley y Chu* (1996) formulan un eficiente heurístico genético. *Almiñana y Pastor* (1997) han construido un nuevo heurístico, denominado RS, que se basa en una combinación de la relajación lagrangiana y de la resolución de problemas subrogados.

Con estas ideas, en las siguientes secciones se va a desarrollar un heurístico para el GSCP, que denotaremos por HGSCP, basado en las siguientes fases:

- a. Obtención de una cota superior mediante un procedimiento heurístico *greedy*.
- b. Cálculo de una cota inferior mediante un heurístico de ascenso dual.
- c. Mejora de la cota inferior mediante un procedimiento subgradiente que comienza con el último vector de multiplicadores lagrangianos calculado (en la primera aplicación del procedimiento se utiliza la solución obtenida en la fase b).
- d. Mejora de la cota superior mediante un procedimiento heurístico *greedy* basado en las variables duales cuando la cota inferior es mejorada en la fase c).

- e. Criterios de parada del procedimiento iterativo. Si los criterios de parada no son satisfechos, las fases c) y d) son repetidas hasta que algún criterio de parada se verifique.

### 3.4.1. Obtención de una cota superior inicial

En esta sección, vamos a desarrollar un heurístico tipo “greedy 1-óptimo” para obtener una solución factible inicial para el GSCP. Antes de describir el heurístico prestaremos atención a los aspectos notacionales. Sea  $N = \{1, 2, \dots, n\}$  el conjunto de índices asociado con los puntos-servicio, que denotaremos por el índice  $j$  y  $V = \{1, 2, \dots, v\}$  el conjunto de índices asociados con el número de vehículos,  $k$ , que se pueden ubicar en cada punto-servicio.

Consideremos, también, los conjuntos:

$I = \{k_{j0} / j \in N\}$  = Cota inferior del número de vehículos a ubicar en cada posición  $j$ ,

$S = \{k_{j1} / j \in N\}$  = Cota superior del número de vehículos a ubicar en cada posición  $j$ ,

entonces la solución parcial en la iteración  $r$ -ésima la identificaremos mediante los conjuntos

$S_r$ , subconjunto de  $N$  que contiene las estaciones seleccionadas hasta el momento y

$U_r = \{k_j / j \in S_r\}$  = Número de vehículos ubicados en cada posición  $j$ . Además, denotaremos

por  $M = \{1, 2, \dots, m\}$  al conjunto de puntos-demanda cuya cantidad de cubrimiento debemos satisfacer y por  $i$  a uno cualquiera de dichos punto-demanda. Finalmente  $F_r$ , denota al

subconjunto de  $M$  que contiene en la iteración  $r$  del algoritmo las filas cuya solicitud de cubrimiento no ha sido totalmente satisfecha por la solución parcial  $S_r$ . Este subconjunto  $F_r$  se

irá reduciendo hasta que se cumpla la condición  $F_r = \emptyset$ . También consideraremos los siguientes conjuntos:

$M_j = \{i \in M / a_{ij} = 1\}$  (filas cubiertas por la columna  $j$ ),

$N_i = \{j \in N / a_{ij} = 1\}$  (columnas que cubren a la fila  $i$ ).

Con esta notación, nuestro heurístico “greedy 1-óptimo”, consta de los siguientes pasos:

**Algoritmo “greedy 1-óptimo”**

Paso 1. Inicialización: hacer  $M = \{1, 2, \dots, m\}$ ,  $V = \{1, 2, \dots, v\}$ ,  $N = \{1, 2, \dots, n\}$ ,  $F_1 = M$ ,  
 $S_1 = \emptyset$ ,  $U_1 = \emptyset$ ,  $I = \{0, 0, \dots, 0\}$ ,  $S = \{v, v, \dots, v\}$ ,  $G_i = c_i$  y  $r = 1$ .

Paso 2. (Comienzo de la iteración r).

Para cada punto-servicio j y número de vehículos k, calcular el cubrimiento efectivo por unidad de coste asociado con la inclusión de la estación j con k vehículos en la solución parcial  $S_r$  y  $U_r$

$$d_{jk} = \frac{\sum_{i \in F_r} \min(G_i, b_{jk}) a_{ij}}{w_{jk}} \quad \text{si } j \notin S_r, \quad k_{j0} < k \leq k_{j1}, \quad (3.18)$$

$$d_{jk} = \frac{\sum_{i \in F_r} \min(G_i, b_{jk} - b_{jk_j}) a_{ij}}{w_{jk}} \quad \text{si } j \in S_r, \quad k_j < k \leq k_{j1}, \quad (3.19)$$

y calcular el máximo  $d_{jk}^*$  de los cubrimientos efectivos por unidad de coste.

Sea  $P^*$  el conjunto de los pares (j,k) donde se obtiene el máximo  $d_{jk}^*$ .

Paso 3. Si  $|P^*| = 1$ , sea  $(j^0, k^0)$  el único par de  $P^*$ . En tal caso, ir al Paso 6; en otro caso, continuar.

Paso 4. Para cada punto-servicio j y número de vehículos k pertenecientes al conjunto  $P^*$  calcular el número de restricciones,  $r_{jk}$ , sin satisfacer que pasan a ser satisfechas por la inclusión de la estación j con k vehículos en la solución parcial  $S_r$  y  $U_r$ .

Sea  $P^{**}$  el conjunto de los pares (j,k) donde obtiene el máximo  $r_{jk}^*$ .

Si  $|P^{**}| = 1$ , sea  $(j^0, k^0)$  el único par de  $P^{**}$  e ir al paso 6.

Si  $|P^{**}| > 1$ , ir al paso 5.



Paso 5. Para cada punto-servicio  $j$  y número de vehículos  $k$  pertenecientes al conjunto  $P^{**}$ , calcular el cubrimiento adicional al efectivo aportado, por unidad de coste, por la inclusión de la estación  $j$  con  $k$  vehículos en la solución parcial  $S_r$  y  $U_r$ .

$$f_{jk} = \frac{\left( \sum_{i \in F_r} a_{ij} \left| \min(G_i - b_{jk}, 0) \right| \right) + \left( \sum_{i \notin F_r} a_{ij} \right) b_{jk}}{w_{jk}} \quad \text{si } j \notin S_r, \quad k_{j_0} < k \leq k_{j_1}, \quad (3.20)$$

$$f_{jk} = \frac{\left( \sum_{i \in F_r} a_{ij} \left| \min(G_i - b_{jk} + b_{jk_j}, 0) \right| \right) + \left( \sum_{i \notin F_r} a_{ij} \right) (b_{jk} - b_{jk_j})}{w_{jk}} \quad \text{si } j \in S_r, \quad k_j < k \leq k_{j_1}, \quad (3.21)$$

y tomar como  $(j^\circ, k^\circ)$  el par de  $P^{**}$  que maximiza dicho cubrimiento total aportado. En caso de empate, elegir el primer par de  $P^{**}$  que obtenga dicho máximo.

Paso 6. Hacer  $r = r+1$ . Añadir  $(j^\circ, k^\circ)$  al conjunto solución y actualizar:

Si  $j^\circ \notin S_r$ , entonces:

$$k_j^\circ = k^\circ, \quad S_r = S_{r-1} \cup \{j^\circ\}, \quad U_r = U_{r-1} \cup \{k_j^\circ\}.$$

Para cada  $i \in M$ :

$$\text{Actualizar } G_i \leftarrow G_i - a_{ij^\circ} (b_{j^\circ k^\circ}).$$

$$\text{Si } (i \in F_r) \wedge (G_i \leq 0), \text{ entonces } F_r = F_{r-1} - \{i\}.$$

Si  $j^\circ \in S_r$ , entonces:

$$S_r = S_{r-1}, \quad U_r = U_{r-1}.$$

Para cada  $i \in M$ :

$$\text{Actualizar } G_i \leftarrow G_i - a_{ij^\circ} (b_{j^\circ k^\circ} - b_{j^\circ k_{j_0}}).$$

$$\text{Si } (i \in F_r) \wedge (G_i \leq 0), \text{ entonces } F_r = F_{r-1} - \{i\}.$$

$$\text{Actualizar } k_{j^\circ} \leftarrow k^\circ.$$

Si  $F_r = \emptyset$  ir al Paso 7; caso contrario volver al Paso 2. (Fin de la iteración  $r$ ).

Paso 7. Eliminación de los vehículos y estaciones redundantes:

- a) Obtener la estación,  $j$ , y los vehículos ubicados en ella,  $k_j$ , con mayor coste.
- b) Verificar si al eliminar un vehículo se sigue teniendo una solución factible. En tal caso actualizar la configuración obtenida, e ir al apartado a).

En caso contrario, elegir secuencialmente la siguiente estación y los vehículos ubicados con mayor coste e ir al apartado b, hasta recorrer todas las estaciones del conjunto  $S_r$ .

Paso 8. Procedimiento de intercambio 1-óptimo:

Para cada estación,  $j$ , y los vehículos ubicados en ella,  $k_j$ , del conjunto  $S_r$  se intenta sustituir por otra estación y unidades de servicio ubicados en ella con menor coste, de manera que se sigan satisfaciendo todas las restricciones. Si es posible, intercambiar y actualizar la configuración obtenida, repitiendo el procedimiento hasta recorrer completamente el conjunto  $S_r$ .

Observemos que, en este algoritmo, las estaciones con un determinado número de vehículos candidatas a entrar en la solución intermedia son aquellas que aportan un mayor cubrimiento efectivo por unidad de coste de la cantidad de cubrimiento demandada por los puntos-demanda que todavía no ha sido completamente satisfecha, lo cual viene determinado por el Paso 2. Notar que en cualquier iteración se permite incrementar el número de vehículos en una estación previamente elegida.

El Paso 4 establece básicamente una regla de desempate entre las estaciones con un determinado número de vehículos candidatos para ser incorporados al conjunto de soluciones  $S_r$  y  $U_r$ , la de maximizar el número de restricciones satisfechas.

En el caso de que exista un empate entre las candidatas, respecto al número de restricciones no satisfechas que pasan a serlo, la regla elegida es la de maximizar la holgura del cubrimiento, con ello se pretende que se produzca un mayor aporte de cubrimiento añadido a las estaciones que entran a formar parte de  $S_r$ , de manera que al aplicar, en el Paso 7, la eliminación de los vehículos redundantes, la probabilidad de que un punto demanda esté cubierto por la unión de las restantes estaciones y vehículos sea alta.

En el último paso, al intentar sustituir una estación y unos vehículos ubicados en ella por otra estación y unidades de servicio ubicados en ella con menor coste, de manera que se sigan satisfaciendo todas las restricciones, se intenta incorporar posibilidades no contempladas a lo largo del algoritmo y obtener soluciones de mejora calidad.

Al final del procedimiento, el valor de la solución factible del GSCP es

$$CS^* = \sum_{j=1}^n \sum_{k=1}^L w_{jk} X_{jk}, \quad (3.22)$$

tomando  $x_{jk}$  el valor 1 si  $j \in S_r$  y  $k \in U_r$  y 0 en otro caso.

### 3.4.2. Obtención de la cota inferior inicial

La relajación lagrangiana es una de las técnicas más usadas en la resolución de problemas de optimización combinatoria. El procedimiento surge de la observación de que ciertos problemas del tipo NP-Complejos pueden contemplarse como el resultado de un problema relativamente fácil de resolver complicado por la presencia de un conjunto adicional de restricciones, donde el problema fácil es susceptible de ser resuelto en un tiempo de computación acotado por un polinomio función del tamaño del problema. Dualizando las restricciones se obtiene un nuevo problema que se conoce como problema dual lagrangiano,

que es más fácil de resolver que el original y que proporciona una cota para el valor óptimo de la función objetivo original, permitiendo la obtención de una buena solución factible en base a la información obtenida durante el procedimiento de obtención de la cota.

Dada la formulación del GSCP:

$$\begin{aligned} \text{Min } \mathbf{w}^t \mathbf{x} & \quad (3.23) \\ \text{s.a. } \mathbf{ABx} \geq \mathbf{c} & \quad (3.24) \\ \text{(P)} \quad \mathbf{Gx} \leq \mathbf{e} & \quad (3.25) \\ \mathbf{x} \in \{0,1\}^p, & \quad (3.26) \end{aligned}$$

definiremos su relajación lagrangiana ( $L_u$ ) como

$$\begin{aligned} \text{(L}_u\text{)} \quad \text{Min } \mathbf{w}^t \mathbf{x} + \mathbf{u}^t (\mathbf{c} - \mathbf{ABx}) & \quad (3.27) \\ \text{s.a. } \mathbf{Gx} \leq \mathbf{e} & \quad (3.28) \\ \mathbf{x} \in \{0,1\}^p & \quad (3.29) \\ \mathbf{u} \geq \mathbf{0}, & \quad (3.30) \end{aligned}$$

donde  $\mathbf{u}$  es vector real con una componente por cada fila de  $\mathbf{AB}$ . Notemos que ( $L_u$ ) se ha obtenido mediante la relajación de las restricciones  $\mathbf{ABx} \geq \mathbf{c}$ . Estas restricciones han dejado de serlo y se han incorporado a la función objetivo con ciertos pesos  $u_i$ . Por otra parte, su dual lagrangiano, es el problema de optimización en  $\mathbf{u}$ , dado por

$$\text{(D}_L\text{)} \quad \text{Sup } \{v(L_u) : \mathbf{u} \geq \mathbf{0}\} \quad (3.31)$$

donde  $v(\cdot)$  representa el valor óptimo de la función objetivo del problema ( $\cdot$ ). El siguiente teorema demuestra que, efectivamente, ( $L_u$ ) es una relajación de (P) y por tanto ( $D_L$ ) satisface las restricciones de la dualidad débil.

**Teorema 1.** (Dualidad lagrangiana débil)

Para todo  $\mathbf{u} \geq \mathbf{0}$  se tiene que:

$$v(L_u) \leq v(P), \text{ y en consecuencia, } v(D_L) \leq v(P).$$

Demostración: Considérese el siguiente problema ( $S_u$ ):

$$\text{Min } \mathbf{w}^t \mathbf{x} \tag{3.32}$$

$$(S_u) \quad \text{s.a. } \mathbf{u}^t (\mathbf{c} - \mathbf{ABx}) \leq 0 \tag{3.33}$$

$$\mathbf{Gx} \leq \mathbf{e} \tag{3.34}$$

$$\mathbf{x} \in \{0,1\}^p \tag{3.35}$$

$$\mathbf{u} \geq \mathbf{0}. \tag{3.36}$$

Se cumple que  $v(P) = v(S_u)$ , dado que es sencillo comprobar que toda solución factible en P es factible en  $S_u$  y recíprocamente. Por otra parte, dado el problema ( $LS_u$ ):

$$\text{Min } \mathbf{w}^t \mathbf{x} + \mathbf{u}^t (\mathbf{c} - \mathbf{ABx}) \tag{3.37}$$

$$(LS_u) \quad \text{s.a. } \mathbf{u}^t (\mathbf{c} - \mathbf{ABx}) \leq 0 \tag{3.38}$$

$$\mathbf{Gx} \leq \mathbf{e} \tag{3.39}$$

$$\mathbf{x} \in \{0,1\}^p \tag{3.40}$$

$$\mathbf{u} \geq \mathbf{0}, \tag{3.41}$$

se tiene que  $v(S_u) \geq v(LS_u)$ , puesto que los conjuntos de factibles de ambos problemas son iguales y

$$\mathbf{w}^t \mathbf{x} \geq \mathbf{w}^t \mathbf{x} + \mathbf{u}^t (\mathbf{c} - \mathbf{ABx}) \quad \forall (\mathbf{u}, \mathbf{x}) \text{ factible.}$$

Por otra parte, se tiene que  $v(LS_u) \geq v(L_u)$ , por ser ( $L_u$ ) una relajación de ( $LS_u$ ). Teniendo en cuenta todo lo anterior se tiene que:

$$v(P) = v(S_u) \geq v(LS_u) \geq v(L_u) \quad \forall \mathbf{u} \geq \mathbf{0}.$$

Por tanto  $V(P) \geq \text{Sup } \{v(L_u): \mathbf{u} \geq \mathbf{0}\} = v(D_L)$  ■

Para obtener, de forma explícita, una solución óptima de ( $L_u$ ), reescribiremos la función objetivo como

$$\text{Min } (\mathbf{w}^t - \mathbf{u}^t \mathbf{AB})\mathbf{x} + \mathbf{u}^t \mathbf{c} \tag{3.42}$$

o, equivalentemente,

$$\text{Min } \sum_{j=1}^n \sum_{k=1}^L \left( w_{jk} - \left[ \sum_{i=1}^m u_i a_{ij} \right] b_{jk} \right) x_{jk} + \sum_{i=1}^m c_i u_i, \tag{3.43}$$

y dada la estructura particular del conjunto factible podemos enunciar la siguiente proposición que se demuestra por inspección.

**Proposición 1.** Una solución óptima  $x_u^*$  de  $(L_u)$  viene dada por:

$$\text{Si } \text{Min}_{k=1,2,\dots,L} \left\{ w_{jk} - \left[ \sum_{i \in M_j} u_i \right] b_{jk} \right\} \geq 0, \text{ entonces } x_{jk} = 0 \quad k = 1, 2, \dots, L$$

$$\text{Si } \text{Min}_{k=1,2,\dots,L} \left\{ w_{jk} - \left[ \sum_{i \in M_j} u_i \right] b_{jk} \right\} < 0, \text{ entonces}$$

$$x_{jk} = 1 \quad \text{para el menor } k \text{ que tome el valor mínimo calculado.}$$

$$x_{jk} = 0 \quad \text{en otro caso.}$$

Siendo el valor óptimo de  $(L_u)$ :

$$v(L_u) = \sum_{j=1}^n \text{Min} \left( 0, \text{Min}_{k=1,2,\dots,L} \left\{ w_{jk} - \left[ \sum_{i \in M_j} u_i \right] b_{jk} \right\} \right) + \sum_{i=1}^m c_i u_i. \quad (3.44)$$

Dado que nuestro interés radica en obtener una solución inicial del problema  $\text{Sup} \{ v(L_u) : \mathbf{u} \geq \mathbf{0} \}$  y observando las características del valor óptimo de  $(L_u)$  para un vector cualquiera  $\mathbf{u} \geq \mathbf{0}$ , podemos resolver el siguiente problema, basado en encontrar un conjunto de multiplicadores que maximicen el segundo sumando de la expresión con la restricción de que todos los sumandos del primer término sean mayores o iguales a cero:

$$\text{Max } \sum_{i=1}^m c_i u_i \quad (3.45)$$

$$\text{s.a. } \sum_{i \in M_j} u_i \leq \text{Min}_{k=1,2,\dots,L} \left\{ \frac{w_{jk}}{b_{jk}} \right\} = d_{jk^*} = \frac{w_{jk^*}}{b_{jk^*}} \quad j = 1, 2, \dots, n \quad (3.46)$$

$$u_i \geq 0 \quad i = 1, 2, \dots, m, \quad (3.47)$$

siendo  $k^*$  el índice  $k$  donde se toma el mínimo.

Con objeto de conseguir rápidamente una cota inferior para el GSCP y un vector de multiplicadores  $\mathbf{u}$  inicial, el problema definido por (3.45), (3.46) y (3.47) es resuelto con un heurístico greedy de ascenso dual, que mejora la función objetivo aumentando tanto como sea posible la variable  $u_i$  elegida sin violar las restricciones.

**Algoritmo greedy de ascenso dual**

Paso 1. Inicializar  $\mathbf{u}^1 = \mathbf{0}$ .

Paso 2. Obtener  $\Delta_i(\mathbf{u}^1) = \text{Min}_{j \in N_i} \left\{ d_{jk^*} - \sum_{i \in M_j} u_i^1 \right\}$   $i = 1, 2, \dots, m$ .

Paso 3. Sea  $I(\mathbf{u}^1) = \{i \in \{1, 2, \dots, m\} : \Delta_i(\mathbf{u}^1) > 0\}$ . Si  $I(\mathbf{u}^1) = \emptyset$ , STOP.

Paso 4. Elegir el  $i^* \in I(\mathbf{u}^1)$  que maximiza  $c_i \Delta_i(\mathbf{u}^1)$ . En caso de empate, romperlo eligiendo al último  $i$  que obtenga dicho máximo.

Paso 5. Actualizar  $u_{i^*}^1 = u_{i^*}^1 + \Delta_{i^*}(\mathbf{u}^1)$  e ir al paso 2.

Observemos que en este algoritmo la variable  $u_i$  es elegida entre aquellas que permiten un aumento en todas las restricciones que cubre, seleccionando aquella que aporta mayor valor añadido a la función objetivo. Al final de este procedimiento, el valor de la solución factible inicial del problema dual lagrangiano de la relajación lagrangiana del GSCP es:

$$CI^* = \sum_{i=1}^m c_i u_{i^*}^1, \tag{3. 48}$$

dado que, por construcción, se asegura que la expresión  $\text{Min}_{k=1, 2, \dots, L} \left\{ w_{jk} - \left[ \sum_{i \in M_j} u_{i^*}^1 \right] b_{jk} \right\}$  de  $V(L_u)$

sea siempre no negativa, y por tanto, el primer sumando de  $V(L_u)$  toma el valor 0.

### 3.4.3. Procedimiento de mejora de la cota inferior

Cualquier proceso de búsqueda para determinar multiplicadores duales óptimos es complicado debido a la posible presencia de lugares llanos o mesetas. Por otro lado, el valor de la relajación lagrangiana,  $v(L_u)$ , es una función cóncava lineal a trozos de  $\mathbf{u}$ . Esta propiedad permite establecer procedimientos eficientes de búsqueda para localizar multiplicadores lagrangianos óptimos (véase *Fisher* (1981)). En particular, utilizaremos las técnicas de búsqueda subgradiente como estrategia de optimización dual lagrangiana, dado que presenta buenas propiedades teóricas y facilidad de uso, aunque existen otros métodos como el método “bundle”, ascenso generalizado o generación de columnas (véase *Wolsey* (1998)).

A continuación describiremos el algoritmo de búsqueda subgradiente utilizado en el HGSCP.

#### Algoritmo de búsqueda subgradiente

**Paso 0.** Inicialización. Tomar

$\mathbf{u}^1$  = último vector de multiplicadores lagrangianos disponible (en la primera aplicación del procedimiento se utiliza la solución obtenida con el Algoritmo greedy de ascenso dual).

CI\* = mejor cota inferior (en la primera aplicación del procedimiento se utiliza la cota inferior obtenida con el Algoritmo greedy de ascenso dual).

CS\* = mejor cota superior (en la primera aplicación del procedimiento se utiliza la cota superior obtenida con el Algoritmo “greedy 1-óptimo” en la fase a),

$k=1$  y  $\alpha_k = 1$ .



**(Comienzo de la iteración k)**

**Paso 1.** Resolver la relajación lagrangiana ( $L_u^k$ ) y hallar un óptimo  $\mathbf{x}_u^k$ .

**Paso 2.** Almacenar las soluciones de mejora:

Si se cumple que  $CI^* < v(L_u^k)$  guardar la nueva cota mejorada de la solución dual,  $CI^* = v(L_u^k)$ .

Actualizar el tamaño del paso  $\alpha_k$ :

Si durante 20 iteraciones la cota inferior no mejora, hacer  $\alpha_k \leftarrow \alpha_k/2$ .

Criterios de parada:

Si se cumple alguna de las tres condiciones siguientes, ir al paso 5.

- a)  $\alpha_k < 0.005$
- b)  $k > 1000$
- c) Si durante 300 iteraciones  $v(L_u^k)$  no mejora.

**Paso 3.** Paso subgradiente.

Calcular un nuevo punto: 
$$\mathbf{u}^{k+1} = \mathbf{u}^k + t_k \frac{\mathbf{c} - \mathbf{A}\mathbf{B}\mathbf{x}_u^k}{\|\mathbf{c} - \mathbf{A}\mathbf{B}\mathbf{x}_u^k\|}$$

donde  $t_k$  es la amplitud de paso calculada como:

$$t_k = \frac{\alpha_k (CS^* - CI^*)}{\|\mathbf{c} - \mathbf{A}\mathbf{B}\mathbf{x}_u^k\|}.$$

**Paso 4.** Proyección.

Proyectar el nuevo  $\mathbf{u}^{k+1}$  en  $\{\mathbf{u} \geq \mathbf{0}\}$  haciendo  $u_i^{k+1} = \max\{0, u_i^{k+1}\}$   $i = 1, 2, \dots, m$ .

Hacer  $k=k+1$  y volver al paso 1.

**Paso 5.** Stop.

En el algoritmo anterior hay tres puntos que merecen ser analizados. En primer lugar, observemos que en el paso 2 la solución,  $v(L_u^k)$ , se almacena en el caso de producirse una mejora. Ello se debe al hecho de que en las distintas iteraciones subgradientes no se garantiza la mejora de la solución encontrada hasta el momento, por lo que en cada iteración guardaremos ésta únicamente en caso de producirse realmente una mejora.

En segundo lugar, en el Paso 2, se establece una actualización de la amplitud de paso, satisfaciendo la desigualdad  $0 < \alpha_k \leq 2$ . Aunque esta elección de la amplitud de paso no satisface la tercera condición suficiente de *Polyak* (1967) para la convergencia óptima, en la práctica, tiene un comportamiento muy eficiente. Como queda enfatizado en *Sherali and Myers* (1988), uno de los factores que más influye en la convergencia del algoritmo es la elección de la amplitud de paso, especialmente la actualización del parámetro  $\alpha_k$  a lo largo del proceso. *Held et al.* (1970) inicializan  $\alpha_k$  como 2 y lo dividen por la mitad cada vez que en un paso del algoritmo no se produce una mejora de la solución. Otros, como *Beasley* (1987), hacen la división cuando, tras un número determinado de iteraciones, no se produce dicha mejora, criterio utilizado en nuestro heurístico.

