

1001000101001010010101011000111010 ISSN Nº 1390 - 3802

1010011001000101010100010100100010101001010110100101

001 101

101 010

010 001

010 010

010 010

001 101

101 010

010 010

010 010

100 100

110 010

011 100

100 100

010 011

010 110

100 100

101 011

110 010

010 101

001 100

101 000

111 111

010 110

010 111

111 000

101 100

100 101

001 101

001 001

010 110

010 010

010 001

010 010

001 010

010 101

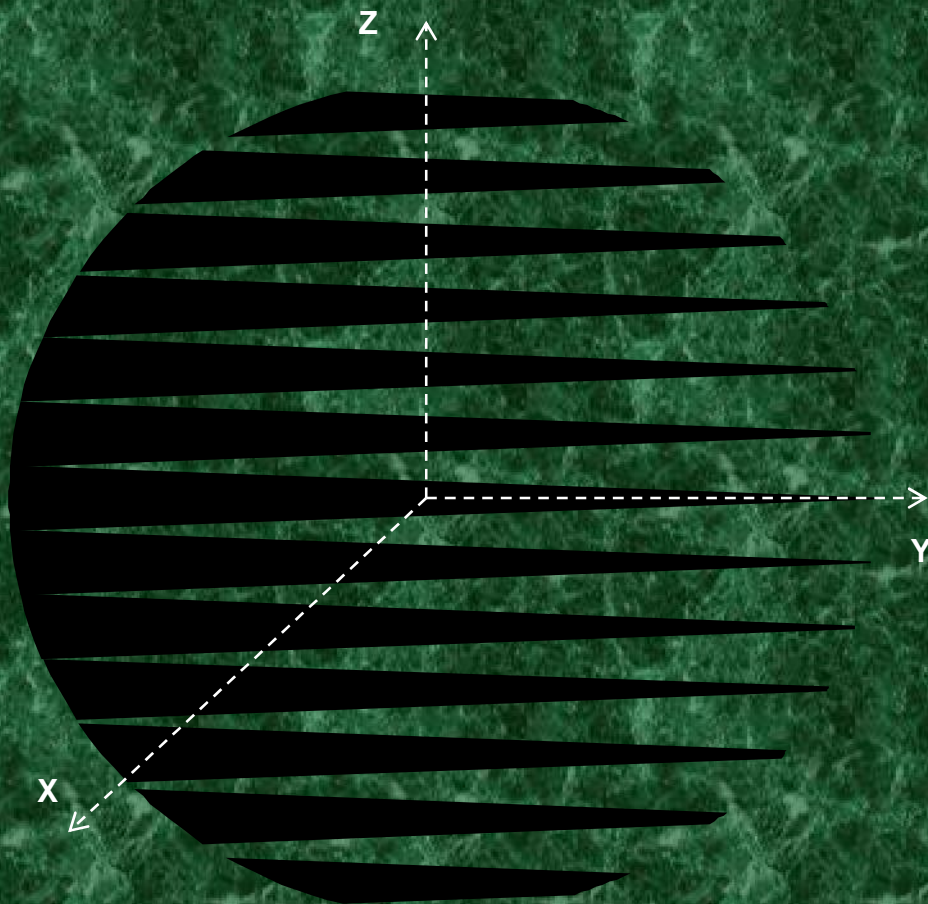
00101010110010001010010100101011000111010100010100

1010011001000101010100010100100010101001010110100101

matemática

UNA PUBLICACIÓN DE FCNM - ESPOL

Volumen 11 Número 2 Octubre 2013



Escuela Superior Politécnica del Litoral - ESPOL
Facultad de Ciencias Naturales y Matemáticas - FCNM

FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICAS

DEPARTAMENTO DE MATEMÁTICAS

El Departamento de Matemáticas (DM) es una unidad académica de la ESPOL. Desde el inicio la función del DM ha sido la docencia en Matemáticas, Ciencias Gráficas e Informática, para la formación de profesionales en ingeniería, tecnología y ciencias, habiendo tenido a su cargo en los albores de la ESPOL, el dictado de 10 materias. Con el transcurso del tiempo y acorde con la era de la información, el Departamento de Matemáticas creó en mayo de 1995 la carrera de “Ingeniería en Estadística Informática”, como alternativa en ingeniería en información y servicios. Posteriormente, con el fin de garantizar la eficiencia en el control y gestión empresarial con profesionales capacitados y de excelencia se creó la carrera de “Auditoría y Control de Gestión” en mayo de 2000. También el Instituto ha incursionado en una de las más importantes ramas de la matemática aplicada que tiene grandes aplicaciones en el mundo moderno, esto es la Investigación de Operaciones, la Teoría de Optimización, y particularmente las aplicaciones logísticas, a través del ofrecimiento de programas de pre-grado y post-grado en estas áreas. Así es como desde el año 2005 se viene ofreciendo la maestría en Control de Operaciones y Gestión Logística y desde el año 2006 la carrera de Ingeniería en Logística y Transporte.

El DM también cuenta con el CENTRO DE INVESTIGACIONES ESTADÍSTICAS, a través del cual, se realizan: estudios de predicción, estudios actuariales, estudios de mercado, diseños de experimentos, planificación y dirección de censos, análisis financieros, bases de datos estadísticos, formulación de proyectos, ingeniería de la calidad, etc.

Entre otras actividades que desarrolla el DM anualmente están: las JORNADAS EN ESTADÍSTICA E INFORMÁTICA que actualmente está en su decimoséptima versión, el CONCURSO INTERCOLEGIAL DE MATEMÁTICAS que se viene realizando en forma continúa desde 1988.



Más información: www.icm.espol.edu.ec o escribirnos al e-mail: icm@espol.edu.ec, jvaldi@espol.edu.ec,
erivaden@espol.edu.ec, 30 ½ vía Perimetral: Edificios 25 – B Planta alta
Telfs.: (593-4) 2269525 – 2269526, fax: (593-4) 853138.
Guayaquil – Ecuador

matemática

UNA PUBLICACIÓN DE LA FCNM – ESPOL

Volumen 11

Número 2

Octubre 2013

Rector ESPOL:

M.Sc. Sergio Flores

Vicerrectora General ESPOL:

Ph.D. Cecilia Paredes

Decano FCNM:

M.Sc. Gaudencio Zurita Herrera

Subdecano FCNM:

M.Sc. Oswaldo Valle Sánchez

**Directora Departamento de
Matemáticas:**

M.Sc. Janet Valdiviezo

Director Departamento de Física:

M.Sc. Hernando Sánchez

Director Departamento de

Química:

Ph.D. David Matamoros

Editor de publicaciones:

M.Sc. Eduardo Rivadeneira Molina

Comité Editorial:

M.Sc. Efrén Jaramillo Carrión

Ph.D. David Matamoros

M.Sc. Luis Rodríguez Ojeda

Ph.D. Francisco Vera

Asesores Editoriales:

Ph.D. Joseph Páez Chávez

Mae. Sandra García Bustos

M.Sc. Olga González Sánchez

M.Sc. Justo Huayamave Navarrete

Mg. Eva María Mera Intriago

Mg. María Nela Pastuizaca

M.Sc. Fernando Sandoya Sánchez

M.Sc. Francisco Torres Andrade

Ing. Antonio Chong Escobar

Edición:

Srta. Carolina Carrasco Salas



matemática es una publicación del Departamento de Matemáticas de la Facultad de Ciencias Naturales y Matemáticas de la Escuela Superior Politécnica del Litoral, y pretende constituirse en un órgano de difusión científico – tecnológico, con el fin de incentivar y motivar el desarrollo y avance de la matemática y sus aplicaciones.

matemática publica artículos teóricos y de tipo experimental tales como ensayos, resúmenes de tesis de grado y trabajos de investigación relacionados con la aplicación de la matemática en los diferentes ámbitos de la realidad.

CONTENIDO

EDITORIAL.....	5
DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO GRASP PARA EL PROBLEMA DE COLORACIÓN DE GRAFOS Delgado Erwin.....	7
IMPLEMENTACIÓN DE UN ALGORITMO GRASP PARA EL PROBLEMA DE COLORACIÓN DE GRAFOS APLICADO A LA CALENDARIZACIÓN DE EXAMENES EN UNA INSTITUCIÓN EDUCATIVA Delgado Erwin.....	15
ATENUACIÓN DE RIZADO EN LA DENSIDAD ESPECTRAL DE POTENCIA CALCULADA EN UNA SEÑAL DE RITMO CARDIACO González Javier, Forero Edwin, Jiménez Fabián, Mariño Iván.....	22
CONSTRUCCIÓN DE KERNELS Y FUNCIONES DE DENSIDAD DE PROBABILIDAD Rodríguez Luís.....	27
ALGORITMOS FACTIBLES, PROBLEMAS TRATABLES Y LA COMPLEJIDAD COMPUTACIONAL DE UNA VARIANTE DEL PROBLEMA DE LA DIVERSIDAD MÁXIMA Sandoya Fernando.....	41

EDITORIAL

Como es evidente a lo largo de la historia, las universidades han sido pilares fundamentales en los procesos de desarrollo recientes, especialmente en aquellos observados en los países del sudeste asiático. En términos generales, se pueden resaltar dos ingredientes fundamentales: facilitar el acceso a la universidad por parte de los sectores de la sociedad tradicionalmente excluidos y proveer una educación superior de excelencia. En el caso particular de nuestro país, el primer punto ha sido incorporado a través de la gratuidad de la educación hasta el tercer nivel, teniendo como único filtro el Examen Nacional de Educación Superior (ENES), el cual evalúa las capacidades numéricas, verbales y de razonamiento abstracto del aspirante. De esta forma se establece un acceso democrático a la educación superior, la cual es sostenida financieramente por toda la sociedad.

Sin embargo, todavía queda camino por recorrer en cuanto a los altos estándares planteados para el quehacer universitario, lo cual pasa por dar el salto cualitativo de ser reproductores y difusores de conocimiento a ser creadores del mismo. Sin duda alguna que para lograr aquello se requiere de ingentes recursos económicos así como también la convicción de que esto se logrará únicamente con el esfuerzo conjunto de todos los actores que formamos parte del sistema de educación superior.

En este sentido, la revista *Matemática* ofrece un espacio de difusión y discusión científica como punto de encuentro de diversas disciplinas de la oferta académica de la ESPOL, cuyo lenguaje de comunicación común para el análisis y entendimiento de los diversos problemas de estudio es precisamente el de las Matemáticas.

DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO GRASP PARA EL PROBLEMA DE COLORACIÓN DE GRAFOS

Delgado Erwin¹

Resumen. Una de las dificultades en la búsqueda de la solución óptima a problemas de optimización combinatoria es el elevado costo computacional para obtenerlas. Por ello se hace necesaria la utilización de algoritmos basados en metaheurísticas para obtener soluciones factibles a un costo computacional razonable. En este contexto, en el presente trabajo se aborda el diseño de una heurística basada en la metodología GRASP para el Problema de Coloración de Grafos. Este algoritmo se lo ha implementado en Mathematica 8.0.4.0 © y ejecutado con diversas instancias del problema para medir la calidad del mismo.

Palabras Claves: GRASP, Coloración de Grafos, Metaheurísticas.

Abstract. One of the difficulties in finding the best solution for combinatorial optimization problems is the high computational cost to obtain them. Therefore it is necessary to use algorithms based on metaheuristics for feasible solutions to a reasonable computational cost. In this context, this paper addresses the design of a heuristic based on the GRASP methodology for graph coloring problem. This algorithm is implemented in Mathematica has 8.0.4.0 © of the problem to measure quality.

Keywords: GRASP, graph coloring, metaheuristics.

Recibido: Julio 2013

Aceptado: Agosto 2013

1. INTRODUCCIÓN

Muchos de los problemas de optimización combinatoria se caracterizan por su complejidad computacional en la determinación de una solución óptima por medio de métodos exactos. Por ello, dada esta particularidad desde hace mucho tiempo se intenta obtener soluciones factibles a estos problemas en un tiempo computacional razonable por medio de la aplicación de diversas metaheurísticas. Precisamente, El Problema de Coloración de Grafos (PCG) es un problema combinatorio el cual para grandes instancias se requiere la implementación de algoritmos que provea soluciones factibles a un costo computacional razonable en comparación a los obtenidos por métodos exactos. En este contexto, en el presente trabajo se aborda el diseño e implementación de una heurística para el PCG. Antes de definir formalmente el PCG es importante mencionar la siguiente definición establecida en [14].

Definición 1.1.

Dado un grafo $G = (V, E)$ donde V es un conjunto de vértices y $E = \{(v_i, v_j): v_i, v_j \in V, v_i \neq v_j\}$ un conjunto de arcos definidos en G . Una coloración propia de G es una función $f: V \rightarrow [1, k]$ tal que $\forall (a, b) \in E$ se cumple que $f(a) \neq f(b)$, es decir, asigna a cada vértice de G un color diferente al asignado a cualquier otro vértice adyacente.

En este contexto, el Problema de Coloración de Grafos, PCG, consiste en determinar el mínimo número k de colores diferentes con el objeto de realizar una coloración propia de G . A este mínimo valor de k se le conoce como número cromático de G , denotado por $\chi(G)$.

2. FORMULACIÓN MATEMÁTICA DEL PROBLEMA DE COLORACIÓN DE GRAFOS

A continuación, se presenta un modelo en programación lineal que permite encontrar la solución óptima al problema de coloración de colores.

2.1 ÍNDICES

Sea $G = (V, E)$ un grafo no dirigido y C un conjunto de colores

i, k : Índices referidos al conjunto de vértices

j : índice referido al conjunto de colores

2.2 VARIABLES

$$x_{ij} = \begin{cases} 1, & \text{si el vértice } i \text{ es coloreado} \\ & \text{con el color } j \\ 0, & \text{si no} \end{cases}$$

¹ Delgado Erwin, M.Sc., Profesor de la Escuela Superior Politécnica del Litoral (ESPOL).
(e_mail: edelgado@espol.edu.ec).

2.3 FORMULACIÓN MATEMÁTICA

$$\text{Min } \sum_i \sum_j jx_{ij} \quad (1)$$

s.a.

$$\sum_j x_{ij} = 1, \quad \forall i \in V \quad (2)$$

$$x_{ij} + x_{kj} \leq 1, \quad \forall (i, k) \in E, \forall j \in C \quad (3)$$

$$x_{ij} \in \{0,1\}, \quad \forall (i, j) \in E \quad (4)$$

La ecuación (1) representa la función objetivo, la cual permite minimizar el número de colores utilizados para realizar una coloración propia. La restricción (2) permite asegurar que cada vértice sea coloreado por algún color. La restricción (3) permite garantizar que para cada arco definido en el grafo, sus vértices sean coloreados con colores distintos. Por último, la restricción (4) indica la naturaleza de las variables de decisión.

