

# Implementación de una hiperheurística para la reducción de conflictos en el bloque quirúrgico

Marina I. Ramos-Martínez, Martín Carpio, Martín Soria-Alcaraz  
y Héctor José Puga-Soberanes

División de Estudios de Posgrado e Investigación, Instituto Tecnológico de León,  
León, Guanajuato, México  
marramos85@gmail.com, jmcarpio61@hotmail.com,  
soajorgea@gmail.com, pugahector@yahoo.com

**Resumen** A la fecha en los hospitales se requiere aprovechar los recursos del bloque quirúrgico, administrando de una forma más eficiente las salas para reducir el tiempo que el paciente tiene que esperar para ser atendido. Los procedimientos por lo general son organizados en papel y hojas de cálculo, lo cual contribuye a no hacer un uso eficiente del quirófano. Debido a esto, se convierte en una necesidad el desarrollo de una herramienta que permita distribuir las cirugías en el bloque quirúrgico de forma óptima. Con la implementación de un Árbol de Decisión para satisfacer las restricciones del bloque quirúrgico y una Hiperheurística la cual tiene como órgano rector un algoritmo memético para generar la combinación de tres heurísticas, se obtiene una solución que permite la optimización del bloque quirúrgico para un conjunto de instancias de prueba simuladas con distribución uniforme.

**Palabras clave:** Satisfacción de restricciones, árbol de decisión, algoritmo memético, hiperheurística.

## 1. Introducción

Gran parte de los problemas de logística hospitalaria son análogos a varios de los problemas que surgen en la industria y que son analizados bajo el enfoque de la optimización de recursos [1]. Por optimización se entiende al acto de obtener el mejor resultado posible dadas ciertas circunstancias [2]. La optimización tiene una enorme utilidad en muchas ramas del conocimiento. En la actualidad, no existe ningún método de optimización para resolver todo tipo de problemas, por lo que a lo largo de los años se han desarrollado diversos métodos para resolver distintas clases de problemas [3].

En los problemas donde el dominio (valores que definen al problema) es típicamente finito, puede parecer trivial encontrar una solución, ya que existe un proceso para determinar la solución óptima buscada, que es realizar una exploración exhaustiva del conjunto de soluciones. En teoría, este método puede llegar a la solución óptima, sin embargo puede llegar a tardar años o nunca encontrar la solución en tiempo razonable. Los algoritmos de tipo heurístico han sido empleados por su capacidad de proporcionar soluciones en tiempos razonables.

Con la implementación de algunas técnicas de Soft Computing, se creó una herramienta que permite obtener una Programación Quirúrgica inicial a partir de instancias de prueba simuladas con distribución uniforme. Las instancias de prueba manejan 3 prioridades para garantizar la selección de pacientes con mayor urgencia,

donde 1 es urgente, 2 moderado y 3 no urgente. La herramienta generada consta de una Hiperheurística que tiene como órgano rector un algoritmo memético y como base un Árbol de Decisión. Dicha herramienta genera la combinación de heurísticas de bajo nivel para maximizar el número de cirugías.

El uso de una herramienta para el control de pacientes en el bloque quirúrgico tiene un alto impacto en la eficiencia de las operaciones, principalmente en el ahorro de costos y la mejora del nivel de servicio ofrecido a los pacientes. El mayor desafío para el área quirúrgica es incrementar la eficiencia del hospital, de los recursos y la reducción en las listas de espera [4].

## 2. Marco Teórico

### 2.1. Satisfacción de Restricciones

La Satisfacción de Restricciones (CSP), es la operación de encontrar valores para las variables para satisfacer un conjunto de restricciones [5]. El modelo CSP permite representar de manera natural, y resolver de manera eficiente, numerosos problemas de Informática en general y de Inteligencia Artificial en particular [6]. La CSP trata con problemas de dominios finitos. La resolución de un CSP consta de dos fases diferentes [1], [7]:

- Modelar el problema como un problema de satisfacción de restricciones. La modelización expresa el problema mediante una sintaxis de CSPs, es decir, mediante un conjunto de variables, dominios y restricciones del CSP.
- Procesar el problema de satisfacción de restricciones resultante. Una vez formulado el problema como un CSP, hay dos maneras de procesar las restricciones:
  - 1) Técnicas de consistencia. Se basan en la eliminación de valores inconsistentes de los dominios de las variables.
  - 2) Algoritmos de búsqueda. Se basan en la exploración sistemática del espacio de soluciones hasta encontrar una solución o probar que no existe tal solución.