En tercer lugar, el procedimiento termina tan pronto como se estima que el proceso converge, por detectarse una tasa de convergencia excesivamente lenta. Notar que si la solución  $\mathbf{x}_u^k$  de la relajación lagrangiana ( $L_u^k$ ) es factible, puesto que es una relajación del problema original, si  $\mathbf{u}^{kt} (\mathbf{c} - \mathbf{AB}\mathbf{x}_u^k) = 0$ , entonces resulta que esta solución debe ser necesariamente óptima (véase el siguiente Teorema de la dualidad lagrangiana fuerte), y, en consecuencia, se tratará de la solución buscada. Un criterio de parada, basado en la obtención del óptimo, podría haberse incorporado en el paso 2, tan sólo verificando el cumplimiento de las condiciones  $(\mathbf{c} - \mathbf{AB}\mathbf{x}_u^k) \geq \mathbf{0}$  y  $\mathbf{u}^{kt} (\mathbf{c} - \mathbf{AB}\mathbf{x}_u^k) = 0$ , pero la experiencia computacional lo ha desaconsejado.

**Teorema 2.** (Dualidad lagrangiana fuerte)

Sea  $\mathbf{x}^*$  una solución óptima de  $(L_{\mathbf{u}^*})$ , para un cierto  $\mathbf{u}^* \geq \mathbf{0}$ , que, además verifica:

$$\begin{aligned} \mathbf{A}\mathbf{B}\mathbf{x}^* &\geq \mathbf{c}, \\ \mathbf{u}^{*t}(\mathbf{c} - \mathbf{A}\mathbf{B}\mathbf{x}^*) &= 0; \end{aligned}$$

entonces  $\mathbf{x}^*$  es una solución óptima de (P).

Demostración: Cualquier óptimo  $\mathbf{x}^*$  cumpliendo las hipótesis del teorema es una solución factible de (P) por la primera condición,  $\mathbf{A}\mathbf{B}\mathbf{x}^* \geq \mathbf{c}$ , y por el hecho de que toda solución de  $(L_{\mathbf{u}^*})$  verifica  $\mathbf{G}\mathbf{x}^* \leq \mathbf{e}$  y  $\mathbf{x}^* \in \{0,1\}^P$ . En particular, esto implica que  $\mathbf{w}^t \mathbf{x}^* \geq v(P)$ . Pero, por la desigualdad débil:  $v(P) \geq v(L_{\mathbf{u}^*}) = \mathbf{w}^t \mathbf{x}^* + \mathbf{u}^{*t}(\mathbf{c} - \mathbf{A}\mathbf{B}\mathbf{x}^*)$  y junto a la segunda condición  $\mathbf{u}^{*t}(\mathbf{c} - \mathbf{A}\mathbf{B}\mathbf{x}^*) = 0$  podemos concluir que  $v(P) \geq v(L_{\mathbf{u}^*}) = \mathbf{w}^t \mathbf{x}^*$ . Así, puesto que  $\mathbf{x}^*$  es factible para (P), se tiene que  $\mathbf{w}^t \mathbf{x}^* \geq v(P)$ , por lo que finalmente, se da la igualdad  $v(P) = \mathbf{w}^t \mathbf{x}^*$ , y resulta que  $\mathbf{x}^*$  es una solución óptima para (P). ■

**3.4.4. Procedimiento de mejora de la cota superior**

Una vez conocida la solución  $\mathbf{x}_u^k$  de la relajación lagrangiana  $(L_{\mathbf{u}^k})$ , siendo  $\mathbf{u}^k$ , el vector de multiplicadores de Lagrange obtenido en el procedimiento subgradiente, se procede a encontrar una solución factible de (P) recurriendo a la utilización del algoritmo “greedy 1-óptimo” usado para la obtención de la cota superior inicial tomando como solución inicial la solución óptima  $\mathbf{x}_u^k$  del  $(L_{\mathbf{u}^k})$ . Obtenida con el procedimiento greedy la solución  $\mathbf{x}^k$ , se comprueba si se cumple que  $CS^* > \mathbf{w}^t \mathbf{x}^k$ :

en caso afirmativo, se guarda la nueva cota mejorada de la solución primal,  $CS^* = \mathbf{w}^t \mathbf{x}^k$ ,

en otro caso, se verifican los criterios de parada.

### 3.4.5. Criterios de parada

Finalmente, se determinan unos criterios de parada que tienen como objetivo determinar el momento en que el algoritmo se detiene. En concreto, dichas reglas son:

- Si el número de veces que se repite la fase c de mejora de la cota de inferior y la fase del d de mejora de la cota superior es mayor que 150.
- Si durante 10 iteraciones consecutivas donde se repite alternativamente la fase c y d, la diferencia entre la cota superior e inferior de la última iteración no ha conseguido disminuir al menos un 10% respecto de la diferencia entre  $CS^*$  y  $CI^*$  de la primera iteración considerada. Es decir, si la diferencia de los valores en la iteración  $k$ , respecto a la iteración  $k+10$  verifica:

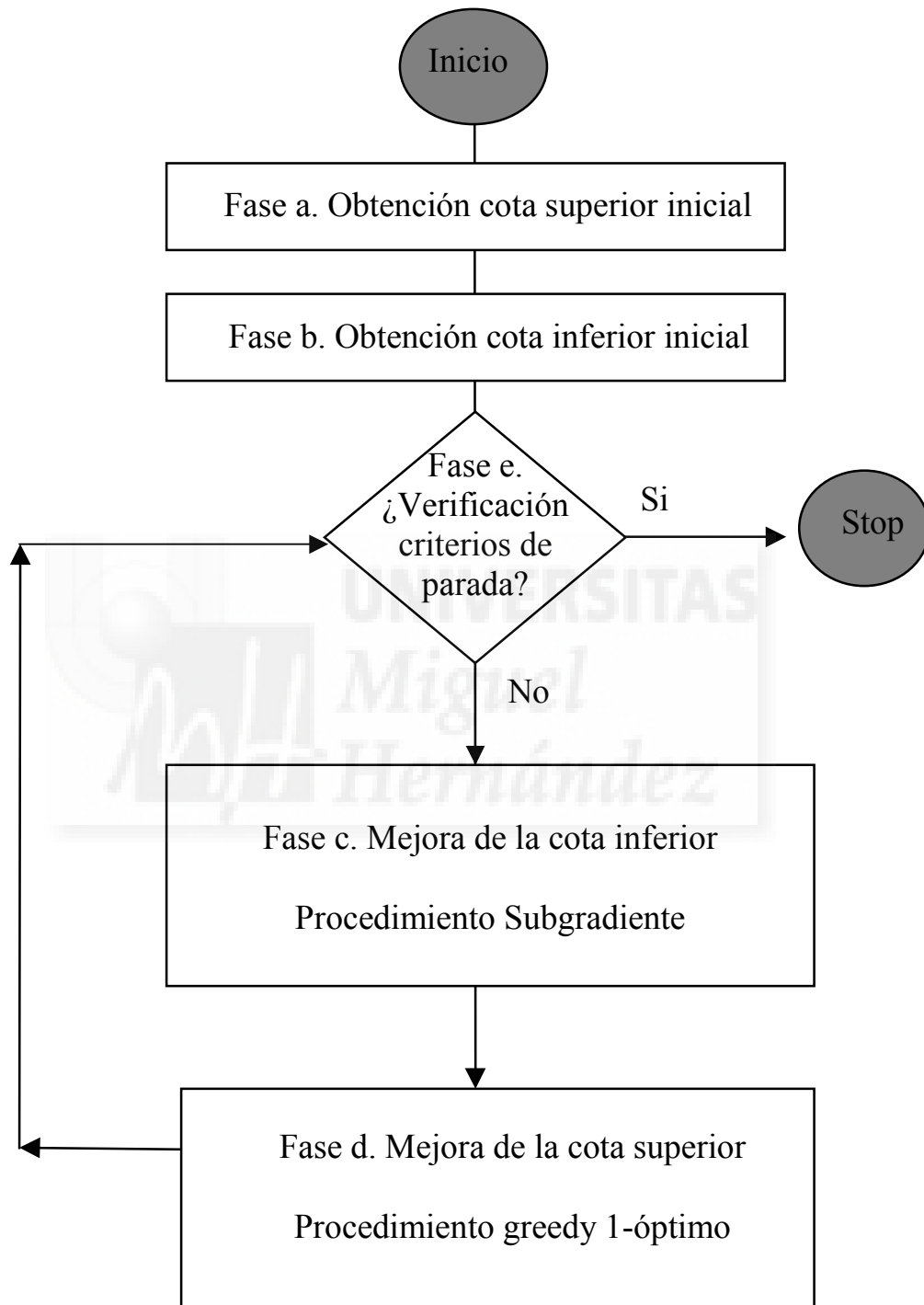
$$\frac{CS_k^* - CI_k^*}{CS_{k+10}^* - CI_{k+10}^*} - 1 < 0.1,$$

el procedimiento se detendrá, por detectarse una tasa de convergencia excesivamente lenta. Este criterio permite detener el proceso sin agotar el total de iteraciones permitidas, al comparar si la diferencia entre la cota superior e inferior tras un número de iteraciones ha mejorado al menos un 10% respecto a la diferencia existente 10 iteraciones atrás.

La elección de estos criterios ha sido guiada por la experiencia computacional para equilibrar la precisión del heurístico y el tiempo necesario para su finalización. Obviamente, si aumentamos el número de iteraciones consecutivas necesarias para realizar la comparación o el grado de mejora a obtener, el tiempo de ejecución del heurístico aumentará.

El diagrama de flujo del algoritmo HGSCP se presenta a continuación:

## Diagrama de Flujo del Heurístico HGSCP



### 3.5. Resultados computacionales

Con objeto de realizar una experiencia computacional lo más real posible hemos decidido resolver los problemas con los costes  $w_{jk}=k$ , para reflejar la situación más plausible de reubicación de la flota existente entre las diferentes posibles ubicaciones de los estacionamientos de los vehículos de emergencia. Esta situación es mucho más compleja que la de resolver el problema conjunto de ubicar las estaciones de emergencias y los vehículos ubicados en ella, ya que los costes de las ubicaciones facilitan la obtención de la solución. Es decir, se trata de resolver la versión probabilística del LSCP, dado que el avance tecnológico permite tener ubicada una unidad de emergencia casi en cualquier lugar, gracias a las conexiones telefónicas inalámbricas.

Para evaluar la eficiencia del heurístico planteado, garantizando la obtención de la solución óptima, se ha realizado una batería de problemas test basados en la red de 79 nodos de Serra(1989) presentada en el apéndice A, considerando 4 configuraciones de llamadas por día y dos distancias máximas  $S$ , 1.5 y 3 kilómetros. El tiempo medio de servicio de cada llamada,  $\bar{t}$ , ha sido establecido en 45 minutos. Para cada sistema particular (red, llamadas por día para cada nodo, y distancia máxima) se establecen 3 niveles de fiabilidad requeridos. Los niveles prefijados han sido  $\alpha=0.80, 0.90$  y  $0.99$ . De los distintos modelos de cubrimiento total probabilístico presentados en el capítulo anterior hemos utilizado 3 formulaciones basadas en cubrimiento múltiple, como el BPLSCP, QPLSCP y PPLSCP, y 3 modelizaciones basadas en cantidad de cubrimiento, como el PRLSCP, QRLSCP y BRLSCP.

Adicionalmente, para evaluar la rapidez del heurístico hemos utilizado dos grandes matrices de adyacencia: una de 511 puntos demanda por 210 posibles ubicaciones con una densidad del 12.32% y otra de 1023 puntos demanda por 330 posibles ubicaciones con una densidad de 8.96%, resolviéndose los mismos modelos que sobre la red de 79 nodos.

El Algoritmo HGSCP ha sido implementado en lenguaje C y ejecutado sobre una estación de trabajo Sun Enterprise 3000 con 3 procesadores UltraSparc a 250MHz, bus de 2GB y 16 Gigas de disco en RAID. La matriz de cubrimiento ha sido eficientemente almacenada, para evitar no disponer de suficiente memoria, construyendo vectores de adyacencias. La

representación de una matriz de cubrimiento, mediante los vectores de adyacencia, se realiza del siguiente modo:

- Para las filas de la matriz consideraremos dos vectores; uno que contiene, en orden secuencial, las columnas que cubren a cada una de las filas, y otro que contiene la información relativa al índice que marca el inicio, en el vector anterior, de las columnas que cubren a cada fila. Este último vector se completa con una posición que indica que el vector primero ha sido completado justo antes de llegar a ella. Estos dos vectores serán de órdenes  $s$  y  $n+1$  respectivamente, siendo  $s$  el número de elementos no nulos de la matriz de cubrimiento.
- Para las columnas, se definirán dos vectores de órdenes  $s$  y  $m+1$  de idéntica estructura que los anteriores, pero conteniendo el primero de ellos la información referida a las filas cubiertas por cada columna.

Para obtener la solución óptima de los problemas generados hemos usado el paquete de programación lineal y entera CPLEX 4.0. Para acelerar el proceso de acotación y ramificación se ha utilizado como solución factible inicial la obtenida mediante el heurístico HGSCP. En la etapa de preproceso se ha obligado a que se redujesen todos los posibles coeficientes de las restricciones para obtener una formulación más fuerte obteniendo un valor de la relajación lineal más cercano al óptimo entero y que se generase un conjunto de desigualdades de cortes de cubrimiento (covers) para ser añadidas al inicio a la formulación original. Respecto a la estrategia de selección de nodos se ha utilizado la búsqueda de la mejor cota y en cuanto a la selección de variables para la ramificación se ha permitido al código que elija en cada momento la mejor opción dependiendo del problema y su progreso. Finalmente, se ha establecido un tope de 18000 segundos de CPU para trabajar en el proceso de ramificación y acotación, que no permite garantizar la optimalidad de la solución, a no ser que el proceso se detenga antes de la cota establecida, debido a que el árbol de ramificación haya sido completamente explorado.

Dicho tope se ha establecido ya que la primera instancia para la matriz de adyacencias de 511 por 210 que se intentó resolver no había conseguido terminar el proceso de Branch & Bound, tras 4 días de computación.

Los resultados obtenidos sobre la batería de problemas test sobre la red de 79 nodos se presentan en las tablas siguientes, donde las columnas recogen los siguientes datos:

PROB	Nombre del Problema resuelto
MODELO	Diferentes formulaciones de modelos de cubrimiento total probabilístico
FIAB	Nivel de fiabilidad
ZUBI	Cota superior inicial
ZLBI	Cota inferior inicial
ZLB	Cota inferior final
ZUB	Cota superior final
NIS	Número de Iteraciones Simplex realizadas por el CPLEX 4.0
NNE	Número de Nodos Explorados en el proceso de ramificación y cota
NCOVERS	Número de desigualdades de cubrimiento añadidos en el preproceso
ZOPT	Solución óptima entera
TMPG	Tiempo en segundos de CPU utilizado para obtener ZLBI
TMPH	Tiempo en segundos de CPU utilizado por HGSCP
TMPC	Tiempo en segundos de CPU utilizado por el CPLEX 4.0.



**Tabla 3.1. Resultados computacionales sobre el escenario de llamadas por día, red 79.1, distancia máxima S=1,5 y 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR1	BPLSCP	0.80	12	10,00	10,90	11	141	0	0	11	0	45	0
PR2	BPLSCP	0.90	16	12,00	12,06	14	422	18	35	13	0	14	1
PR3	BPLSCP	0.99	20	15,00	16,99	18	264	11	32	17	0	69	0
PR4	QPLSCP	0.80	12	8,00	10,00	11	221	0	0	11	0	80	0
PR5	QPLSCP	0.90	16	12,00	12,06	14	422	18	35	13	0	14	0
PR6	QPLSCP	0.99	21	15,00	18,10	20	2846	62	120	19	0	43	1
PR7	PPLSCP	0.80	12	10,00	10,90	11	141	0	0	11	0	45	0
PR8	PPLSCP	0.90	16	12,00	12,06	14	422	18	35	13	0	14	0
PR9	PPLSCP	0.99	21	15,00	18,10	20	2846	62	120	19	0	42	2
PR10	PRLSCP	0.80	16	4,28	5,62	11	862	18	85	10	0	17	1
PR11	PRLSCP	0.90	16	6,12	8,12	14	1692	34	129	13	1	29	1
PR12	PRLSCP	0.99	25	10,89	14,38	22	38847	711	404	19	0	26	36
PR13	QRLSCP	0.80	14	4,19	5,50	12	1485	35	61	10	0	25	1
PR14	QRLSCP	0.90	15	5,99	7,90	14	2501	72	159	13	0	20	2
PR15	QRLSCP	0.99	24	10,79	14,24	21	76407	1447	462	19	0	33	67
PR16	BRLSCP	0.80	15	3,74	4,87	14	834	56	69	11	0	19	0
PR17	BRLSCP	0.90	16	5,35	7,07	14	2383	38	121	13	0	20	1
PR18	BRLSCP	0.99	24	9,00	11,81	20	17584	362	168	18	0	33	11

**Tabla 3.2. Resultados computacionales sobre el escenario de llamadas por día, red 79.2, distancia máxima S=1,5 y 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR19	BPLSCP	0.80	16	12,00	12,07	14	397	14	43	13	0	13	0
PR20	BPLSCP	0.90	16	11,00	12,40	13	143	0	0	13	1	46	0
PR21	BPLSCP	0.99	24	5,22	6,66	24	554	36	34	19	0	17	1
PR22	QPLSCP	0.80	15	12,00	12,00	15	263	4	37	12	0	10	0
PR23	QPLSCP	0.90	17	10,00	13,00	14	214	0	0	14	0	38	0
PR24	QPLSCP	0.99	23	17,00	19,84	22	27813	526	429	21	0	68	33
PR25	PPLSCP	0.80	16	12,00	12,07	14	397	14	43	13	0	14	0
PR26	PPLSCP	0.90	17	10,00	13,22	15	3266	128	92	14	0	43	2
PR26	PPLSCP	0.99	23	17,00	19,84	22	27813	526	429	21	0	68	32
PR28	PRLSCP	0.80	17	4,51	7,44	15	2656	178	144	13	0	17	2
PR29	PRLSCP	0.90	18	5,57	9,18	15	2319	36	81	14	0	31	2
PR30	PRLSCP	0.99	31	10,04	16,51	26	8075	377	100	22	0	27	6
PR31	QRLSCP	0.80	16	4,29	7,08	14	2340	58	124	13	0	25	2
PR32	QRLSCP	0.90	17	5,45	8,93	15	2978	79	128	14	0	25	1
PR33	QRLSCP	0.99	28	9,91	16,26	26	727404	21482	462	22	0	24	1095
PR34	BRLSCP	0.80	17	3,92	6,44	15	4807	99	160	13	0	16	2
PR35	BRLSCP	0.90	18	4,47	7,26	16	632	27	37	14	0	23	1
PR36	BRLSCP	0.99	27	7,82	12,69	25	1164	31	70	20	0	26	1

**Tabla 3.3. Resultados computacionales sobre el escenario de llamadas por día, red 79.3, distancia máxima S=1,5 y 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR37	BPLSCP	0.80	18	13,00	16,41	18	1101	15	54	17	0	86	1
PR38	BPLSCP	0.90	24	17,00	19,52	22	25916	738	326	20	0	41	29
PR39	BPLSCP	0.99	34	7,10	9,61	34	6418	154	106	27	0	35	5
PR40	QPLSCP	0.80	18	12,00	15,84	18	4626	119	80	17	0	59	2
PR41	QPLSCP	0.90	26	18,00	20,63	22	899	23	63	21	0	67	1
PR42	QPLSCP	0.99	36	26,00	28,96	32	20340	641	353	30	0	172	25
PR43	PPLSCP	0.80	22	13,00	18,09	20	53101	1240	448	19	0	80	51
PR44	PPLSCP	0.90	26	16,00	21,20	23	78948	1856	462	22	0	64	108
PR45	PPLSCP	0.99	36	26,00	28,96	32	20340	641	353	30	0	171	26
PR46	PRLSCP	0.80	26	8,28	11,46	21	1707	33	91	20	0	40	1
PR47	PRLSCP	0.90	32	10,41	14,28	31	582	12	66	23	0	28	0
PR48	PRLSCP	0.99	49	17,23	23,60	37	2177	58	58	35	0	40	2
PR49	QRLSCP	0.80	20	7,72	10,54	19	397900	12327	462	18	0	31	532
PR50	QRLSCP	0.90	30	11,04	15,33	24	790810	16033	462	21	0	28	918
PR51	QRLSCP	0.99	41	16,90	23,33	37	3974	100	104	33	0	56	5
PR52	BRLSCP	0.80	24	5,83	8,00	20	542	16	17	18	0	26	0
PR53	BRLSCP	0.90	32	8,34	11,60	26	1728	66	61	21	0	28	1
PR54	BRLSCP	0.99	41	14,44	19,76	36	827	21	66	28	0	43	1

**Tabla 3.4. Resultados computacionales sobre el escenario de llamadas por día, red 79.4, distancia máxima S=1,5 y 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR55	BPLSCP	0.80	28	22,00	22,01	28	1375	24	72	23	0	17	1
PR56	BPLSCP	0.90	29	11,98	15,67	29	31823	918	378	26	1	33	39
PR57	BPLSCP	0.99	41	13,12	18,24	36	40304	1131	462	33	1	53	68
PR58	QPLSCP	0.80	25	16,00	20,79	23	35887	851	443	22	0	84	38
PR59	QPLSCP	0.90	33	23,00	25,92	30	93639	2888	462	27	1	81	158
PR60	QPLSCP	0.99	47	30,00	38,96	44	1083845	45648	462	40	0	122	2309
PR61	PPLSCP	0.80	29	23,00	23,46	28	42995	939	272	25	0	51	40
PR62	PPLSCP	0.90	35	27,00	27,02	35	77841	2912	462	28	0	18	91
PR63	PPLSCP	0.99	45	30,00	38,65	45	9229907	413003	462	41	0	113	18069
PR64	PRLSCP	0.80	36	12,23	14,76	33	2281	60	83	28	1	32	3
PR65	PRLSCP	0.90	45	15,63	18,85	35	1538	56	41	33	0	40	1
PR66	PRLSCP	0.99	62	26,28	31,45	61	4600	124	65	47	0	45	6
PR67	QRLSCP	0.80	33	12,68	15,31	26	820077	17022	462	23	1	35	1074
PR68	QRLSCP	0.90	46	16,37	19,76	36	301901	8097	462	30	0	41	511
PR69	QRLSCP	0.99	57	25,68	30,80	50	9253	259	243	45	0	70	14
PR70	BRLSCP	0.80	35	9,19	11,05	28	577	14	40	25	0	28	0
PR71	BRLSCP	0.90	43	11,12	13,25	35	1021	45	80	29	1	34	0
PR72	BRLSCP	0.99	58	19,71	23,41	47	501	28	15	37	1	34	1

**Tabla 3.5. Resultados computacionales sobre el escenario de llamadas por día, red 79.1, distancia máxima S=3 y 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR73	BPLSCP	0.80	6	5,00	5,00	6	313	0	0	6	0	13	0
PR74	BPLSCP	0.90	7	5,00	5,08	7	41	0	0	7	0	13	1
PR75	BPLSCP	0.99	10	7,00	7,07	10	51	0	0	10	0	16	0
PR76	QPLSCP	0.80	6	3,00	5,01	6	43	0	0	6	0	72	0
PR77	QPLSCP	0.90	7	5,00	5,07	7	31	0	0	7	0	13	0
PR78	QPLSCP	0.99	10	8,00	8,03	10	27	0	0	10	0	17	1
PR79	PPLSCP	0.80	7	5,00	5,00	6	34	0	0	6	0	12	1
PR80	PPLSCP	0.90	7	5,00	5,07	7	367	0	0	7	0	13	0
PR81	PPLSCP	0.99	10	8,00	8,03	10	27	0	0	10	0	17	0
PR82	PRLSCP	0.80	9	2,09	3,51	6	191	4	90	6	0	41	1
PR83	PRLSCP	0.90	9	2,99	5,03	8	691	8	84	7	0	34	1
PR84	PRLSCP	0.99	17	4,45	8,12	11	46	0	30	11	0	51	1
PR85	QRLSCP	0.80	7	1,90	3,36	6	500	30	101	6	0	37	1
PR86	QRLSCP	0.90	10	2,72	4,79	8	681	5	123	7	0	33	1
PR87	QRLSCP	0.99	14	4,31	7,99	11	137	18	6	10	1	53	1
PR88	BRLSCP	0.80	6	1,59	1,96	6	32	0	0	6	0	13	0
PR89	BRLSCP	0.90	10	2,27	3,86	7	45	0	0	7	0	31	0
PR90	BRLSCP	0.99	13	3,58	6,75	10	80	0	44	10	0	42	1

**Tabla 3.6. Resultados computacionales sobre el escenario de llamadas por día, red 79.2, distancia máxima S=3 y 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR91	BPLSCP	0.80	7	4,00	4,13	7	223	0	0	7	0	15	0
PR92	BPLSCP	0.90	8	6,00	6,01	8	602	0	0	8	0	15	1
PR93	BPLSCP	0.99	11	3,07	4,76	11	81	0	0	11	0	46	1
PR94	QPLSCP	0.80	7	5,00	5,04	7	49	0	0	7	0	13	1
PR95	QPLSCP	0.90	8	6,00	6,00	8	111	0	0	8	0	14	1
PR96	QPLSCP	0.99	12	9,00	9,00	12	64	0	0	12	0	18	1
PR97	PPLSCP	0.80	7	4,00	6,03	7	542	0	0	7	0	111	1
PR98	PPLSCP	0.90	8	6,00	6,00	8	91	0	0	8	0	14	0
PR99	PPLSCP	0.99	12	9,00	9,00	12	64	0	0	12	0	18	1
PR100	PRLSCP	0.80	7	2,91	4,32	7	250	0	0	7	0	50	0
PR101	PRLSCP	0.90	12	3,51	5,49	10	141	8	63	9	0	57	1
PR102	PRLSCP	0.99	19	5,48	9,25	13	65	0	0	13	0	74	0
PR103	QRLSCP	0.80	9	2,89	4,62	7	487	8	159	7	0	39	1
PR104	QRLSCP	0.90	10	3,63	5,72	9	1585	65	76	8	0	32	2
PR105	QRLSCP	0.99	18	5,80	9,80	14	950	18	250	13	0	68	3
PR106	BRLSCP	0.80	7	2,00	2,41	7	65	0	0	7	0	15	1
PR107	BRLSCP	0.90	10	2,86	4,42	10	92	6	48	8	0	36	0
PR108	BRLSCP	0.99	19	4,54	7,56	14	326	5	70	11	0	36	1