3. REVISIÓN DE LITERATURA

Uno de los enfoques utilizados para resolver el PCG es por métodos exactos. Así, Isabel Méndez et al [8] propone un modelo de programación entera, el mismo que es resuelto por un algoritmo de planos cortantes. De igual manera, Hans Bodlaender [10] desarrolla un modelo exacto incorporando memoria polinomial mientras que C. Lucet et al [7] propone un método exacto basado en una descomposición lineal del grafo. Sin embargo estos métodos tienen limitaciones por la cardinalidad del conjunto de vértices del grafo. Precisamente, el PCG es uno de los problemas de optimización combinatoria, categorizado como NP-Hard [3], por tal motivo múltiples heurísticas se han diseñado con el objeto de obtener soluciones factibles de buena calidad en tiempos de ejecución adecuados. Uno de ellos es el realizado por [14] quién implementó un algoritmo glotón, ordenando en primer lugar los vértices de acuerdo al grado de adyacencia y asignando en cada iteración un color cada vértice obteniendo así una solución factible. De igual manera, Cédric Avanthay et al [5], desarrollaron una búsqueda local en un vecindario variable. Uno de los trabajos a destacar, dentro de la categoría de métodos de búsqueda local, es el propuesto en [15], en el cual se implementa una búsqueda local con una función de evaluación de la coloración de un grafo, la misma que está en términos del tamaño de cada partición del conjunto de vértices, así como del número de arcos que violen la condición de coloración.

Dentro de los trabajos que se fundamentan en algoritmos evolutivos, podemos destacar los realizados por Zhipeng Lu [4], quien desarrolló un algoritmo memético para este problema, resolviendo iterativamente el k -coloring desde un

valor de k , disminuyéndolo hasta que no exista alguna solución factible. Este algoritmo combina un genético con una búsqueda tabú, incorporando un operador de cruce adaptativo; y el realizado por A. Eiben et al [9] que desarrolla un algoritmo evolutivo para la determinación de buenas soluciones factibles. Incorpora un mecanismo adaptativo que permite cambiar el fitness de la función objetivo en cada iteración. Por otro lado, A. Hertz et al [6], diseña un algoritmo fundamentado en la búsqueda tabú, la misma que la comparó con los resultados obtenidos por algunos trabajos desarrollados con recocido simulado observando un mayor performance para instancias de gran cardinalidad.

4. GRASP

El GRASP² es una metaheurística en la que cada iteración consiste de dos etapas: construcción y búsqueda local [16]. El objetivo de la fase de construcción es la determinación de una solución factible la cual es mejorada en la fase de búsqueda local, lo que se muestra en el pseudocódigo mostrado a continuación, considerando un problema de minimización:

Algorithm 1 GRASP

```

1: procedure GRASP(MaxIter,Seed)
2:   for k=1,...,MaxIter do
3:     Solución1 ← EtapaConstrucción(Seed)
4:     Solución ← BúsquedaLocal(Solución1)
5:     if Solución < BestSolución then
6:       BestSolución = Solución
7:     end if
8:   end for
9:   return BestSolución
10: end procedure

```

El fundamento de la fase de construcción de una solución factible es un algoritmo glotón no determinístico, en la que iterativamente se incorporan los elementos a una solución parcial.

El elemento a ser incorporado en la solución parcial es escogido aleatoriamente desde una lista RCL, la cual está formada por aquellos elementos que produzcan los más bajos costos incrementales en la función objetivo. Luego que el elemento seleccionado es incorporado en la solución parcial, esta es actualizada y se procede a iterar nuevamente.

² Greedy Randomized Adaptive Search Procedure.

5. DISEÑO DE HEURÍSTICA BASADA EN LA METODOLOGÍA GRASP PARA EL PROBLEMA DE COLORACIÓN DE GRAFOS

A continuación se detalla cada componente del algoritmo basado en la metodología GRASP para el problema de coloración de grafos.

5.1. DISEÑO DE FASE DE CONSTRUCCIÓN

El objetivo principal en la fase de construcción, es la determinación de una solución factible del problema de coloración de grafos. Esta solución se la ha de determinar por medio de un algoritmo glotón no determinístico. Para tal efecto, se realizó una adaptación al algoritmo desarrollado por Daniel Brélaz [14] para el problema de coloración de grafos. A continuación se muestra el pseudocódigo de la fase de construcción de la solución inicial:

Algorithm 2 Fase de construcción

```

1: procedure CONSTRUCCIÓN(Seed)
2:   solución ←  $\phi$ 
3:   Ordenar los vértices de acuerdo a su grado en
   orden decreciente.
4:   Construir la lista de candidatos
   restringidas  $RCL$  definida por
    $\{v \in V | grad(v) \geq C_{min} + \alpha(C_{max} - C_{min})\}$ ,
   donde  $C_{max}$  y  $C_{min}$  son el máximo y mínimo
   grado de entre todos los vértices.
5:   Seleccionar y colorear aleatoriamente un
   vértice  $s$  de la lista  $RCL$ 
6:   while  $|solución| \neq |V|$  do
7:     Ordenar los vértices no coloreados de
     acuerdo a su grado de saturación en orden de-
     creciente.
8:     Construir la lista de candida-
     tos restringidas  $RCL$  definida por
      $\{v \in V | grad(v) \geq C_{min} + \alpha(C_{max} - C_{min})\}$ ,
     donde  $C_{max}$  y  $C_{min}$  son el máximo y mínimo
     grado de saturación de entre todos los vértices
     no coloreados.
9:     Seleccionar y colorear aleatoriamente un
     vértice  $s$  de la lista  $RCL$ , utilizando el mínimo
     número de colores para obtener una solución fac-
     tible parcial.
10:    solución ← solución  $\cup \{s\}$ 
11:  end while
12:  return solución
13: end procedure

```

Antes de explicar en detalle la adaptación realizada, es necesario introducimos en la siguiente definición establecida en [14].

Definición 5.1. Sea G un grafo simple y C una coloración parcial de G . Se define al grado de saturación de un vértice como el número de diferentes colores asignados a sus vértices adyacentes.

5.1.1. CRITERIO GLOTÓN

Una de las componentes del algoritmo GRASP es la construcción de un algoritmo glotón. Diversos enfoques pueden ser considerados para el efecto. Por ejemplo, se podría priorizar el grado de los vértices en el momento de colorear un vértice. En el presente trabajo inicialmente se procede a ordenar de forma decreciente los vértices de acuerdo a su grado de saturación. Este ordenamiento nos permite construir una lista de «buenos» candidatos a ser considerados en la siguiente iteración. Sin embargo, este criterio cambia a partir del momento en que se va a colorear el segundo vértice de la solución parcial ya que el ordenamiento se lo realiza con base en los grados de saturación de los vértices no coloreados. Esta idea surge del hecho de que vértices de un número elevado de arcos incidentes se los debe colorear primero con el objeto de minimizar el número de colores utilizados para realizar una coloración propia.

5.1.2. LONGITUD DE LA LISTA RCL

Una de las características del algoritmo GRASP es que la construcción de la solución factible no es determinística ya que en cada iteración se incorpora un vértice a la solución parcial de forma aleatoria, el mismo que pertenece a una lista, denominada RCL, de buenos candidatos. Diversas estrategias pueden ser consideradas para la determinación de la longitud de la lista de candidatos; así, puede considerarse una lista con longitud fija en todas las iteraciones o de longitud variable. Por ejemplo, sean C_{max} y C_{min} el máximo y mínimo grado de saturación de entre todos los vértices (o el grado de los vértices) entonces la lista RCL está definida como:

$$RCL = \left\{ v \in V \mid grad(v) \geq C_{min} + \alpha(C_{max} - C_{min}) \right\} \quad (5)$$

donde α es un parámetro a considerar.

5.1.3. ELECCIÓN DEL PRÓXIMO VÉRTICE A SER COLOREADO

Luego de construida la lista RCL formada por los vértices considerados buenos candidatos y que no

han sido coloreados, se procede a escoger aleatoriamente un vértice v de esta lista. Debido a que el GRASP debe realizarse un número MAX_ITER de iteraciones, la función generadora de números aleatorios está en términos de una variable entera $SEED$ que cambiará en cada iteración para garantizar la exploración de otras regiones. Al vértice v escogido se le asignará un color que minimice el número de colores utilizados para realizar una coloración parcial del grafo y que garantice la factibilidad del mismo.

Este procedimiento se lo realizará iterativamente hasta que todos los vértices sean coloreados, obteniendo así una solución factible inicial.

Sin embargo, no necesariamente la solución factible obtenida en la etapa de construcción es la óptima, por lo que se hace imprescindible un algoritmo de mejora de la solución actual, objetivo que se cumple con la fase de búsqueda local.

5.2. DISEÑO DE FASE DE BÚSQUEDA LOCAL

Como se ha explicado anteriormente, la fase de búsqueda local tiene por objeto mejorar una solución actual obteniendo así, en cada iteración, una solución de mejor calidad.

En este contexto, la fase de búsqueda local empieza con una solución inicial dada por la fase de construcción.

5.2.1. VECINDARIO DE UNA SOLUCIÓN

Uno de los requerimientos de cualquier búsqueda local es la forma de construcción de vecindades a una solución actual. En este sentido, una solución vecina a la actual es obtenida al colorear cualquier vértice escogido aleatoriamente por un color diferente, sea éste uno ya utilizado o por medio de otro color.

5.2.2. EVALUACIÓN DE UNA SOLUCIÓN

Conforme a la construcción del vecindario de una solución cualquiera, es evidente que al realizar un intercambio de colores podría generar soluciones no factibles, es decir asignar a un vértice un color que viole la restricción que cualquier vértice adyacente sea coloreado de la misma manera.

Sean $C = \{C_1, C_2, \dots, C_k\}$ una partición de colores en el grafo $G = (V, E)$ y $E_i, \forall i = 1, 2, \dots, k$ el conjunto formado por los arcos que tienen sus vértices adyacentes en la partición C_i , entonces el valor de la coloración definida por C está dada por [15]:

$$f(C) = -\sum_{i=1}^k |C_i|^2 + \sum_{i=1}^k |C_i| |E_i| \quad (6)$$

Un hecho a considerar en la ecuación 6 es que esta función de evaluación promueve la creación de particiones de gran cardinalidad, penalizando aquellas que contenga un gran conjunto de arcos que violen las restricciones dadas.

5.2.3. ACTUALIZACIÓN DE LA SOLUCIÓN ACTUAL

Existen dos maneras de actualizar la solución actual. Una de ellas es explorar todo el vecindario de la solución actual y reemplazarla por la solución vecina que posea el menor valor en la función definida en la sección anterior. Sin embargo, una de las dificultades de tal estrategia es el consumo de recursos, especialmente tiempo computacional por lo que en el presente trabajo se ha considerado reemplazar la solución actual por la primera solución vecina que mejore el costo de la coloración actual.

Es importante destacar que la exploración se la realiza considerando coloraciones factibles e infactibles y que esta etapa se la ejecuta durante un tiempo máximo.

A continuación se muestra el pseudocódigo de la fase de búsqueda local

Algorithm 3 Búsqueda Local

```

1: procedure BÚSQUEDALOCAL(solución) inicial
   ← f(solución)
2:   while Time_max no se cumpla do Determine
   una solución  $s \in N(solución)$  por medio del
   reemplazo del color de un vértice escogido alea-
   toriamente por un nuevo color.
3:     if  $f(s) < inicial$  then
       solución ← s
       inicial ← f(s)
4:     end if
5:   end while return solución
6: end procedure

```

5.3. CALIBRACIÓN

Uno de los aspectos a analizar en la presente sección es la estrategia a seguir con respecto al tamaño de la lista RCL. Para tal efecto, se ejecutó el algoritmo desarrollado con algunas instancias del problema de coloración de grafos³ durante 10 iteraciones cada una de ellas con una duración de 60 segundos.

³<http://mat.gsia.cmu.edu/COLOR/instances.html>XXLEI

En la tabla I se muestra un resumen de los resultados obtenidos al ejecutar el algoritmo en tres instancias específicas. En la primera columna se observa el nombre de la instancia, en la segunda columna el número de colores óptimo para realizar una coloración propia del grafo dado. En la tercera columna se muestran los porcentajes de aciertos con respecto al óptimo tanto de la estrategia con longitud de lista fija (*F*) y longitud variable (*V*) y por último los tiempos promedios en la obtención del mínimo número de colores requeridos, obtenidos en la ejecución de cada instancia. Con base en los resultados obtenidos, tanto tiempos como en calidad de la solución, se ha considerado la longitud de la lista RC L como variable, es decir, la lista RC L estará definida como en la ecuación 5

TABLA I

Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos

Calibración: Análisis de resultados

Instancia	# óptimo de colores	Resultados			
		% de aciertos		tiempos(seg.) ^a	
		F	V	F	V
myciel5.col	6	100	100	3.852	3.392
myciel6.col	7	60	70	30.157	31.727
myciel7.col	8	0	0	49.600	46.425

^a Tiempo promedio que toma en alcanzar el mínimo

6. RESULTADOS COMPUTACIONALES

Luego de la calibración de los parámetros del algoritmo GRASP desarrollado en la sección anterior, se ejecutó el mismo con diversas instancias en una PC con procesador INTEL CORE i3 3.07 GHz de 64 bits, durante un máximo de MAX_ITER, cada una de ellas ejecutadas durante un tiempo *t_max*, cuyos resultados se muestran en la tabla II.

En las figuras 1,2 y 3 se muestra la tendencia del número de colores requeridos para una coloración propia en cada instancia utilizada en función del tiempo. Como se puede observar, el algoritmo es eficiente en la búsqueda de soluciones de buena calidad, especialmente para instancias de pequeña escala.

A continuación, en las figuras 4, 5 y 6 se muestra la tendencia del número de colores requeridos para una coloración propia, pero sólo en el mínimo valor encontrado luego de ejecutar el número máximo de iteraciones. Como se puede observar, luego que se ha determinado una solución factible inicial, el tiempo que toma en alcanzar el mínimo

local es razonable en comparación con los que se toman al aplicar un modelo que los resuelva en forma exacta.

TABLA II

Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos

Resultados de la ejecución del algoritmo para diversas instancias

Instancia	# óptimo de colores	# de nodos	# de arcos	Parámetros		# de colores
				MAX_ITER	t_max(seg)	
queen8_8.col	9	64	728	400	45	10
queen9_9.col	10	81	2112	280	90	12
queen11_11.col	11	121	3960	255	120	14
myciel3.col	4	11	20	320	10	4
myciel4.col	5	23	71	320	10	5
myciel5.col	6	47	236	320	10	6
myciel6.col	7	95	755	320	10	7
myciel7.col	8	191	2360	320	10	9

FIGURA 1

Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos

Gráfica de la evolución del número de colores requeridos en la instancia queen8_8.col

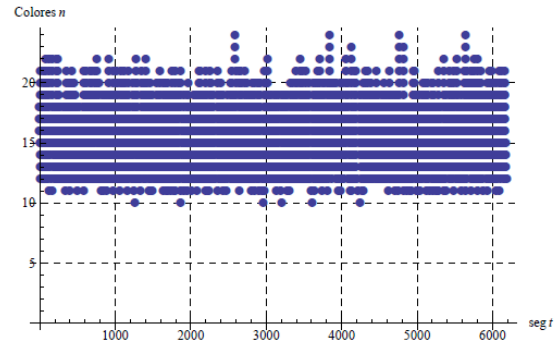


FIGURA 2

Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos

Gráfica de la evolución del número de colores requeridos en la instancia queen9_9.col

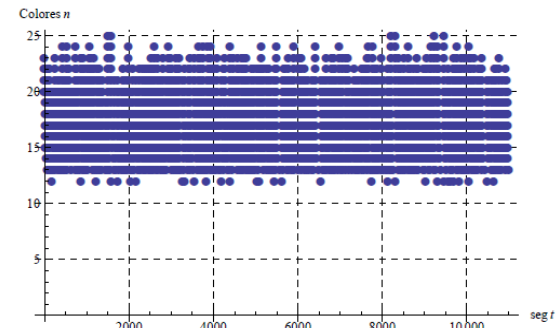


TABLA III

Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos

Comparación entre heurística propuesta y el algoritmo de Brélaz

Instancia	Heurística Propuesta	Algoritmo Brélaz
queen8_8.col	10	15
queen9_9.col	12	15
queen11_11.col	14	16
myciel3.col	4	4
myciel4.col	5	5
myciel5.col	6	6
myciel6.col	7	7
myciel7.col	9	8

Otra forma de validar los resultados que se obtienen al ejecutar el algoritmo propuesto para diversas instancias, es por medio de la comparación de los mismos con los que se obtienen al aplicar una función específica de la librería de Mathematica 8.0.4.0 ® denominada «VertexColoring», la misma que consiste en la implementación del algoritmo de Brélaz [14].