Un problema CSP es una terna  $(X, D, C)$  donde  $X$  es un conjunto de  $n$  variables  $\{X_1, \dots, X_n\}$ .

$D = \langle D_1, \dots, D_n \rangle$  es un vector de dominios. La  $i$ -ésima componente  $D_i$  es el dominio que contiene todos los posibles valores que se le pueden asignar a la variable  $X_i$ .

$C$  es un conjunto finito de restricciones. Cada restricción  $n$ -aria ( $C_n$ ) está definida sobre un conjunto de variables  $\{X_1, \dots, X_n\}$  restringiendo los valores que las variables pueden simultáneamente tomar.

Una solución a un CSP es una asignación de valores a todas las variables de forma que se satisfagan todas las restricciones [8]. Es decir, una solución es una tupla consistente que contiene todas las variables del problema. Por tanto un problema es consistente, si existe al menos una solución, es decir una tupla consistente  $(a_1, a_2, \dots, a_n)$ .

Los objetivos que deseamos obtener de un CSP se centran en encontrar:

- Una solución, sin preferencia alguna.
- Todas las soluciones.

- Una óptima, o al menos una buena solución dando alguna función objetivo definida en términos de algunas o todas las variables.

Existen dos tipos de restricciones [6]:

- Restricciones Duras. Son condiciones de obligatorio cumplimiento.
- Restricciones Blandas. Son restricciones que en realidad denotan preferencias del usuario y se desea que se cumplan en la medida de lo posible.

En esta investigación se implementó un algoritmo de búsqueda como lo es el Árbol de Decisión para la satisfacción de restricciones.

## 2.2. Árbol de decisión

Los Árboles de Decisión (AD) han sido usados desde hace muchos años en diversas disciplinas, como estadística, ingeniería, la teoría de decisión, etc. En todas esas áreas de aplicación, los árboles de decisión han sido utilizados para hacer exploración de datos para cubrir alguno de los siguientes objetivos: para representar el conjunto de observaciones de una manera más concisas y eficiente, para descubrir si el conjunto de ejemplos contiene agrupaciones de objetos claramente separables, para descubrir una función entre las variables independientes o atributos, entre otros [1], [9].

Un AD construye diagramas basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para dar solución a un problema dado. Los valores que pueden entrar o salir del árbol pueden ser discretos o continuos. Los árboles de decisión permiten extraer y representar reglas de clasificación que subyacen a un conjunto de ejemplos observados mediante una estructura jerárquica secuencial que particiona recursivamente el conjunto de ejemplos. En un AD cada camino desde la raíz hasta las hojas representa una regla conjuntiva de clasificación. Cada nodo contiene una pregunta sobre un determinado atributo, el arco hacia el nodo siguiente un valor posible para este atributo y la hoja final una predicción sobre la clase que debe asignarse a los ejemplos que cumplan las condiciones establecidas en el camino [1], [9].

Los AD permiten satisfacer las restricciones [7], en este caso las restricciones en el problema de la Programación Quirúrgica, sin embargo necesita de métodos metaheurísticos para lograr la optimización del bloque quirúrgico.

## 2.3. Algoritmos meméticos

Los algoritmos meméticos (MA), son metaheurísticas basadas en población. Esto quiere decir que mantienen un conjunto de soluciones candidatas para el problema considerado [10]. Los orígenes de los MA se remontan a finales de los años ochenta y deben su nombre al concepto de “meme” introducido por R. Dawkins [11]. La idea básica de los MA es combinar conceptos y estrategias de diferentes metaheurísticas para aunar las ventajas de las mismas. El principio de operación de esta metaheurística es el siguiente: “De la misma forma que en una población se transmiten los genes de los padres a los hijos, los memes se transmiten de cerebro a cerebro de la población” [11]. Desde un punto de vista algorítmico, es común considerar al MA como Algoritmos Genéticos al que se añade un procedimiento de búsqueda local.