**Tabla 3.7. Resultados computacionales sobre el escenario de llamadas por día, red 79.3, distancia máxima S=3 y 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR109	BPLSCP	0.80	13	4,15	5,75	13	2937	53	90	12	0	48	4
PR110	BPLSCP	0.90	13	4,33	4,33	13	159	0	0	13	0	23	1
PR111	BPLSCP	0.99	16	4,67	5,17	16	902	0	0	16	0	25	1
PR112	QPLSCP	0.80	10	7,00	7,00	10	1060	0	0	10	0	21	1
PR113	QPLSCP	0.90	13	8,00	8,37	13	333	0	0	13	0	25	1
PR114	QPLSCP	0.99	20	12,00	12,50	20	77	0	0	20	0	34	2
PR115	PPLSCP	0.80	13	8,00	8,17	13	171	0	0	13	0	24	1
PR116	PPLSCP	0.90	14	9,00	9,00	14	2171	28	34	13	0	25	3
PR117	PPLSCP	0.99	20	12,00	12,16	20	92	0	0	20	0	32	1
PR118	PRLSCP	0.80	18	5,02	6,81	14	92	2	71	14	0	71	1
PR119	PRLSCP	0.90	25	6,51	8,93	16	72	0	0	15	1	57	1
PR120	PRLSCP	0.99	30	9,11	14,22	24	156	55	72	22	1	67	2
PR121	QRLSCP	0.80	16	4,76	7,08	11	225	0	0	11	0	57	1
PR122	QRLSCP	0.90	22	6,17	9,43	14	90	0	0	14	1	88	1
PR123	QRLSCP	0.99	31	9,87	13,76	22	109	0	56	22	1	94	3
PR124	BRLSCP	0.80	21	3,63	3,63	21	144	6	68	12	0	22	0
PR125	BRLSCP	0.90	26	4,02	5,55	14	48	0	0	13	0	63	0
PR126	BRLSCP	0.99	24	7,55	10,70	18	83	0	0	18	0	90	1

**Tabla 3.8. Resultados computacionales sobre el escenario de llamadas por día, red 79.4, distancia máxima S=3 y 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR127	BPLSCP	0.80	28	22,00	22,01	28	1375	24	72	23	0	17	1
PR128	BPLSCP	0.90	29	11,98	15,67	29	31823	918	378	26	1	33	39
PR129	BPLSCP	0.99	41	13,12	18,24	36	40304	1131	462	33	1	53	68
PR130	QPLSCP	0.80	25	16,00	20,79	23	35887	851	443	22	0	84	38
PR131	QPLSCP	0.90	33	23,00	25,92	30	93639	2888	462	27	1	81	158
PR132	QPLSCP	0.99	47	30,00	38,96	44	1083845	45648	462	40	0	122	2309
PR133	PPLSCP	0.80	29	23,00	23,46	28	42995	939	272	25	0	51	40
PR134	PPLSCP	0.90	35	27,00	27,02	35	77841	2912	462	28	0	18	91
PR135	PPLSCP	0.99	45	30,00	38,65	45	9229907	413003	462	41	0	113	18069
PR136	PRLSCP	0.80	36	12,23	14,76	33	2281	60	83	28	1	32	3
PR137	PRLSCP	0.90	45	15,63	18,85	35	1538	56	41	33	0	40	1
PR66	PRLSCP	0.99	62	26,28	31,45	61	4600	124	65	47	0	45	6
PR67	QRLSCP	0.80	33	12,68	15,31	26	820077	17022	462	23	1	35	1074
PR68	QRLSCP	0.90	46	16,37	19,76	36	301901	8097	462	30	0	41	511
PR69	QRLSCP	0.99	57	25,68	30,80	50	9253	259	243	45	0	70	14
PR70	BRLSCP	0.80	35	9,19	11,05	28	577	14	40	25	0	28	0
PR71	BRLSCP	0.90	43	11,12	13,25	35	1021	45	80	29	1	34	0
PR72	BRLSCP	0.99	58	19,71	23,41	47	501	28	15	37	1	34	1

Con objeto de evaluar la eficiencia del heurístico “greedy 1-óptimo” y del HGSCP se ha obtenido la desviación porcentual media respecto de la solución óptima y se ha agrupado para los diferentes modelos. La primera columna muestra la desviación porcentual promedio entre la solución obtenida por el “greedy 1-óptimo” y la solución óptima dada por CPLEX 4.0, la segunda columna presenta la desviación entre la solución calculada por el HGSCP presentado y la solución óptima, y por último, la tercera columna valora la mejora obtenida por la utilización de la relajación lagrangiana respecto al “greedy 1-óptimo”.

**Tabla 3.9. Desviaciones porcentuales medias de las soluciones obtenidas por los heurísticos para la batería de problemas test basados en la red 79 para formulaciones basadas en cubrimiento múltiple**

MODELO	Nº de instancias	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZOPI_i}{ZOPI_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBF_i - ZOPI_i}{ZOPI_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZUBF_i}{ZUBF_i} 100$
BPLSCP	24	10,00	5,84	3,95
QPLSCP	24	8,64	3,80	4,61
PPLSCP	24	9,41	4,31	4,85
TOTAL	72	9,35	4,65	4,47

**Tabla 3.10. Desviaciones porcentuales medias de las soluciones obtenidas por los heurísticos para la batería de problemas test basados en la red 79 para formulaciones basadas en cantidad de cubrimiento**

MODELO	Nº de instancias	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZOPI_i}{ZOPI_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBF_i - ZOPI_i}{ZOPI_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZUBF_i}{ZUBF_i} 100$
PRLSCP	24	37,21	11,27	24,26
QRLSCP	24	32,61	8,80	22,15
BRLSCP	24	39,83	17,91	19,43
TOTAL	72	38,52	14,59	21,85

La observación de las tablas anteriores permite extraer la clara conclusión de que el procedimiento utilizado se comporta razonablemente bien para las formulaciones basadas en cubrimiento múltiple y mal para las formulaciones basadas en cantidad de cubrimiento. Este hecho obliga a modificar el heurístico “greedy 1-óptimo” utilizado en la fase a del HGSCP, que es posteriormente implementado en la fase d de mejora de la cota superior, para modelizaciones basadas en cantidad de cubrimiento. Las modificaciones necesarias se describen a continuación:

**Algoritmo “greedy 1-óptimo modificado”**

Paso 1. Inicialización: hacer  $M = \{1, 2, \dots, m\}$ ,  $V = \{1, 2, \dots, v\}$ ,  $N = \{1, 2, \dots, n\}$ ,  $F_1 = M$ ,  $S_1 = \emptyset$ ,  $U_1 = \emptyset$ ,  $I = \{0, 0, \dots, 0\}$ ,  $S = \{v, v, \dots, v\}$ ,  $G_i = c_i$  y  $r = 1$ .

Paso 2. (Comienzo de la iteración  $r$ ).

Para cada punto-servicio  $j$  y número de vehículos  $k$ , calcular el número de restricciones,  $r_{jk}$ , sin satisfacer que pasan a ser satisfechas por la inclusión de la estación  $j$  con  $k$  vehículos en la solución parcial  $S_r$  y  $U_r$ .

Sea  $P^*$  el conjunto de los pares  $(j, k)$  donde obtiene el máximo  $r_{jk}^*$ .

Si  $|P^*| = 1$ , sea  $(j^\circ, k^\circ)$  el único par de  $P^*$  e ir al paso 4.

Si  $|P^*| > 1$ , ir al paso 3.

Paso 3. Para cada punto-servicio  $j$  y número de vehículos  $k$  pertenecientes al conjunto  $P^*$ , calcular el cubrimiento adicional al efectivo aportado, por unidad de coste, por la inclusión de la estación  $j$  con  $k$  vehículos en la solución parcial  $S_r$  y  $U_r$ .

$$f_{jk} = \frac{\left( \sum_{i \in F_r} a_{ij} \left| \min(G_i - b_{jk}, 0) \right| \right) + \left( \sum_{i \in F_r} a_{ij} \right) b_{jk}}{w_{jk}} \quad \text{si } j \notin S_r, \quad k_{j0} < k \leq k_{j1},$$

$$f_{jk} = \frac{\left( \sum_{i \in F_r} a_{ij} \left| \min(G_i - b_{jk} + b_{jk_j}, 0) \right| \right) + \left( \sum_{i \in F_r} a_{ij} \right) (b_{jk} - b_{jk_j})}{w_{jk}} \quad \text{si } j \in S_r, \quad k_j < k \leq k_{j1},$$

y tomar como  $(j^\circ, k^\circ)$  el par de  $P^*$  que maximiza dicho cubrimiento total aportado. En caso de empate, elegir el primer par de  $P^*$  que obtenga dicho máximo.

Paso 4. Hacer  $r = r+1$ . Añadir  $(j^\circ, k^\circ)$  al conjunto solución y actualizar:

Si  $j^\circ \notin S_r$ , entonces:

$$k_j^\circ = k^\circ, \quad S_r = S_{r-1} \cup \{j^\circ\}, \quad U_r = U_{r-1} \cup \{k_j^\circ\}.$$

Para cada  $i \in M$ :

$$\text{Actualizar } G_i \leftarrow G_i - a_{ij}^\circ (b_j^\circ - k_j^\circ).$$

$$\text{Si } (i \in F_r) \wedge (G_i \leq 0), \text{ entonces } F_r = F_{r-1} - \{i\}.$$

Si  $j^\circ \in S_r$ , entonces:

$$S_r = S_{r-1}, \quad U_r = U_{r-1}.$$

Para cada  $i \in M$ :

$$\text{Actualizar } G_i \leftarrow G_i - a_{ij}^\circ (b_j^\circ - b_j^\circ)$$

$$\text{Si } (i \in F_r) \wedge (G_i \leq 0), \text{ entonces } F_r = F_{r-1} - \{i\}.$$

$$\text{Actualizar } k_j^\circ \leftarrow k^\circ.$$

Si  $F_r = \emptyset$  ir al Paso 5; caso contrario volver al Paso 2. (Fin de la iteración  $r$ )

Paso 5. Eliminación de los vehículos y estaciones redundantes:

- a) Obtener la estación,  $j$ , y los vehículos ubicados en ella,  $k_j$ , con mayor coste.
- b) Verificar si al eliminar un vehículo se sigue teniendo una solución factible. En tal caso actualizar la configuración obtenida, e ir al apartado a).

En caso contrario, elegir secuencialmente la siguiente estación y los vehículos ubicados con mayor coste e ir al apartado b, hasta recorrer todas las estaciones del conjunto  $S_r$ .

Paso 6. Procedimiento de intercambio 1-óptimo:

Para cada estación,  $j$ , y los vehículos ubicados en ella,  $k_j$ , del conjunto  $S_r$  se intenta sustituir por otra estación y unidades de servicio ubicados en ella con menor coste, de manera que se sigan satisfaciendo todas las restricciones. Si es posible, intercambiar y actualizar la configuración obtenida, y continuar repitiendo el procedimiento recorriendo hasta el final el conjunto  $S_r$ .

Notemos que las modificaciones introducidas en este algoritmo, respecto al algoritmo “greedy 1-óptimo” son la eliminación de la segunda y tercera etapa, y los cambios necesarios para un correcto desarrollo del procedimiento.

Con objeto de evaluar la eficiencia del heurístico “greedy 1-óptimo modificado” y del HGSCP modificado se ha obtenido la desviación porcentual media respecto de la solución óptima para la batería de problemas test basados en la red 79 nodos para las formulaciones basadas en cantidad de cubrimiento, obteniéndose la siguiente tabla.

**Tabla 3.11. Desviaciones porcentuales medias de las soluciones obtenidas por los heurísticos para la batería de problemas test basados en la red 79 para formulaciones basadas en cantidad de cubrimiento**

MODELO	Nº de instancias	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZOPI_i}{ZOPI_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBF_i - ZOPI_i}{ZOPI_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZUBF_i}{ZUBF_i} 100$
PRLSCP	24	18,60	2,81	15,32
QRLSCP	24	16,69	2,22	14,12
BRLSCP	24	17,96	3,35	14,35
TOTAL	72	17,75	2,79	14,60

Los resultados obtenidos por la modificación propuesta han significado aproximadamente un 50% de mejora respecto a la anterior desviación porcentual promedio



del óptimo de los valores obtenidos por el “greedy 1-óptimo” y un 80% de mejora sobre los calculados por el HGSCP.

Veamos ahora el efecto, sobre diferentes configuraciones, de los criterios de parada de la fase e) del heurístico HGSCP, considerando para cada grupo de modelos la versión del heurístico greedy que mejor resultados obtiene. Los resultados alcanzados se presentan en la siguiente tabla:

**Tabla 3.12. Desviaciones porcentuales medias respecto del óptimo y tiempos de computación de las soluciones obtenidas por los heurísticos HGSCP para la batería de problemas test basados en la red 79 para diferentes configuraciones de criterios de parada en la fase e).**

MODELO	N° de instancias	Fase e con límite de 150 iteraciones y 10 iteraciones consecutivas con un porcentaje de mejora 10%.		Fase e con límite de 150 iteraciones y 20 iteraciones consecutivas con un porcentaje de mejora 5%.		Fase e con límite de 150 iteraciones y 20 iteraciones consecutivas con un porcentaje de mejora 1%.	
		Desviación promedio del óptimo.	Tiempo de computación promedio (segundos).	Desviación promedio del óptimo.	Tiempo de computación promedio (segundos).	Desviación promedio del óptimo.	Tiempo de computación promedio (segundos).
BPLSCP	24	5.84	39.21	4.46	142.08	4.10	175.17
QPLSCP	24	3.80	50.75	3.14	248.75	3.14	258.04
PPLSCP	24	4.31	45.92	2.59	233.21	2.59	285.75
PRLSCP	24	2.81	54.88	2.52	83.25	2.22	104.08
QRLSCP	24	2.22	46.57	2.01	85.04	2.01	98.13
BRLSCP	24	3.35	39.83	2.13	87.00	1.64	99.79
TOTAL	72	3.72	46.59	2.81	144.94	2.62	170.16

Como era de esperar al relajar los criterios de parada, el procedimiento diseñado mejora los resultados ligeramente, aunque con un esfuerzo computacional notablemente mayor.

Los dos criterios con los que se pretendía evaluar la bondad del heurístico eran la proximidad de la que se encuentre su valor del óptimo y la rapidez con que se localicen dichas soluciones. En esta batería de problemas sobre la red de 79 nodos no se han presentado estadísticas resumen sobre tiempos de computación en comparación con el tiempo empleado por el CPLEX 4.0, ya que para garantizar la obtención de la solución óptima era necesario trabajar sobre instancias de pequeño tamaño, que este software resuelve en tiempos relativamente pequeños. En cambio, para evaluar conjuntamente la rapidez del HGSCP y la eficiencia del heurístico utilizaremos las matrices de adyacencias de 511 por 210 y 1023 por 330.

Las instancias basadas en modelos de cubrimiento múltiple han sido resueltas por el HGSCP y las instancias basadas en cantidad de cubrimiento han sido obtenidas mediante el HGSCP modificado con criterios de parada en la fase e) de 150 iteraciones globales como máximo y 10 iteraciones consecutivas con al menos una mejora del 10% en la diferencia entre la cota superior e inferior. Las tablas siguientes presentan los resultados obtenidos para la matriz de 511 por 210.

**Tabla 3.13. Resultados computacionales de diferentes instancias sobre un escenario de 511 X 210 con una distribución espacial de llamadas para 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR145	BPLSCP	0.80	40	4,00	25,22	36	2602320	44726	894	36	1	1145	18177
PR146	BPLSCP	0.90	57	4,00	37,76	57	2072807	16026	1424	57	2	1078	18219
PR147	BPLSCP	0.99	62	6,00	38,92	59	1895672	23976	550	59	2	974	18206
PR148	QPLSCP	0.80	34	4,00	21,75	33	3041116	46350	862	33	1	943	18199
PR149	QPLSCP	0.90	56	4,00	38,16	53	2031000	15355	1392	53	2	1206	18203
PR150	QPLSCP	0.99	71	7,00	44,49	69	2099434	20708	1174	69	3	1214	18198
PR151	PPLSCP	0.80	40	4,00	25,22	36	2367142	40839	833	36	2	1144	18204
PR152	PPLSCP	0.90	57	5,00	36,63	53	1833207	11429	1494	53	2	1301	18186
PR153	PPLSCP	0.99	71	7,00	46,17	70	1966883	20852	1091	70	3	1350	18239
PR154	PRLSCP	0.80	93	18,30	37,40	84	3323911	59640	436	84	8	843	18088
PR155	PRLSCP	0.90	94	22,07	45,09	87	2179852	25516	1648	87	9	1183	18061
PR156	PRLSCP	0.99	155	38,80	81,74	140	2328128	6923	438	133	16	1024	18195
PR157	QRLSCP	0.80	76	16,47	34,60	66	1545867	11928	2163	66	7	1076	18288
PR158	QRLSCP	0.90	93	23,56	49,85	90	2523509	37927	963	90	9	923	18159
PR159	QRLSCP	0.99	153	37,51	79,56	145	3104583	82097	254	145	16	1438	18154
PR160	BRLSCP	0.80	80	14,01	26,55	74	4108733	17249	376	72	8	866	18145
PR161	BRLSCP	0.90	85	20,04	39,53	78	4020922	19092	421	75	8	730	18140
PR162	BRLSCP	0.99	110	31,36	65,41	102	3415542	14361	483	100	12	1140	18171

Con objeto de evaluar la eficiencia y la rapidez del heurístico “greedy 1-óptimo” para las formulaciones de cubrimiento múltiple y del heurístico “greedy 1-óptimo modificado” para las modelizaciones basadas en cantidad de cubrimiento dentro del HGSCP se ha obtenido la desviación porcentual media respecto de la solución óptima y los tiempos promedio computacionales, agrupados para los diferentes modelos.

**Tabla 3.14. Desviaciones porcentuales medias de las soluciones obtenidas por los heurísticos para las diferentes instancias sobre el escenario de 511 X 210 con una distribución espacial de llamadas para 3 niveles de fiabilidad para cada modelo.**

MODELO	Nº de instancias	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZOPT_i}{ZOPT_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBF_i - ZOPT_i}{ZOPT_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZUBF_i}{ZUBF_i} 100$
BPLSCP	3	5,40	0,00	5,40
QPLSCP	3	3,86	0,00	3,86
PPLSCP	3	6,70	0,00	6,70
PRLSCP	3	11,77	1,75	9,82
QRLSCP	3	8,00	0,00	8,00
BRLSCP	3	11,48	2,93	8,31
TOTAL	18	7,87	0,78	7,02

**Tabla 3.15. Tiempos promedios computacionales de las diferentes instancias sobre el escenario de 511 X 210 con una distribución espacial de llamadas para 3 niveles de fiabilidad para cada modelo.**

MODELO	Nº de instancias	TMPG	TMPH	TMPC
BPLSCP	3	1,67	1065,67	18200,67
QPLSCP	3	2,00	1121,00	18200,00
PPLSCP	3	2,33	1265,00	18209,67
PRLSCP	3	11,00	1016,67	18114,67
QRLSCP	3	10,67	1145,67	18200,33
BRLSCP	3	9,33	912,00	18152,00
TOTAL	18	6,17	1087,67	18179,56

La observación de los resultados presentados indica que en 14 de las 18 instancias, la solución obtenida por el heurístico HGSCP no pudo ser mejorada por el CPLEX 4.0 tras 18000 segundos, solamente en 4 instancias consiguió obtener una mejor solución. En cuanto, a la comparación conjunta entre tiempo y eficiencia de las dos versiones del greedy, hay que hacer constar que en promedio conseguimos soluciones cuya desviación promedio respecto del óptimo disponible es de aproximadamente un 8%, observándose una diferencia

significativa entre las instancias de cubrimiento múltiple con aproximadamente un 5% y un 11% para las instancias de cantidad de cubrimiento, con un tiempo promedio global de 6 segundos, así como unas diferencias significativas entre la resolución de problemas de cubrimiento múltiple y de cantidad de cubrimiento, con una razón de tiempos aproximada de 1 a 5. Por otra parte, los tiempos promedios computacionales del heurístico HGSCP están supeditados a los parámetros definidos en la fase e) del heurístico, pero se observa que con aproximadamente una dieciochoava parte del tiempo empleado por el CPLEX 4.0, se consiguen soluciones muy cercanas al óptimo.

Las tablas siguientes presentan los resultados obtenidos para la matriz de 1023 por 330.

**Tabla 3.16. Resultados computacionales de diferentes instancias sobre un escenario de 1023 X 330 con una distribución espacial de llamadas para 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	FIAB	ZUBI	ZLBI	ZLBF	ZUBF	NIS	NNE	NCOVERS	ZOPT	TMPG	TMPH	TMPC
PR163	BPLSCP	0.80	24	5,00	18,05	23	199907	1579	485	21	2	4669	766
PR164	BPLSCP	0.90	24	6,00	17,78	23	422703	3758	667	21	2	6176	2130
PR165	BPLSCP	0.99	35	7,00	26,73	34	1968751	22832	1569	34	3	5841	18217
PR166	QPLSCP	0.80	25	5,00	17,82	24	360679	3740	566	21	2	3869	1255
PR167	QPLSCP	0.90	25	6,00	18,41	24	2740866	19659	1347	22	2	6677	18221
PR168	QPLSCP	0.99	35	9,00	27,02	34	1910068	15215	1578	34	4	10531	18194
PR169	PPLSCP	0.80	24	6,00	17,53	24	266584	2314	520	21	1	5605	966
PR170	PPLSCP	0.90	26	6,00	18,43	24	2351199	21447	1330	24	2	6038	18219
PR171	PPLSCP	0.99	35	9,00	27,06	34	1704591	16047	1562	34	4	11256	18210
PR172	PRLSCP	0.80	52	18,11	20,25	45	2351634	7916	1638	42	19	1726	18069
PR173	PRLSCP	0.90	55	22,04	23,48	55	2022205	5408	1312	55	26	1147	18074
PR174	PRLSCP	0.99	80	34,32	36,54	80	1511769	3888	1647	73	38	1420	18065
PR175	QRLSCP	0.80	43	15,82	17,08	41	1651562	20155	2126	41	18	1340	18072
PR176	QRLSCP	0.90	49	19,91	23,16	48	1952142	30687	1220	48	21	1980	18066
PR177	QRLSCP	0.99	80	32,74	37,81	74	1698444	5273	1412	74	38	1939	18073
PR178	BRLSCP	0.80	45	12,81	13,61	45	953418	4339	580	39	17	967	4190
PR179	BRLSCP	0.90	49	18,32	19,58	49	244168	1352	401	43	19	1045	926
PR180	BRLSCP	0.99	65	24,02	29,07	65	2511711	7217	952	63	27	2053	18066

**Tabla 3.17. Desviaciones porcentuales medias de las soluciones obtenidas por los heurísticos para las diferentes instancias sobre el escenario de 1023 X 330 con una distribución espacial de llamadas para 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZOPI_i}{ZOPI_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBF_i - ZOPI_i}{ZOPI_i} 100$	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZUBF_i}{ZUBF_i} 100$
37 -39	BPLSCP	10,50	6,35	3,88
40 - 42	QPLSCP	11,88	7,79	3,76
43 -45	PPLSCP	8,52	4,76	3,76
46 -48	PRLSCP	11,13	5,58	5,19
49 -51	QRLSCP	5,02	0,00	5,02
52 -54	BRLSCP	10,84	10,84	0,00
	TOTAL	9,65	5,89	3,60

**Tabla 3.18. Tiempos promedios computacionales de las diferentes instancias sobre un escenario de 1023 X 330 con una distribución espacial de llamadas para 3 niveles de fiabilidad para cada modelo.**

PROB	MODELO		TMPG	TMPH	TMPC
37 -39	BPLSCP		2,33	5562,00	7037,67
40 - 42	QPLSCP		2,67	7025,67	12556,67
43 -45	PPLSCP		2,33	7633,00	12465,00
46 -48	PRLSCP		27,67	1431,00	18069,33
49 -51	QRLSCP		25,67	1753,00	18070,33
52 -54	BRLSCP		21,00	1355,00	7727,33
	TOTAL		13,61	4126,61	12654,39

La observación de los resultados presentados indica que en 8 instancias, la solución obtenida por el heurístico HGSCP no pudo ser mejorada por el CPLEX 4.0 tras 18000 segundos. En cuanto, a la comparación conjunta entre tiempo y eficiencia de las dos versiones del greedy, hay que observar que en promedio conseguimos soluciones cuya desviación promedio respecto del óptimo disponible es de aproximadamente un 10%, con un tiempo promedio global de aproximadamente 14 segundos, observándose unas diferencias significativas entre la resolución de problemas de cubrimiento múltiple y de cantidad de cubrimiento, con una razón de tiempos aproximada de 1 a 10. Por otra parte, los tiempos promedios computacionales del heurístico HGSCP están supeditados a los parámetros definidos en la fase e) del heurístico, pero se observa que existe un comportamiento desigual entre las instancias de cubrimiento múltiple y las instancias de cantidad de cubrimiento, con una razón de tiempos aproximada de 5 a 1.

A la vista de los resultados presentados, tanto por lo que se refiere a desviaciones respecto de la solución óptima como a los tiempos de cálculo, el heurístico descrito manifiesta un buen comportamiento que permitirá en un futuro proporcionar buenas soluciones, con tiempos moderados, para grandes instancias de los modelos presentados en el capítulo anterior.