En la tabla III se muestra los resultados al ejecutar diversas instancias tanto con la heurística propuesta como con el algoritmo de Brélaz.

FIGURA 3

Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos

Gráfica de la evolución del número de colores requeridos en la instancia queen11_11.col

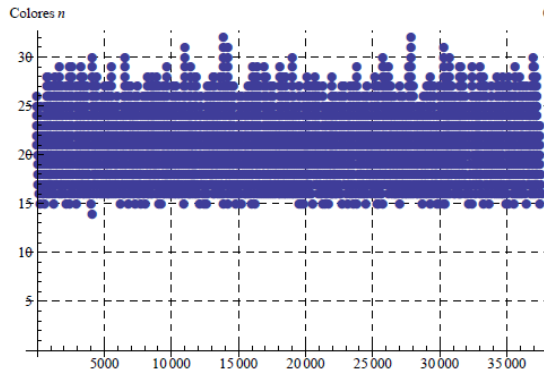


FIGURA 4

Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos

Evolución del número de colores requeridos en la instancia queen8_8.col

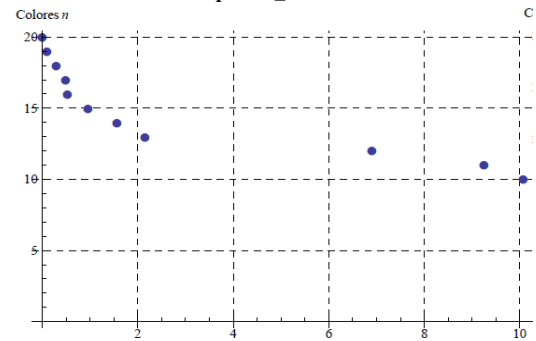


FIGURA 5

Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos

Evolución del número de colores requeridos en la instancia queen9_9.col

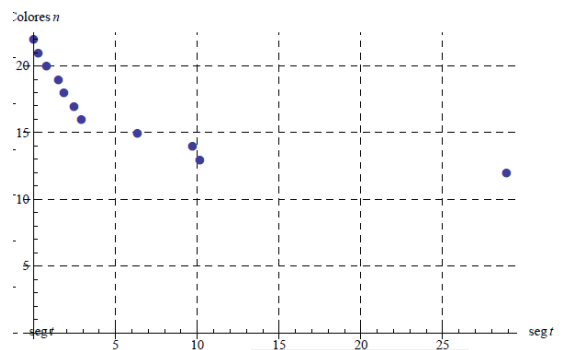
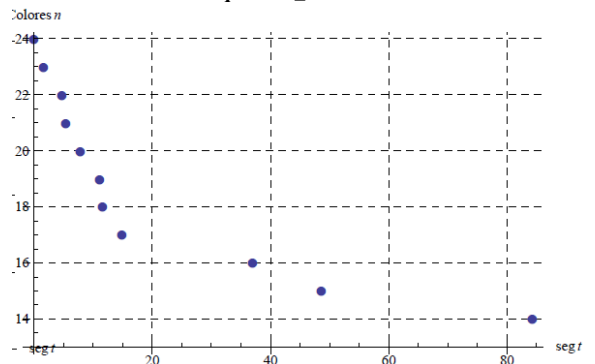


FIGURA 6

Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos

Evolución del número de colores requeridos en la instancia queen11_11.col



7. CONCLUSIONES Y RECOMENDACIONES

Conforme a los resultados obtenidos al aplicar la heurística propuesta al problema de coloración de grafos, es evidente la gran utilidad de aplicar

heurísticas en la búsqueda de soluciones factibles de buena calidad siendo computacionalmente mejores en tiempos de ejecución en comparación a los obtenidos por métodos exactos. Sin embargo, para problemas de gran cardinalidad aún se observa

una gran variación con respecto a los valores óptimos.

Esto se debe principalmente por la no utilización de estructuras de memorias a corto y largo plazo, por lo que se recomienda en futuros trabajos incorporarlos.

REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS

- [1]. **DING-ZHUE DU AND PANOS M. PARDALOS.** (2005). “*Handbook of combinatorial optimization*”. Volume B, Springer.
- [2]. **MCCOLLUM B., MCMULLAN P.** (2007). “*The Second International Timetabling Competition: Examination Timetabling Track*”. Technical Report. Queen’s University Belfast.
- [3]. **DING-ZHUE DU AND PANOS M. PARDALOS,** (2005). “*Handbook of combinatorial optimization*”, Volume B, Springer.
- [4]. **ZHIPENG LU, JIN-KAO HAO.** (2009). “*A memetic algorithm for graph coloring*”, European Journal of Operations Research, ELSEVIER.
- [5]. **AVANTHAY C., HERTZ A., ZUFFEREY N.** (2003). “*A variable neighborhood search for graph coloring*”. European Journal of Operational Research, ELSEVIER.
- [6]. **A. HERTZ, D. DE WERRA.** (1987). “Using Tabu Search Techniques for Graph Coloring. Computing”. By Springer-Verlag.
- [7]. **C.LUCET, F. MENDES, A. MOUKRIM.** (2004). “*An Exact method for graph coloring. Computer & Operations Research*”, ELSEVIER.
- [8]. **MÉNDEZ I., ZABALA P.** (2007). “*A cutting plane algorithm for graph coloring. Discrete Applied Mathematics*”. ELSEVIER.
- [9]. **EIBEN A., VAN DER HAUW J., VAN HENNERT J.** “*Graph Coloring with Adaptive Evolutionary Algorithms*”. Journal of Heuristics, volume 4:1.
- [10]. **BODLAENDER H., KRATSCH D.** (2006). “An exact algorithm for graph coloring with polynomial memory”. Utrecht University.
- [11]. **JOHNSON D., ARAGÓN C., MCGEOCH L., SCHEVON C.** (1990). “*Optimization by simulated annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partition, Operations Research Society of America*”.
- [12]. **SOPENA E.** (2001). “*Oriented Graph Coloring*”, ELSEVIER.
- [13]. **WERRA, D.** (1990). “*Heuristics for Graph Coloring, Computational Graph Theory, Compute*”. Suppl. 7, Springer, Vienna, 191-208.
- [14]. **BRÉLAZ, D.** (1979). “*New methods to color the vertices of a graph*”, Communications of the Assoc. of Com- put. Machinery 22, 251-256.
- [15]. **JOHNSON, D. S., ARAGON, C. R., MCGEOCH, L.A., AND SCHEVON, C.** “*Optimization by simulated annealing: An experimental evaluation; part ii, graph coloring and number partitioning*”. Operations Research, 39(3):378406.
- [16]. **RESENDE M., RIBEIRO C.** (2002). “*Greedy Randomized Adaptive Search Procedures*”, AT&T Labs Research Technical Report. Hanbook in Metaheuristic, Fred Glover.
- [17]. **GLOVER FRED, KOCHENBERGER.** (2003). “*HandBook of Metaheuristics*”, International Series in Operations Research & Management Science.
- [18]. **TALBI EL-GHAZALI.** (2009). “*Metaheuristics from design to implementation*”, John Wiley & Sons, Inc., Hoboken, New Jersey.

IMPLEMENTACIÓN DE UN ALGORÍTMO GRASP PARA EL PROBLEMA DE COLORACIÓN DE GRAFOS APLICADO A LA CALENDARIZACIÓN DE EXÁMENES EN UNA INSTITUCIÓN EDUCATIVA

Delgado Erwin¹

Resumen. Una de las tareas que enfrentan las instituciones educativas cada año, es la planificación de los horarios de clases y exámenes. Su dificultad radica en que diversas restricciones operativas surgen en el momento de la planificación. Dada la naturaleza del problema descrito anteriormente, la calendarización de exámenes pertenece al conjunto de problemas de optimización combinatoria categorizado NP-Duro por lo que resulta complejo resolverlo por métodos exactos. La ventaja es que la calendarización de exámenes es un problema operativo por lo que bastaría con obtener soluciones factibles de gran calidad, no necesariamente la óptima, en tiempos computacionales razonables. Una de las herramientas utilizadas para el efecto, es la construcción de heurísticas basadas en metaheurísticas por la fortaleza en la exploración inteligente en el espacio de soluciones. Con base en lo anterior, en el presente trabajo se desarrollará un algoritmo heurístico basado en la metodología GRASP el mismo que se lo aplicará en la confección de horarios de exámenes sujetos a un conjunto de restricciones de diversas índoles.

Palabras Claves: GRASP, Coloración de Grafos, Metaheurísticas, Calendarización de exámenes.

Abstract. One of the tasks facing educational institutions each year is planning class schedules and exams. His difficulty is that various operational constraints arise at the time of planning. Given the nature of the problem described above, the test scheduling belongs to the set of combinatorial optimization problems categorized NP-Hard, making it complex to solve by exact methods. The advantage is that the scheduling of examinations is an operational problem it would be sufficient to obtain high quality feasible solutions, not necessarily optimal, in reasonable computational times. One of the tools used for this purpose is the construction of metaheuristics based heuristics for intelligent exploration strength in the solution space. Based on the above, in this paper we develop a heuristic algorithm based on the GRASP methodology the same as it applied in the preparation of test schedules subject to a set of constraints of various kinds.

Keywords: GRASP, graph coloring, metaheuristics, timetabling examinations.

Recibido: Agosto 2013

Aceptado: Septiembre 2013

1. INTRODUCCIÓN

Una de las tareas que enfrentan las instituciones educativas cada año, es la planificación de los horarios de clases y exámenes. Su dificultad radica en que diversas restricciones operativas surgen en el momento de la planificación. Específicamente, en el caso de la calendarización de exámenes, generalmente se intenta elaborarlos considerando diversas restricciones como por ejemplo que los mismos deben ser receptados en uno y sólo un periodo de tiempo y aula previamente definida, la misma que debe admitir un número máximo de estudiantes. De igual manera, la calidad de los calendarios está basado en promover aquellos que impidan que estudiantes rindan más de un examen en un mismo día, entre otras restricciones operativas. Considerando lo anterior, y el hecho de que la calendarización de exámenes pertenece al conjunto de problemas combinatorios con elevada complejidad computacional, hace que éste sea complejo de resolver por métodos exactos.

La ventaja, es que la calendarización de exámenes es un problema operativo por lo que bastaría con obtener soluciones factibles de gran calidad, no necesariamente la óptima, en tiempos computacionales razonables. Una de las herramientas utilizadas para el efecto, es la construcción de heurísticas basadas en metaheurísticas por la fortaleza en la exploración inteligente en el espacio de soluciones.

Considere un conjunto de N exámenes $E = \{e_1, e_2, \dots, e_N\}$, los mismos que deben ser planificados en algún periodo $p \in \{p_1, p_2, \dots, p_m\}$ Un calendario factible es aquel que satisface las siguientes restricciones fuertes:

- Todos los exámenes planificados son asignados a algún periodo de tiempo
- Todos los exámenes planificados son asignados a alguna aula.
- Ningún estudiante podrá rendir más de un examen en un mismo periodo de tiempo.
- En cada periodo de tiempo, no puede asignarse un aula para receptor dos o más exámenes
- La capacidad de las aulas no debe ser excedida.
- Alguna otra restricción operativa definida por la organización.

¹ Delgado Erwin, M.Sc., Profesor de la Escuela Superior Politécnica del Litoral (ESPOL).
(e_mail: edelgado@espol.edu.ec).

La calidad de un calendario factible está definida por el valor de la función objetivo que, entre otras cosas, penaliza alguna de las siguientes restricciones suaves:

- Materias de un mismo nivel no pueden ser planificadas en el mismo día
- A cada estudiante se le receptorá máximo un examen por día.
- En caso de que a un estudiante se le recepte más de un examen por día, estos no deben ser consecutivos.
- Alguna otra restricción operativa definida por la organización.

En este contexto, el problema de calendarización de exámenes consiste en la asignación de aula y horario a cada uno de los exámenes planificados en un periodo académico, satisfaciendo cada una de las restricciones fuertes establecidas por la organización, minimizando las penalizaciones por el no cumplimiento de las restricciones suaves.

2. FORMULACIÓN MATEMÁTICA DEL PROBLEMA DE CALENDARIZACIÓN DE EXÁMENES

Antes de abordar el modelo matemático para el problema de calendarización de exámenes es importante establecer algunas componentes del mismo.

2.1 CONJUNTOS Y PARÁMETROS

1. Exámenes:
 - a) E : Conjunto de exámenes.
 - b) se_i : Número de estudiantes registrados en el examen $i \in E$
2. Estudiantes:
 - a) S : Conjunto de estudiantes
 - b) $t_{is} \begin{cases} 1 & \text{si el estudiante } s \text{ debe rendir} \\ & \text{el examen } i \\ 0 & \text{si no} \end{cases}$
3. Aulas
 - a) R conjunto de aulas
 - b) s_r : Capacidad del aula $r \in R$
4. Periodos
 - a) P : Conjunto de periodos
 - b) H^{aft} : Conjunto de pares de exámenes.
 $\forall (e_1, e_2) \in H^{aft}$ examen e_1 debe receptorse estrictamente después del examen e_2 .
 - c) H^{coin} : Conjunto de pares de exámenes.
 $\forall (e_1, e_2) \in H^{coin}$ los exámenes e_1 y e_2 deben ser receptados en el mismo periodo.
 - d) H^{excl} : Conjunto de pares de exámenes.

$\forall (e_1, e_2) \in H^{excl}$ los exámenes e_1 y e_2 no deben ser receptados en el mismo periodo.

5. Penalizaciones

- a) w^{2R} : penalización por exámenes consecutivos.
- b) w^{2D} : penalización por exámenes receptados en el mismo día.

2.2 VARIABLES

$$1. xp_{ip} = \begin{cases} 1, & \text{si el examen } i \text{ debe programarse} \\ & \text{en el periodo } p \\ 0, & \text{si no} \end{cases}$$

$$2. xr_{ir} = \begin{cases} 1, & \text{si el examen } i \text{ debe programarse} \\ & \text{en el aula } r \\ 0, & \text{si no} \end{cases}$$

3. $C2R_s$ Variable que indica el número de ocasiones que el estudiante s tiene dos exámenes consecutivos en un día.

4. $C2D_s$ Variable que indica el número de ocasiones que el estudiante s tiene dos exámenes planificados en un día.