Los procedimientos de búsqueda local son sencillos métodos de mejora, en los que en cada iteración se sustituye la solución por una solución vecina mejor [12]. Más concretamente en este caso, sea  $S$  una solución, definimos el vecindario de  $S$ ,  $N(S)$ , como sigue:

Es decir, el conjunto de soluciones que se obtiene intercambiando un elemento que está fuera de  $S$  con otro que está dentro. Nuestro procedimiento de búsqueda local se describe en el algoritmo 2.1:

---

Algoritmo 2.1 Algoritmo memético

---

```

1 Leer solución inicial  $S$ 
2 Repetir
3 Hacer  $valor\_ant=f(S)$ 
4   Buscar  $f(S^*)= \max \{f(S')/ S' \in N(S)\}$ 
5   Si  $f(S^*) > f(S)$  entonces hacer  $S=S^*$ 
6   Hasta  $f(S^*) \leq valor\_ant$ 

```

---

La idea central de los MA subyace a mejoras individuales de las soluciones en cada uno de los agentes junto con procesos de cooperación y competiciones de tipo poblacional [13].

## 2.4. Hiperheurística

El proceso de investigación muestra que es posible la implementación de Hiperheurísticas (HH) para tratar de obtener la optimización del bloque quirúrgico. Las HH han surgido como parte de un nuevo enfoque. El término HH fue introducido por Cowling, en donde las define de la manera siguiente:

Enfoques que funcionan a un nivel más alto de abstracción que las heurísticas y gestionan la elección de lo que debería ser un método heurístico de bajo nivel para ser aplicado en un momento dado, dependiendo de las características de la región del espacio de soluciones en exploración [14].

A partir de esta definición, un algoritmo hiperheurístico básico se muestra en 2.2:

---

Algoritmo 2.2 Hiperheurística

---

```

/* Conjunto de heurísticas */
1  $H = \{h_1, h_2, \dots, h_n\}$ 
2  $i \leftarrow 0$ 
3 while (Criterio de terminación no satisfecho) do
4   Seleccionar la heurística  $h_j$  más adecuada, donde  $j \in \{1, 2, \dots, |H|\}$ 
5   Transformar el estado actual  $S_i$  al estado  $S_{i+1}$  aplicando  $h_j$ 
6    $i \leftarrow i + 1$ 

```

---

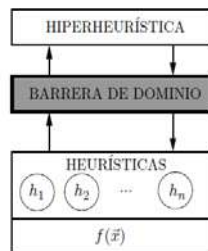
La idea básica que reside detrás de las HH consiste en utilizar un conjunto de métodos heurísticos que actúan sobre un problema determinado (heurísticas de bajo

nivel), de tal forma que en cada momento la HH decida cuál es el mejor de ellos para resolver el problema. En otras palabras, las HH operan en el espacio de las heurísticas mientras que las heurísticas trabajan directamente sobre el espacio de las soluciones del problema [15].

La HH no es la encargada de buscar la mejor solución al problema. Lo que hace, es seleccionar en cada paso del proceso de búsqueda las heurísticas de bajo nivel más prometedoras que sean potencialmente capaces de mejorar la solución actual. Por otra parte, si no hay mejora, la HH debe ser capaz de diversificar la búsqueda a otras regiones mediante la selección de otras heurísticas, buscando de esta manera poder escapar de óptimos locales.

Las HH trabajan a un nivel más alto de abstracción y comúnmente no incluyen información alguna sobre el dominio del problema, por lo que son desarrollados como métodos de optimización general y de esta manera pueden aplicarse a una gama más amplia de problemas. Idealmente, una HH sólo requiere como parámetros el conocer el número de heurísticas de bajo nivel así como el tipo de función objetivo a optimizar (mín  $f(x)$  o máx  $f(x)$ ) [15].

Las HH pueden ser descritas como estrategias diseñadas para controlar la aplicación de un conjunto de heurísticas durante el proceso de búsqueda y suelen verse como cajas negras (Figura 1) [16].