## Capítulo 4. Conclusiones. Líneas Futuras de investigación

### 4.1. Conclusiones.

Todos los modelos de localización de servicios de emergencias presentados en el capítulo anterior pueden ser expresados por un modelo general que llamaremos Generalized Set Covering Problem (GSCP). Para comprender el proceso de generalización llevado a cabo, se comenzará con la presentación de la formulación del Set Covering Problem (SCP):

$$\text{Min } \mathbf{w}'\mathbf{x} \quad (3.1)$$

$$\text{s.a. } \mathbf{Ax} \geq \mathbf{e} \quad (3.2)$$

$$x_j \in \{0,1\} \quad (j = 1,2,\dots,n) \quad (3.3)$$

donde  $\mathbf{A} = [a_{ij}]$  es una matriz denominada matriz del cubrimiento de dimensión  $m \times n$  cuyos elementos  $a_{ij}$  valen 1 ó 0, dependiendo de que la fila o punto demanda  $i$  esté cubierta o no por la columna o punto servicio  $j$ ,  $\mathbf{e} \in \mathfrak{R}^m$  es el vector formado por  $m$  unos,  $\mathbf{w} \in \mathfrak{R}_+^n$  es el vector de costes y  $x_j$  son las variables a determinar, siendo

$$x_j = \begin{cases} 1, & \text{si la columna } j \text{ está en la solución} \\ 0, & \text{en caso contrario} \end{cases}$$

## 4.2. Líneas de investigación futuras

Este trabajo forma parte de un proyecto más extenso, que requerirá la profundización y extensión de algunas de las líneas de investigación aquí estudiadas. Las direcciones en las que se pretende orientar el trabajo futuro están relacionadas con las siguientes áreas :

- Técnicas de Preprocesamiento aplicadas a la resolución del GSCP
- Nuevos Modelos de Optimización de Cubrimiento Probabilístico
- Ubicaciones óptimas respecto de dos objetivos de cubrimiento probabilísticos

En las siguientes secciones se presenta el estado de las líneas de investigación futuras al concluir esta memoria.

### 4.2.1. Técnicas de preprocesamiento aplicadas a la resolución del GSCP

El éxito de los métodos de Ramificación y Acotación depende de lo cerca que esté la solución óptima correspondiente a la relajación lineal de la solución entera del problema original y de la rapidez con la que se encuentre una solución factible. Una vez obtenida una buena solución factible mediante el heurístico desarrollado en la sección anterior, nos centraremos en mejorar la relajación lineal mediante técnicas de preproceso. Dentro de estas técnicas se incluyen todas aquellas que permitan reformular el problema con el fin de obtener una formulación más fuerte, es decir, que el conjunto de soluciones lineales factibles de la relajación lineal se acerquen más a la envoltura convexa del conjunto de soluciones enteras de las restricciones del problema. Así, en el caso de que la formulación más fuerte definiese las facetas de dicha envoltura convexa, la solución óptima de la relajación lineal sería también solución óptima para el problema combinatorio.

Las técnicas de preproceso que pueden ser utilizadas son el redondeo de coeficientes por defecto en restricciones existentes y la identificación de nuevas desigualdades válidas que son añadidas a la formulación del problema.

- **Redondeo de coeficientes por defecto en restricciones existentes**

Para las restricciones del tipo

$$\sum_{j=1}^n \sum_{k=1}^L a_{ij} b_{jk} x_{jk} \geq c_i \quad i = 1, 2, \dots, m \quad (4.1)$$



donde las variables  $x_{jk}$  toman valores 0 o 1 podemos reemplazar  $a_{ij}b_{jk}$  por  $c_i$  si  $a_{ij}b_{jk} > c_i$ . Dado que cualquier solución entera factible que verifique la restricción verificará la restricción con los coeficientes disminuidos.

**Ejemplo 4.1.** Dado el problema reducido del ejemplo 3.1 y considerando la restricción de cantidad de cubrimiento necesario de la fila 1 tenemos

$$9x_{11} + 11x_{12} + 13x_{13} + 8x_{41} + 11x_{41} + 13x_{43} \geq 12$$

que será redondeada por defecto a

$$9x_{11} + 11x_{12} + 12x_{13} + 8x_{41} + 11x_{41} + 12x_{43} \geq 12$$

- **Identificación de desigualdades de cubrimiento**

Para las restricciones del tipo

$$\sum_{j=1}^n \sum_{k=1}^L a_{ij} b_{jk} x_{jk} \geq c_i \tag{4.2}$$

donde las variables  $x_{jk}$  toman valores 0 o 1, hay al menos una variable  $x_{jk}$  que debe tomar el valor 1. Por otra parte, si existe un subconjunto F de variables 0-1 verificando que

$$\sum_{(j,k) \in F} b_{jk} < c_i \tag{4.3}$$

entonces claramente podemos añadir la desigualdad de cubrimiento

$$\sum_{(j,k) \notin F} x_{jk} \geq 1 \tag{4.4}$$

**Ejemplo 4.2.** Dado el problema reducido del ejemplo 3.1 y considerando la restricción de cantidad de cubrimiento necesario de la fila 6 tenemos

$$7x_{31} + 13x_{32} + 14x_{33} + 5x_{61} + 6x_{62} + 13x_{63} \geq 18$$

Si consideramos el conjunto  $F = \{x_{31}, x_{61}\}$  se puede comprobar que  $b_{31} + b_{61} = 7 + 5 = 12 < 18$  que genera la desigualdad de cubrimiento  $x_{32} + x_{33} + x_{62} + x_{63} \geq 1$

Si consideramos el conjunto  $F = \{x_{31}, x_{62}\}$  se puede comprobar que  $b_{31} + b_{62} = 7 + 6 = 13 < 18$  que genera la desigualdad de cubrimiento  $x_{32} + x_{33} + x_{61} + x_{63} \geq 1$

- **Identificación de desigualdades de cubrimiento fuertes**

Las restricciones del tipo

$$\sum_{j=1}^n \sum_{k=1}^L a_{ij} b_{jk} x_{jk} \geq c_i \quad (4.5)$$

junto con las restricciones

$$\sum_{k=1}^L x_{jk} \leq 1 \quad j = 1, 2, \dots, n \quad (4.6)$$

permiten derivar desigualdades de cubrimiento más fuertes. Considerando el conjunto

$F = \{(j, k) : b_{jk} \leq b^*\}$ , donde

$$b^* = \max \left\{ b' : \sum_{j: a_{ij}=1} \max_k \{ b_{jk} : b_{jk} \leq b' \} < c_i \right\} \quad (4.7)$$

podemos establecer la siguiente desigualdad

$$\sum_{(j,k) \in F} x_{jk} \geq 1 \quad (4.8)$$

Esta desigualdad está basada en que si elegimos el  $b_{jk}$  menor o igual que  $b^*$  para cada  $j$ , entonces la suma es menor que  $c_i$ . Dado que solo un  $x_{jk}$  de los  $k=1, 2, \dots, L$  puede tomar el valor 1, entonces alguna de las variables del conjunto  $\{(j, k) : b_{jk} > b^*\}$  debe tomar el valor 1

**Ejemplo 4.3.** Dado el problema reducido del ejemplo 3.1 y considerando la restricción de cantidad de cubrimiento necesario de la fila 6 tenemos

$$7x_{31} + 13x_{32} + 14x_{33} + 5x_{61} + 6x_{62} + 13x_{63} \geq 18$$

Si consideramos  $b^*=7$  y el conjunto  $F = \{(j, k) : b_{jk} \leq 7\}$  se genera la desigualdad de cubrimiento fuerte

$$x_{32} + x_{33} + x_{63} \geq 1$$

- **Identificación de conjuntos de corte minimales**

Para un nodo demanda  $i$ , llamaremos conjunto de corte minimal a un conjunto de índices  $C = \{(j, k)\}$  satisfaciendo

$$\sum_{j: a_{ij}=1} \max_k \{ b_{jk} : (j, k) \notin C \} < c_i \quad (4.9)$$

que permite establecer la siguiente desigualdad

$$\sum_{(j,k) \notin C} x_{jk} \geq 1 \tag{4.10}$$

El siguiente ejemplo ilustra el proceso para obtener un conjunto de corte minimal.

**Ejemplo 4.4.** Dado el problema reducido del ejemplo 3.1, consideremos la restricción de cantidad de cubrimiento necesario de la fila 6

$$2x_{11} + 3x_{12} + 5x_{13} + 3x_{21} + 4x_{22} + 8x_{23} + 1x_{31} + 5x_{32} + 6x_{33} \geq 18.$$

El proceso para generar un conjunto de corte minimal, C, puede ser ilustrado como sigue:

	Corte 1:	Corte 2:	Corte 3:
Unidad de servicio 1:	2 <b>3</b> 5	2    3 <b>5</b>	2    3 <b>5</b>
Unidad de servicio 2:	3    4 <b>8</b>	3 <b>4</b> 8	3 <b>4</b> 8
Unidad de servicio 3:	1 <b>5</b> 7	1    5 <b>7</b>	1 <b>5</b> 7

Obviamente,  $C1 = \{x_{13}, x_{33}\}$  es un conjunto de corte minimal, ya que

$$\sum_{j: a_{ij}=1} \max_k \{b_{jk} : (j,k) \notin C\} = 3 + 8 + 5 = 16 < 18 = c_i$$

permitiendo generar la desigualdad  $x_{31} + x_{33} \geq 1$ .

Análogamente, podemos encontrar un segundo conjunto de corte  $C2 = \{x_{23}\}$ , que fija la variable  $x_{23} = 1$  dado que son variables binarias.

El tercer conjunto de corte C3 no es minimal ya que incluye el conjunto de corte C2.

Para asegurar que un conjunto de corte es minimal, solamente debemos comprobar que sustituyendo el máximo  $b_{jk}$  no perteneciente al conjunto de corte por  $b_{jk+1}$  la propiedad no se cumple.

En general, la identificación de cortes minimales para una restricción de cubrimiento generalizado crece exponencialmente con el número de unidades de servicio que pueden atender al nodo demanda.

#### **4.2.2. Nuevos Modelos de Optimización de Cubrimiento Probabilístico**



### 4.2.3. Ubicaciones óptimas respecto de dos objetivos de cubrimiento probabilístico

Cuando se pretenden estudiar problemas de ubicaciones de estaciones y vehículos de emergencia que se adapten más a la realidad, la modelización debe considerar más de un objetivo para que el decisor pueda evaluar los méritos de distintas configuraciones alternativas. Una nueva línea de investigación será centrarnos en los problemas bicriterio. En general, no hay una única solución que sea óptima para cada uno de los dos objetivos considerados en un problema bicriterio, por tanto se trata de encontrar un conjunto de soluciones eficientes (óptimos de Pareto), de forma que el decisor pueda seleccionar la solución de mejor preferencia o de mejor compromiso a la vista del conjunto de soluciones eficientes generado.

La modelización de nuestro sistema de emergencias con  $p$  vehículos está basado en las siguientes hipótesis:

- Los  $I$  nodos demanda son generadores de llamadas independientes siguiendo una distribución de Poisson con media  $f_i$  llamadas por unidad de tiempo.
- La media del tiempo del servicio es  $\bar{t}$ , independiente de la identidad del servidor, de la localización del demandante y de la historia del sistema.
- Hay un orden determinístico y estacionario que determina la preferencia de cada nodo demanda para ser atendido por las distintas estaciones abiertas, que depende del tiempo de viaje
- El vehículo más cercano libre es asignado a la llamada de emergencia.
- Solamente un vehículo es asignado a cada incidente.
- Todas las llamadas son contestadas por un vehículo aparcado en la estación de emergencias y no volviendo de un servicio.

- Si todos los vehículos del sistema están ocupados y llega una llamada, entonces es desviada a otro sistema de emergencia distinto

El propósito del modelo es maximizar dos medidas del cubrimiento probabilístico, una relacionada con la eficiencia global del sistema como la probabilidad que una llamada cualquiera del sistema sea atendida dentro del tiempo máximo  $S$

$$\sum_{i \in I} f_i \left( \sum_{j \in N_i} P_{ji} \right) / \sum_{i \in I} f_i \quad (4.1)$$

y otra relacionada con la equidad del sistema como es la probabilidad de ser servido dentro de  $S$  del peor nodo demanda atendido,

$$\text{Min}_{i \in I} \sum_{j \in N_i} P_{ji} \quad (4.2)$$

donde

$d_{ij}$  es el tiempo de viaje que separa el nodo  $i$  del potencial puesto de servicio  $j$

$S$  es el tiempo máximo permitida para que una llamada de emergencia se considere cubierta;

$N_i = \{j \in J / d_{ij} \leq S\}$  es el conjunto de ubicaciones potenciales que pueden servir al nodo  $i$  dentro del tiempo máximo o distancia permitida  $S$ .

$P_{ji}$  = la probabilidad que el vehículo de la estación  $j$  sea asignado a una llamada de la zona  $i$ .

Denotando

$y_j$  = 1 si la estación  $j$  es abierta, 0 en otro caso.

$u_{jki}$  = 1 si la estación  $j$  es la  $k$ -ésima base más cercana a la zona demanda  $i$ , 0 en otro caso.

$x_j$  = número de vehículos estacionados en la base  $j$ .

$f_i$  = número o frecuencia de llamadas del nodo demanda  $i$ , en llamadas por día.

$\bar{t}$  = duración media de un servicio de emergencia, en horas.

$\rho_j$  = fracción de tiempo que cada vehículo de la estación  $j$  está ocupado sirviendo llamadas, suponiendo que la carga de trabajo total de la estación es distribuida igualitariamente entre todas las unidades de emergencia

Si en el modelo añadimos que la probabilidad que un vehículo esté ocupado no depende del estado del sistema, entonces la probabilidad que el vehículo de la estación  $j$  sea asignado a una llamada de la zona  $i$ ,  $P_{ji}$ , si  $j$  es la estación  $k$ -ésima preferida para la zona  $i$  puede ser calculada como

$$P_{ji} = (1 - \rho_j^{x_j}) \left[ \prod_{l=1}^{k-1} \rho_{a_{il}}^{x_{a_{il}}} \right] \quad (4.3)$$

donde  $a_{il}$  es la estación  $l$ -ésima preferida para el nodo demanda  $i$ .

La fórmula de cálculo supone que los vehículos de las  $k-1$  más preferidas estaciones para servir el nodo  $i$  están ocupados y hay alguna unidad de emergencia en la base  $j$ -ésima libre para poder atender el servicio. La probabilidad conjunta se obtiene como el producto de las marginales basada en la independencia. *Batta et al (1989)* muestran que en un sistema de dos vehículos hay siempre dependencia y que el producto de probabilidades marginales difiere del de la probabilidad conjunta. Por otra parte, un reciente artículo de *Birge and Pollock (1989)* da evidencias empíricas que el sesgo causado por la hipótesis de independencia es bastante pequeño.

La formulación del modelo suponiendo independencia entre las unidades de emergencia, basado en el presentado por Goldberg y Paz (1991), puede ser formulada como

$$\text{Max} \sum_{i \in I} f_i \left( \sum_{j \in N_i} \sum_k u_{jki} (1 - \rho_j^{x_j}) \left[ \prod_{l=1}^{k-1} \sum_{r \in N_i} u_{rli} \rho_r^{x_r} \right] \right) / \sum_{i \in I} f_i \quad (4.4)$$

$$\text{Max} \text{Min}_{i \in I} \left( \sum_{j \in N_i} \sum_k u_{jki} (1 - \rho_j^{x_j}) \left[ \prod_{l=1}^{k-1} \sum_{r \in N_i} u_{rli} \rho_r^{x_r} \right] \right) \quad (4.5)$$

$$\text{s.a.} \sum_{j \in J} u_{jki} = 1 \quad \text{para cada posible par } (k, i) \quad (4.6)$$

$$\sum_k u_{jki} \leq 1 \quad \text{para cada posible par } (j, i) \quad (4.7)$$

$$\sum_k u_{jki} \leq x_j \quad \text{para cada posible par } (j, i) \quad (4.8)$$

$$x_j \leq p y_j \quad \forall j \in J \quad (4.9)$$

$$\sum_{j \in J} x_j = N \quad (4.10)$$

$$\sum_k k u_{jki} \leq \sum_k k u_{rki} + p(1 - y_r) \quad \text{para cada } i \text{ y cada par de} \quad (4.11)$$

bases  $(j, r)$  donde  $j$  sea preferido  
a  $r$  para  $i$

$$\rho_j = \frac{\sum_{i \in I} f_i \left( \sum_k u_{jki} (1 - \rho_j^{x_j}) \left[ \prod_{l=1}^{k-1} \sum_{r \in J} u_{rli} \rho_r^{x_r} \right] \right)}{24 x_j} \quad \forall j \in J \quad (4.12)$$

$$\rho_j \leq y_j \quad \forall j \in J \quad (4.13)$$

$$y_j \in \{0,1\}, x_j \in \{0,1,\dots,N\}, \rho_j \geq 0 \quad \forall j \in J \quad (4.14)$$

$$u_{jki} \in \{0,1\} \quad \forall j \in J, k = 1,2,\dots,N, \forall i \in I \quad (4.15)$$

Las funciones objetivos (4.4) y (4.5) maximizan la probabilidad de que una llamada cualquiera del sistema sea atendida dentro de un tiempo  $S$  y la probabilidad de ser servido dentro de  $S$  del peor nodo demanda atendido. El conjunto de restricciones (4.6) aseguran que cada estación abierta toma un rango de preferencia particular  $k$  para cada nodo  $i$ . (4.7) asegura que al menos un rango puede ser asignado a cada estación abierta para alguna zona particular  $i$ . (4.8) permite asignar rangos de proximidad solo a estaciones abiertas, ya que si  $x_j = 0$ , entonces asegura que  $u_{jki} = 0$  para todos los pares  $(k,i)$ . El conjunto (4.9) limita el tamaño



de la estación al número de vehículos a ubicar y no asigna unidades de emergencia si la estación no está abierta. (4. 10) fija el número de vehículos a ubicar. (4. 11) garantiza que si la base  $j$  es preferida a la base  $r$  para el nodo demanda  $i$ , entonces  $j$  debe tener un rango de cercanía menor si ambas están abiertas. Cuando solo la estación  $j$  es seleccionada el lado derecho se convierte en  $p$ , y entonces la estación  $j$  puede tomar cualquier rango para la zona  $i$ . El conjunto de restricciones no lineales (4. 12) presenta las ecuaciones de utilización para los vehículos de cada estación abierta, basada en la definición de carga de trabajo de un vehículo, i.e., el tiempo esperado que la estación está ocupado durante el día dividido por el tiempo disponible de trabajo de todas las unidades de emergencia ubicadas en la estación. (4. 13) permite que la carga de trabajo  $\rho_j$  sea igual a 0 cuando la base no está abierta. El conjunto de restricciones (4. 14) y (4. 15) explicita el carácter binario de las variables  $y_j$ ,  $u_{jki}$ . El carácter entero de las variables  $x_j$  y la no negatividad de las variables.

Dada una configuración de los vehículos de emergencia, entonces todas las variables  $u_{jki}$  son determinadas fácilmente, ya que solo estaciones abiertas obtienen rangos de preferencia, ver restricciones (4. 8), cada base abierta toma solo un rango para cada zona (ver restricciones (4. 6)) y el rango es determinado comprimiendo los rangos de preferencia para todas las estaciones en una que incluye solo bases abiertas (ver (4. 11)). Por tanto, solo resta resolver las ecuaciones no lineales, (4. 12), para obtener las fracciones de ocupación  $\rho_j$  y evaluar los objetivos. Esta estructura particular del problema nos permite plantear un heurístico para su resolución. Su estructura es:

- Fase inicial: Conversión en una sola función utilizando el método de los pesos  

$$z = (4-w) \cdot \text{eficiencia} + w \cdot \text{equidad}, \quad 0 < w < 1$$
- Fase a: Obtención de una configuración de vehículos inicial resolviendo un problema lineal entero mixto.

- Fase b: Evaluación del objetivo ponderado con la configuración inicial obtenida, una vez determinados los rangos de preferencia, resolviendo el sistema de ecuaciones no lineales (4. 12) con un proceso iterativo de punto fijo.
- Fase c: Procedimiento de intercambio 1-óptimo, donde en cada reubicación del vehículo se resuelve la fase b, guardándose la configuración si se produce una mejora.

La solución inicial del intercambio 1-óptimo está basada en el siguiente problema lineal entero mixto, que utiliza las siguientes hipótesis

- La carga de cada vehículo es considerada la misma y estimada a priori, a través del cociente entre tiempo esperado de servicio y el tiempo disponible para el servicio con la expresión propuesta por Daskin (1982)

$$\rho = \frac{\bar{t} \sum_{i \in I} f_i}{24p}$$

- No existen rangos de preferencia para las llamadas realizadas desde un punto demanda, sólo se establece si la estación que la sirve está dentro de un radio de tiempo de viaje  $S$  o no.

La formulación del BIMEXCLP( Biobjective Maximum Expected Covering Location Problem), fácilmente transformable a un único objetivo ponderado, es

$$\text{Max} \quad \sum_{i \in I} f_i \left( \sum_{k=1}^p (1-\rho)\rho^{k-1} y_{ik} \right) \quad (4.16)$$

$$\text{Max} \quad E \quad (4.17)$$

$$\text{s.a.} \quad \sum_{j \in N_i} \sum_{k=1}^p kx_{jk} = \sum_{k=1}^p y_{ik} \quad \forall i \in I \quad (4.18)$$

$$y_{ik} \leq y_{i,k-1} \quad \forall i \in I, k = 2,3,\dots, p \quad (4.19)$$

$$\sum_{j \in N_i} \sum_{k=1}^p b_{jk} x_{jk} \geq E \quad \forall i \in I \quad (4.20)$$

$$\sum_{k=1}^p x_{jk} \leq 1 \quad \forall j \in J \quad (4.21)$$

$$\sum_{j \in J} \sum_{k=1}^p kx_{jk} = p \quad \forall j \in J \quad (4.22)$$

$$y_{ik}, x_{jk} \in \{0,1\}, \quad \forall i \in I, \forall j \in J, k = 1,2,\dots, p \quad (4.23)$$

$$E \geq 0 \quad (4.24)$$

donde

$B_j = \{i \in I / d_{ji} \leq S\}$  es el conjunto de nodos demanda que pueden ser atendidos por la estación j.

$$b_{jk} = -\ln(r_{jk})^k \quad \text{siendo} \quad r_{jk} = \frac{\bar{t} \sum_{i \in B_j} f_i}{24k}$$

$$x_{jk} = \begin{cases} 1, & \text{si } k \text{ vehículos son ubicados en la estación } j \\ 0, & \text{en caso contrario} \end{cases}$$

$$y_{ik} = \begin{cases} 1, & \text{si el nodo } i \text{ es cubierto al menos por } k \text{ vehículos} \\ 0, & \text{en caso contrario} \end{cases}$$

El objetivo (4. 16) maximiza el cubrimiento esperado de llamadas que son atendidas dentro de un tiempo S y (4. 17) maximiza el opuesto del logaritmo neperiano de la

probabilidad de no ser atendido del nodo peor servido de la red. Las restricciones (4. 18) aseguran que el número de veces que un nodo demanda es cubierto es igual al número de vehículos ubicados en su entorno. Las desigualdades (4. 19) obligan a que un nodo demanda no puede ser cubierto al menos  $k$  veces si no es cubierto  $k-1$  veces. Las restricciones del tipo (4. 20) permiten asegurar un nivel de cubrimiento esperado mínimo para todos los nodos demanda. El grupo (4. 21) de desigualdades aseguran que o  $k$  vehículos son ubicados en la localización  $j$  o ningún vehículo es estacionado en  $j$ . La igualdad (4. 22) indica el número de vehículos a ubicar. (4. 23) y (4. 24) determinan el carácter de las variables.

El objetivo (4. 16) más el conjunto de restricciones (4. 18), (4. 21), (4. 22) y (4. 23) son equivalentes al Maximum expected Covering Location Problem, MEXCLP (Daskin, 1983). El objetivo (4. 17) junto con el conjunto de restricciones (4. 20) son las encargadas de conseguir para todos los nodos demanda un nivel de equidad determinado.

El bloque de restricciones (4. 19) es innecesario, ya que los coeficientes asociados a las variables  $y_{ik}$  son decrecientes y por tanto siempre esta asegurado que para conseguir ser cubierto  $k$  veces debe haber sido cubierto  $k-1$  veces. La incorporación de este conjunto se introduce porque se consigue que la relajación lineal este más cerca de la envoltura convexa del conjunto de soluciones factibles para este problema y por tanta el proceso de ramificación y cota es menor.

Obtenida la solución óptima del problema anterior, y considerando la variable entera  $x_j^0$  = número de vehículos a ubicar en la base  $j$  y obtenidos los ordenes de preferencia para ser atendidos para cada nodo demanda a partir de los tiempos de viaje a las estaciones abiertas, podemos plantear el conjunto de restricciones (4. 12) para la configuración obtenida como

$$\rho_j^0 = \frac{\bar{t} \sum_{i \in I} f_i \left( \left( 1 - (\rho_j^0)^{x_j^0} \right) \left[ \prod_{l=1}^{k-1} \rho_{a_{il}}^{x_{a_{il}}^0} \right] \right)}{24 x_j^0} \quad \forall j \in J^0 \quad (4. 25)$$

donde

$J^0 = \{j \in J / x_j^0 > 0\}$  = conjunto de estaciones abiertas

$a_{il}$  denota el subíndice asociado con el  $l$ -ésimo vehículo preferido para el nodo  $i$ , análogamente  $j = a_{ik}$ .