2.3 FORMULACIÓN MATEMÁTICA

A continuación se presenta un modelo de programación no lineal presentado en [2] que permite encontrar una solución óptima con la mínima penalización por violación de las restricciones suaves.

$$\text{Min } \sum_{s \in S} w^{2R} C2R_s + w^{2D} C2D_s \quad (1)$$

$$\sum_{r \in R} xr_{ir} = 1, \quad \forall i \in E \quad (2)$$

$$\sum_{p \in P} xp_{ip} = 1, \quad \forall i \in E \quad (3)$$

$$se_i xp_{ip} xr_{ir} \leq sr_r, \quad \forall p \in P, \forall i \in E, \forall r \in R \quad (4)$$

$$\sum_{i \in E} t_{is} xp_{ip} \leq 1, \quad \forall p \in P, \forall s \in S \quad (5)$$

$$xp_{ip} + xp_{jq} \leq 1, \quad \forall p, q \in P, p \leq q, \forall (i, j) \in H_{aft} \quad (6)$$

$$xp_{ip} = xp_{jq}, \quad \forall p \in P, \forall (i, j) \in H_{coin} \quad (7)$$

$$xp_{ip} + xp_{jq} \leq 1, \quad \forall p \in P, \forall (i, j) \in H_{excl} \quad (8)$$

$$C2R_s = \sum_{i \neq j} \sum_{\substack{p, q \in P, \\ q = p + 1, \\ ypq = 1}} t_{is} t_{js} xp_{ip} xp_{jq} \quad (9)$$

$$C2D_s = \sum_{i \neq j} \sum_{\substack{p, q \in P, \\ q > p + 1, \\ ypq = 1}} t_{is} t_{js} xp_{ip} xp_{jq} \quad (10)$$

$$xp_{ip}, xr_{ir} \in \{0, 1\} \quad (11)$$

$$C2R_s, C2D_s \in \mathbb{Z}, C2R_s \geq 0, C2D_s \geq 0 \quad \forall s \in S \quad (12)$$

La ecuación (1) establece la función objetivo la cual es una combinación lineal del número de estudiantes que tienen exámenes consecutivos, así como exámenes que se toman en el mismo día. La restricción (2) establece que cada examen sea receptado en alguna aula. La restricción (3) establece que cada examen sea receptado en algún

periodo. La restricción (4) garantiza que ningún examen sea receptado en un aula con capacidad menor al número de alumnos que deben rendir la prueba. La restricción (5) garantiza que cada estudiante en cualquier periodo rinda a lo mucho un examen. La restricción (6) garantiza el cumplimiento de que para cada par ordenado de exámenes que pertenezcan a Haft, uno debe estar planificado estrictamente después del otro. De manera similar se analiza las restricciones (7) y (8). Las ecuaciones (9) y (10) permiten determinar el número de estudiantes que tienen planificado más de un examen en el mismo día, así como los que rinden dos exámenes consecutivos. Por último las restricciones (11) y (12) se refieren al carácter de las variables declaradas en el modelo.

Uno de los enfoques utilizado para obtener soluciones factibles de buena calidad es por medio de la utilización de métodos que permite resolver el problema de coloración de grafos. Este es el enfoque que se ha considerado en el presente proyecto.

3. REVISIÓN DE LITERATURA

Uno de los enfoques utilizados para resolver el PCG es por métodos exactos. Así, Isabel Méndez et al [9] propone un modelo de programación entera, el mismo que es resuelto por un algoritmo de planos cortantes. De igual manera, Hans Bodlaender [11] desarrolla un modelo exacto incorporando memoria polinomial mientras que C. Lucet et al [8] propone un método exacto basado en una descomposición lineal del grafo. Sin embargo estos métodos tienen limitaciones por la cardinalidad del conjunto de vértices del grafo. Precisamente, el PCG es uno de los problemas de optimización combinatoria, categorizado como NP-Hard [4], por tal motivo múltiples heurísticos se han diseñado con el objeto de obtener soluciones factibles de buena calidad en tiempos de ejecución adecuados. Uno de ellos es el realizado por [15] quién implementó un algoritmo glotón, ordenando en primer lugar los vértices de acuerdo al grado de adyacencia y asignando en cada iteración un color cada vértice obteniendo así una solución factible. De igual manera, Cédric Avanthay et al [6], desarrollaron una búsqueda local en un vecindario variable. Uno de los trabajos a destacar, dentro de la categoría de métodos de búsqueda local, es el propuesto en [16], en el cual se implementa una búsqueda local con una función de evaluación de la coloración de un grafo, la misma que está en términos del tamaño de cada partición del conjunto de vértices, así como del número de arcos que violen la condición de coloración.

Dentro de los trabajos que se fundamentan en algoritmos evolutivos, podemos destacar los realizados por Zhipeng Lu [5], quien desarrolló un algoritmo memético para este problema, resolviendo iterativamente el k-coloring desde un valor de k, disminuyéndolo hasta que no exista alguna solución factible. Este algoritmo combina un genético con una búsqueda tabú, incorporando un operador de cruce adaptativo; y el realizado por A. Eiben et al [10] que desarrolla un algoritmo evolutivo para la determinación de buenas soluciones factibles. Incorpora un mecanismo adaptativo que permite cambiar el fitness de la función objetivo en cada iteración. Por otro lado, A. Hertz et al [7], diseñó un algoritmo fundamentado en la búsqueda tabú, la misma que la comparó con los resultados obtenidos por algunos trabajos desarrollados con recocido simulado observando un mayor performance para instancias de gran cardinalidad.

4. GRASP

El GRASP² es una metaheurística en la que cada iteración consiste de dos etapas: construcción y búsqueda local [17]. A continuación se muestra el pseudocódigo del algoritmo GRASP, considerando un problema de minimización:

Algorithm 1 GRASP

```

1: procedure GRASP(MaxIter,Seed)
2:   for k=1,...,MaxIter do
3:     Solución1 ← EtapaConstrucción(Seed)
4:     Solución ← BúsquedaLocal(Solución1)
5:     if Solución < BestSolución then
6:       BestSolución = Solución
7:     end if
8:   end for
9:   return BestSolución
10: end procedure

```

El objetivo de la fase de construcción es la determinación de una solución factible la cual es mejorada en la fase de búsqueda local. En este sentido, el fundamento de la fase de construcción de una solución factible es un algoritmo glotón no determinístico, en la que iterativamente se incorporan los elementos a una solución parcial. El elemento a ser incorporado en la solución parcial es escogido aleatoriamente desde una lista RCL, la

² Greedy Randomized Adaptive Search Procedure.

cual está formada por aquellos elementos que produzcan los más bajos costos incrementales en la función objetivo. Luego que el elemento seleccionado es incorporado en la solución parcial, esta es actualizada y se procede a iterar nuevamente hasta que todos los vértices sean coloreados respectando las restricciones establecidas. Luego de la obtención de la solución inicial, se procede a realizar una búsqueda local, la misma que en cada iteración mejora la solución actual hasta alcanzar un óptimo local. En el presente trabajo se ha utilizado una adaptación del algoritmo GRASP desarrollado por [3], el cual permite encontrar soluciones factibles al problema de coloración de grafos.

5. IMPLEMENTACIÓN DEL ALGORITMO GRASP AL PROBLEMA DE CALENDARIZACIÓN DE EXÁMENES

Una de las aplicaciones de los algoritmos diseñados para el Problema de Coloración de Grafos es en la confección de horarios de exámenes en una institución educativa. En este contexto, se ha considerado aplicar una adaptación del algoritmo desarrollado en [3] en la planificación de los horarios de exámenes en un semestre específico de estudios, de una carrera específica de una institución educativa de nuestro medio, por lo que se hace necesario establecer diversas consideraciones operativas propias de la organización que son importantes en el diseño de horarios factibles.

5.1. RESTRICCIONES

Diversas restricciones operativas se presentan en el momento de elaborar cualquier planificación. Específicamente, en la confección de los horarios de exámenes en la que se pretende aplicar una adaptación del algoritmo desarrollado en [3], se han considerado las siguientes restricciones.

1. Restricciones fuertes.

a) Uno de los mayores problemas que tiene el estudiantado de cualquier institución educativa es la que exámenes considerados fuertes se planifiquen el mismo día. Por ello, y como una forma de impulsar

su desempeño académico, se ha considerado que los exámenes de las materias fuertes ubicadas en el mismo nivel se lo recepten en días diferentes.

b) De igual manera, se ha considerado un máximo de 30 aulas disponibles en un día, por lo que en cada día no se puede programar materias que en conjunto superen el número máximo de aulas disponibles.

2. Restricciones suaves.

a) Se ha considerado el hecho de que en cada semestre planificado se penalice la posibilidad de evaluar más dos exámenes del mismo nivel.

Con base en lo establecido anteriormente, es necesario conocer las materias y el número de paralelos aperturados por cada materia. En la tabla 1 se muestran las materias por nivel. De igual manera, entre paréntesis se muestra el número de paralelos por cada materia.

TABLA I

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Materias impartidas por niveles

Nivel	Materias				
1	MatI914(14)	MatI2496(1)	MatI039(1)		
2	MatI985(11)	MatI640(13)	MatI910(1)	MatI749(10)	MatI236(2)
3	MatI185(6)	MatI623(2)	MatI966(10)	MatI444(1)	MatI947(11)
4	MatI127(1)	MatI415(1)	MatI166(8)		
5	MatI402(1)	MatI496(1)	MatI458(1)		
6	MatF177(1)	MatI591(1)	MatF115(1)	MatF144(1)	MatI477(1)
7	MatI514(3)				
8	MatI326(2)	MatI484(2)			
9	MatI548(1)	MatI913(1)	MatI529(1)	MatI618(1)	

De igual manera, en la tabla 2 en cada fila se muestran las materias en la que las evaluaciones de la primera columna deben estar planificadas en días distintos a la restante por su nivel de complejidad.

6. RESULTADOS COMPUTACIONALES

A continuación se exponen los resultados obtenidos de la implementación del algoritmo GRASP en el problema de calendarización de exámenes:

TABLA II

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Materias que no deben evaluarse en el mismo día

Materia	Materias que deben evaluarse en días diferentes a la de la columna 1					
MatI548	MatI185	MatI947				
MatI127	MatI966	MatI166	MatI947			
MatF115	MatI591	MatI402	MatI477			
MatI913	MatI548	MatI185				
MatI591	MatF115	MatI477				
MatI185	MatI548	MatI913	MatI947			
MatI402	MatF115					
MatI749	MatI910	MatI640	MatI914	MatI985		
MatI477	MatF115	MatI591	MatI947			
MatI966	MatI127	MatI640	MatI947			
MatI166	MatI127	MatI947				
MatI640	MatI910	MatI966	MatI914	MatI985		
MatI914	MatI910	MatI749	MatI640			
MatI985	MatI910	MatI749	MatI640			
MatI947	MatI548	MatI127	MatI185	MatI477	MatI966	MatI166

TABLA III

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Horario de exámenes de materias del primer nivel

	Lu	Ma	Mi	Ju	Vi
8:30 10:30		MATF099			
11:00 13:00					
13:30 15:30			MATI914		MATI2496
16:00 18:00					

TABLA VI

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Horario de exámenes de materias del cuarto nivel

	Lu	Ma	Mi	Ju	Vi
8:30 10:30					MATI166
11:00 13:00		MATI415			
13:30 15:30			MATH127		
16:00 18:00					

TABLA IV

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Horario de exámenes de materias del segundo nivel

	Lu	Ma	Mi	Ju	Vi
8:30 10:30		MATI640		MATI910	
11:00 13:00					
13:30 15:30	MATI749		MATI985		MATI296
16:00 18:00					

TABLA VII

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Horario de exámenes de materias del quinto nivel

	Lu	Ma	Mi	Ju	Vi
8:30 10:30					MATI496
11:00 13:00		MATI402			
13:30 15:30					MATI458
16:00 18:00					

Como se puede observar en cada una de las tablas III, IV, V, VI, VII, VIII, IX, X, XI, se cumplen cada una de las restricciones fuertes establecidas anteriormente.

TABLA V

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Horario de exámenes de materias del tercer nivel

	Lu	Ma	Mi	Ju	Vi
8:30 10:30	MATI947	MATI185		MATI966	
11:00 13:00					
13:30 15:30			MAT1623		MATI444
16:00 18:00					

TABLA VIII

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Horario de exámenes de materias del sexto nivel

	Lu	Ma	Mi	Ju	Vi
8:30 10:30		MATF177		MATI477	
11:00 13:00		MATI591			
13:30 15:30	MATF115				MATF144
16:00 18:00					

TABLA IX

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Horario de exámenes de materias del séptimo nivel

	Lu	Ma	Mi	Ju	Vi
8:30 10:30					MATI514
11:00 13:00					
13:30 15:30					
16:00 18:00					

TABLA X

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Horario de exámenes de materias del octavo nivel

	Lu	Ma	Mi	Ju	Vi
8:30 10:30					MATI484
11:00 13:00					
13:30 15:30			MATI926		
16:00 18:00					

TABLA XI

Implementación de un algoritmo GRASP para el problema de coloración de grafos aplicado a la calendarización de exámenes en una institución educativa

Horario de exámenes de materias del noveno nivel

	Lu	Ma	Mi	Ju	Vi
8:30 10:30	MATI618				MATI913
11:00 13:00					
13:30 15:30		MATI529		MATI548	
16:00 18:00					

7. CONCLUSIONES Y RECOMENDACIONES

En el presente trabajo se ha utilizado una adaptación del algoritmo basado en la metodología GRASP desarrollado en [3] aplicado en el problema de calendarización de exámenes. En este contexto, se consideró aplicarlo en la planificación de un semestre académico de una carrera de pregrado de una institución educativa. Al observar los resultados obtenidos se ha podido evidenciar la utilidad de heurísticas en problemas de optimización combinatoria con elevada complejidad por cuanto nos entrega soluciones factibles de gran calidad a un costo computacional razonable. Sin embargo, sólo se ha considerado como restricciones fuertes el hecho de que materias de complejidad elevada sean evaluadas en días distintos, así como el número de paralelos evaluados en un día específico no supere un límite, por lo que se debe considerar restricciones adicionales como el hecho de existen materias que los estudiantes dejan rezagadas periodo a periodo y que en definitiva afecta la confección de calendarios factibles.

REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS

- [1]. **DING-ZHUE DU, PANOS M. PARDALOS.** (2005), “*Handbook of combinatorial optimization*”. Volumen B, Springer.
- [2]. **BARRY MCCOLLUM, PAUL MCMULLAN.** (2007), “*The Second International Timetabling Competition: Examination Timetabling Track*”. Technical Report. Queen’s University Belfast.
- [3]. **E. DELGADO,** “*Diseño e implementación de un algoritmo GRASP para el problema de coloración de grafos*”.
- [4]. **DING-ZHUE DU, PANOS M. PARDALOS.** (2005). “*Handbook of combinatorial optimization*”, Volumen B, Springer.
- [5]. **ZHIPENG LU, JIN-KAO HAO.** (2009). “*A memetic algorithm for graph coloring*”, European Journal of Operations Research, **ELSEVIER**.
- [6]. **CÉDRIC AVANTHAY, ALAIN HERTZ, NICOLAS ZUFFEREY.** (2003). “*A variable neighborhood search for graph coloring*”. European Journal of Operational Research, **ELSEVIER**.
- [7]. **A. HERTZ, D. DE WERRA.** (1987). “*Using Tabu Search Techniques for Graph Coloring*”. Computing by Springer-Verlag.
- [8]. **C.LUCET, F. MENDES, A. MOUKRIM.** (2004). “*An Exact method for graph coloring*”. Computer & Operations Research, **ELSEVIER**.
- [9]. **ISABEL MÉNDEZ, PAULA ZABALA.** (2007). “*A cutting plane algorithm for graph coloring*”. Discrete Applied Mathematics. **ELSEVIER**.
- [10]. **A. EIBEN, J. VAN DER HAUW, J. VAN HENNERT.** “*Graph Coloring with Adaptive Evolutionary Algorithms*”. Journal of Heuristics, volume 4:1.
- [11]. **HANS BODLAENDER, DIETER KRATSCH.** (2006). “*An exact algorithm for graph coloring with polynomial memory*”. Utrecht University.
- [12]. **DAVID JOHNSON, CECILIA ARAGÓN, LYLE MCGEOCH, CATHERINE SCHEVON,** (1990). “*Optimization by simulated annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partition*”, Operations Research Society of America.
- [13]. **ERIC SOPENA,** (2001). “*Oriented Graph Coloring*”, **ELSEVIER**.
- [14]. **WERRA, D.** (1990). “*Heuristics for Graph Coloring*”, Computational Graph Theory, Comput. Suppl. 7, Springer, Vienna, 191-208.
- [15]. **BRÉLAZ, D.,** (1979). “*New methods to color the vertices of a graph*”, Communications of the Assoc. of Comput. Machinery 22, 251-256.
- [16]. **JOHNSON, D. S., ARAGON, C. R., MCGEOCH, L. A., AND SCHEVON, C.** “*Optimization by simulated annealing: An experimental evaluation*”; part ii, graph coloring and number partitioning. Operations Research, 39(3):378406.
- [17]. **RESENDE M., RIBEIRO C.,** (2002). Greedy Randomized Adaptive Search Procedures, AT&T Labs Research Technical Report. Handbook in Metaheuristic, Fred Glover.
- [18]. **GLOVER FRED, KOCHENBERGER.** (2003). “*HandBook of Metaheuristics*”, International Series in Operations Research & Management Science.
- [19]. **TALBI EL-GHAZALI.** (2009). Metaheuristics from design to implementation, John Wiley & Sons, Inc., Hoboken, New Jersey.

DE RIZADOS EN LA DENSIDAD ESPECTRAL DE POTENCIA CALCULADA EN UNA SEÑAL DE RITMO CARDIACO

González Javier¹, Forero Edwin², Jiménez Fabián³, Marino Iván⁴

Resumen. El análisis de la variabilidad de la frecuencia cardiaca está basado en el estudio de los cambios de cada ciclo cardiaco. Estos cambios son estudiados a partir de la señal de ritmo cardiaco compuesta por los tiempos entre cada onda de la señal electrocardiográfica. El análisis de la señal de ritmo cardiaco es realizado en dos métodos: cálculos estadísticos (dominio del tiempo) y estimación de la densidad espectral de potencia (dominio de la frecuencia). La estimación de la densidad espectral de potencia (PSD) de una señal de ritmo cardiaco requiere métodos orientados para señales con muestreo irregular. Para este caso se ha implementado la estrategia de Lomb. El objetivo de este artículo es la implementación del método de promediado de espectros para atenuar los rizados que aparecen en el cálculo de la PSD. El procedimiento final se basa en la aplicación de la misma técnica de tomar señales de ritmo cardíacos adquiridos de ritmo sinus normal del repositorio "Physionet". Los resultados obtenidos de estos experimentos mostraron una disminución de la ondulación en el PSD y la variación de los parámetros en el dominio de la frecuencia.

Palabras Claves: Densidad espectral de potencia, tiempo RR, muestreo irregular.

Abstract. The analysis of heart rate variability is based on the study of changes detected in each cardiac cycle. These changes have been studied from the cardiac rhythm signal and it is composed of data acquired from the time measured between the R waves of electrocardiographic signal. The cardiac rhythm signal analysis is based on two kinds of methods: statistical calculation (time domain) and the power spectrum density estimation (frequency domain). Power spectrum density (PSD) estimation from cardiac rhythm signal, can be done through math methods for signals with non-regular sampling time. For this case, in the literature has been registered the use of Lomb method. The main goal of this paper is the presentation of results obtained from the implementation of a technical based on spectrum averaging oriented to ripple decrease of the PSD estimation in cardiac signal rhythm. The final procedure is based on the application of the same technique taking cardiac rhythm signals acquired from normal sinus rhythm database "Physionet". The results obtained from these experiments showed a decrease of ripple in the PSD and variation of parameters in the frequency domain.

Keywords: Power Spectral Density, RR time, non-regular sampling.

Recibido: Agosto 2013

Aceptado: Septiembre 2013

1. INTRODUCCIÓN

El análisis de la variabilidad de la Frecuencia cardiaca (VFC) está basado en el estudio de las fluctuaciones de la frecuencia cardiaca instantánea. Estos cambios son calculados a través de la detección de la onda R del electrocardiograma (ECG) y de la cuantificación del valor del tiempo entre cada onda. Los valores de tiempo son almacenados en un arreglo denominado la señal de ritmo cardiaco y a la vez se presentan con muestreo irregular [1]. Para el análisis de la VFC, se toman los valores de la señal de ritmo cardiaco para la realización de cálculos estadísticos (análisis en el

dominio del tiempo) y para la medición de indicadores sobre el espectro de esta señal (análisis en el dominio de la frecuencia) [2]. Para el cálculo de los indicadores del espectro de la señal de ritmo cardiaco se utilizan técnicas tradicionales basadas en la interpolación lineal y remuestreo de la señal de ritmo cardiaco con la finalidad de calcular la densidad espectral de potencia (PSD) a través de la Transformada Rápida de Fourier (FFT).

El método basado en la FFT, puede causar un efecto pasa bajos en la PSD de la señal de ritmo cardiaco [5]. Para evitar el efecto pasa bajos, se ha optado por estrategias no convencionales para la estimación de la (PSD), aplicadas a datos irregularmente muestreados. Siendo la técnica de Lomb, utilizada por sus beneficios para estimar la PSD a partir de las muestras obtenidas a tiempos irregulares sin la necesidad de un remuestreo e interpolación. El método de Lomb [3] toma un arreglo de datos $x_n(t_n)$ el cual contiene las muestras adquiridas para cada tiempo t_n y procede a calcular la densidad espectral de potencia para cada valor de frecuencia f como se observa en (1).

¹ González B. Javier E. Ing., Facultad de Ingeniería Electrónica. Universidad Santo Tomás. Bogotá D.C. Colombia. (e_mail: javier.gonzalez@ieee.org).

² Forero Edwin F. Ing., Facultad de Ingeniería Electrónica. Universidad Santo Tomás. Bogotá D.C. Colombia. (e_mail: fedwinforero@usantotomas.edu.co).

³ Jiménez Fabián R. Ing., Facultad de Ingeniería Electrónica. Universidad Santo Tomás. Bogotá D.C. Colombia. (e_mail: fabianjimenez@usantotomas.edu.co).

⁴ Marino M. Iván, Ing., Facultad de Ingeniería de Telecomunicaciones. Universidad Santo Tomás. Bogotá D.C. Colombia. (e_mail: marinobolivar@yahoo.com.co).

$$P(f) = \frac{1}{2\delta^2} \left[\frac{\left(\sum_{n=1}^N (x_n(t_n) - \bar{x}) \cdot \cos(2\pi f((t_n) - \tau)) \right)^2}{\sum_{n=1}^N \cos^2(2\pi f((t_n) - \tau))} + \frac{\left(\sum_{n=1}^N (x_n(t_n) - \bar{x}) \cdot \sin(2\pi f((t_n) - \tau)) \right)^2}{\sum_{n=1}^N \sin^2(2\pi f((t_n) - \tau))} \right] \quad (1)$$

En la ecuación 2, δ^2 corresponde a la a la varianza de los datos $x_n(t_n)$ y el cálculo de la variable τ se realiza a través de la expresión (2).

$$\tan(4\pi\tau) = \frac{\sum_{n=1}^N \sin(4\pi t_n)}{\sum_{n=1}^N \cos(4\pi t_n)} \quad (2)$$

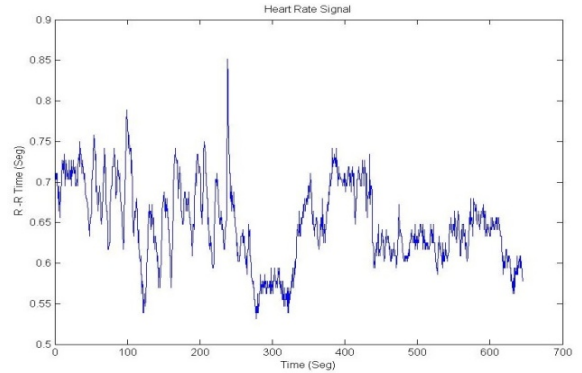
El método de Lomb ha tenido gran utilidad en aplicaciones del procesamiento de la señal de ritmo cardiaco, la cual presenta un escenario de datos (tiempo de duración entre latido y latido) que se presentan de forma irregular en el tiempo [5,6], también se ha aplicado en el estudio de componentes de audio[7], análisis de señales de múltiples canales [8], análisis de componentes armónicas de sistemas de potencia eléctrica [9], estudios de grabaciones con componentes superpuestas [10], análisis de adquisiciones no regulares de sistemas laser [11] y análisis de la frecuencia respiratoria [12].

En el cálculo de la PSD, es común la presencia de componentes de rizados que puede distorsionar el espectro de los datos bajo estudio. Por lo cual, el objetivo de este artículo es ilustrar los resultados obtenidos al aplicar el promediado de espectros de señales de ritmo cardiaco para la disminución de rizados y obtener un suavizado de la PSD estimada. La aplicación de la técnica de Lomb a través del promediado de espectros ha presentado un excelente resultado sin causar un efecto pasa bajos en la PSD comparado con la técnica basada en la interpolación lineal y el remuestreo.

2. MATERIALES Y MÉTODOS

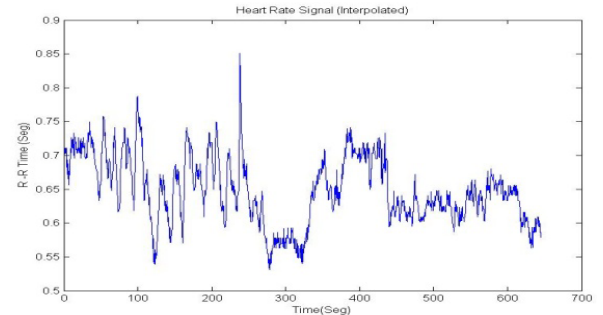
En el estudio en el dominio de la frecuencia de una señal de ritmo cardiaco, también se cuenta con el escenario de una serie de tiempo con muestreo irregular. En la figura 1 se puede apreciar una señal de ritmo cardiaco adquirida de la base de datos señales electro fisiológicas de personas con ritmo sinusal normal del repositorio Physionet.

FIGURA 1
De rizados en la densidad espectral de potencia calculada en una señal de ritmo cardiaco
Señal de ritmo cardiaco tomada de la base de datos Physionet



Como se pudo apreciar en la figura 1, los datos contenidos en la señal de ritmo cardiaco presentan un muestreo irregular. Tradicionalmente, para estimar la PSD y poder hacer un análisis en el dominio de la frecuencia, se procede a realizar un proceso de interpolación lineal y remuestreo.

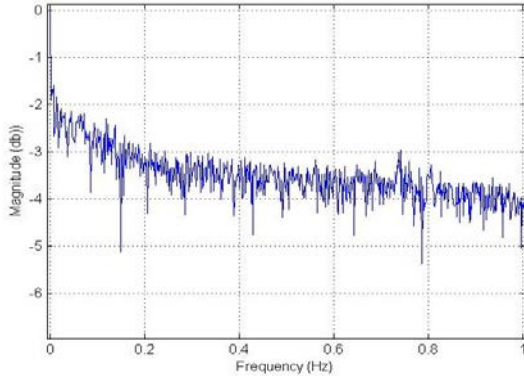
FIGURA 2
De rizados en la densidad espectral de potencia calculada en una señal de ritmo cardiaco
Resultado obtenido después de interpolar y remuestrear la señal de ritmo cardiaco



La figura 2 permite observar la señal de ritmo cardiaco después de un proceso de interpolación lineal y remuestreo. El periodo de muestreo se escogió tomando la mitad del menor tiempo de separación que presentó la señal de ritmo cardiaco.

FIGURA 3

De rizados en la densidad espectral de potencia calculada en una señal de ritmo cardiaco
Cálculo de la PSD a través del cálculo de la FFT



La figura 3 contiene el resultado de estimar la PSD a través de la Transformada Rápida de Fourier. Se puede observar que a partir del valor de frecuencia equivalente 0.3 Hz, se experimenta una atenuación de -3db, a partir de la componente de 0.4 Hz la atenuación se incrementa a -3.5db y en la componente de 0.8 Hz la atenuación ya tiene un valor de 6-db, como consecuencia del proceso de interpolación lineal y remuestreo. La figura 3 también permite observar la cantidad de rizados presente en la PSD. El efecto pasa bajo y los rizados pueden alterar los índices estadísticos calculados para el estudio en el dominio de la frecuencia de la V.F.C.

A partir de la señal con frecuencia de muestreo fija, se puede generar una estrategia basada en el periodograma de Welch para la estimación de la PSD a partir de promedios de espectros de l segmentos de la señal con tamaño L . La PSD para cada segmento se puede calcular a través de la ecuación 3.

$$XF_l = \left| \sum_{k=0}^{L-1} x_l(n) e^{j\frac{2\pi nk}{L}} \right|^2 \quad (3)$$

Donde la variable l es un número entero que cumple la condición: $0 < l < L$. La PSD promedio se calcula tomando cada resultado XF_l como se puede ver en la ecuación 4.

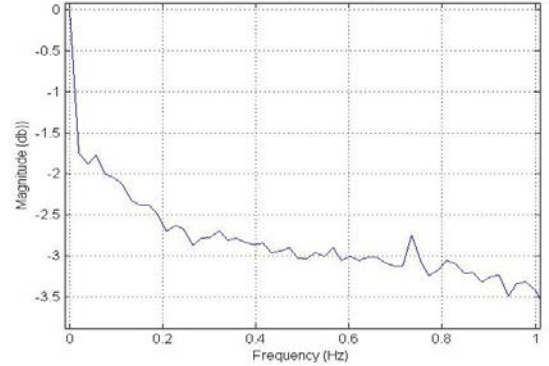
$$P_A = \frac{1}{L} \sum_{l=0}^{L-1} XF_l \quad (4)$$

La figura 4 permite observar el resultado de estimar la PSD a través del promediado de espectros. En este caso se ha segmentado la señal de ritmo cardiaco para el cálculo de la PSD a través del

promedio de los espectros de cada segmento. El resultado contenido en la figura 4 indica que el promediado de espectros ha disminuido el contenido de rizados, pero mantiene el efecto pasa bajos en la PSD.

FIGURA 4

De rizados en la densidad espectral de potencia calculada en una señal de ritmo cardiaco
Cálculo de la PSD a través del promedio de espectros usando la FFT



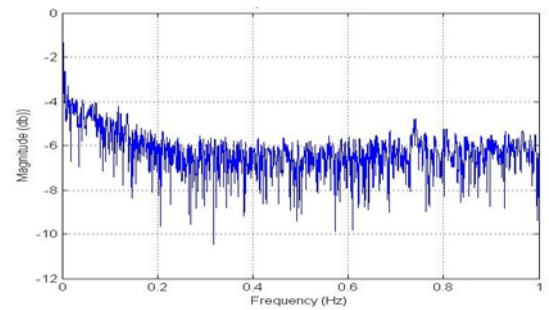
3. RESULTADOS

El principal objetivo de este trabajo es poder comprobar los beneficios de la técnica de Lomb para el cálculo de la PSD de la señal de ritmo cardiaco y la atenuación de los rizados.