**Figura 1.** Marco de una HH tomado de [15].

Actualmente existen diversos mecanismos de selección diseñados para las HH, los cuales, según Chakhlevitch y Cowling [15], se pueden clasificar en cuatro grandes grupos:

1. Hiperheurísticas basadas en selección aleatoria
2. Hiperheurísticas codiciosas
3. Hiperheurísticas basadas en metaheurísticas
4. Hiperheurísticas con aprendizaje

Se proponen 3 características del enfoque hiperheurístico [17][18] que aparecen implícitas en la mayoría de las aplicaciones, mas no son definidas explícitamente, éstas características se enuncian a continuación:

a) Repetitividad de la implementación de la solución otorgada: el resultado de una HH es un algoritmo que soluciona  $n$  instancias de un problema dado, es necesario entonces que la solución HH posea una estructura que sea fácilmente aplicable a diversas instancias y resolver con un rendimiento adecuado.

b) Dirección en la búsqueda de solución: el espacio de búsqueda del enfoque hiperheurístico al manejar una representación no directa podrá ser menor que el espacio de búsqueda original del problema, aunque el espacio de búsqueda nunca es en realidad disminuido, el proceso de evaluación si se acelera al dirigir la búsqueda.

c) Adquisición de experiencia por parte del investigador: la respuesta es un algoritmo de solución, este algoritmo definido como el conjunto de heurísticas que resuelven un problema, puede ser analizado por el investigador, denotando pues elementos claves que resuelven su problema, aumentando así su experiencia personal, a diferencia de un enfoque metaheurístico donde se resuelve el problema sin la información del cómo se resolvió.

### **3. Implementación del árbol de decisión y la hiperheurística**

Se implementó una heurística para la organización de las cirugías, la cual consiste en agendar la cirugía programada en el día y turno del médico especialista correspondiente, asignado la primera cirugía independientemente del tiempo que dure en la sala 1, la segunda cirugía en la sala 2 y posteriormente la tercera cirugía en la sala 3. La finalidad de la heurística es distribuir las cirugías en las 3 salas de tal forma que no queden salas vacías. Como restricción de programación se van sumando los tiempos por sala, día y turno, si al sumar la nueva cirugía excede en más de 360 minutos está no puede ser agendada en esa sala por lo que busca si puede ser programada en la siguiente sala y así sucesivamente. En caso de no encontrar una sala disponible ese día busca en el siguiente día que opere el médico especialista, en caso de no poder ser asignada en otro día, se crea una nueva semana y se sigue el mismo procedimiento. El siguiente algoritmo 3.1 [1] muestra como se programan las cirugías:

Esta heurística permite la satisfacción de las restricciones. Para la optimización del bloque quirúrgico se implementó una HH.

La parte medular del enfoque hiperheurístico es el conjunto de heurísticas usadas a fin de crear un algoritmo de solución, se eligieron 3 heurísticas para ser implementadas en la HH, dichas heurísticas son [1]:

- Heurística 1 Orden Ascendente: Esta heurística consiste en ordenar la instancia de prueba de acuerdo al tiempo de las cirugías, seleccionar las cirugías una por una de menor a mayor tiempo para ser programada de ser posible en alguna de las salas.
- Heurística 2 Orden Descendente: Esta heurística consiste en ordenar la instancia de prueba de acuerdo al tiempo de las cirugías, seleccionar las cirugías una por una de mayor a menor para ser programada de ser posible en alguna de las salas.
- Heurística 3 Ordenar Mediante la Mediana: Esta heurística consiste en ordenar la instancia de prueba de acuerdo al tiempo de las cirugías, seleccionar las cirugías una por una de acuerdo a la mediana para ser programada de ser posible en alguna de las salas.
- La HH consiste en generar una mezcla de las 3 heurísticas anteriores, de tal forma que en cada iteración se busque la combinación de estas que genere mejores resultados, es decir, que dé como resultado un mayor número de cirugías programadas por semana. La HH consta de 2 fases, una de entrenamiento y una de aplicación. En el entrenamiento se obtiene la combinación de heurísticas de bajo nivel que de como resultado un mayor número de cirugías, mientras que en la fase de aplicación se genera la Programación Quirúrgica a partir de la solución

---

Algoritmo 3.1 AD para la programación quirúrgica

---

```
1 Mientras existan nuevos registros
2 Inicializar población
   Día de operación del médico especialista
   Tiempo promedio de la cirugía
3 Si médico especialista opera día (lunes, martes, miércoles, jueves, viernes, sábado o
   domingo)
4 Tiempo de nueva cirugía + tiempo acumulado sala 1 < 360
5 Nueva cirugía <-- sala 1
6 Si no
7 Tiempo de nueva cirugía + tiempo acumulado sala 2 < 360
8 Nueva cirugía <-- sala 2
9 Si no
10 Tiempo de nueva cirugía + tiempo acumulado sala 3 < 360
11 Nueva cirugía <-- sala 3
12 Si no
13 Buscar otro día que opere el médico especialista regresar al paso 2
14 Si no
15 Crear nueva semana regresar al paso 2
16 Fin mientras
```

---

encontrada en el entrenamiento, esto independientemente que las instancias de prueba no sean las mismas que con las que se entreno (barrera de dominio).