El procedimiento usado para la resolución es un método iterativo de punto fijo donde

a) la solución inicial es  $r_j^c = 0$  para  $j \in J^0$

b) las cargas de trabajo de cada vehículo pueden ser aproximadas por

$$r_j^{c+1} = \frac{V_j^c}{1 + V_j^c} \quad \forall j \in J^0 \quad (4.26)$$

donde

$$V_j^c = \frac{\bar{t}}{24x_j^0} \left( 1 + r_j^c + \dots + (r_j^c)^{x_j^0 - 1} \right) \sum_{i \in I} f_i \prod_{l=1}^{k-1} (r_{a_{il}}^c)^{x_{a_{il}}^0} \quad \forall j \in J^0 \quad (4.27)$$

c) Si no se cumple el criterio de convergencia, ir a paso b.

$$\sum_{j=1}^N |r_j^{c+1} - r_j^c| \leq \varepsilon \quad (4.28)$$

Obtenidas las fracciones de ocupación,  $r_j^0$ , de cada vehículo en cada estación, podemos evaluar los objetivos con

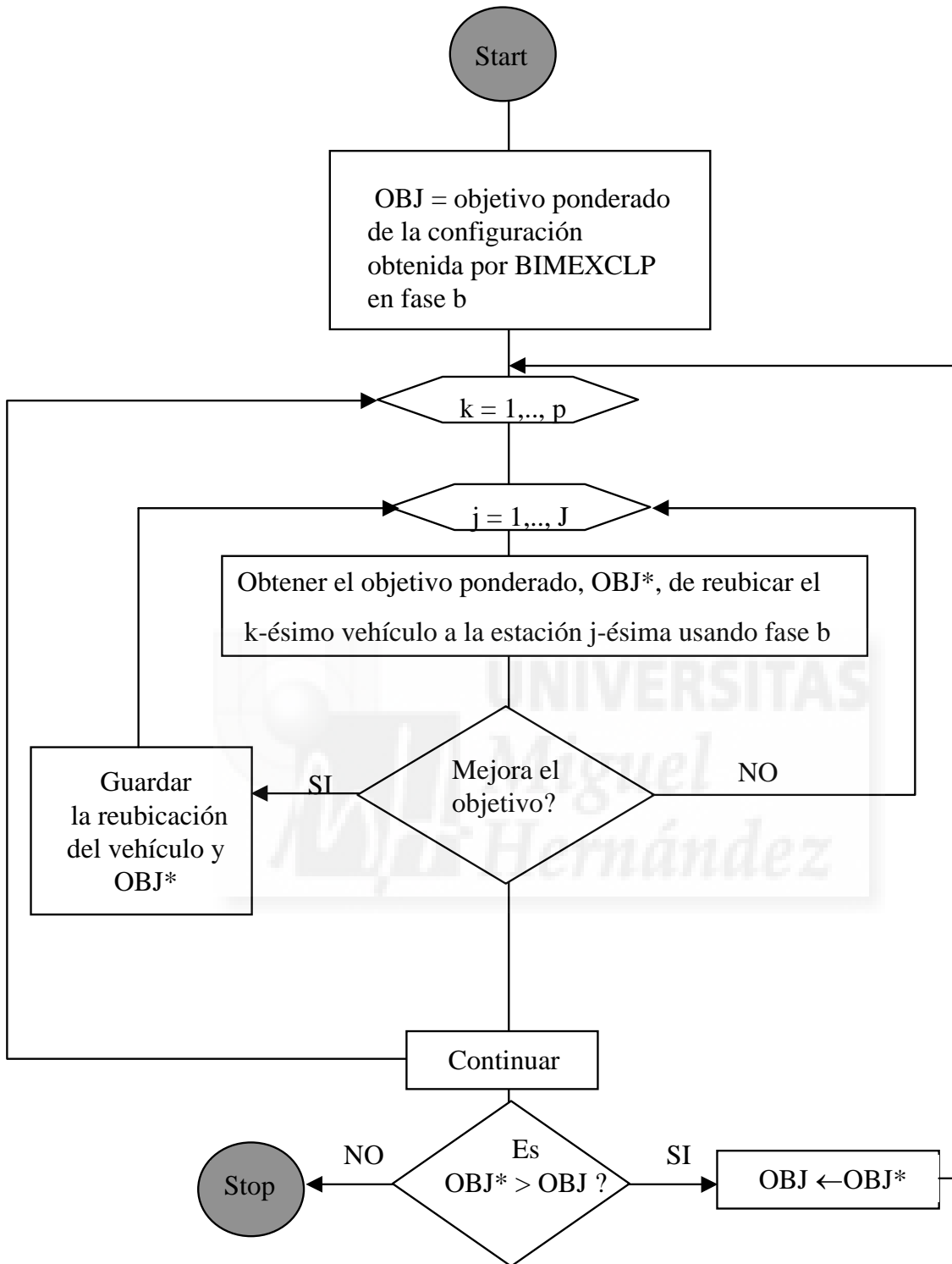
$$\sum_{i \in I} f_i \left( \sum_{j \in N_i^0} (1 - (r_j^0)^{x_j^0}) \prod_{l=1}^{k-1} (r_{a_{il}}^0)^{x_{a_{il}}^0} \right) / \sum_{i \in I} f_i \quad (4.29)$$

$$\text{Min}_{i \in I} \left( \sum_{j \in N_i^0} (1 - (r_j^0)^{x_j^0}) \prod_{l=1}^{k-1} (r_{a_{il}}^0)^{x_{a_{il}}^0} \right) \quad (4.30)$$

donde

$N_i^0 = \{j \in N_i \cap J^0\}$  = conjunto de estaciones abiertas que pueden servir al nodo  $i$  dentro del tiempo máximo  $S$ .

Una vez obtenido el objetivo ponderado, podemos comenzar con el procedimiento 1-óptimo, cuyo mecanismo interno está desarrollado en el siguiente diagrama de flujo





## Capítulo 4. Conclusiones.

Esta memoria se ha dedicado a profundizar en el estudio de modelos de cubrimiento total probabilístico, incluyendo tanto su diseño como su resolución computacional. Las principales aportaciones realizadas han sido:

- Introducción de la verificación a posteriori del nivel de fiabilidad prefijado, sobre la base de la configuración de vehículos obtenida, resolviendo un modelo lineal entero, usando para tal fin, un simulador de sistemas de emergencia y/o la resolución de un sistema de ecuaciones no lineales, correspondiente a sistemas en equilibrio de modelos de la teoría de colas.
- Estudio empírico de las diferencias existentes entre el simulador y el sistema de ecuaciones no lineales del sistema, bajo las hipótesis alternativas de independencia o dependencia entre las unidades de servicio. Sorprendentemente, la no verificación de la fiabilidad prefijada, pronosticada mediante el sistema de ecuaciones, concuerda aproximadamente un 98% de las veces con los resultados obtenidos mediante la simulación.
- Estudio computacional de las deficiencias de los modelos existentes en la literatura revisada sobre una batería de problemas test, obteniéndose los siguientes resultados:



**Tabla 4.1. Número de instancias y porcentaje global de instancias que no verifican el nivel de fiabilidad prefijado para diferentes modelos de cubrimiento total probabilístico para un total de 36 instancias resueltas en cada celda (9 fiabilidades en 4 escenarios distintos).**

	BPLSCP		QPLSCP		PRLSCP	
	MLR -I	MLR -D	MLR -I	MLR -D	MLR -I	MLR -D
Red de 55 y S=1.5	24	24	19	22	0	0
Red de 55 y S=3	19	28	10	25	0	0
Red de 79 y S=1.5	11	15	9	10	0	0
Red de 79 y S=3	15	23	14	19	0	0
<b>Total</b>	69	90	52	76	0	0
<b>%</b>	47.9	62.5	36.1	52.8	0	0

Básicamente, los modelos existentes en la literatura basados en el cubrimiento múltiple no verifican, en la mayoría de los casos, los niveles prefijados de fiabilidad. Resaltar que el comportamiento del QPLSCP es mejor que el del BPLSCP, dado que el porcentaje global de instancias que no verifican los niveles requeridos es del 36.1% frente al 47.9%, bajo la hipótesis de independencia y de un 52.8% frente a un 62.5%, bajo la de dependencia. Por otra parte, el primer modelo basado en cantidad de cubrimiento, el PRLSCP es un modelo que garantiza la obtención de una fiabilidad local mínima por encima del nivel de fiabilidad prefijado, tanto suponiendo independencia como dependencia, debido a que ubica, por lo general, más vehículos de los necesarios. En otras palabras, la cota superior probabilística establecida para cada nodo demanda es, en muchas ocasiones, excesiva para garantizar la fiabilidad impuesta.

- Construcción de nuevos modelos basados en cubrimiento múltiple:
  1. Formulación del Poisson Probabilistic Location Set Covering Problem (PPLSCP), basado en la hipótesis “la generación de llamadas de cada nodo demanda sigue una

distribución de Poisson”, que permite encontrar el mínimo número de vehículos necesario para verificar la fiabilidad impuesta con la expresión

$$b_i = \min \left\{ n \in \mathbb{N} / \sum_{k=0}^{n-1} e^{-\lambda_i/\mu_i} \frac{1}{k!} \left( \frac{\lambda_i}{\mu_i} \right)^k \geq \alpha \right\} \quad \text{donde} \quad \frac{\lambda_i}{\mu_i} = \frac{\bar{t} \sum_{k \in M_i} f_k}{24}$$

2. Revisión del BPLSCP, que denotamos por RBPLSCP, basada en una nueva definición de la fracción de ocupación local específica del nodo demanda  $i$ , que necesita para su implementación del conocimiento de la ubicación de una configuración inicial de vehículos,  $x_j^0 \quad \forall j \in J^0$ , y de la resolución del sistema de ecuaciones no lineales, bajo la hipótesis de independencia,

$$\rho_j^0 = \frac{\bar{t} \sum_{i \in I} f_i \left( \left( 1 - (\rho_j^0)^{x_j^0} \right) \left[ \sum_{k: a_{ik}=j} \prod_{\ell=1}^{k-1} (\rho_{a_{i\ell}}^0)^{x_{a_{i\ell}}^0} \right] \right)}{24x_j^0} \quad \forall j \in J^0,$$

cuya solución,  $r_j^0 \quad \forall j \in J^0$ , es obtenida mediante un método iterativo de punto fijo. A continuación, se establece un procedimiento de resolución de problemas de cubrimiento múltiple, que determina el mínimo número de vehículos necesario para verificar la fiabilidad fijada con la expresión:

$$b_i^* = \min \left\{ n \in \mathbb{N} / \left( \frac{\bar{t} \sum_{j \in J^0 \cap N_i} \sum_{i \in I} f_i P_{ji}^0}{24n} \right)^n \leq 1 - \alpha \right\} \quad \forall i \in I,$$

donde  $P_{ji} = \left( 1 - (r_j^0)^{x_j^0} \right) \left[ \prod_{\ell=1}^{k-1} (r_{a_{i\ell}}^0)^{x_{a_{i\ell}}^0} \right]$ , si la estación  $j$ -ésima es la  $k$ -ésima preferida para el nodo demanda  $i$ .

3. Revisión del PPLSCP, que denotamos por RPPLSCP, basada en una nueva definición de la fracción de ocupación local específica del nodo demanda  $i$ , que necesita para su implementación del conocimiento de la ubicación de una configuración inicial de vehículos,  $x_j^0 \quad \forall j \in J^0$ , y de la resolución del sistema de ecuaciones no lineales, bajo la hipótesis de dependencia,

$$\rho_j^0 = \frac{\bar{t} \sum_{i \in I} f_i \left( (1 - \rho_j^0) \sum_{k: a_{ik}=j} \left( \sum_{s=0}^{x_j^0-1} Q(N, \rho, \sum_{\ell=1}^{k-1} x_{a_{i\ell}}^0 + s) (\rho_j^0)^s \right) \left( \prod_{\ell=1}^{k-1} (\rho_{a_{i\ell}}^0)^{x_{a_{i\ell}}^0} \right) \right)}{24 x_j^0} \quad \forall j \in J^0$$

$$\frac{\sum_{j \in J^0} \rho_j^0 x_j^0}{N} = \rho(1 - P_N),$$

cuya solución,  $r_j^0 \quad \forall j \in J^0$ , es obtenida mediante un método iterativo de punto fijo. A continuación, se establece un procedimiento de resolución de problemas de cubrimiento múltiple, que determina el mínimo número de vehículos necesario para verificar la fiabilidad impuesta con la expresión:

$$b_i^* = \min \left\{ n \in \mathbb{N} / \sum_{k=0}^{n-1} e^{-\lambda_i^0 / \mu_i^0} \frac{1}{k!} \left( \frac{\lambda_i^0}{\mu_i^0} \right)^k > \alpha \right\}$$

donde

$$\frac{\lambda_i^0}{\mu_i^0} = \frac{\bar{t} \sum_{j \in J^0 \cap N_i} \sum_{i \in I} f_i P_{ji}^0}{24} \quad \forall i \in I$$

$$P_{ji}^0 = (1 - r_j^0) \left( \sum_{s=0}^{x_j^0-1} Q(N, \rho, \sum_{\ell=1}^{k-1} x_{a_{i\ell}}^0 + s) (r_j^0)^s \right) \prod_{\ell=1}^{k-1} (r_{a_{i\ell}}^0)^{x_{a_{i\ell}}^0}, \text{ si la estación } j\text{-ésima es la } k\text{-ésima preferida para el nodo demanda } i.$$

- Estudio computacional del comportamiento de los nuevos modelos de cubrimiento múltiple sobre la batería de problemas test, obteniéndose los siguientes resultados:

**Tabla 4.2. Número de instancias y porcentaje global de instancias que no verifican el nivel de fiabilidad prefijado para diferentes modelos de cubrimiento total probabilístico para un total de 36 instancias resueltas en cada celda (9 fiabilidades en 4 escenarios distintos).**

	PPLSCP		RBPLSCP		RPPLSCP	
	MLR -I	MLR -D	MLR -I	MLR -D	MLR -I	MLR -D
Red de 55 y S=1.5	10	14	0	8	0	0
Red de 55 y S=3	1	10	0	18	0	0
Red de 79 y S=1.5	1	3	0	8	0	0
Red de 79 y S=3	6	10	0	18	0	0
<b>Total</b>	18	37	0	52	0	0
<b>%</b>	12.5	25.69	0	36.11	0	0

Se observa claramente un mejor comportamiento respecto a la verificación de las restricciones de fiabilidad impuestas en comparación con los modelos de cubrimiento total probabilístico basados en cubrimiento múltiple existentes (véase tabla 4.1).

- Construcción de nuevos modelos basados en cantidad de cubrimiento:
  1. Formulación del Binomial Reliability Location Set Covering Problem (BRLSCP), basada en una hipótesis de independencia, como

$$\begin{aligned} \text{Min} \quad & \sum_{j \in J} \sum_{k=k_{0j}}^L kx_{jk} \\ \text{s.a.} \quad & \sum_{j \in N_i} \sum_{k=k_{0j}}^L -\log(r_{jk}^k) x_{jk} \geq -\log(1-\alpha) \quad \forall i \in I \\ & \sum_{k=k_{0j}}^L x_{jk} \leq 1 \quad \forall j \in J \\ & x_{jk} \in \{0,1\} \quad \forall j \in J, \quad k = 1,2,\dots,L, \end{aligned}$$

donde

$$\begin{aligned} r_{jk} &= \frac{\bar{t} \sum_{i \in B_j} f_i}{24k} \\ L &= \text{Max}_{j \in J} \left\{ \min \left\{ k / -\log(r_{jk}^k) > -\log(1-\alpha) \right\} \right\} \\ k_{0j} &= \min \{ k / k \in \{1,2,\dots,L\} \wedge r_{jk} \leq 1 \}. \end{aligned}$$

2. Formulación del Queueing Reliability Location Set Covering Problem (QRLSCP), donde se modeliza el comportamiento en cada entorno de cada estación como un sistema de colas con llegadas distribuidas Poisson, tiempos de servicio exponenciales y pérdida de llamadas cuando el sistema está saturado. La formulación obtenida es un refinamiento del PRLSCP de *Ball y Lin* (1993), calculando los coeficientes de las restricciones de cantidad de cubrimiento a satisfacer mediante la distribución de Poisson truncada.

- Estudio computacional del comportamiento de los modelos basados en cantidad de cubrimiento sobre la batería de problemas test, obteniéndose los siguientes resultados:

**Tabla 4.3. Número de instancias y porcentaje global de instancias que no verifican el nivel de fiabilidad prefijado para diferentes modelos de cubrimiento total probabilístico para un total de 36 instancias resueltas en cada celda (9 fiabilidades en 4 escenarios distintos).**

	BRLSCP		QRLSCP	
	MLR -I	MLR -D	MLR -I	MLR -D
Red de 55 y S=1.5	1	2	0	1
Red de 55 y S=3	0	12	0	0
Red de 79 y S=1.5	0	2	0	1
Red de 79 y S=3	0	17	1	6
<b>Total</b>	1	33	1	8
<b>%</b>	0.69	22.92	0.69	5.56

Se observa claramente que los resultados obtenidos por los nuevos modelos de cubrimiento total probabilístico, basados en cantidad de cubrimiento, tienen un comportamiento ligeramente inferior que el PRLSCP (véase tabla 4.1), pero los alcanza ubicando un número de vehículos menor.

- Estudio computacional de la eficiencia de los diferentes modelos de cubrimiento total probabilístico estudiados en esta memoria; las siguientes dos tablas presentan un resumen a este respecto.

**Tabla 4.4. Número total de instancias y porcentaje global de instancias que verifican el nivel de fiabilidad prefijado con el menor número de vehículos conocido para diferentes modelos de cubrimiento total probabilístico, suponiendo independencia y dependencia, para un total de 144 instancias resueltas (9 fiabilidades en 16 escenarios distintos).**

	INDEPENDENCIA		DEPENDENCIA	
	total	%	total	%
BPLSCP	62	43.056	43	29.861
QPLSCP	46	31.944	45	31.250
PPLSCP	31	21.528	40	27.778
RBPLSCP	119	82.639	71	49.306
RPPLSCP	34	23.611	49	34.028
PRLSCP	31	21.528	37	25.694
BRLSCP	73	50.694	91	63.194
QRLSCP	96	66.667	80	55.556

**Tabla 4.5. Número total de instancias y porcentaje global de instancias que verifican el nivel de fiabilidad prefijado siendo la mejor y la única solución conocida para diferentes modelos de cubrimiento total probabilístico, suponiendo independencia y dependencia, para un total de 144 instancias resueltas (9 fiabilidades en 16 escenarios distintos).**

	INDEPENDENCIA		DEPENDENCIA	
	total	%	total	%
BPLSCP	0	0.000	0	0.000
QPLSCP	1	0.694	5	3.472
PPLSCP	0	0.000	0	0.000
RBPLSCP	15	10.417	6	4.167
RPPLSCP	0	0.000	2	1.389
PRLSCP	0	0.000	1	0.694
BRLSCP	6	4.167	17	11.806
QRLSCP	7	4.861	10	6.944

A la vista de los resultados presentados, tanto por lo que se refiere a configuraciones con menos vehículos cumpliendo la fiabilidad, como a la obtención de la mejor solución conocida, podemos concluir que nuestros modelos presentan un mejor comportamiento que el resto de los modelos existentes hasta la fecha en la literatura.

- Construcción de un modelo general que incluye todos los modelos de cubrimiento total probabilístico presentados, tanto modelos basados en el cubrimiento múltiple, como los basados en cantidad de cubrimiento, con la formulación general:

$$\text{Min } \sum_{j=1}^n \sum_{k=1}^L w_{jk} x_{jk}$$

$$\text{s.a. } \sum_{j=1}^n \sum_{k=1}^L a_{ij} b_{jk} x_{jk} \geq c_i \quad i = 1, 2, \dots, m$$

$$\sum_{k=1}^L x_{jk} \leq 1 \quad j = 1, 2, \dots, n$$

$$x_{jk} \in \{0, 1\} \quad j = 1, 2, \dots, n; \quad k = 1, 2, \dots, L,$$

donde:

$L$  es el número máximo de facilidades que pueden ubicarse en cualquier punto servicio  $j$ , para  $j=1,2,\dots,n$ .

$w_{jk}$  es el coste de emplazar  $k$  facilidades en el puesto de servicio  $j$  para  $j=1,2,\dots,n$  y  $k=1,2,\dots,L$ .

$a_{ij}$  vale 1 ó 0, dependiendo de que la fila o punto demanda  $i$  esté cubierta o no por la columna o punto servicio  $j$ , para  $i=1,2,\dots,m$  y  $j=1,2,\dots,n$ .

$b_{jk}$  es la cantidad de cubrimiento que aporta la ubicación de  $k$  unidades de emergencia en el punto de servicio  $j$ , para  $j=1,2,\dots,n$  y  $k=1,2,\dots,L$ .

- Diseño de un heurístico para obtener soluciones para grandes instancias de los modelos construidos en un tiempo de ejecución razonable, denotado HGSCP, basado en las siguientes etapas:

a) Obtención de una cota superior inicial mediante un procedimiento heurístico “greedy 1-óptimo”, basada en que las estaciones con un determinado número de vehículos candidatas a entrar en la solución intermedia son aquellas que aportan un mayor cubrimiento efectivo por unidad de coste de la cantidad de cubrimiento demandada por los puntos-demanda que todavía no ha sido completamente satisfecha, complementada con una regla de desempate cuyo objetivo es maximizar el número de restricciones sin satisfacer que pasan a ser satisfechas. Como último criterio de desempate secuencial se recurre a maximizar el cubrimiento adicional aportado por unidad de coste. Una vez construida una solución factible, se procede a un proceso de eliminación de estaciones y vehículos redundantes, terminando con un procedimiento de intercambio 1-óptimo con candidatos no pertenecientes a la solución factible, que con menor coste sigan satisfaciendo las restricciones impuestas.

La experiencia computacional posterior sobre diferentes instancias, donde los costes no incorporan el coste de la estación, sólo el de los vehículos, ha permitido establecer notables diferencias entre el comportamiento de los modelos basados en cubrimiento múltiple ( $b_{jk} = k$  para  $k=1,2,\dots,L$  y la cantidad de cubrimiento a

satisfacer,  $c_i$ , es un cantidad entera) y los modelos basados en cantidad de cubrimiento ( $b_{jk} \in \mathcal{R}$  y  $c_i \in \mathcal{R}$ ). Estas desviaciones en los resultados ha obligado a modificar el heurístico “greedy 1-óptimo” para las instancias basadas en cantidad de cubrimiento. Como primer criterio de selección de estaciones candidatas a entrar en la solución intermedia, se utiliza la maximización del número de restricciones sin satisfacer que pasan a ser satisfechas, complementada con una regla de desempate como la de maximizar la holgura del cubrimiento necesario por unidad de coste. Finalmente, se incorpora de nuevo un proceso de eliminación de estaciones y vehículos redundantes y un procedimiento de intercambio 1-óptimo.

b) Cálculo de una cota inferior inicial mediante un heurístico de ascenso dual, basado en encontrar un conjunto de multiplicadores que maximicen el segundo sumando del valor óptimo de la relajación lagrangiana ( $L_u$ )

$$v(L_u) = \sum_{j=1}^n \text{Min} \left( 0, \text{Min}_{k=1,2,\dots,L} \left\{ w_{jk} - \left[ \sum_{i \in M_j} u_i \right] b_{jk} \right\} \right) + \sum_{i=1}^m u_i c_i,$$

con la restricción de que todos los sumandos del primer término sean mayores o iguales a cero. En el algoritmo presentado, la variable  $u_i$  es elegida entre aquellas que permiten un aumento en todas las restricciones que cubre, seleccionando aquella que aporta mayor valor añadido a la función objetivo.

c) Mejora de la cota inferior mediante un procedimiento subgradiente que comienza con el último vector de multiplicadores lagrangianos calculado,  $\mathbf{u}^0$ , (en la primera aplicación del procedimiento se utiliza la solución obtenida en la etapa b), y genera una secuencia del vector de multiplicadores  $\mathbf{u}^0, \mathbf{u}^1, \dots, \mathbf{u}^k, \dots$ , usando la siguiente fórmula de actualización



$$\mathbf{u}^{k+1} = \max \left\{ \mathbf{u}^k + \alpha_k \frac{(CS^* - CI^*)}{\|\mathbf{c} - \mathbf{A}\mathbf{B}\mathbf{x}_u^k\|^2} (\mathbf{c} - \mathbf{A}\mathbf{B}\mathbf{x}_u^k), \mathbf{0} \right\},$$

donde en cada iteración se resuelve la relajación lagrangiana ( $L_{\mathbf{u}^k}$ ), actualizándose  $CI^*$

si  $v(L_{\mathbf{u}^k}) = \sum_{j=1}^n \text{Min} \left( 0, \text{Min}_{k=1,2,\dots,L} \left\{ w_{jk} - \left[ \sum_{i \in M_j} u_i^k \right] b_{jk} \right\} \right) + \sum_{i=1}^m u_i^k c_i$  es mayor que la mejor cota inferior  $CI^*$  obtenida hasta este momento. Si durante 20 iteraciones la cota inferior no mejora, se actualiza el tamaño del paso,  $\alpha_k \leftarrow \alpha_k/2$ . Por último, el procedimiento se detiene si se cumple alguna de las tres condiciones siguientes: el tamaño del paso es  $\alpha_k < 0.005$ , el número de iteraciones es mayor de 1000 o si durante 300 iteraciones  $v(L_{\mathbf{u}^k})$  no mejora.

d) Mejora de la cota superior mediante el procedimiento heurístico “greedy 1-óptimo” descrito en la etapa a), dependiendo del tipo de instancia resuelta, tomando como solución inicial la solución óptima  $\mathbf{x}_u^k$  de la relajación lagrangiana ( $L_{\mathbf{u}^k}$ ), actualizándose la cota superior cuando se obtiene una solución factible con un coste menor que la actual cota superior.

e) Las fases c) y d) son repetidas hasta que se verifique que el número de veces que se repiten las dos fases de mejora sea mayor que 150 o si la diferencia entre la cota superior e inferior no ha mejorado al menos un 10% respecto a la diferencia existente 10 iteraciones atrás, al detectarse una tasa de convergencia excesivamente lenta.

- Valoración de la eficiencia de la implementación del algoritmo desarrollado comparando su solución con la solución óptima obtenida mediante un proceso de ramificación y acotación. La siguiente tabla presenta diferentes desviaciones porcentuales medias con objeto de evaluar la eficiencia “greedy 1-óptimo” para las

formulaciones de cubrimiento múltiple y del heurístico “greedy 1-óptimo modificado” dentro del HGSCP para las modelizaciones basadas en cantidad de cubrimiento, agrupados para los diferentes modelos.