Al tomar los datos de señal de ritmo cardiaco (Figura 1), sin aplicar procesos de interpolación lineal y remuestreo, es posible calcular la PSD a través de la técnica de Lomb. La figura 5 ilustra el resultado obtenido.

FIGURA 5

De rizados en la densidad espectral de potencia calculada en una señal de ritmo cardiaco
PSD estimada a través del método de Lomb de los datos contenidos en la señal de ritmo cardiaco adquirida de la base de datos Physionet



La figura 5 permite observar el resultado del cálculo de la PSD a través de la técnica de Lomb, con gran cantidad de rizados. A partir de la componente de

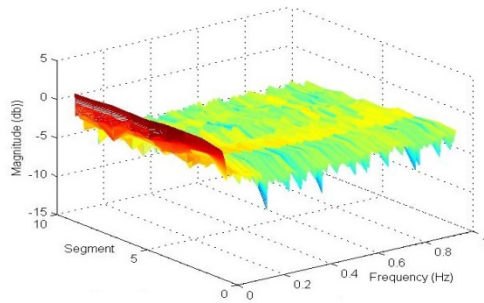
0.2 Hz existe un efecto pasa bajos que se mantiene constante a diferencia del resultado obtenido con la técnica basada en la FFT.

Con la finalidad de poder obtener una atenuación de los rizados, se ha procedido a segmentar la señal de ritmo cardiaco. A cada segmento se ha calculado su respectiva PSD usando la técnica de Lomb. El resultado obtenido se puede apreciar en la figura 6.

FIGURA 6

De rizados en la densidad espectral de potencia calculada en una señal de ritmo cardiaco

Cálculo de PSD de cada segmento de la señal de ritmo cardiaco

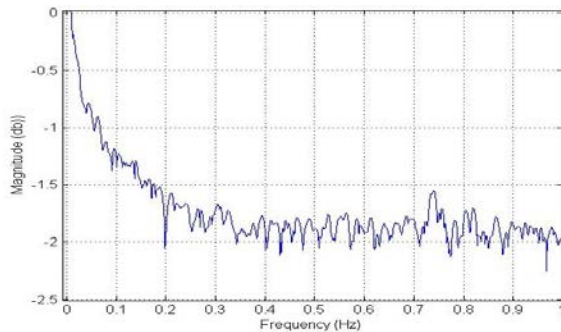


La figura 6 ilustra un efecto pasa bajos constante a partir de 0.2 Hz. También se puede apreciar los rizados de la PSD de cada segmento.

FIGURA 7

De rizados en la densidad espectral de potencia calculada en una señal de ritmo cardiaco

Resultado obtenido después del proceso de promediado de espectros calculados a través de la técnica de Lomb para estimación de PSD

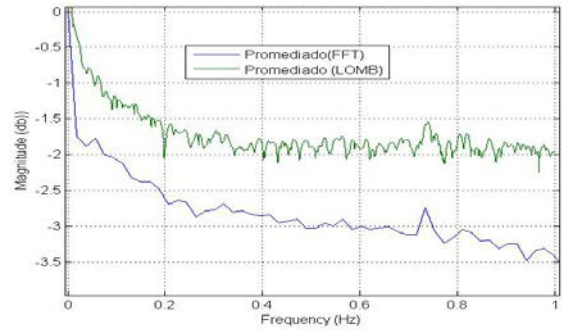


El resultado final del proceso se puede evidenciar en la figura 7. En el cual se puede observar el efecto de suavizado en la PSD calculada a través del promediado de espectros.

FIGURA 8

De rizados en la densidad espectral de potencia calculada en una señal de ritmo cardiaco

Comparación de resultados obtenidos por ambos métodos: FFT y Lomb



La comparación de los resultados obtenidos con ambos procesos de estimación de la PSD a través de promediados de espectros: Lomb y FFT, se pueden observar en la figura 8. Los resultados obtenidos de la comparación entre ambos métodos permiten ver la ventaja que ofrece el uso de la técnica de Lomb en el proceso de atenuación de rizados en la estimación de la PSD.

4. CONCLUSIONES

El análisis en el dominio de la frecuencia de la señal de ritmo cardiaco exige el uso de técnicas orientadas a la estimación de la Densidad Espectral de Potencia para datos irregularmente muestreados. La técnica de Lomb permite calcular la PSD, tomando los datos de la señal de ritmo cardiaco sin importar el tiempo de ocurrencia de cada uno. Al estimar la PSD por medio de la técnica de Lomb, se puede evidenciar la presencia de rizados en el espectro calculado.

En este artículo se ha elaborado con señales de ritmo cardiaco y se ha evidenciado que el promediado de espectros calculados a partir del método de Lomb puede aportar en el suavizado de la densidad espectral de potencia.

El suavizado del espectro, permite facilitar el análisis en el dominio de la frecuencia de la señal de ritmo cardiaco y permite tener una mejor caracterización de la variabilidad de la frecuencia cardiaca.

5. AGRADECIMIENTO

Los resultados obtenidos en este trabajo fueron logrados dentro de los objetivos del proyecto, financiado por la convocatoria interna de proyectos de grupos de investigación, realizada por la unidad de investigación y postgrados de la Universidad Santo.

REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS

- [1]. NIMUNKAR, A.J.; TOMPKINS, W.J. (2007). "R-peak Detection and Signal Averaging for Simulated Stress ECG using EMD". Engineering in Medicine and Biology Society. EMBS. 29th Annual International Conference of the IEEE, vol., no., pp.1261-1264, 22-26 Aug.
- [2]. MALIK M. (2010). "Clinical Guide to Cardiac Autonomic Tests". Kluwer Academic Publisher. Netherland. Pag 149.
- [3]. LOMB N. R. (1976). "Least-squares frequency analysis of unequally spaced data" Astrophysics and Space Science, vol. 39, no. 2, pp. 447-462.
- [4]. PRESS W. (2007). "Numerical Recipes" 3r Edition. Hong Kong, China: Cambridge University Press. Pp685.
- [5]. LAGUNA P., G. B. MOODY AND R.G. MARK (1998). "Power Spectra Density of Unevenly Sampled Data by Least-Square Analysis: Performance and Application to Heart Rate Signals". IEEE Trans. On Biomedical Engineering Vol. 45, No. 6. June. pp. 698-715.
- [6]. SHAO-YEN TSENG; WAI-CHI FANG, (2010). "An effective heart rate variability processor design based on time-frequency analysis algorithm using windowed Lomb periodogram" Biomedical Circuits and Systems Conference (BioCAS), IEEE , vol., no., pp.82-85, 3-5 Nov.
- [7]. ZHIGUO ZHANG; SHING-CHOW CHAN. (2005). "Robust adaptive Lomb periodogram for time-frequency analysis of signals with sinusoidal and transient components" Acoustics, Speech, and Signal Processing. Proceedings. (ICASSP '05). IEEE International Conference on, vol.4, no., pp. iv/493- iv/496 Vol. 4, 18-23.
- [8]. ZHANG, Z.G.; CAI, X.L.; CHAN, S.C.; HU, Y.; HU, L.; CHANG, C.Q. (2009). "Time-frequency coherence analysis of multi-channel eventrelated potential using adaptive windowed Lomb periodogram" Neural Engineering. NER '09. 4th International IEEE/EMBS Conference on, vol., no., pp.657-660, April 29 2009-May 2 2009.
- [9]. ZHANG, Z.G.; CHAN, S.C. (2010). "Harmonic analysis of power system signals using a new regularized adaptive windowed Lomb periodogram," Green Circuits and Systems (ICGCS), 2010 International Conference on, vol., no., pp.567-572, 21-23 June.
- [10]. THONG, T.; MCNAMES, J.; ABOY, M. (2004). "Lomb-Wechperiodogram for non-uniform sampling" Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE, vol.1, no., pp.271-274, 1-5 Sept.
- [11]. HUI-BOMENG; YAN-LI GAO; ZHI-QIANG LIU; YAN-FANG YU; JIAN-HUA WU. (2010). "Analysis of Turbulent Characteristics of Unevenly Velocity Signals in KSM Used Empirical Mode Decomposition and Lomb Periodogram" Information Engineering and Computer Science (ICIECS), 2010 2nd International Conference on, vol., no., pp.1-4, 25-26 Dec.
- [12]. VASU, V.; FOX, N.; HENEGHAN, C.; SEZER, S. (2010). "Using the Lomb periodogram for non-contact estimation of respiration rates" Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, vol., no., pp.2407-2410, Aug. 31 2010-Sept. 4.

CONSTRUCCIÓN DE KERNELS Y FUNCIONES DE DENSIDAD DE PROBABILIDAD

Luis Rodríguez Ojeda¹

Resumen. En este artículo se describen los detalles del conocido método kernel para construir una función de densidad de probabilidad para una muestra univariada. Se proponen nuevas formas para los kernels y criterios adicionales para su elección y para estimar el ancho de banda óptimo. Como soporte para esta investigación se instrumentó un software para experimentación y obtención de resultados gráficos y numéricos.

Palabras clave: Kernel. Densidad de Probabilidad. Ancho de banda. Estimación no Paramétrica

Abstract. This paper describes the details of the well known kernel method used to construct a probability density function for a univariate sample. New forms for the kernels are proposed and additional criteria for their election and for estimating the optimal bandwidth. As support for this research a software was implemented for experimentation and obtaining of graphical and numerical results.

Keywords: Kernel. Probability Density. Bandwidth. Nonparametric Estimation

Recibido: Agosto 2013

Aprobado: Septiembre 2013

1. INTRODUCCIÓN

La distribución de probabilidad de una variable aleatoria continua X se describe mediante una función denominada función de densidad $f(x)$ con la cual se pueden determinar valores de probabilidad con la definición:

$$P(a \leq X \leq b) = \int_a^b f(x) dx$$

Si $f(x)$ no es un modelo conocido, es de interés para la investigación poder estimar $f(x)$ a partir de una muestra de observaciones x_1, x_2, \dots, x_n que suponemos son resultados independientes y tienen la misma distribución de probabilidad. Como ocurre frecuentemente en problemas de ingeniería, la obtención de estos datos se basa en ensayos que involucran tiempo y costo, por lo tanto su cantidad es limitada. El método kernel utiliza un conjunto de datos que provienen de una distribución continua, univariada y desconocida para aproximar esta función. Los fundamentos matemáticos son conocidos pero la investigación aún continua en la selección de los parámetros de ajuste adecuados para su aplicación. En este documento se describe en detalle la formulación y se proponen algunos criterios para la aplicación de este método. Los kernels son funciones que se asocian a cada uno de los datos. Entonces, la suma ponderada de estas funciones es un estimador para aproximar la función de densidad desconocida. Estas funciones son objetos matemáticos conocidos, pero en esta contribución se desarrollan nuevas formas basadas en consideraciones geométricas y se establecen otros criterios para compararlos.

Como ocurre en muchas áreas del conocimiento, la formulación desarrollada no es adecuada para el tratamiento manual, por lo que se ha construido un programa computacional para su aplicación. Existen programas para usar el método Kernel, pero no incluyen los modelos para experimentar como se describe en este trabajo.

El programa usa como soporte el lenguaje MATLAB y constituye un pequeño laboratorio con el que se pueden probar diferentes modelos para construir el estimador de la función $f(x)$ realizando pruebas con los parámetros de ajuste. Los resultados que se muestran son gráficos, simbólicos y numéricos.

2. CONSTRUCCIÓN DE KERNELS

2.1 Definición de kernel

Un kernel es una función de variable real: $K: R \rightarrow R$ con las siguientes propiedades

- a) $K(x) \in [0, \infty)$, $x \in [-1, 1]$
- b) $K(x) = 0$, $x \notin [-1, 1]$
- c) $K(x) = K(-x)$
- d) $\int_{-1}^1 K(x) dx = 1$
- e) $\int_{-1}^1 xK(x) dx = 0$
- f) $\int_{-1}^1 x^2 K(x) dx \in R^+$

De aquí en adelante, nos referiremos solamente al intervalo en el cual K no es negativo. En la definición este intervalo es $[-1, 1]$ pero puede modificarse mediante un parámetro. Además, es deseable que K sea diferenciable

2.2 PARAMETRIZACIÓN DE KERNELS

Sea $h \in \mathbb{R}^+$, el kernel parametrizado en h es

¹ Rodríguez Luís, M.Sc., Departamento de Matemáticas ESPOL. (e_mail:lrodrig@espol.edu.ec).

$$K_h(x) = \frac{1}{h} K\left(\frac{x}{h}\right), \quad x \in [-h, h]$$

Esta modificación mantiene las propiedades anteriores, pero referidas al intervalo $[-h, h]$. h se denomina el ancho de banda de K y es la semi-amplitud del kernel en el intervalo de interés.

La propiedad d) se prueba mediante la sustitución: $u = x/h: x=-h \Rightarrow u = -1, x=h \Rightarrow u=1$

$$x=hu, \quad dx=hdu$$

$$\int_{-h}^h \frac{1}{h} K\left(\frac{x}{h}\right) dx = \int_{-1}^1 \frac{1}{h} K(u) hdu$$

$$= \int_{-1}^1 K(u) du = 1$$

Es importante interpretar el rol de h en $K_h(x)$

Si h se incrementa, $h > 1$, la amplitud de K aumenta, pero el factor $1/h$, reduce el rango de K para mantener el área igual a 1

Si h se reduce, $h < 1$, la amplitud de K se reduce, pero el factor $1/h$ incrementa el rango de K para mantener el área igual a 1.

2.3 TRASLACIÓN DE KERNELS

El kernel se puede centrar en cualquier punto $x_i \in \mathbb{R}$. El kernel parametrizado en h y centrado en x_i es:

$$K_h(x) = \frac{1}{h} K\left(\frac{x-x_i}{h}\right), \quad x \in [x_i - h, x_i + h]$$

Esta modificación mantiene las propiedades anteriores en el nuevo intervalo de interés:

$$[x_i - h, x_i + h]$$

Para verificar la propiedad d) se realiza la sustitución: $u = \frac{x-x_i}{h}: x = x_i - h \Rightarrow u = -1,$

$$x = x_i + h \Rightarrow u = 1, \quad dx = hdu$$

$$\int_{x_i-h}^{x_i+h} \frac{1}{h} K\left(\frac{x-x_i}{h}\right) dx = \int_{-1}^1 \frac{1}{h} K(u) hdu$$

$$= \int_{-1}^1 K(u) du = 1$$

Se concluye que la función $K_h(x) = \frac{1}{h} K\left(\frac{x-x_i}{h}\right), x \in [x_i - h, x_i + h]$ es una función de densidad de probabilidad centrada en el punto x_i .

3. MODELOS DE KERNELS

Se describen a continuación algunos kernels que son bien conocidos.