- La HH implementada se basa en el MA, donde primero se genera la población inicial para evaluar la función. La población es inicializada con  $n$  valores aleatorios entre 1 y 3 (genes), donde cada gen representa una heurística de bajo nivel. La variable  $n$  representa el tamaño de la instancia de prueba.
- La población es evaluada para calcular el fitness (cantidad de cirugías programadas), los fitness son ordenados y el mayor resultado representa la primera solución. Para formar la nueva generación, el padre y la madre son seleccionados mediante una ruleta para realizar la cruce y formar a los dos descendientes. Los dos hijos son mutados con cierta probabilidad, es decir, se cambia un gen al azar del individuo. Posteriormente se evalúan, el que tenga el mayor fitness es seleccionado para formar parte de la nueva generación (figura 2).

La figura 2 muestra el funcionamiento interno de la HH.

#### 4. Pruebas y resultados

Para evaluar la solución generada por la HH (es decir la combinación de heurísticas de bajo nivel que permitiera obtener un mayor número de cirugías programadas), se

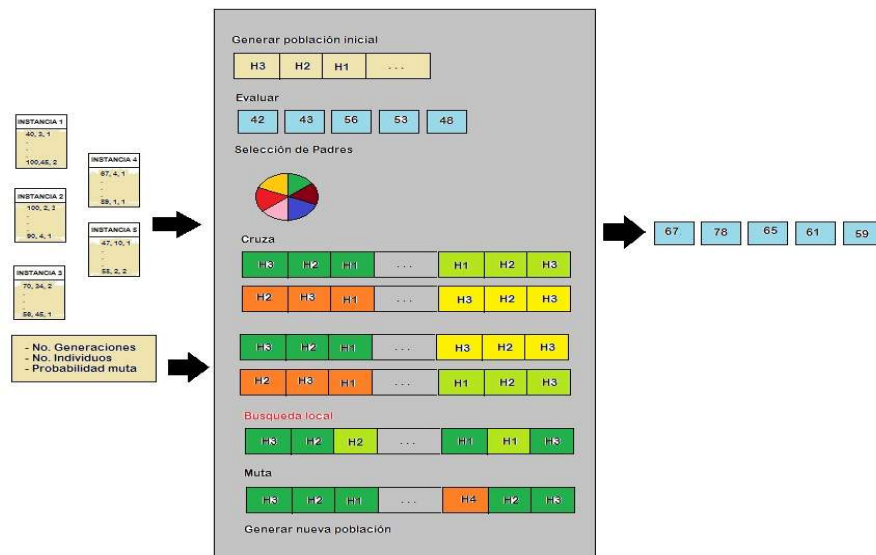


Figure 2. Diagrama conceptual de la HH con MA como órgano rector.

generaron 10 instancias de prueba con distribución uniforme, las cuales cuentan con 150 registros cada una. Cada registro tiene 3 valores, el primer valor es el tiempo estimado de la cirugía que va desde 60 a 360 minutos, el segundo la clave del médico especialista y por ultimo el grado de prioridad 1, 2 ó 3. Se realizaron un total de 10 pruebas, donde en cada prueba se generó una solución a partir de seleccionar 5 instancias aleatorias y correr el MA en 20,000 generaciones (Entrenamiento). Posteriormente se aplicó la solución generada por la HH en el entrenamiento a las 5 instancias de prueba restantes seleccionadas en forma aleatoria uniforme (Aplicación). Como se puede observar en la tabla 1, la cantidad de cirugías programadas es muy cercana a la cantidad máxima encontrada por la mejor heurística de bajo nivel. En la figura 3 se muestra una gráfica comparativa de los resultados obtenidos para las 10 instancias en cada una de las pruebas, donde la serie 11 es el mejor valor encontrado por las heurísticas de bajo nivel.