**Tabla 4.6. Desviaciones porcentuales medias de las soluciones obtenidas por los heurísticos para las diferentes instancias sobre los escenarios de 511 X 210 y 1023 X 330 con una distribución espacial de llamadas para 3 niveles de fiabilidad para cada modelo.**

MODELO	Nº de instancias	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZOPT_i}{ZOPT_i} \cdot 100$	$\frac{1}{n} \sum_i \frac{ZUBF_i - ZOPT_i}{ZOPT_i} \cdot 100$	$\frac{1}{n} \sum_i \frac{ZUBI_i - ZUBF_i}{ZUBF_i} \cdot 100$
BPLSCP	6	7.95	3.17	4.64
QPLSCP	6	7.87	3.90	3.81
PPLSCP	6	7.61	2.38	5.23
PRLSCP	6	11.45	3.67	7.51
QRLSCP	6	6.51	0.00	6.51
BRLSCP	6	11.16	6.88	4.15
TOTAL	36	8.76	3.33	5.31

La desviación porcentual promedio entre la solución obtenida por el “greedy 1-óptimo” en sus dos versiones y la solución óptima dada por CPLEX 4.0 es globalmente de aproximadamente un 9%, y la desviación entre la solución calculada por el HGSCP presentado y la solución óptima es aproximadamente de un 3.5%, resultados que permitirán en un futuro proporcionar buenas soluciones, con tiempos moderados, para grandes instancias de los modelos de cubrimiento total probabilísticos presentados en esta memoria.



## Referencias

- Almiñana M. and Pastor J.T. Two New Heuristics for the Location Set Covering problem. *Top*, 2, 315-328, 1994
- Almiñana M. and Pastor J.T., An adaptation of SH heuristic to the location set covering problem, *European Journal of Operational Research* 100(3), 586-593, 1997
- Almiñana M. and Pastor J.T. A comparison of algorithm RS with algorithm OPTSOL70. *Top*, 5(2), 213-219, 1997
- Balas E. and Ho H. Set Covering Algorithms using cutting planes, heuristics and subgradient optimization: a computational study. *Mathematical Programming*, 12, 37-60, 1980.
- Balas E. and Carrera M.C. A dynamic subgradient-based branch and bound procedure for the set covering. *Operations Research*, 44, 875-890, 1996.
- Balinski D.M. Integer programming: methods, uses, computation. *Management Science*, 12(3), 253-313, 1965.
- Ball M. and Lin F. A reliability model applied to emergency service vehicle location. *Operations Research*, 41, 18-36, 1993.
- Batta R., Dolan J.M. and Krishnamurthy N.N. The Maximal Expected Covering Location Problem: revisited. *Transportation Science*, 23, 277-287, 1989.
- Batta R. and Mannur N.R. Covering–location models for emergency situations that require multiple response units. *Management Science*, 36(1), 16-23, 1990.

- Beasley J.E. An algorithm for the Set Covering Problem. *European Journal of Operational Research*, 31, 85-93, 1987.
- Beasley J.E. A lagrangean heuristic for the Set Covering Problem. *Naval Research Logistic*, 37, 151-164, 1990.
- Beasley J.E. and Jörnsten K. Enhancing an algorithm for Set Covering Problems. *European Journal of Operational Research* 58, 293-300, 1992.
- Beasley J.E. and Chu P.C. A genetic algorithm for the set covering problem. *European Journal of Operational Research* (94)2, 392-404, 1996.
- Benedict J.M. Three hierarchical objective models which incorporate the concepts of excess coverage to locate EMS vehicles or hospitals. M.S. Thesis, Department of Civil Engineering, Northwestern University, Evanston, IL, 1983.
- Berlin G.R. and Liebman J.C. Mathematical analysis of emergency ambulance location. *Socio-Economic Planning Sciences*, 8(6), 323-328, 1974.
- Birge J. and Pollock S. Using parallel iteration for approximate analysis of a multiple server queueing system. *Operations Research*, 37, 769-779, 1989.
- Burwell T.H., Jarvis J.P. and McKnew M.A. Modeling co-located servers and dispatch ties in the hypercube model. *Computers and Operations Research*, 20(2), 113-119, 1993.
- Burwell T.H., McKnew M.A. and Jarvis J.P. An application of a spatially distributed queuing model to an ambulance system. *Socio-Economic Planning Sciences*, 26(4), 289-300, 1992.
- Caprara A., Fischetti M. and Toth P. A heuristic method for the set covering problem. *Operations Research*, 47(5) 730-743, 1999.
- Carnegie Commission on Higher Education. Higher Education and the nation's Health. McGraw-Hill, New York, 1970.

- Ceria S. P., Nobili A. and Sassano A. A lagrangian-based heuristic for large-scale set covering problem. Technical R.406, IASI-CNR, Roma. To appear in *Mathematical Programming*, 1995.
- Chaiken J.M. Transfer of emergency service deployment models to operating agencies. *Management Science*, 24(7), 719- 731, 1978.
- Chapman S. and White J. Probabilistic formulation of emergency service facilities location problems. Paper presented at the ORSA/TIMS Conference, San Juan, Puerto Rico, 1974.
- Church R. and ReVelle C. The Maximal Covering Location Problem. *Papers of the Regional Science Associaton*, 32,101-118, 1974.
- Church R. and Roberts K. L. Generalized coverage models and public facility location. *Papers of the Regional Science Associaton*, 53,117-135, 1983.
- Daskin M.S. A maximum expected covering location problem: formulation, properties and heuristic solution. *Transportation Science*, 17,48-70,1983.
- Daskin M.S. and Stern E.H. A hierarchical objective set covering model for emergency medical service vehicle deployment. *Transportation Science*, 15(2), 137-152, 1981.
- Daskin M.S. Application of an expected covering model to emergency medical service design. *Decision Sciences*, 13(3), 416-439, 1982.
- Daskin M.S., Hogan, K. and ReVelle C. Integration of multiple, excess, backup, and expected covering models. *Environment and Planning B*, 15,15-35,1988.
- Eaton D.J., Daskin M.S., Simmons D., Bulloch B. and Jansma, G. Determining emergency medical service vehicle deployment in Austin, Texas. *Interfaces*, 15(1), 96-108, 1986.
- Eaton D.J., Hector M., Sanchez V., Lantigua R. and Morgan, J. Determining ambulance the deployment in Santo Domingo, Dominican Republic. *Journal of the Operational Research Society*, 37(2), 113-126, 1986.

- Erkut H and Polat S. A simulation model for an urban fire fighting system. *OMEGA: International Journal of Management Science*, 20(4), 535-542, 1992.
- Fisher M.L. The lagrangian relaxation method for solving integer programming problems. *Management Science*, 27, 1-18, 1981.
- Fisher M. L. and Kedia P. Optimal Solutions of Set Covering / Partitioning Problems using dual heuristics. *Management Science*, 36(6), 674-688, 1990.
- Fitzsimmons J.A. A methodology for emergency ambulance deployment. *Management Science*, 19(6), 627- 636, 1973.
- Fujiwara O., Makjamroen T. and Gupta K. K. Ambulance deployment analysis: A case study of Bangkok. *European Journal of Operational Research*, 31, 9-18, 1987.
- Garfinkel R. S. and Newhauser G. L. Integer programming. *Wiley*, NewYork, 1972.
- Goldberg J., Dietrich R., Ming J. and Mitwasi M.G. A simulation model for evaluating a set of emergency vehicle base locations: Development, validation and usage. *Socio-Economic Planning Sciences*, 24(2), 125-141, 1990a.
- Goldberg J., Dietrich R., Ming J. and Mitwasi M.G. Validating and applying a model for locating emergency medical vehicles in Tucson, AZ. *European Journal of Operational Research*, 49, 308-324, 1990b.
- Goldberg J., and Szidararovszky F. Methods for solving equations used in evaluating emergency vehicle busy probabilities. *Operations Research*, 39(6), 903-916, 1991.
- Goldberg J. and Paz L. Locating emergency vehicle bases when service time depends on call location. *Transportation Science*, 25,264-280, 1991.
- Gonsalvez D. J., Hall N.G., Rhee W. T. and Siferd S. P. Heuristic solutions and confidence intervals for the multicovering problem. *European Journal of Operational Research*, 31, 94-101, 1987.
- Hall N.H. and Hochbaum D. S. A fast approximation algorithm for the multicovering problem. *Discrete Applied Mathematics*, 15, 35-40, 1986.

- Hakimi S. L. Optimum distribution of switching centers and the absolute centers and medians of a graph. *Operations Research*, 12, 450-459, 1963.
- Hakimi S. L. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research*, 13, 462-475, 1964.
- Held M., Wolfe P. and Crowder H. Validation of subgradient optimization. *Mathematical Programming*, 6, 62-88, 1970.
- Hogan K. and ReVelle C. Concepts and applications of backup coverage. *Management Science*, 32(11), 1434- 1444, 1986.
- Insurance Services Office. *Grading schedule for municipal fire protection*, Insurance Services Office, New York, 1974
- Jarvis J. P. Optimization in Stochastics Systems with Distinguishables Servers, TR-19-15, *Operations Research Center, MIT*, 1975.
- Jarvis J. P. Approximating the equilibrium behavior of multi-server loss systems. *Management Science*, 31, 235- 239, 1985.
- Karp R. M. Reducibility among combinatorial problems. Muller R.E. and Thatcher J.W. eds., *Complexity of computer*, Plenum Press, New York, 1972.
- Kendall D.G. Stochastic Processes occurring in the theory of queues and their analysis by the method of imbedded markov chains. *Annals Mathematical Statistics*, 24, 338-354, 1953.
- Kroll P. The multiple-depot, multiple-tour and multiple-stop delivery problem. Ph.D. Dissertation, The Johns Hopkins University, Baltimore, MD (1988)
- Larson R. C. A hypercube queuing model for facility location and redistricting in urban emergency services. *Computers Operations Research*, 1, 67-95, 1974.
- Larson R. C. Approximating the performance of Urban Emergency Service Systems. *Operations Research*, 23, 845-868, 1975.



- Lemke C., Salkin H. and Spielberg K. Set covering by single branch and enumeration with Linear Programming Subproblems. *Operations Research*, 19(4), 998-1022, 1971.
- Liu M.S. and Lee J.T. A simulation of a hospital emergency call system using SLAM. *Simulation* 51(6), 216-221, 1988.
- Lorena L. A. N. and Lopes F. B. A surrogate heuristic for set covering problems, *European Journal of Operational Research*, 79(1),138-150, 1994.
- Marianov V. and ReVelle C. The Standard Response Fire Protection Siting Problem. *Information Systems and Operations Research*, 29, 116-129, 1991.
- Marianov V. and ReVelle C. A probabilistic fire-protection siting model with joint vehicle reliability requirements. *Papers in Regional Science: the journal of the RSAI*, 71, 217-241, 1992.
- Marianov V. and ReVelle C. The Queuing Probabilistic Location Set Covering Problem and some extensions. *Socio-Economic Planning Science*, 28, 67-178, 1994.
- Marianov V. and ReVelle C. Chapter 10. Siting emergency services, Zvi Drezner(editor) *Facility location: a survey of applications and methods*, Springer Series in Operations Research, 1995.
- Marianov V. and ReVelle C. The Queueing Maximal Availability Location Problem: A model for the siting of emergency vehicles, *European Journal of Operational Research* 93(1), 110-120, 1996.
- Nobili P. and Sassano A. A separation routine for the set covering polytope. Balas, E. Cornuejols G., Kannan R. Eds. *Integer programming and combinatorial optimization*. Proceedings of the 2<sup>nd</sup> IPCO conference, Carnegie-Mellon University Press, Pittsburgh, PA, 1992.
- Pastor J.T. Fractions in the Location Set Covering Problem: A brief Note. *Environment and Planning B: Planning and Design*, 20, 479-480, 1993.
- Pezzella F. and Faggioli E. Solving large set covering problems for crew scheduling. *Top*, 5(1), 41-59, 1997.

- Poliak B. T. A general method of solving extremum problems. *Soviet Mathematics Doklady*, 8, 593-597, 1967.
- Repede J.F. and Bernardo J.J. Developing and validating a decision support system for locating emergency medical vehicles in Louisville, Kentucky. *European Journal of Operational Research*, 75, 567-581, 1994.
- ReVelle C. and Hogan K. A reliability constrained siting model with local estimates of busy fractions. *Environment and Planning B*, 15, 143-152, 1988.
- ReVelle C. and Hogan K. The Maximum Availability Location Problem. *Transportation Science*, 23, 192-200, 1989a.
- ReVelle C. and Hogan K. The maximum reliability location problem and  $\alpha$ -reliable p-center problem: derivatives of the probabilistic location set covering problem. *Annals of Operations Research*, 18, 155-174, 1989b.
- ReVelle C. Review, extension and prediction in Emergency Service Siting Models. *European Journal of Operational Research*, 40, 58-69, 1989.
- ReVelle C. and Marianov V. A probabilistic FLEET model with individual vehicle reliability requirements. *European Journal of Operational Research*, 53, 93-105, 1991.
- ReVelle C. Facility siting and integer-friendly programming. *European Journal of Operational Research*, 65, 147-158, 1993.
- ReVelle C. and Snyder S. Integrated Fire and Ambulance Siting: A deterministic Model. *Socio-Economic Planning Sciences*, 29(4), 261-271, 1995.
- Roth R. Computer solutions to minimum-cover problems. *Operations Research*, 17, 455-465. 1969.
- Saydam C. and McKnew M. A. A separable programming approach to expected coverage: an application to ambulance location. *Decision Sciences*, 16, 381-398, 1985
- Savas E. Simulation and cost-effectiveness analysis of New York's emergency ambulance service. *Management Science*, 15(12), 608- 627, 1969.

- Schilling D., Elzinga D., Cohon J. and ReVelle C. The Team/FLEET models for simultaneous facility and equipment siting. *Transportation Science*, 13, 163-175, 1979.
- Schilling D., Jayaraman V. and Barkhi R. A review of covering problems in facility location. *Location Science*, 1, 25-55, 1993.
- Serra D. The pq-median problem: location and districting of hierarchical facilities. PhD thesis, Department of Geography and Environmental Engineering. John Hopkins University, 1989.
- Sherali H.D. and Myers P.C. Dual formulations and subgradient optimization strategies for linear programming relaxations of mixed-integer programs. *Discrete Applied Mathematics*, 20, 51-68, 1988.
- Storbeck J.E. Slack, natural slack and location covering. *Socio-Economic Planning Sciences*, 16(3), 99- 105, 1982.
- Storbeck J.E. and Vohra R.V. A simple trade-off model for maximal and multiple coverage. *Geographical Analysis*, 20, 220- 230, 1988.
- Swain R. A decomposition Algorithm for a Class of Facility Location Problems. PhD thesis, Cornell University, Ithaca, N.Y., 1971.
- Swoveland C., Uyeno D., Vertinsky I. and Vickson R. Ambulance location: a probabilistic enumeration approach. *Management Science*, 20(4), 686- 698, 1973.
- Toregas C., Swain R. ReVelle C. and Bergman L. The location of emergency service facilities. *Operations Research*, 19, 1363-1373, 1971.
- Toregas C. and ReVelle C. Optimal location under time or distance constraints. *Papers of the Regional Science Association*, 28, 133-1143, 1972
- Toregas C. and ReVelle C. Binary logic solutions to a class of location problems. *Geographical Analysis*, 5, 145-155, 1973
- Uyeno D. And Seeberg C. A practical methodology for ambulance location. *Simulation*, 43(2), 79-87, 1984.

- Van Slyke R. Covering problems in CCCI Systems, report to the Air Force Office of Scientific Research, 1981.
- Wedelin D. The design of a 0-1 integer optimizer and its application in the Carmen system, *European Journal of Operational Research* 87(3), 722-730, 1995.
- White J. and Case K. On covering problems and the central facility location problem. *Geographical Analysis*, 6(3), 281, 1974.
- Wolsey L. A. Integer programming. *John Wiley & Sons*, 1998.
- Wrightson C. W. Stochastic models for Emergency Service Systems, Operations Research Center, ORC 76-18, University of California, Berkeley, 1976.





## **Apéndice A. Listado de redes utilizadas**

En los diferentes estudios computacionales se han utilizado 2 redes diferentes; una de 55 nodos y una de 79 nodos. Con objeto de que fuesen comparables se realizó una translación de las redes originales a un cuadrante de 36 km cuadrados, y se han utilizado dos distancias máximas  $S$  de 1.5 y 3 kilómetros para definir el cubrimiento de un nodo demanda por una estación de servicio. La duración promedio del servicio de una emergencia ha sido considerada de 45 minutos, y se han analizado 4 escenarios de llamadas por día generados aleatoriamente, denotados por red x.1, x.2, x.3 y x.4, siendo x el número de nodos de la red utilizada.

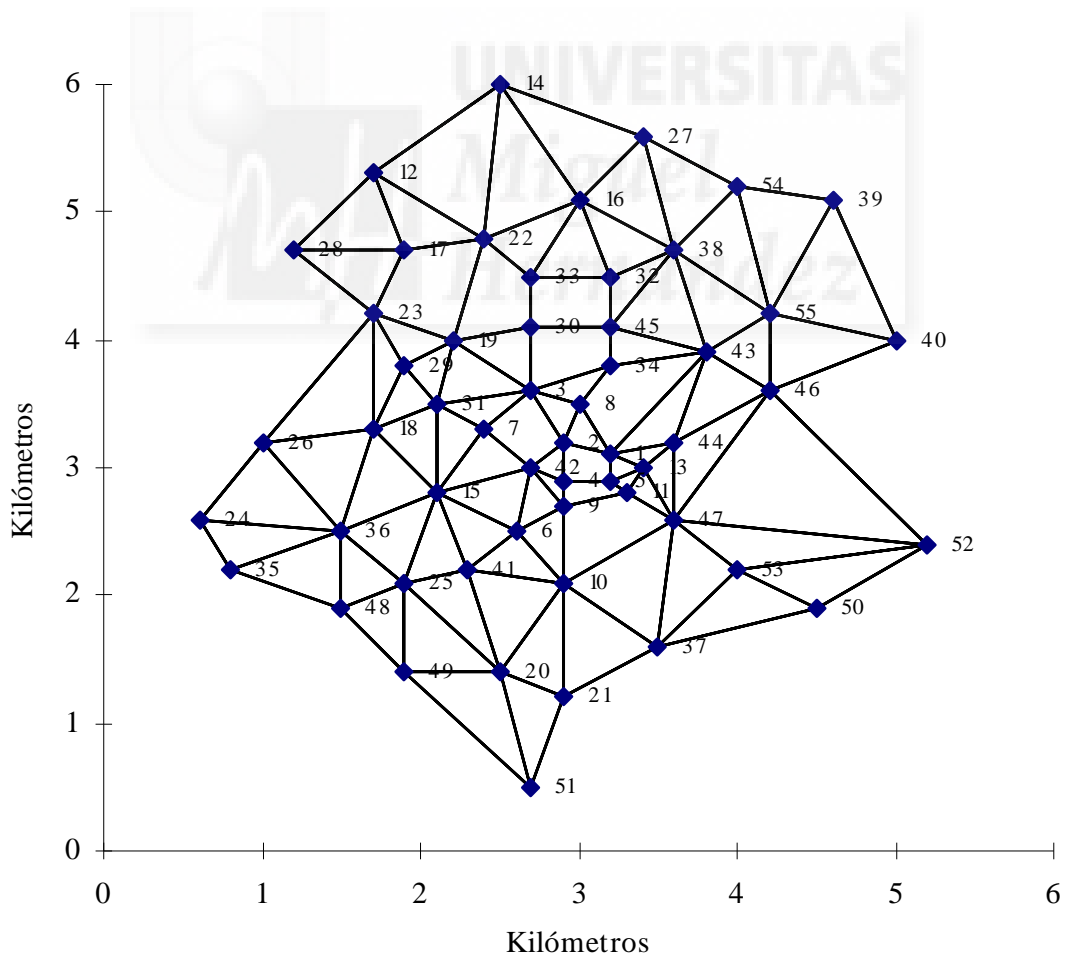
En las páginas siguientes se presentan las coordenadas de los nodos demanda, y el número de llamadas por día para cada escenario, así como una representación gráfica de la red que permite obtener la matriz de adyacencia entre los diferentes nodos de la red y un gráfico de líneas para cada escenario de llamadas distinto.

## a) Red de 55 nodos

<i>Nodo</i>	<i>X</i>	<i>Y</i>	<i>LLlamadas/día red 55.1</i>	<i>Llamadas/día red 55.2</i>	<i>LLlamadas/día red 55.3</i>	<i>LLlamadas/día red 55.4</i>
1	3.2	3.1	2.4406	0.8820	4.4989	0.5795
2	2.9	3.2	2.1313	1.7617	1.3361	5.8359
3	2.7	3.6	1.9250	0.6414	0.8290	1.0922
4	2.9	2.9	1.3406	1.5211	2.6663	6.3483
5	3.2	2.9	1.2031	1.3386	2.1593	1.6046
6	2.6	2.5	0.7219	0.2183	3.9966	6.8610
7	2.4	3.3	0.6875	1.0980	0.8337	2.1173
8	3	3.5	0.6531	1.9777	0.3268	7.3737
9	2.9	2.7	0.5844	0.8574	2.1640	2.6300
10	2.9	2.1	0.5844	0.6747	1.6571	7.8864
11	3.3	2.8	0.5500	1.5544	3.4943	3.1424
12	1.7	5.3	0.5156	0.4341	0.3316	8.3987
13	3.4	3	0.4813	1.3138	4.8245	3.6551
14	2.5	6	0.4125	1.1312	1.6618	4.2230
15	2.1	2.8	0.4125	0.0109	1.1548	9.4794
16	3	5.1	0.3781	0.8907	2.9919	4.7357
17	1.9	4.7	0.3438	1.7704	4.8293	9.9918
18	1.7	3.3	0.3438	0.6501	4.3222	5.2481
19	2.2	4	0.3094	0.4675	1.1595	0.5045
20	2.5	1.4	0.3094	1.3472	0.6525	5.7608
21	2.9	1.2	0.3094	0.2269	2.4898	1.0172
22	2.4	4.8	0.2750	1.1066	4.3269	6.2735
23	1.7	4.2	0.2750	0.9240	3.8200	1.5298
24	0.6	2.6	0.2750	1.8036	0.6572	6.7859
25	1.9	2.1	0.2750	0.6833	0.1503	2.0422
26	1	3.2	0.2406	1.5630	1.9875	7.2986
27	3.4	5.6	0.2063	0.4428	3.8248	2.5549
28	1.2	4.7	0.2063	0.2602	3.3177	3.1229
29	1.9	3.8	0.2063	1.1399	0.1549	8.3792
30	2.7	4.1	0.2063	0.0196	4.6480	3.6353
31	2.1	3.5	0.2063	0.8993	1.4851	8.8916
32	3.2	4.5	0.1719	0.7167	3.3224	4.1479
33	2.7	4.5	0.1719	1.5964	2.8154	9.4043
34	3.2	3.8	0.1719	0.4761	4.6527	4.6606
35	0.8	2.2	0.1719	1.3558	4.1457	9.9170
36	1.5	2.5	0.1719	0.2355	0.9830	5.1730
37	3.5	1.6	0.1719	0.0529	2.8201	0.4294
38	3.6	4.7	0.1375	0.9326	2.3132	5.6857
39	4.6	5.1	0.1375	1.8123	4.1504	0.9421
40	5	4	0.1375	0.6920	3.6433	6.1984
41	2.3	2.2	0.1375	0.5094	0.4807	1.4548
42	2.7	3	0.1375	1.3891	2.3178	2.0227
43	3.8	3.9	0.1375	0.2688	1.8109	7.2787
44	3.6	3.2	0.1375	1.1485	3.6481	2.5351

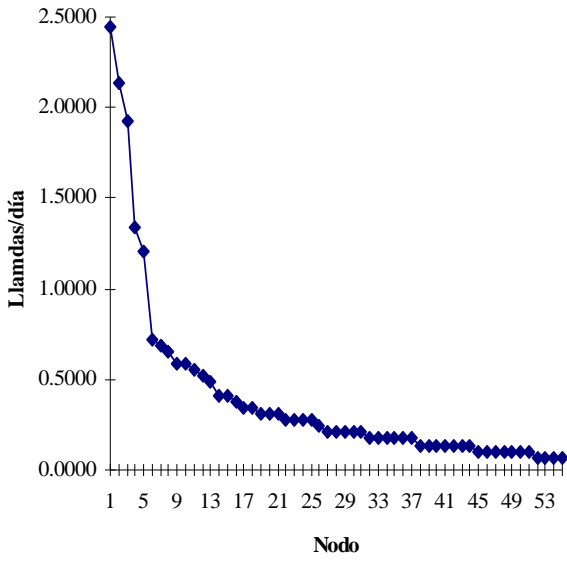
Nodo	X	Y	LLlamadas/día red 55.1	Llamadas/día red 55.2	LLlamadas/día red 55.3	LLlamadas/día red 55.4
45	3.2	4.1	0.1031	0.0282	3.1412	7.7914
46	4.2	3.6	0.1031	1.8456	4.9783	3.0478
47	3.6	2.6	0.1031	0.7253	1.8156	8.3041
48	1.5	1.9	0.1031	1.6050	1.3086	3.5605
49	1.9	1.4	0.1031	0.4847	3.1459	8.8165
50	4.5	1.9	0.1031	0.3021	2.6389	4.0729
51	2.7	0.5	0.1031	1.1818	4.4762	9.3292
52	5.2	2.4	0.0688	0.0615	1.3133	4.5856
53	4	2.2	0.0688	0.9412	0.8063	9.8419
54	4	5.2	0.0688	1.8209	2.6436	5.0983
55	4.2	4.2	0.0688	1.6383	2.1365	0.3543

Red de 55 nodos (Kroll)