3.1 KERNEL RECTANGULAR O UNIFORME

Es un rectángulo que se coloca sobre cada punto. Al interactuar con los kernels de los otros puntos, el efecto en la suma es un cambio abrupto.

$$K_h(x) = 0.5, \quad x \in [-1, 1]$$

3.2 KERNEL TRIANGULAR

Es un triángulo que se coloca sobre cada punto. Al interactuar con los otros kernels el efecto combinado es lineal pero más liso que los rectángulos

$$K(x) = 1 - |x|, \quad x \in [-1, 1]$$

3.3 KERNEL DE EPANECHNIKOV

Es el kernel más estudiado. Es un segmento del perfil de un arco de parábola que se coloca sobre cada punto.

$$K(x) = \frac{3}{4}(1-x^2), \quad x \in [-1, 1]$$

3.4 KERNEL GAUSSIANO

Este kernel es un caso especial. Para este kernel se define como intervalo el conjunto \mathbb{R} , por lo que cada kernel influye en todos los otros kernels colocados en los puntos de la muestra. La suma resultante es continua y suave.

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad x \in (-\infty, +\infty)$$

3.5 KERNEL BIWEIGHT O CUÁRTICO

$$K(x) = \frac{15}{16}(1-x^2)^2, \quad x \in [-1, 1]$$

3.6 KERNEL ARCO COSENO

$$K(x) = \frac{\pi}{4} \cos\left(\frac{\pi}{2}x\right), \quad x \in [-1, 1]$$

4. CONSTRUCCIÓN DE KERNELS

El diseño de nuevas formas para los kernels es un ejercicio matemático. El diseño se basa en consideraciones geométricas. Estos objetos matemáticos son los componentes con los que se construye el estimador de la función de densidad. Las formas propuestas difieren en algún aspecto las formas conocidas que normalmente son funciones de potencia par. Las funciones propuestas usan polinomios cúbicos entre otros.

4.1 KERNEL CÚBICO NATURAL

El perfil son dos segmentos de un polinomio cúbico que se conectan manteniendo continuidad hasta la primera derivada y con los extremos libres, sin curvatura. Se ilustra su obtención. Similarmente se obtienen los otros. Polinomio cúbico en el tramo derecho:

$$K(x) = ax^3 + bx^2 + cx + d, \quad x \in [0, 1]$$

$$K'(x) = 3ax^2 + 2bx + c, \quad K''(x) = 6ax + 2b$$

Condiciones geométricas:

a) $K(1) = 0 \Rightarrow a + b + c + d = 0$

b) $K'(0) = 0 \Rightarrow c = 0$

c) $K''(1) = 0 \Rightarrow 3a + b = 0$

La altura t del kernel es el parámetro para convertirlo en función de densidad:

d) $K(0) = t \Rightarrow d = t$

Resolviendo y sustituyendo se obtiene

$$K(x) = \frac{t}{2}x^3 - \frac{3}{2}tx^2 + t$$

Para que K sea función de densidad:

$$\int_0^1 K(x)dx = \frac{1}{2} \Rightarrow t = \frac{4}{5}$$

Entonces

$$K(x) = \frac{2}{5}(x^3 - 3x^2 + 2), \quad x \in [0,1]$$

$$K(-x) = \frac{2}{5}(-x^3 - 3x^2 + 2), \quad x \in [-1,0]$$

Se pueden escribir con una regla:

$$K(x) = \frac{2}{5}(|x|^3 - 3x^2 + 2), \quad x \in [-1,1]$$

4.2 KERNEL CÚBICO PLANO

El perfil son dos segmentos de un polinomio cúbico que se conectan manteniendo continuidad hasta la segunda derivada

$$K(x) = \frac{2}{3}(1 - |x|^3), \quad x \in [-1,1]$$

4.3 KERNEL CÚBICO SUJETO

El perfil son dos segmentos de un polinomio cúbico que se conectan manteniendo continuidad hasta la primera derivada, pero con los extremos horizontales (primera derivada nula), con el objetivo de que al combinarse con otros kernels, la interacción sea lisa.

$$K(x) = 2|x|^3 - 3x^2 + 1, \quad x \in [-1,1]$$

4.4 KERNEL COSENO

El perfil es un segmento modulado de la función coseno.

$$K(x) = \frac{1}{2}(\cos(\pi x) + 1), \quad x \in [-1,1]$$

4.5 KERNEL ARCO CIRCULAR

El perfil es un arco de una circunferencia:

$$K(x) = \sqrt{1.145834 - x^2} - 0.381882, \quad x \in [-1,1]$$

En la Figura 18 al final de este artículo se muestra el perfil de algunos kernels formulados.

5. CONSTRUCCIÓN DE FUNCIONES DE DENSIDAD DE PROBABILIDAD

Sea X una variable aleatoria con distribución de probabilidad continua, univariada y

desconocida $f(x)$ de la cual se dispone de una muestra aleatoria de n observaciones independientes: x_1, x_2, \dots, x_n . El objetivo es usar estos datos para obtener un estimador $\hat{f}(x)$ de la función de densidad de probabilidad $f(x)$

El método clásico para construir $\hat{f}(x)$ es el histograma que agrupa los datos en clases con amplitud que debe elegirse. La representación gráfica son rectángulos excluyentes cuya altura es el conteo de observaciones en cada clase y se denomina frecuencia de clase.

El histograma es una función continua pero cambia con saltos entre clases. La altura de cada rectángulo puede asociarse a valores de probabilidad. Esta altura solo depende de la cantidad de datos incluidos en cada clase ignorando la influencia de los datos adyacentes aunque estén muy próximos.

5.1 EL MÉTODO KERNEL

Un kernel es una función de densidad. Si se coloca un kernel en cada uno de los datos de la muestra, la suma ponderada de estas funciones también será una función de densidad de probabilidad. Esta suma es una función continua que suaviza el perfil de la distribución captando la influencia de los datos cercanos y constituye el estimador $\hat{f}(x)$ del modelo teórico del cual provienen los datos, permitiendo observar diferencias que los rectángulos del histograma no puede mostrar.

$$\text{Sea } K_h(x) = \frac{1}{h}K\left(\frac{x-x_i}{h}\right), x \in [x_i - h, x_i + h]$$

kernel parametrizado y centrado en cada punto $x_i, i = 1, 2, 3, \dots, n$.

K_h es una función de densidad de probabilidad. Si cada kernel se multiplica por $1/n$, entonces la suma de los n kernels también será una función de densidad de probabilidad.

Definición: Estimador por kernels $\hat{f}(x)$:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_h(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x-x_i}{h}\right)$$

$$x \in [x_i - h, x_i + h] \text{ en cada kernel } i$$

Intervalo de $\hat{f}(x)$: $[x_i - h, x_i + h]$

Se supondrá que $x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n$

El ancho de banda h es el parámetro de ajuste o suavizado de $\hat{f}(x)$ su elección es crítica para el modelo.

Mientras más pequeño es h , más concentrada está la contribución del kernel en cada punto x_i . Mientras más grande es h , mayor es la influencia e interacción del kernel hacia los puntos vecinos.

En el límite, cuando $h \rightarrow 0$, la contribución de cada kernel estará concentrada en cada punto x_i así el estimador $\hat{f}(x)$ tendrá una distribución puntual concentrada en cada dato. Por otra parte, cuando $h \rightarrow \infty$, la distribución de $\hat{f}(x)$ se aplanará, con un solo cúmulo y con mayor

dispersión. Es necesario buscar un ancho de banda adecuado para construir el estimador.

Ejemplo. Dados los siguientes datos de una muestra aleatoria (variables independientes y con la misma distribución de probabilidad), analizar un modelo de densidad de probabilidad:

X : 1.1, 2.1, 2.3, 2.7, 3.8

Todos los resultados gráficos y numéricos que se muestran a continuación fueron obtenidos con el software KDEN desarrollado para esta investigación. El programa puede mostrar también el modelo matemático del estimador $\hat{f}(x)$ y algunas medidas estadísticas de interés.

FIGURA 1

Construcción de kernels y funciones de densidad de probabilidad
Diagrama de puntos

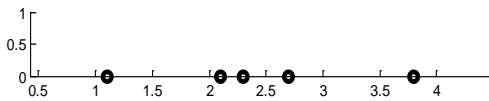
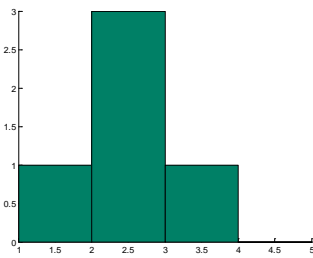


FIGURA 2

Construcción de kernels y funciones de densidad de probabilidad
Histograma, amplitud de clase = 1

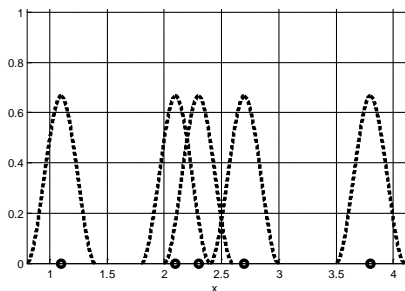


La representación del histograma muestra solamente parte de la información de los datos.

Kernel elegido: Cúbico Sujeto

FIGURA 3

Construcción de kernels y funciones de densidad de probabilidad
Gráfico de kernels con $h=0.3$



Si h es muy pequeño, los kernels están concentrados en cada punto y no interactúan con los otros. La suma es la función de densidad $\hat{f}(x)$ y su perfil se muestra superpuesto al gráfico de los kernels en la siguiente figura.

FIGURA 4

Construcción de kernels y funciones de densidad de probabilidad
Kernels y la función de densidad $\hat{f}(x)$ con $h=0.3$

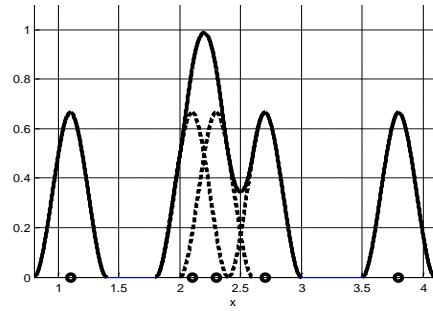
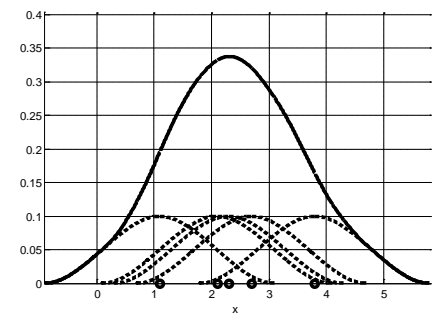


FIGURA 5

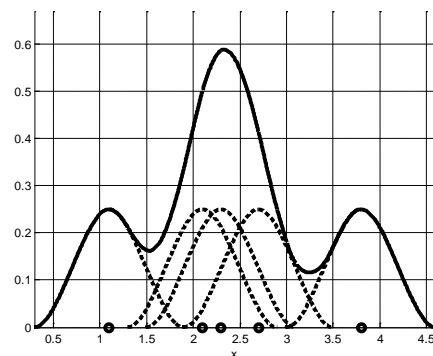
Construcción de kernels y funciones de densidad de probabilidad
Kernels y la función de densidad $\hat{f}(x)$ con $h=2.0$



Si h es muy grande hay un sobre ajuste y la distribución sumada será más plana, con un solo cúmulo y con mayor dispersión

FIGURA 6

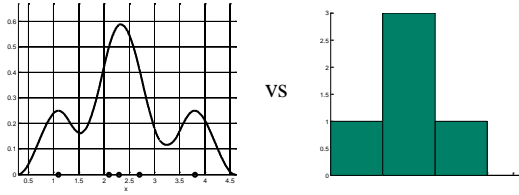
Construcción de kernels y funciones de densidad de probabilidad
Kernels y la función de densidad $\hat{f}(x)$ con $h=0.8$



Si el valor de h es adecuado, la distribución se suaviza y permite observar más detalles de la distribución de probabilidad.

FIGURA 7

Construcción de kernels y funciones de densidad de probabilidad
Función de densidad $\hat{f}(x)$ con $h=0.8$ y el histograma



Cálculo de probabilidad

Calcular la probabilidad que la variable X tome un valor entre 1.5 y 2.5

$$P(1.5 \leq X \leq 2.5) = \int_{1.5}^{2.5} \hat{f}(x) dx = 0.3973$$

Medidas estadísticas de $\hat{f}(x)$

Para el ejemplo anterior:

Media	2.4000
Varianza	0.8533
Sesgo	0.0064
Rango	[0.3, 4.6]
Amplitud	4.3000
Mediana	2.3579
Primer Cuartil	1.8299
Tercer Cuartil	2.9229

En resumen, el método kernel proporciona un estimador continuo $\hat{f}(x)$. Si se elige adecuadamente el ancho de banda h los rectángulos del histograma son reemplazados por cúmulos suavizados que se solapan e interactúan de tal manera que al sumarlos producen una función que presenta detalles que el histograma no puede mostrar.

5.2 PROPIEDADES DE LA VARIABLE ALEATORIA CON DENSIDAD $\hat{f}(X)$

Sea X : variable aleatoria con densidad $\hat{f}(x)$

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n K_h(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x-x_i}{h}\right),$$

$K_h(x), x \in [x_i - h, x_i + h]$, Kernel

h : Ancho de banda

$\hat{f}(x), x \in [x_1 - h, x_n + h]$

$x_1 \leq x_2 \leq x_3 \leq \dots \leq x_n$

5.2.1 $\hat{f}(X)$ ES UNA FUNCIÓN DE DENSIDAD

Demostración

$$\begin{aligned} \int_{x_1-h}^{x_n+h} \hat{f}(x) dx &= \int_{x_1-h}^{x_n+h} \frac{1}{n} \sum_{i=1}^n \frac{1}{h} K\left(\frac{x-x_i}{h}\right) dx \\ &= \frac{1}{nh} \sum_{i=1}^n \int_{x_i-h}^{x_i+h} K\left(\frac{x-x_i}{h}\right) dx \end{aligned}$$

Con la sustitución: $u = \frac{x-x_i}{h} : x = x_i - h \Rightarrow u = -1$
 $x = x_i + h \Rightarrow u = 1, dx = hdu$

$$\begin{aligned} \frac{1}{nh} \sum_{i=1}^n \int_{x_i-h}^{x_i+h} K\left(\frac{x-x_i}{h}\right) dx &= \frac{1}{n} \sum_{i=1}^n \int_{-1}^1 K(u) du \\ &= \frac{1}{n} (n) = 1 \end{aligned}$$

5.2.2 VALOR ESPERADO DE LA VARIABLE ALEATORIA

Sea $x \in [x_1 - h, x_n + h]$ una variable aleatoria con distribución $\hat{f}(x)$. Su valor esperado:

$$E(x) = \int_{x_1-h}^{x_n+h} x \hat{f}(x) dx = \frac{1}{nh} \sum_{i=1}^n \int_{x_i-h}^{x_i+h} x K\left(\frac{x-x_i}{h}\right) dx$$

Mediante la sustitución

$$u = \frac{x-x_i}{h} : x = x_i - h \Rightarrow u = -1, x = x_i + h \Rightarrow u = 1, x = hu+x_i, dx = hdu$$

$$\begin{aligned} E(x) &= \frac{1}{n} \sum_{i=1}^n \int_{-1}^1 (hu + x_i) K(u) du = \\ &= \frac{1}{n} \sum_{i=1}^n \int_{-1}^1 u K(u) du + \frac{1}{n} \sum_{i=1}^n x_i \int_{-1}^1 K(u) du \\ &= \frac{h}{n} (0) + \frac{1}{n} \sum_{i=1}^n x_i (1) = \frac{1}{n} \sum_{i=1}^n x_i = \bar{X} \end{aligned}$$

$$E(x) = \bar{X}$$

El valor esperado de la variable aleatoria x con densidad $\hat{f}(x)$ coincide con la media muestral, independientemente del kernel K .