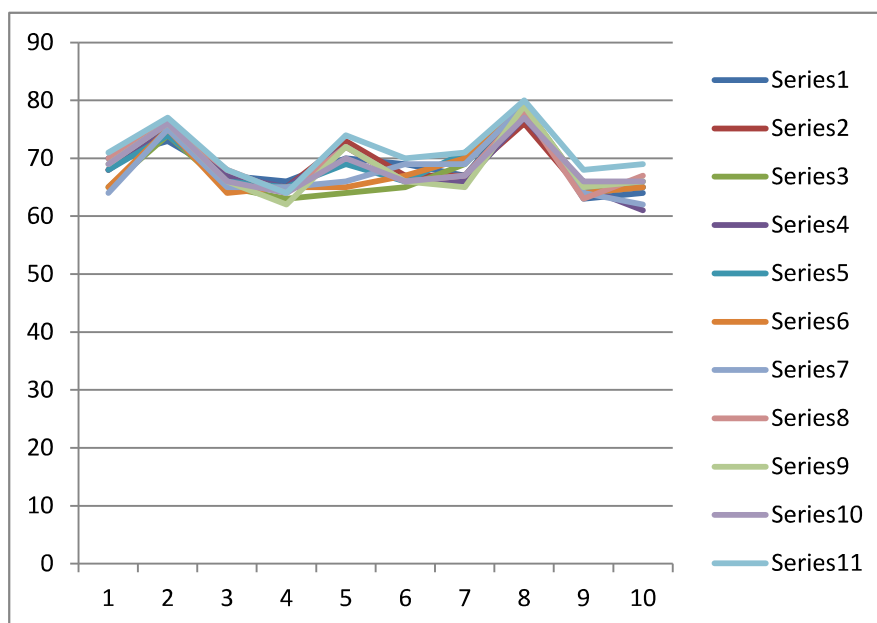
## 5. Conclusiones

La optimización de la Programación Quirúrgica es un problema que a la fecha tienen algunos hospitales. Para el caso de estudio analizado, aun es un gran problema, ya que actualmente realizan la programación manualmente con la ayuda de herramientas de cálculo. Esto genera inconsistencia en los datos, pérdida de información e incluso un mal aprovechamiento del bloque quirúrgico.

La HH logró generar la combinación de heurísticas de bajo nivel que permitió la selección del mayor número de cirugías tomando en cuenta toda la lista destinada a programar. La herramienta diseñada permite encontrar una solución en tiempo real, sin embargo por ser un método heurístico no garantiza encontrar la solución óptima.

**Tabla 1.** Pruebas simuladas

Prub	Valor Encontrado por las Heurísticas										Valor Encontrado por la HH										Entrena	Aplica	Dif1	Dif2
	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10				
1	71	77	68	64	74	70	71	80	68	69	70	73	67	66	70	69	67	79	63	64	4 6 2 1 8	5 10 7 9 3	5	-1
2	71	77	68	64	74	70	71	80	68	69	68	75	65	65	73	67	67	76	64	65	2 1 9 5 4	10 7 3 8 6	9	2
3	71	77	68	64	74	70	71	80	68	69	65	74	65	63	64	65	69	79	65	66	10 2 9 8 7	3 4 5 6 1	12	8
4	71	77	68	64	74	70	71	80	68	69	68	75	67	65	72	66	66	77	65	61	4 9 5 2 3	10 1 7 6 8	7	7
5	71	77	68	64	74	70	71	80	68	69	68	74	66	65	69	66	71	78	64	65	1 7 4 9 2	3 6 10 8 5	9	-1
6	71	77	68	64	74	70	71	80	68	69	65	75	64	65	65	67	70	79	64	65	2 4 9 7 8	1 3 10 5 6	7	7
7	71	77	68	64	74	70	71	80	68	69	64	75	65	65	66	69	69	79	64	62	7 2 8 4 6	10 1 5 3 9	5	8
8	71	77	68	64	74	70	71	80	68	69	70	76	68	64	72	66	67	78	63	67	2 5 3 10 1	8 6 7 9 4	6	1
9	71	77	68	64	74	70	71	80	68	69	69	76	66	62	72	66	65	79	65	66	9 10 8 5 2	1 7 6 4 3	10	1
10	71	77	68	64	74	70	71	80	68	69	69	76	66	64	70	66	67	77	66	66	5 9 8 10 2	7 3 6 4 1	13	-1