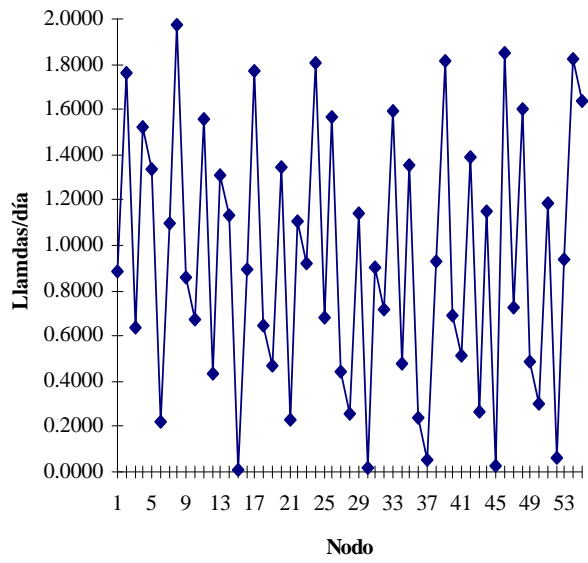




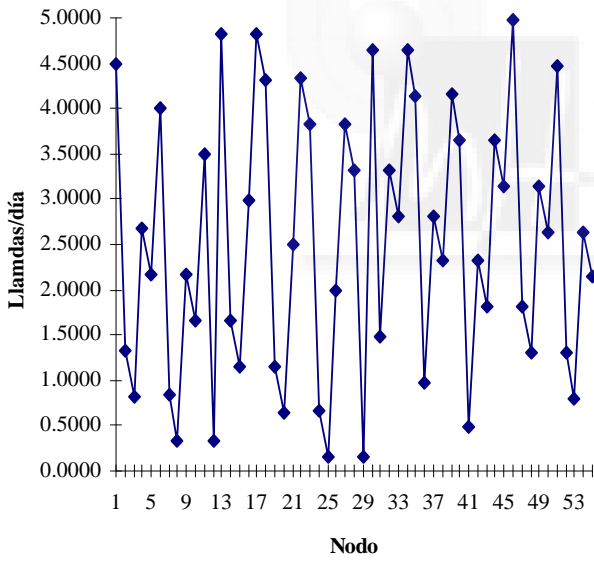
red55.1



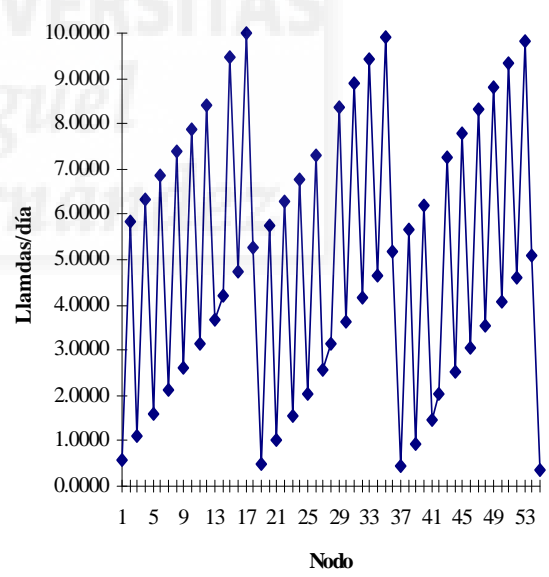
red 55.2



red 55.3



red55.4

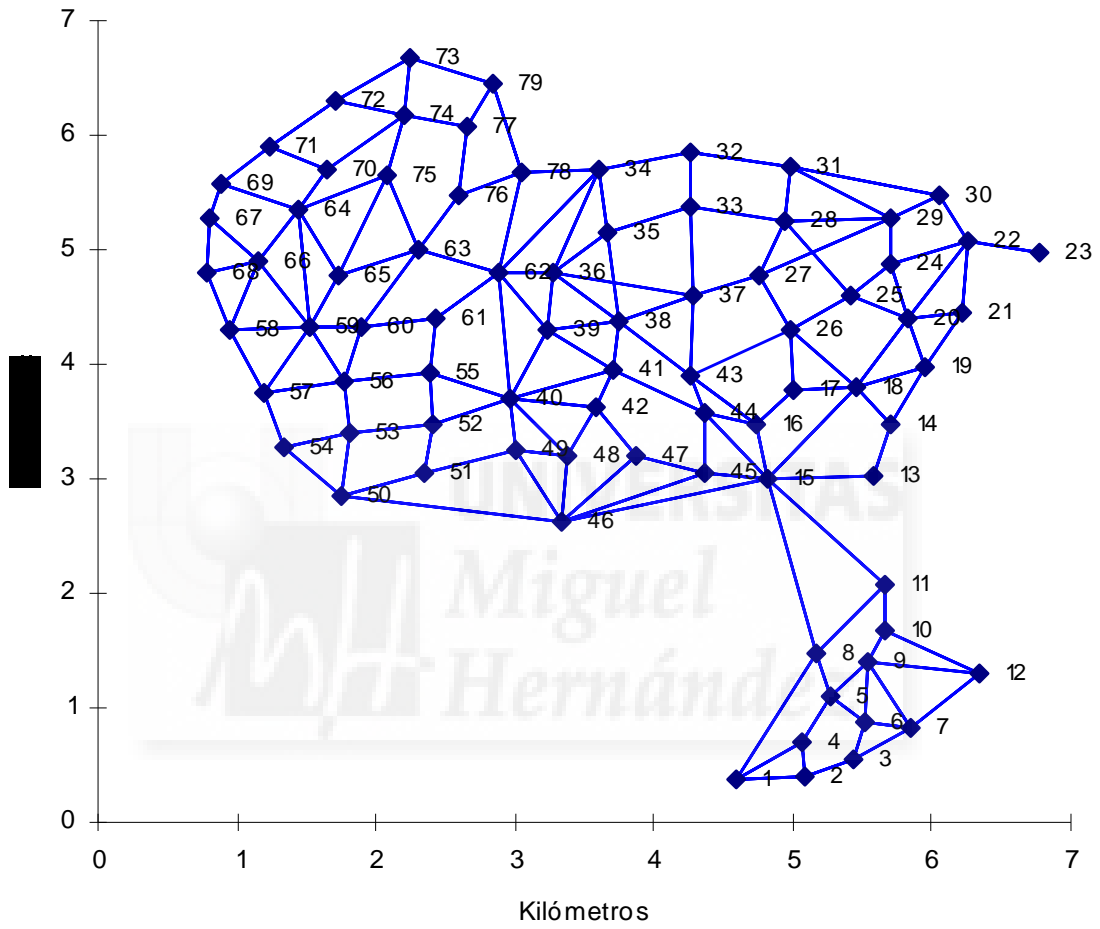


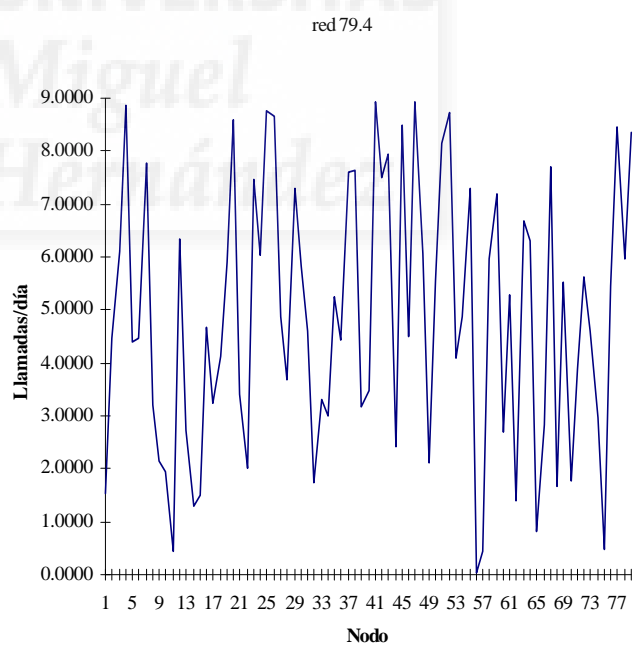
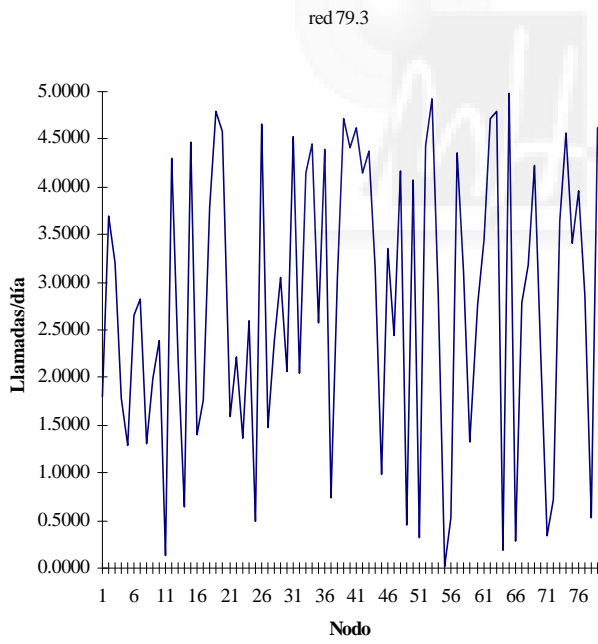
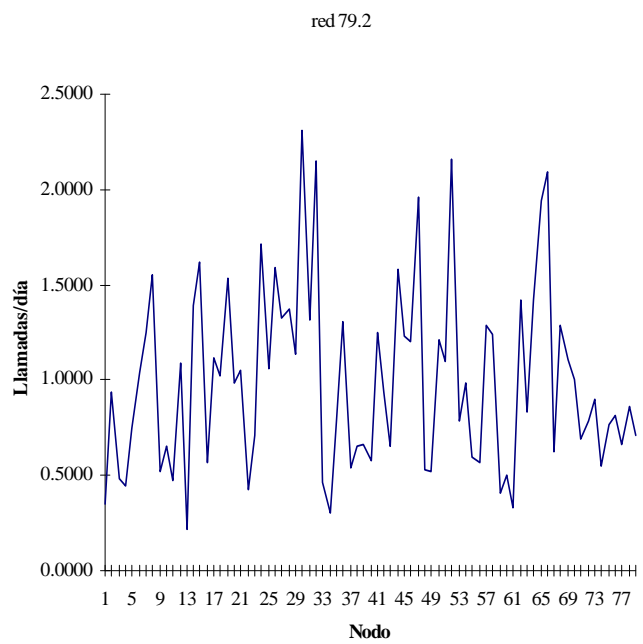
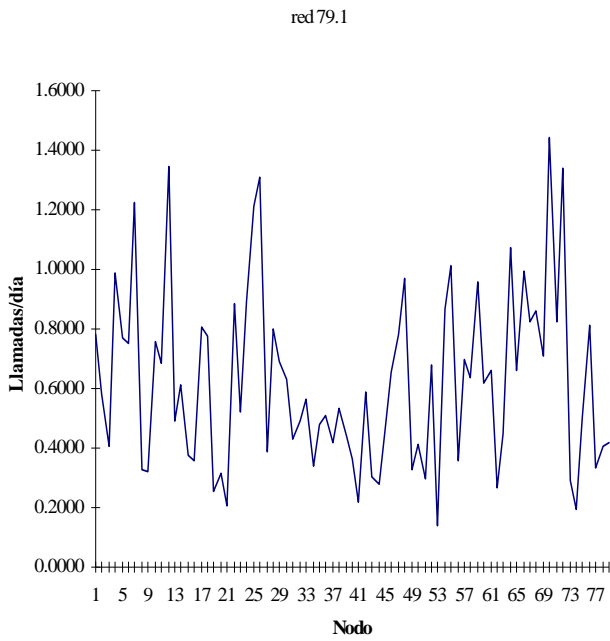
**b) Red de 79 nodos**

<i>Nodo</i>	<i>X</i>	<i>Y</i>	<i>LLlamadas/día red 79.1</i>	<i>LLlamadas/día red 79.2</i>	<i>LLlamadas/día red 79.3</i>	<i>LLlamadas/día red 79.4</i>
1	4.5970	0.3831	0.7815	0.3490	1.8076	1.5215
2	5.0778	0.3961	0.5835	0.9373	3.6877	4.4560
3	5.4329	0.5500	0.4085	0.4811	3.1993	6.0893
4	5.0664	0.7047	0.9860	0.4490	1.7855	8.8610
5	5.2660	1.1116	0.7710	0.7508	1.2918	4.4082
6	5.5091	0.8685	0.7520	1.0446	2.6430	4.4754
7	5.8512	0.8213	1.2225	1.2527	2.8169	7.7685
8	5.1586	1.4720	0.3285	1.5568	1.3151	3.2084
9	5.5457	1.4011	0.3230	0.5219	1.9836	2.1572
10	5.6668	1.6747	0.7575	0.6555	2.3815	1.9316
11	5.6699	2.0687	0.6860	0.4779	0.1373	0.4522
12	6.3336	1.2998	1.3480	1.0902	4.3001	6.3398
13	5.5761	3.0265	0.4915	0.2185	2.1688	2.7409
14	5.7118	3.4829	0.6145	1.3879	0.6393	1.2829
15	4.8187	3.0113	0.3735	1.6201	4.4619	1.4995
16	4.7433	3.4822	0.3570	0.5715	1.4083	4.6720
17	5.0077	3.7717	0.8055	1.1198	1.7552	3.2379
18	5.4535	3.7931	0.7775	1.0181	3.7811	4.1160
19	5.9472	3.9706	0.2535	1.5320	4.7840	5.9115
20	5.8322	4.3928	0.3155	0.9885	4.5856	8.5979
21	6.2216	4.4438	0.2070	1.0558	1.5921	3.3953
22	6.2681	5.0862	0.8875	0.4234	2.2091	1.9974
23	6.7801	4.9735	0.5195	0.7148	1.3595	7.4700
24	5.7026	4.8698	0.8880	1.7129	2.5857	6.0499
25	5.4199	4.6054	1.2130	1.0566	0.4978	8.7569
26	4.9848	4.2990	1.3100	1.5896	4.6579	8.6667
27	4.7600	4.7723	0.3885	1.3223	1.4817	4.8919
28	4.9475	5.2492	0.8020	1.3767	2.3849	3.6918
29	5.7095	5.2752	0.6915	1.1398	3.0456	7.2922
30	6.0486	5.4702	0.6295	2.3092	2.0577	5.8396
31	4.9886	5.7156	0.4325	1.3199	4.5200	4.6113
32	4.2686	5.8528	0.4900	2.1451	2.0546	1.7541
33	4.2526	5.3819	0.5610	0.4634	4.1506	3.3001
34	3.6033	5.6889	0.3405	0.3058	4.4452	3.0071
35	3.6567	5.1410	0.4770	0.7980	2.5765	5.2632
36	3.2749	4.8080	0.5085	1.3031	4.4023	4.4231
37	4.2914	4.5924	0.4160	0.5379	0.7293	7.5939
38	3.7458	4.3760	0.5360	0.6491	3.0500	7.6420
39	3.2239	4.2937	0.4445	0.6676	4.7163	3.1663
40	2.9600	3.7085	0.3615	0.5787	4.4210	3.4757
41	3.7123	3.9447	0.2180	1.2511	4.6231	8.9309
42	3.5866	3.6193	0.5855	0.9341	4.1542	7.5133
43	4.2526	3.8975	0.3005	0.6539	4.3740	7.9434
44	4.3630	3.5668	0.2805	1.5784	3.1473	2.4090
45	4.3615	3.0433	0.4690	1.2342	0.9879	8.5045
46	3.3351	2.6287	0.6525	1.2038	3.3577	4.5085

<i>Nodo</i>	<i>X</i>	<i>Y</i>	<i>LLlamadas/día red 79.1</i>	<i>LLlamadas/día red 79.2</i>	<i>LLlamadas/día red 79.3</i>	<i>LLlamadas/día red 79.4</i>
47	3.8624	3.1987	0.7825	1.9570	2.4420	8.9244
48	3.3732	3.1903	0.9725	0.5259	4.1680	6.0602
49	3.0074	3.2490	0.3260	0.5171	0.4514	2.1246
50	1.7547	2.8421	0.4095	1.2126	4.0682	5.5458
51	2.3529	3.0516	0.2985	1.0982	0.3185	8.1558
52	2.4154	3.4852	0.6810	2.1579	4.4251	8.7134
53	1.8042	3.4007	0.1365	0.7868	4.9213	4.1021
54	1.3333	3.2642	0.8670	0.9837	2.8514	4.8749
55	2.3978	3.9165	1.0120	0.5979	0.0147	7.3109
56	1.7776	3.8434	0.3570	0.5715	0.5273	0.0264
57	1.2015	3.7611	0.6995	1.2895	4.3628	0.4559
58	0.9546	4.2952	0.6360	1.2447	3.1112	5.9691
59	1.5299	4.3356	0.9570	0.4058	1.3249	7.2076
60	1.8911	4.3265	0.6175	0.5051	2.7744	2.6909
61	2.4375	4.4004	0.6595	0.3314	3.4420	5.2894
62	2.8817	4.7974	0.2645	1.4207	4.7168	1.4035
63	2.3003	5.0016	0.4465	0.8316	4.7830	6.6673
64	1.4347	5.3438	1.0700	1.4215	0.1857	6.2901
65	1.7280	4.7806	0.6600	1.9418	4.9852	0.8120
66	1.1550	4.9080	0.9930	2.0971	0.2904	2.8185
67	0.7944	5.2799	0.8260	0.6219	2.7810	7.7023
68	0.7778	4.8050	0.8600	1.2839	3.1813	1.6646
69	0.8868	5.5868	0.7120	1.1070	4.2151	5.5094
70	1.6412	5.7103	1.4425	1.0077	2.2477	1.7707
71	1.2320	5.9122	0.8245	0.6924	0.3502	3.8713
72	1.7029	6.3084	1.3400	0.7844	0.7130	5.6151
73	2.2394	6.6650	0.2895	0.8981	3.6288	4.5904
74	2.1990	6.1751	0.1910	0.5451	4.5695	2.9498
75	2.0839	5.6417	0.4985	0.7636	3.4007	0.4861
76	2.6005	5.4725	0.8140	0.8140	3.9504	5.4569
77	2.6539	6.0730	0.3360	0.6660	2.8632	8.4478
78	3.0471	5.6767	0.4055	0.8581	0.5217	5.9609
79	2.8444	6.4441	0.4170	0.7116	4.6275	8.3427

Red de 79 nodos(Serra)





## **Apéndice B. Listado de programas**

En este apéndice están recogidos los listados de algunos de los algoritmos programados utilizando el lenguaje FORTRAN y C, implementados para realizar los estudios computacionales, cuyos resultados se exponen en los diferentes capítulos de esta memoria. Únicamente vamos a reflejar en este anexo los procedimientos implementados para realizar:

- a) El Simulador del Sistema de Emergencias
- b) La Resolución del sistema de ecuaciones no lineales del Sistema de Emergencia sin hacer mención al resto de herramientas utilizadas, como el heurístico HGSCP para resolver el problema del cubrimiento total generalizado, la resolución exacta del Generalized Set Covering Problem utilizando la librería de funciones del CPLEX, generador aleatorio de redes, etc.

En las páginas siguientes se presentan los listados de los programas con diferentes comentarios para una mejor comprensión y seguimiento de los mismos

### a) El Simulador del Sistema de Emergencias

```

CC      SIMULACION DE UN SISTEMA DE AYUDA MEDICA URGENTE (SAMU)
CC
CC      el fichero de entrada configs tiene la siguiente estructuracc
CC      TSERV 0=CONSTANTE 1=EXPONENCIAL K = ERLANG K
CC      n_configuraciones periodo_de_ref(dias) repetit_periodo num nodos
CC      nombre_configuración
CC      Valor_de_S estaciones_configuracion
CC      n°_estación n°_engines
CC      ..... valores 1| configuración
CC      .....
CC      n°_estación n°_engines
CC      nombre_configuración
CC      Valor_de_S estaciones_configuracion
CC      n°_estación n°_engines
CC      ..... valores 2| configuración
CC      .....
CC      n°_estación n°_engines

CC      llamada(i,1)= tiempo de aparición de la llamada
CC      llamada(i,2)= nodo donde se produce la llamada
CC      llamada(i,3)= tiempo de servicio de la llamada

CC      emergencia(i,1)= tiempo de vuelta del vehículo de emergencia
CC      emergencia(i,2)= estacion a la que pertenecen el vehiculo

CC      a(i,j) matriz de incidencias
CC      ae(i,j) matriz incidencia presentado para cada nodo el número de
CC      engines

CC      servicio(i,1) =número de llamadas desde el nodo i en una repetición
CC      servicio(i,2)=número de llamadas servidas del nodo i en el tiempo S
CC      servicio(i,3)=número de llamadas no servidas en el tiempo S de el
CC      nodo i

CC      servicio(i,4)=numero de llamadas no servidas
CC      servicio(i,5)= fiabilidad ex-post del nodo i
CC      servicio(nd+1,1)=número total de llamadas en una repetición del
CC      periodo de referencia
CC      servicio(nd+1,2)=número total de llamadas servidas en una repetición
CC      del periodo de referencia
CC      servicio(nd+1,3)=número total de llamadas no servidas en el tiempo
CC      en una repetición del periodo de referencia
CC      servicio(nd+1,4)=número total de llamadas no servidas en una
CC      repetición del periodo de referencia
CC      servico(nd+1,5)= fiabilidad ex-post global
CC      pr periodo de referencia
CC      rpr repeticioens periodo de referncia

call principal
end

```

subroutine principal

```

integer ndd,pila
PARAMETER (Ndd=55, pila=16500)
integer nd,stations,tmp,s1,nconfig,vehiculos,tserv,STAT(NDD)
Integer*1 a(ndd,ndd),v(ndd,2),
+ae(ndd,ndd),aex(ndd,ndd),veh(ndd)

Real f(ndd),temp(pila,3),total,minll,foc,alfa,fiabalcaest
Real D(NDD,ndd),S,llamada(pila/2,3),emergencia(pila/2,2),pr,rpr
Real enginesdisp,servicio(ndd+1,5),RJ(NDD,2),MEANRJ(55,25)
REAL SEMEANRJ(55,25),meanexpost(55,ndd+1)
real semeanexpost(55,ndd+1)

Real tm,r,tiempol,tiempo2
character nomf*12,nomi*12,nomsal1*12,
CC +nomconfigur*13,nomsal2*12,NOMSAL3*12,NOMSAL4*12,NOMSAL5*12
+nomconfigur*7,nomsal2*12,NOMSAL3*12,NOMSAL4*12,NOMSAL5*12

CALL GETDAT(IYR,IMON,IDAY)
CALL SETTIM(0,0,0,0)

OPEN(9,FILE='configs')
READ(9,*) TSERV
READ(9,*) nconfig,pr,rpr,nd
close(unit=9)

CC abrir ficheros de lectura y escritura
CC
nomsal1='simuglo.sal'
nomsal2='simuind.sal'
NOMSAL3='SIMUEST.SAL'
NOMSAL4='SIMMEAN.SAL'
NOMSAL5='SIMSE.SAL'
nomi='matriz.dis'
nomf='nodos.frc'
OPEN(13,FILE=nomi)
OPEN(2,FILE=nomf)
OPEN(3,FILE=nomsal1)
OPEN(4,FILE=nomsal2)
OPEN(5,FILE=NOMSAL3)
OPEN(6,FILE=nomsal4)
OPEN(7,FILE=NOMSAL5)

CC lectura FICHERO MATRIZ DE distancias
OPEN (Unit=13,FILE=nomi)
DO I=1,ND
  READ(13,'(10(F10.4,1X))') (D(I,J),J=1,nd)
ENDDO
CLOSE(UNIT=13)

CC ENTRADA FICHERO DEFINICION EPSC
CC 1 FILA tm formato F10.5
CC SIGUIENTES FILAS f(i)
OPEN (Unit=2,FILE=nomf)
total=0
read(2,'(F10.5)') tm
Do i=1,nd
  Read(2,73) f(i)
total=total+f(i)
enddo

```



```

CC      write(*,*) 'total llamadas por dia  por nodo en media=',total/nd
73      Format(F10.5)

CC      crear repeticiones del periodo que se este considerando
do sl=1,rpr
  write(*,*) '      repetición',sl
  OPEN(9,FILE='configs')
  READ(9,*) TSERV
  READ(9,*) nconfig,pr,rpr,nd

CC      inicializar a cero

      do i=1,k2
        do j=1,3
          llamada(i,j)=0
        enddo
      enddo

      k2=0
      call GETTIM(IHR,IMIN,ISEC,I100TH)
      TMP=IHR*3600+IMIN*60+ISEC+i100th
CC      Generar las eventos de llamada
CC      llamadas poisson
CC      (tiempo entre llegadas exponencial de media 1/lambda)

      k=1
      do I=1,nd
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC      simular numero aleatorio en el PC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      call seed(tmp*i*k)
      call random(r)
      do while (r.le.+1.1-20)
        call random(r)
      enddo
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC      simular numero aleatorio en el SUN 3000
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC      call random_number(r)

CC      producir el evento de llamada inicial para el nodo i

      llamada(k,1)=-alog(r)/f(i)
      llamada(k,2)=i

      if (TSERV.EQ.0) THEN
CC TIEMPO DE SERVICIO CONTASTE
      llamada(k,3)=tm/1440
CC      write(*,*) llamada(k,3)
      ELSEif (TSERV.EQ.1) THEN
CC TIEMPO DE SERVICIO EXPONENCIAL
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC      simular numero aleatorio en el PC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
      call random(r)
      do while (r.le.+1.1E-20)
        call random(r)
      enddo
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC      simular numero aleatorio en el SUN 3000
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC      call random_number(r)

      llamada(k,3)=-alog(r)*tm/1440
CC      write(*,*) llamada(k,3)

```



```

CC          call random_number(r)

          llamada(k,3)=-alog(r)*tm/1440
CC          write(*,*) llamada(k,3)
          ELSEIF (TSERV .GE.2 ) THEN
CC  TIEMPO DE SERVICIO ERLANG K (SUMA DE K EXPONENCIALES DE MEDIA 1/KMU)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC          simular numero aleatorio en el PC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          DO ID1=1,TSERV
          call random(r)
          do while (r.le.+1.1E-20)
          call random(r)
          enddo
          llamada(k,3)=llamada(K,3)-alog(r)*tm/(1440*TSERV)
          ENDDO
CC          write(*,*) llamada(k,3)
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC          simular numero aleatorio en el SUN 3000
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
          DO ID1=1,TSERV
          call call random_number(r)
          llamada(k,3)=llamada(K,3)-alog(r)*tm/(1440*TSERV)
          ENDDO
          write(*,*) llamada(k,3)

          endif

          else
          goto 1000
          endif
          enddo
1000      k=k+1
          enddo

          k2=k-1
          write(*,*) 'número de llamadas simulados' ,k2
          escritura de simulacion de llamadas
          do i=1,k2
          write(*, '(3f10.5)') (llamada(i,j),j=1,3)
          enddo
          pause

          iconfig=1
          do while (iconfig.le. nconfig)
CC          READ(9,'(13a)') nomconfigur
CC          read(9,333) s,stations,vehiculos,alfa,FIABALCAest,
CC          + tiempo1,tiempo2
CC          STAT(ICONFIG)=STATIONS
CC          S=S/10
CC333      FORMAT(F5.2,1X,I6,I6,3X,F7.4,1X,F7.4,1X,F20.2,1X,F20.2)
CC          read(9,444) (VEH(j2),j2=1,vehiculos)
CC444      FORMAT(60(1X,I2))
          icon=1
          do I=1,stations
          v(i,1)=veh(icon)
          icon=1
          do j2=icon+1,vehiculos
          if (veh(j2).eq.v(i,1)) then
          icon=icon+1
          endif
          enddo
          v(i,2)=icon
          icon=icon+icon