**Figura 3.** Distribución de las cirugías.

Como trabajo futuro se propone el uso de esta herramienta para generar una agenda dinámica que permita tener un mayor control sobre los procedimientos quirúrgicos, dando como resultado un mayor aprovechamiento del bloque quirúrgico. Dicha herramienta puede ser ligada a un servidor para poder ser monitoreada desde cualquier dispositivo móvil ó sistema de cómputo para que el especialista esté informado en

tiempo real de lo que ocurre en el bloque quirúrgico y sepa la hora en que le toca operar al siguiente paciente.

**Agradecimientos.** Al Instituto Tecnológico de León y a CONACyT por el apoyo brindado para el desarrollo de la investigación tecnológica y científica.

## Referencias

1. Marina I Ramos: Implementación de Heurísticas Para la Reducción de Conflictos en el Bloque Quirúrgico. Instituto Tecnológico de León. IX Encuentro Participación de la Mujer en la Ciencia, León Guanajuato (2012)
2. Rao, S. S.: Engineering Optimization, Theory and Practice. Third ed. John Wiley & Sons, Inc. (1996)
3. Köppen M.: No-Free-Lunch Theorems and the Diversity of Algorithms. Pascalstr 8-9, 10587 Berlin, Germany (2004)
4. C. Núñez: La información del bloque quirúrgico, clave de sanidad para el siglo XXI. TIC en sanidad, No. 248, pág. 432-433 (2008)
5. Alfonso, María: Un modelo de integración de técnicas de Clausura y CSP de restricciones temporales: Aplicación a problemas de Scheduling. Tesis Médico especialista, Universidad de Alicante (2001)
6. Larrosa Javier: Restricciones Blandas: Modelos y Algoritmos. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial. No. 20 (2003)
7. Barber, Federico: Introducción a la Programación de Restricciones. Departamento de Sistemas Informáticos y Computación. Universidad Politécnica de Valencia, España (2004)
8. Mányá Felip: Técnicas de Resolución de Problemas de Satisfacción de Restricciones. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 7 número 019. España (2003)
9. Joaquim Llisterri Boix: Tratamiento del lenguaje natural: tecnología de la lengua oral y escrita. Fundación Duques de Soria (2002)
10. Cotta, Carlos: Una Visión General de los MA. Dept. Lenguajes y Ciencias de la Computación. ETSI Informática, University of Málaga, España (2004)
11. L. Meléndez, Valadés: Metaheurísticas. Editorial DYKINSON, Universidad Rey, Juan Carlos, Madrid (2007)
12. Bonrostro, Joaquín: Algoritmo Memético para Selección de Variables en el Análisis Discriminante. Departamento de Economía Aplicada. Universidad de Burgos (2004)
13. Moscato, Pablo: Una Introducción a los MA. School of electrical Engineering and Computer Science. University of Newcastle Callaghan, Australia (2002)
14. Cowling, P., Kendall, G., and Soubeiga, E.: A hyperheuristic approach to scheduling a sales summit. In: Proceedings of the Third Metaheuristic International Conference (MIC 2001), Porto, Portugal, vol. 2079, Springer-Verlag, pp. 127–131 (2000)
15. Chakhlevitch, K., and Cowling, P.: Hyperheuristics: Recent Developments. Vol. 136 of Studies in Computational Intelligence. Springer Berlin, pp. 3–29 (2008)
16. Villela Tinoco, Carlos: Mecanismo de selección y control de una hiperheurística basada en Evolución Diferencial para optimización en espacios restringidos.
17. Soria Alcaraz, Jorge Alberto: Diseño de horarios con respecto al alumno mediante técnicas de cómputo evolutivo. Instituto Tecnológico de León (2010)
18. Soria-Alcaraz Jorge A, Carpio Martin, Terashima-Marin Hugo.: Academic Timetabling Design Using Hyper-Heuristics. Soft Computing for Intelligent Control and Mobile Robotics. Studies in Computational Intelligence, Springer Berlin /Heidelberg, Vol 318, 43-56 (2012)