```

```

CC          enddo

          read(9,333) nomconfigur,s,stations,vehiculos,
+          (v(i,1),v(i,2),i=1,stations)

          STAT(ICONFIG)=STATIONS
          S=S/10
333  FORMAT(a7,1x,f5.2,1X,I2,I9,20(i4,i4))

          engines=0
          do i=1,stations
            RJ(I,1)=V(I,1)
            RJ(I,2)=0
            engines= engines+v(i,2)
          enddo
          write(*,'(13a)') nomconfigur
CC          write(*,*) 'numero total de vehiculos',engines
CC          pause
          do I=1,k1-k2
            do j=1,2
              emergencia(i,j)=0
            enddo
          enddo
          do i=k2+1,k1
            temp(i,1)=0.
            temp(i,2)=0.
            temp(i,3)=0.
          enddo
          do k=1,k2
            temp(k,1)=llamada(k,1)
            temp(k,2)=1
            temp(k,3)=k
          enddo
          k1=k2
          call ordena(1,k2,temp)

CC          do k4=1,k2
CC            write(*,'(3f10.5)') (temp(k4,j),j=1,3)
CC          enddo

          do i=1,nd+1
            do j=1,4
              servicio(i,j)=0.
            enddo
          enddo

          do i=1,nd
            do j=1,stations
              a(i,j)=0
              ae(i,j)=0
              aex(i,j)=0
            enddo
          enddo

CC          crear matriz de cubrimiento de la solución actual
          do i=1,nd
            do j=1,stations
              if (d(i,v(j,1)) .le. s) then
                a(i,j)=1
                ae(i,j)=v(j,2)
                aex(i,j)=ae(i,j)
              endif
            enddo
          enddo

```

```

    enddo
CC    simular entrada de llamadas de emergencia, asignar vehiculo,etc

    illamada=1
    iemer=1
    do while (illamada .le. k1)
        ll=int(temp(illamada,2))
        indice=int(temp(illamada,3))

        if (ll .eq. 0) then
CC            vuelta de vehiculos de emergencia
CC            write(*,*) 'Vuelta emergencia:'
CC            write(*, '(2f8.4,i5)') (emergencia(indice,j),j=1,2),indice
CC            vuelta de engines vehiculos

            do j3=1,stations
                if (int(emergencia(indice,2)) .eq.v(j3,1) )then
                    v(j3,2)=v(j3,2)+1
                    do j=1,nd
                        if (a(j,j3) .eq. 1) then
                            aex(j,j3)=aex(j,j3)+1
                        endif
                    enddo
                endif
            enddo

        else
CC            llamada corresponde a un llamada
CC            write(*,*) 'Configuration llamada:'
CC            write(*, '(3f10.4,2i4),')(llamada(indice,j),j=1,3),indice,illamada

CC            el llamada se produce en el nodo ilocal
            ilocal=int(llamada(indice,2))
CC            write(*,*) 'llama nodo', ilocal
            servicio(ilocal,1)=servicio(ilocal,1)+1
CC            determinar si hay vehiculos para poder atender la llamada
            enginesdisp=0
            do j=1,stations
                enginesdisp=enginesdisp +aex(ilocal,j)
            enddo
CC            write(*,*) 'vehiculos engines disponibles=',enginesdisp
            if( enginesdisp .ge. 1) then
CC                write(*,*) 'llamada servida'
                servicio(ilocal,2)=servicio(ilocal,2)+1
                emergencia(iemer,1)=llamada(indice,1)+llamada(indice,3)
CC            determinar que vehiculos atienden la llamada

CC            determinar que engines y de donde sirven la llamada

            minll=1.0e10
            do j=1,stations
                if (aex(ilocal,j) .ge.1) then
                    minll=min(d(ilocal,v(j,1)),minll)
                    if (minll .eq. d(ilocal,v(j,1))) ele1=j
                endif
            enddo
            emergencia(iemer,2)=v(ele1,1)
CC            write(*,*) 'sirve vehiculo estacion dentro de s',
CC            + emergencia(iemer,2)
CC            pause
            RJ(ELE1,2)=RJ(ELE1,2)+llamada(indice,3)
CC            eliminar dicho engine durante el tiempo de servicio
            v(ele1,2)=v(ele1,2)-1
            do I=1,nd

```

```

        if (a(i,ele1) .eq. 1) then
            aex(i,ele1)=aex(i,ele1)-1
        endif
    enddo

CC      colocar una bandera para saber cuando vuelve el equipo de
CC      emergencia
    temp(k1+1,1)=emergencia(iemer,1)
    temp(k1+1,2)=0
    temp(k1+1,3)=iemer
    iemer=iemer+1
    k1=k1+1
    call ordena(illamada,k1,temp)
else
CC      no existen vehiculos para cubrir la llamada en la distancia standar
CC      guardar que la llamada no se ha servido en el tiempo
CC      write(*,*) 'no se atiende la llamada en el el tiempo requerido'
CC      determinar si hay vehiculos en el sistema para poder atender la
CC      llamada aunque sea fuera del tiempo requerido
    enginesdisp=0
    do j=1,stations
        enginesdisp=enginesdisp+v(j,2)
    enddo
CC      write(*,*) 'vehiculos engines disponibles fuera tiempo='
CC      +      ,enginesdisp

CC      se sirven si tenemos vehiculos disponibles
    if( enginesdisp .ge. 1) then
        servicio(ilocal,3)=servicio(ilocal,3)+1
        emergencia(iemer,1)=llamada(indice,1)+llamada(indice,3)
CC      determinar que vehiculos atienden la llamada

CC      determinar que engines y de donde sirven la llamada
        minll=1.0e10
        do j=1,stations
            if (v(j,2) .ge. 1) then
                minll=min(d(ilocal,v(j,1)),minll)
                if (minll .eq. d(ilocal,v(j,1))) ele1=j
            endif
        enddo
        emergencia(iemer,2)=v(ele1,1)
        RJ(ELLE1,2)=RJ(ELLE1,2)+llamada(indice,3)
CC      write(*,*) 'sirve vehiculo estacion fura de tiempo S'
CC      +      ,emergencia(iemer,2)
CC      pause
CC      eliminar dicho engine durante el tiempo de servicio
        v(ele1,2)=v(ele1,2)-1
        do I=1,nd
            if (a(i,ele1) .eq. 1) then
                aex(i,ele1)=aex(i,ele1)-1
            endif
        enddo

CC      colocar una bandera para saber cuando vuelve el equipo de
CC      emergencia
    temp(k1+1,1)=emergencia(iemer,1)
    temp(k1+1,2)=0
    temp(k1+1,3)=iemer
    iemer=iemer+1
    k1=k1+1
    call ordena(illamada,k1,temp)
else
CC      llamada no servida por carecer de recursos el sistema
        servicio(ilocal,4)=servicio(ilocal,4)+1

```

```

        endif
    endif
endif
illamada=illamada+1
enddo
servicio(nd+1,5)=1
do I=1,nd
    SERVICIO(I,5)=1
    if (servicio(i,1) .ge.1) then
        servicio(i,5)=servicio(i,2)/servicio(i,1)
        meanexpost(iconfig,i)=meanexpost(iconfig,i)+servicio(i,5)
        semeanexpost(iconfig,i)=semeanexpost(iconfig,i)+servicio(i,5)**2
    endif
    servicio(nd+1,1)= servicio(nd+1,1)+servicio(i,1)
    servicio(nd+1,2)= servicio(nd+1,2)+servicio(i,2)
    servicio(nd+1,3)= servicio(nd+1,3)+servicio(i,3)
    servicio(nd+1,4)= servicio(nd+1,4)+servicio(i,4)
    servicio(nd+1,5)=min(servicio(i,5),servicio(nd+1,5))
enddo

meanexpost(iconfig,nd+1)=meanexpost(iconfig,nd+1)+
+ servicio(nd+1,2)/servicio(nd+1,1)
semeanexpost(iconfig,nd+1)=semeanexpost(iconfig,nd+1)+
+ (servicio(nd+1,2)/servicio(nd+1,1))**2

do i=1,stations
    RJ(I,2)=RJ(I,2)/(V(I,2)*PR)
enddo
WRITE(*,'(i4,1x,60(1x,f6.0))') iconfig,(RJ(i,1),i=1,STATIONS)
WRITE(*,'(i4,1x,60(1x,f6.4))') iconfig,(RJ(i,2),i=1,STATIONS)

write(*,'(i4,1x,60(1x,f6.4))') iconfig,(servicio(i,5),i=1,nd)
CC    PAUSE
write(3,'(5f10.4,2x,f8.6,2x,f8.6,2x,f8.6,2x,2x,i4,1x,f5.2,
+1x,i2,1x,i2,f7.4,1x,f7.4,1x,f20.2,1x,f20.2,1x,13a)')
+(servicio(nd+1,j),j=1,5),
+servicio(nd+1,2)/servicio(nd+1,1),
+servicio(nd+1,3)/servicio(nd+1,1),
+servicio(nd+1,4)/servicio(nd+1,1),
+iconfig,s,stations,vehiculos,alfa,fiabalcaest,
+tiempol,tiempo2,nomconfigur

write(4,'(i4,1x,60(1x,f6.4))') iconfig,(servicio(i,5),i=1,nd)
CC    write(5,'(i4,1x,60(1x,f6.4))') iconfig,(RJ(i,2),i=1,STATIONS)

CC    pause
do j=1,stations
meanrj(iconfig,j)=meanrj(iconfig,j)+rj(j,2)
SEMEANRJ(iconfig,j)=Semeanrj(iconfig,j)+rj(j,2)*RJ(J,2)
enddo

iconfig=iconfig+1

enddo
close(unit=9)
enddo

WRITE(*,*) ' SOLUCIÓN DEL SISTEMA DE ECUACIONES '
do il=1,NCONFIG
    write(*,'(i4,1x,60(1x,f6.4))') il,
+ (MEANRJ(il,J)/RPR,J=1,STAT(IL))
    write(*,'(i4,1x,60(1x,f10.8))') il,

```

```

+   ( (SEMEANRJ(I1,J)-(MEANRJ(i1,J)**2)/RPR)/(RPR**2-RPR)
+   ,J=1,STAT(I1))

      write(6,'(i4,1x,60(1x,f6.4))') i1,
+   (MEANexpost(i1,J)/RPR,J=1,nd+1)
      write(7,'(i4,1x,60(1x,f10.8))') i1,
+   ((SEMEANexpost(I1,J)-(MEANexpost(i1,J)**2)/RPR)/(RPR**2-RPR)
+   ,J=1,nd+1)

      ENDDO

      CALL GETDAT(IYR,IMON,IDAY1)
      CALL GETTIM(IHR,IMIN,ISEC,I100TH)
      TIEMPO2=(IDAY1-IDAY)*8640000+IHR*360000+IMIN*6000+ISEC*100+I100TH

      write(*,*) 'TIEMPO EN CENTESIMAS DE SEGUNDO=',TIEMPO2

      return
      end

      SUBROUTINE ordena(nc,NI,AORD)
      integer pila
      PARAMETER (pila=16500)
      INTEGER NI,nc
CC   NI=NUMERO DE INPUTS
CC   A = maTRIZ A DOMINAR
      REAL AORD(pila,3), tEMPARRAY(3)
      do 120 I=nc,NI
          TEMPARRAY(1)=Aord(I,1)
          TEMPARRAY(2)=Aord(I,2)
          TEMPARRAY(3)=Aord(I,3)
          DO 110 J=I,nc+1,-1
              IF ( (AORD(J-1,1) - tEMPARRAY(1)) .gt.0.0000000001 ) THEN
                  AORD(J,1)=AORD(J-1,1)
                  AORD(J,2)=AORD(J-1,2)
                  AORD(J,3)=AORD(J-1,3)
              else
                  exit
              ENDIF
110          continue
          aORD(J,1)=TEMPARRAY(1)
          AORD(J,2)=TEMPARRAY(2)
          AORD(J,3)=TEMPARRAY(3)
120          continue
      return
      end

```



## b) La Resolución del sistema de ecuaciones no lineales del Sistema de Emergencia

/\* Programa para la obtención de la fiabilidad mínima para una configuración de vehículos determinada y para un escenario dado

n= Número de nodos demanda

m= Número de estaciones

a(i,j)= matriz de incidencia

d(j,k)= matriz de distancias

f[i]= vector de llamadas por día

tm = duración promedio del servicio

x1[j]=k número k de vehículos en la estación j

KK =número máximo de vehículos por estación\*/

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <string.h>
```

```
#include <time.h>
```

```
#include <stdlib.h>
```

```
#include <float.h>
```

```
#define NN 201
```

```
#define MM 201
```

```
#define KK 15
```

```
main()
```

```
{
int n,m,a[NN][MM],i5,i6,k,i,j,tipo,x1[MM],j0,k0;
int iconjunts,ip,vehiculos,itest;
int estacion,tiempo,gen_clique_cuts,clique_cuts,cover_cuts;
int nitersim,nnodossim;
double f[NN],tm,d[NN][MM],nivelalfa,p1,p,var1,var2;
double num,den,pmn,lt,sum,p0,rho,minfiab,fiabglob;
char model[8],tros[4];
FILE *filein, *filein1, *filein2, *filein3;
char vial[13],via2[13],via3[13],texto[65];
```

```
/* definición de funcioens usadas en le programa*/
```

```
double q( int vehiculos, double rho, int kint);
```

```
double pmno(int vehiculos, double rho, int kint);
```

```
double factd(int le);
```

```
double max(double dplus, double ier);
```

```
void fiab(int n, int m,int a[][MM],double f[NN],double tm,
```

```
double d[][MM],int x1[MM],double minfiab,
```

```
double fiabglob,int tipo);
```

```
/* lectura de las configuraciones de vehículos y de los escenarios*/
```

```
filein=fopen("calcfiab.dat","r");
```

```
fscanf(filein,"%d\n",&itest);
```

```
fscanf(filein,"%d\n",&iconjunts);
```

```
printf("%d\n",&iconjunts);
```

```
for(i6=0; i6< itest;i6++)
```

```
{
```

```

fscanf(filein,"%13s %13s %13s\n",via3,vial,via2);
printf("%13s %13s %13s\n",via3,vial,via2);

for(i5=0;i5<iconjunts;i5++){

  /* fscanf(filein,"PRO%4d %8s %13s %13s %13s",&ip,model,via3,vial,via2);
  fscanf(filein," %5.3lf %8d %8.0lf %20d %8d %8d %8d %8d %8d ",
    &nivelalfa,&estacion,&vehiculos,&tiempo,&nitersim,&nnodossim,
    &gen_clique_cuts,&clique_cuts,&cover_cuts);*/

  fscanf(filein,"PRO%4d %8s %lf %8d %8d %20d %8d %8d %8d %8d %8d",
    &ip,model,&nivelalfa,&estacion,&vehiculos,&tiempo,&nitersim,
    &nnodossim,&gen_clique_cuts,&clique_cuts,&cover_cuts);

  fscanf(filein,"\n");

  for(j=0;j< estacion;j++)
  {
    fscanf(filein," %4d %4d",&j0,&k0);
    j0=j0-1;
    x1[j0]=k0;
  }

  fscanf(filein,"\n");

  filein1=fopen(via1,"r");
  filein2=fopen(via2,"r");
  filein3=fopen(via3,"r");

  fscanf(filein1,"%3d%3d\n",&n,&m);
  for(i=0;i<=n-1;i++)
  {
    for(j=0;j<=m-1;j++)
    {
      fscanf(filein1,"%ld ", &a[i][j]);
    }
    fscanf(filein1,"\n");
  }
  fclose(filein1);

  fscanf(filein2,"%lf\n",&tm);

  for(i=0;i<=n-1;i++)
  {
    fscanf(filein2,"%lf\n", &f[i]);
  }
  fclose(filein2);

  for(i=0;i<=n-1;i++)
  {
    for(j=0;j<=m-1;j++)
    {
      fscanf(filein3,"%lf ", &d[i][j]);

      if ((j+1)%10 == 0)

```

```

        {
            fscanf(filein3, "\n");
        }

    }

    fscanf(filein3, "\n");
}
fclose(filein3);

printf("PRO%04d %8s", ip, model);
printf(" %5.3lf %3d %3d %15d %8d %8d %8d %8d ",
        nivelalfa, estacion, vehiculos, tiempo, nitersim, mmodossim,
        gen_clique_cuts, clique_cuts, cover_cuts);

fiab(n, m, a, f, tm, d, x1, minfiab, fiabglob, 1);
fiab(n, m, a, f, tm, d, x1, minfiab, fiabglob, 2);

for(j=0; j<=m-1; j++)
{
    if (x1[j] > 0)
    {
        printf(" %3d %3d", j+1, x1[j]);
    }
}
printf("\n");

for(j=0; j<=m-1; j++)
{
    x1[j]=0;
}

}

}

double max(double p, double q )
{
    if(p >= q) return p;
    else return q;
}

void fiab(int n, int m, int a[][MM], double f[NN], double tm,
          double d[][MM], int x1[MM], double minfiab,

```

```

        double fiabglob,int tipo)
    {
        int vehiculos,pref[NN][MM],stations,i,j,ksum,l,iteration0;
        int ia,ipass,iter,i2,k;
        double ord[NN][2],p[NN][MM],r1[MM],carga,error,gamma,tot;
        double r0[MM],r[MM],qlarson[MM],sum,temparray[2];
        double fiabglod,fiabglon,rho,pmn;
        double fiab[NN],fiabn[NN],fiabd[NN];

        double q( int ,double valor,int );
        double pmno(int ,double num,int );
        stations=0;
        vehiculos=0;
        for(j=0;j<=m-1;j++)
        {
            r0[j]=0;
            r1[j]=0;
            r[j]=0;
            if (x1[j] >= 0.1)
            {
                stations=stations+1;
                vehiculos=vehiculos+x1[j];
            }
        }

        for(i=0;i<=n-1;i++)
        {
            for(j=0;j<=m-1;j++)
            {
                ord[j][1]=0;
                ord[j][2]=0;
                p[i][j]=0;
            }
            l=0;
            for(j=0;j<=m-1;j++)
            {
                if (x1[j] >=0.1)
                {
                    ord[l][1]=d[i][j];
                    ord[l][2]=j;
                    l=l+1;
                }
            }
        }

        /* ordena los scores*/

        for (ipass=0; ipass<=stations-1;ipass++)

            for (ia=0;ia<=stations-2;ia++)
                if (ord[ia][1] >ord[ia+1][1] )
                    {
                        temparray[1]=ord[ia][1];
                        temparray[2]=ord[ia][2];
                        ord[ia][1]=ord[ia+1][1];
                        ord[ia][2]=ord[ia+1][2];
                        ord[ia+1][1]=temparray[1];
                        ord[ia+1][2]=temparray[2];
                    }

        /* termina la ordenación*/

        for(j=0;j<=stations-1;j++)
        {

```

```

        pref[i][j]=ord[j][2];
    }
}

carga=0;
for(i=0;i<=n-1;i++)
{
    carga=carga+f[i];
}

/*
    for(i=0;i<=n-1;i++)
    {
        printf(" %d ", i+1);
        for(j=0;j<=stations-1;j++)
        {
            printf(" %5d ",pref[i][j]+1);
        }
        printf("\n");

        printf(" %d ", i+1);
        for(j=0;j<=stations-1;j++)
        {
            printf(" %5.3lf ",d[i][pref[i][j]]);
        }
        printf("\n");
    }
*/

/* resolución con la iteración con r igual al vector 0*/

carga=(carga*tm)/(1440*vehiculos);
rho=carga;
iter=0;

for(j=0;j<=vehiculos-1;j++)
{
    qlarson[j]=0;
    if (tipo==1) qlarson[j]=1;
    if (tipo==2) qlarson[j]=q(vehiculos,rho,j);
}

for(j=0;j<=m-1;j++)
{
    if (x1[j] !=0)
    {
        r0[j]=0;
    }
}

iteration0=1;
error=1.0;

while( error >0.00001)
{
    for(j=0;j<=m-1;j++)
    {
        if (x1[j] !=0)
        {
            for(i=0;i<=n-1;i++)
            {
                ksum=0;
                for(k=0;k<=stations-1;k++)
                {

```

```

        ksum=ksum+x1[pref[i][k]];
        if (pref[i][k] == j)
        {
            ksum=ksum-x1[j];
            p[i][j]=0;
            for(i2=0;i2<=x1[j]-1;i2++)
            {
                p[i][j]=p[i][j]+pow(r0[j],i2)*qlarson[ksum+i2];
            }
            for(l=0;l<=k-1;l++)
            {
                p[i][j]=p[i][j]*pow(r0[pref[i][l]],x1[pref[i][l]]);
            }
        }
    }
}

for(j=0;j<=m-1;j++)
{
    if (x1[j]!=0)
    {
        r1[j]=(tm/(1440*x1[j]));
        sum=0;
        for(i=0;i<=n-1;i++)
        {
            sum=sum+f[i]*p[i][j];
        }
        r1[j]=r1[j]*sum;
        r1[j]=r1[j]/(1+r1[j]);
    }
}

if (tipo==2 )
{
/*
    proceso de normalización de las rj*/

    gamma=0;
    for(j=0;j<=m-1;j++)
    {
        if (x1[j]!=0)
        {
            gamma=gamma+r1[j]*x1[j];
        }
    }
    gamma=1/gamma;
    gamma=gamma*vehiculos*rho*(1-pmno(vehiculos,rho,vehiculos));
    for(j=0;j<=m-1;j++)
    {
        if (x1[j]!=0)
        {
            r1[j]=gamma*r1[j];
        }
    }

/*
    fin proceso de normalizacion de las rj*/
}

error=0;

for(j=0;j<=m-1;j++)
{
    error=error+fabs(r1[j]-r0[j]);
}

```

```

    }
    iteration0=iteration0+1;
    for(j=0;j<=m-1;j++)
    {
        if (x1[j] !=0)
        {
            r0[j]=r1[j];
        }
    }
}

/*    printf("\ncarga de los vehiculos\n");
for(j=0;j<=m-1;j++)
{
    if (x1[j]!=0)
    {
        printf(" %d    %6.4lf\n",j+1, r1[j]);
    }
}
printf("\n");*/

for(j=0;j<=m-1;j++)
{
    if (x1[j]!=0)
    {
        r[j]=r1[j];
    }
}

/*cc calculo de las probabilidades p[i][j]*/
for(j=0;j<=m-1;j++)
{
    if (x1[j] !=0)
    {
        for(i=0;i<=n-1;i++)
        {
            ksum=0;
            for(k=0;k<=stations-1;k++)
            {
                ksum=ksum+x1[pref[i][k]];
                if (pref[i][k] == j)
                {
                    ksum=ksum-x1[j];
                    p[i][j]=0;
                    for(i2=0;i2<=x1[j]-1;i2++)
                    {
                        p[i][j]=p[i][j]+pow(r[j],i2)*qlarson[ksum+i2];
                    }
                    p[i][j]=p[i][j]*(1-r[j]);
                    for(l=0;l<=k-1;l++)
                    {
                        p[i][j]=p[i][j]*pow(r[pref[i][l]],x1[pref[i][l]]);
                    }
                }
            }
        }
    }
}
}
}
}

```

```

        if (tipo==2)
        {
/*cc proceso de normalización de las pij (primero de larson)*/
pmn=1-pmno(vehiculos,rho,vehiculos);
for(i=0;i<=n-1;i++)
{
    tot=0;
    for(j=0;j<=m-1;j++)
    {
        if (x1[j] !=0)
        {
            tot=tot+p[i][j];
        }
    }

    for(j=0;j<=m-1;j++)
    {
        if (x1[j] !=0)
        {
            p[i][j]=p[i][j]*pmn/tot;
        }
    }
}

/* fin proceso de normalización de las pij*/
}

/* impresion calculo de las pij */
/*
for(i=0;i<=n-1;i++)
{
    printf(" %d ", i+1);
    for(j=0;j<=stations-1;j++)
    {
        printf(" %5.3lf ",p[i][pref[i][j]]);
    }
    printf("\n");
    printf(" %d ", i+1);
    for(j=0;j<=stations-1;j++)
    {
        printf(" %5d ",pref[i][j]+1);
    }
    printf("\n");
} */

/*calculo de las fiabilidades minima y global*/

fiabglod=0;
for(i=0;i<=n-1;i++)
{
    fiabglod=fiabglod+f[i];
}

fiabglon=0;
for(i=0;i<=n-1;i++)
{
    for(j=0;j<=m-1;j++)
    {
        if (x1[j] !=0)
        {
            fiabglon=fiabglon+f[i]*p[i][j]*a[i][j];
        }
    }
}

```



```
}
fiabglob=fiabglon/fiabglod;

if (fiabglob >1) fiabglob=0.999999999999;

minfiab=1;
for(i=0;i<=n-1;i++)
{
    fiabn[i]=0;
    fiabd[i]=0;
    for(j=0;j<=m-1;j++)
    {
        if (x1[j]!=0 )
        {
            fiabn[i]=fiabn[i]+p[i][j]*a[i][j];
        }
    }
    fiab[i]=fiabn[i];
    if (fiab[i] < minfiab) minfiab=fiab[i];
}

printf(" %8.6lf ",minfiab);
printf(" %8.6lf ",fiabglob);
}
```