5.2.3 VARIANZA DE LA VARIABLE ALEATORIA

Sea $x \in [x_1 - h, x_n + h]$ una variable aleatoria con densidad $\hat{f}(x)$

Su varianza:

$$\begin{aligned} \sigma_x^2 &= E(x^2) - E^2(x) \\ E(x^2) &= \int_{x_1-h}^{x_n+h} x^2 \hat{f}(x) dx = \frac{1}{nh} \sum_{i=1}^n \int_{x_i-h}^{x_i+h} x^2 K\left(\frac{x-x_i}{h}\right) dx \end{aligned}$$

Mediante la sustitución:

$$u = \frac{x-x_i}{h} : x = x_i - h \Rightarrow u = -1, x = x_i + h \Rightarrow u = 1, x = hu+x_i, dx = hdu$$

$$\begin{aligned} E(x^2) &= \frac{1}{n} \sum_{i=1}^n \int_{-1}^1 (hu + x_i)^2 K(u) du \\ &= \frac{1}{n} \sum_{i=1}^n \int_{-1}^1 u^2 h^2 K(u) du + \\ &= \frac{1}{n} \sum_{i=1}^n \int_{-1}^1 2uhx_i K(u) du + \\ &= \frac{1}{n} \sum_{i=1}^n x_i^2 \int_{-1}^1 K(u) du \\ &= \frac{h^2}{n} \sum_{i=1}^n \int_{-1}^1 u^2 K(u) du + \\ &= \frac{2h}{n} \sum_{i=1}^n x_i \int_{-1}^1 u K(u) du + \frac{1}{n} \sum_{i=1}^n x_i^2 \int_{-1}^1 K(u) du \\ &= \frac{h^2}{n} \sum_{i=1}^n \sigma_K^2 + \frac{2h}{n} \sum_{i=1}^n x_i (0) + \frac{1}{n} \sum_{i=1}^n x_i^2 (1) \\ &= h^2 \sigma_K^2 + \frac{1}{n} \sum_{i=1}^n x_i^2 \end{aligned}$$

$$\sigma_x^2 = E(x^2) - E^2(x)$$

$$= h^2 \sigma_K^2 + \frac{1}{n} \sum_{i=1}^n x_i^2 - \left(\frac{1}{n} \sum_{i=1}^n x_i \right)^2$$

$$\sigma_x^2 = h^2 \sigma_K^2 + \hat{S}_x^2$$

En donde

σ_K^2 es la varianza del Kernel original

\hat{S}_x^2 es un valor asociado a la muestra (varianza muestral)

La varianza σ_x^2 de la variable aleatoria depende linealmente de la varianza del kernel, pero cuadráticamente del ancho de banda h , por ello este es el factor crítico.

5.3 CRITERIOS PARA ELEGIR EL KERNEL

5.3.1 VARIANZA

El primer criterio para elegir al kernel más eficiente es seleccionar el de menor varianza. Este valor se suma al construir el estimador $\hat{f}(x)$ y aumentará su dispersión. Los valores calculados están en la primera columna de la Tabla I al final de este artículo

5.3.2 ENLACE

Definimos el coeficiente $c = e^{-|d|}$, en donde d es el valor de la tangente en el borde.

El valor más alto es 1 como en el modelo normal, que se conecta con suavidad a los otros kernels. El menor valor es 0, como en el kernel rectangular. Los valores calculados están en la segunda columna de la Tabla I.

5.3.3 EFECTO DEL FACTOR ENLACE

Al sumar kernels el perfil resultante puede ser liso y continuo como el caso del kernel gaussiano, o cambiar abruptamente como el caso del kernel rectangular. Se debe seleccionar el kernel y el ancho de banda que permitan detectar detalles y una apariencia aceptable. A esto contribuye también el factor enlace. El siguiente ejemplo muestra dos kernels con diferente factor de enlace. Se observa la diferencia significativa alrededor de $x = 3$.

Suponer una muestra $X: 2, 4, 5$

FIGURA 8

Construcción de kernels y funciones de densidad de probabilidad
Kernel Epanechnikov, $h = 1.5$, enlace=0.2231

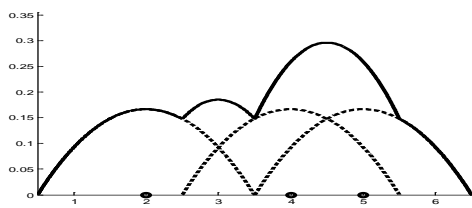
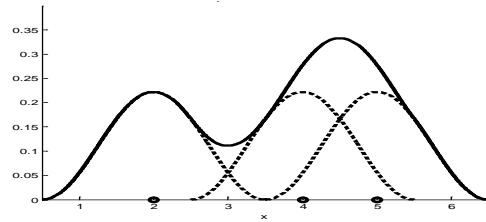


Figura 9

Construcción de kernels y funciones de densidad de probabilidad
Kernel Coseno, $h = 1.5$, enlace = 1



Como se verá más adelante, la valoración de los kernels cambia al analizarlos mediante otros criterios. Para esto se considerarán las restantes columnas de la Tabla I.

En esta primera valoración, es mejor el kernel cuya varianza sea menor, adicionalmente el coeficiente de enlace debe ser alto para que el perfil del estimador sea liso.

6. EFICIENCIA DEL ESTIMADOR $\hat{f}(x)$ CON RESPECTO A $f(x)$

Sean

$f(x)$: Función de densidad de probabilidad teórica (desconocida)

$\hat{f}(x)$: Estimador de $f(x)$ basado en los datos y el kernel elegido

Definiciones

$B(\hat{f}(x)) = E[\hat{f}(x)] - f(x)$: Sesgo del estimador $\hat{f}(x)$ con respecto a $f(x)$

$V(\hat{f}(x)) = E[\hat{f}(x) - E(\hat{f}(x))]^2$: Varianza del estimador $\hat{f}(x)$ con respecto a $E(\hat{f}(x))$

$ECM(\hat{f}(x)) = E[\hat{f}(x) - f(x)]^2$: Error cuadrático medio. (Medición de la diferencia puntual)

Si se desarrolla el cuadrado y se sustituyen las definiciones se obtiene

$$ECM(\hat{f}(x)) = B^2(\hat{f}(x)) + V(\hat{f}(x))$$

La siguiente definición determina la exactitud global del estimador, integrando ECM

$$ECMI(\hat{f}(x)) = \int_R ECM(\hat{f}(x)) dx = \int_R B^2(\hat{f}(x)) dx + \int_R V(\hat{f}(x)) dx$$

Desarrollo de los componentes

$$E(\hat{f}(x)) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} E \left[K \left(\frac{x-x_i}{h} \right) \right] = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} \int_R K \left(\frac{x-t}{h} \right) f(t) dt = \frac{1}{h} \int_R K \left(\frac{x-t}{h} \right) f(t) dt$$

Con las sustituciones: $z = \frac{x-t}{h}$, $t = x-hz$, $dt = -h dz$

$$E(\hat{f}(x)) = \int_R K(z) f(x-hz) dz$$

Esto muestra que $E(\hat{f}(x)) \neq f(x)$ pero $E(\hat{f}(x)) \rightarrow f(x)$ cuando $h \rightarrow 0$ siempre que f sea continua y acotada. Entonces $\hat{f}(x)$ es asintóticamente insesgado. Este resultado parece extraño pues cuando $h \rightarrow 0$, la distribución de $\hat{f}(x)$ se hace puntual.

Para obtener alguna aproximación, y suponiendo que f es diferenciable alrededor de x se desarrolla $f(x-hz)$ con la serie de Taylor. x es la variable aleatoria del estimador $\hat{f}(x)$, t, z son variables aleatorias del modelo teórico $f(x)$ pero x también depende de t, z

$$f(x - hz) = f(x) - hzf'(x) + \frac{1}{2}(hz)^2 f''(x) + O(h^3)$$

$$E(\hat{f}(x)) = \int_R K(z)f(x)dz - \int_R K(z)hzf'(x)dz + \int_R \frac{1}{2}K(z)(hz)^2 f''(x)dz + O(h^3)$$

$$E(\hat{f}(x)) = f(x)(1) - hf'(x)(0) + \frac{h^2}{2}f''(x) \int_R z^2 K(z)dz + O(h^3) = f(x) + \frac{h^2}{2}f''(x)\sigma_K^2 + O(h^3)$$

6.1 SESGO DEL ESTIMADOR $\hat{f}(X)$

Sustituyendo en $B(\hat{f}(x))$ y siendo σ_K^2 la varianza del kernel original se obtiene

$$B(\hat{f}(x)) \approx \frac{h^2}{2}f''(x)\sigma_K^2$$

Sesgo del estimador $\hat{f}(x)$ con respecto a $f(x)$

El sesgo de $\hat{f}(x)$ con respecto a $f(x)$ depende linealmente de la varianza del kernel K y cuadráticamente del ancho de banda h . Adicionalmente, aparece un nuevo factor, la derivada de la densidad teórica desconocida f . Se puede notar que el sesgo de $\hat{f}(x)$ no depende del tamaño muestral. También se observa que $\lim(B(\hat{f}(x))) = 0$, cuando $h \rightarrow 0$. Este resultado parece contradecir el hecho que si $h \rightarrow 0$, la distribución de $\hat{f}(x)$ se hace puntual.

6.2 VARIANZA DEL ESTIMADOR $\hat{f}(X)$

$$\begin{aligned} V(\hat{f}(x)) &= V\left(\frac{1}{n}\sum_{i=1}^n \frac{1}{h}K\left(\frac{x-x_i}{h}\right)\right) \\ &= \frac{1}{n^2 h^2} \sum_{i=1}^n V\left(K\left(\frac{x-x_i}{h}\right)\right) \\ x_i \text{ independientes, con igual distribución} \\ V\left(K\left(\frac{x-x_i}{h}\right)\right) &= E\left(K^2\left(\frac{x-x_i}{h}\right)\right) - [E\left(K\left(\frac{x-x_i}{h}\right)\right)]^2 \\ &= \int_R K^2\left(\frac{x-x_i}{h}\right)f(t)dt - \left[\int_R K\left(\frac{x-t}{h}\right)f(t)dt\right]^2 \\ V(\hat{f}(x)) &= \frac{1}{n^2 h^2} \sum_{i=1}^n \int_R K^2\left(\frac{x-x_i}{h}\right)f(t)dt - \frac{1}{n^2 h^2} \sum_{i=1}^n \left[\int_R K\left(\frac{x-t}{h}\right)f(t)dt\right]^2 \\ &= \frac{1}{n} \int_R \frac{1}{h^2} K^2\left(\frac{x-x_i}{h}\right)f(t)dt - \frac{1}{n} \left(\int_R K\left(\frac{x-t}{h}\right)f(t)dt\right)^2 \end{aligned}$$

Mediante la sustitución:

$$z = \frac{x-t}{h} \Rightarrow t = x-hz, dt = -hdz$$

$$V(\hat{f}(x)) = \frac{1}{nh} \int_R K^2(z)f(x-hz)dz - \frac{1}{n} \left(\int_R K(z)f(x-hz)dz\right)^2$$

Desarrollo de los componentes usando dos términos de la Serie de Taylor

$$\begin{aligned} \int_R K^2(z)f(x-hz)dz &= \int_R K^2(z)[f(x) - hzf'(x) + O(h^2)]dz \\ &= \int_R K^2(z)[f(x) - hzf'(x)]dz + O(h^2) \\ \int_R K(z)f(x-hz)dz &= \int_R K(z)[f(x) - hzf'(x) + O(h^2)]dz \\ &= \int_R K(z)f(x)dz - \int_R K(z)hzf'(x)dz + O(h^2) \\ &= f(x) \int_R K(z)dz - hf'(x) \int_R zK(z)dz + O(h^2) \\ &= f(x)(1) - hf'(x)(0) + O(h^3) = f(x) + O(h^2) \end{aligned}$$

Sustituyendo en la definición de varianza:

$$\begin{aligned} V(\hat{f}(x)) &= \frac{1}{nh} \int_R K^2(z)f(x-hz)dz - \frac{1}{n} \left(\int_R K(z)f(x-hz)dz\right)^2 \\ &= \frac{1}{nh} \left[\int_R K^2(z)[f(x) - hzf'(x)]dz + O(h^2)\right] - \frac{1}{n} [f(x) + O(h^2)]^2 \end{aligned}$$

Si se supone que n es grande y h pequeño se llega a la siguiente aproximación

$$V(\hat{f}(x)) \approx \frac{1}{nh} f(x) \int_R K^2(z)dz$$

Según este resultado $V(\hat{f}(x))$ aumenta si h se reduce.

6.3 MEDICIÓN DE LA EXACTITUD DE $\hat{f}(X)$

Sustituyendo $V(\hat{f}(x))$ en la definición de $ECM(\hat{f}(x))$

$$ECM(\hat{f}(x)) = B^2(\hat{f}(x)) + V(\hat{f}(x)) \approx \left(\frac{h^2}{2}f''(x)\sigma_K^2\right)^2 + \frac{1}{nh}f(x) \int_R K^2(z)dz$$

Finalmente, integrando sobre x

$$ECMI(\hat{f}(x)) \approx \frac{1}{4}h^4(\sigma_K^2)^2 \int_R (f''(x))^2 dx + \frac{1}{nh} \int_R K^2(z)dz \int_R f(x)dx$$

Pero $\int_R f(x)dx = 1$, y se llega a la siguiente expresión

$$ECMI(\hat{f}(x)) \approx \frac{1}{4}h^4(\sigma_K^2)^2 \int_R (f''(x))^2 dx + \frac{1}{nh} \int_R K^2(z)dz$$

Se puede ver que ambos componentes actúan en forma inversa al variar h .

Para determinar el ancho de banda h , tal que $ECMI(\hat{f}(x))$ sea mínimo:

$$\frac{dECMI(\hat{f}(x))}{dh} = 0$$

$$h^3(\sigma_K^2)^2 \int_R (f''(x))^2 dx - \frac{1}{nh^2} \int_R K^2(z)dz = 0$$

De donde se obtiene la siguiente fórmula para estimar h que minimiza a $ECMI(\hat{f}(x))$

$$h^* = \left[\frac{1 (\sigma_K^2)^{-2} \int_R K^2(z) dz}{n \int_R (f''(x))^2 dx} \right]^{1/5}$$

Separando en sus tres componentes:

$$h^* = n^{-\frac{1}{5}} \left[\int_R (f''(x))^2 dx \right]^{-\frac{1}{5}} \left[\frac{\int_R K^2(z) dz}{(\sigma_K^2)^2} \right]^{\frac{1}{5}}$$

Se observa que el valor óptimo h^* que minimiza el valor de $ECMI(\hat{f}(x))$ depende de tres componentes independientes: el tamaño de la muestra n , el modelo de densidad teórico desconocido $f''(x)$, y el kernel elegido: K y σ_K^2 . Los métodos usados en la actualidad para obtener h^* se basan en métodos para estimar $f''(x)$ mediante una aproximación basada en los mismos datos muestrales y en suposiciones acerca del modelo $f(x)$. Es importante anotar que la validez de esta fórmula requiere n grande y h pequeño.

En resumen, existe bastante incertidumbre en la estimación de h^* pero si se dispone de software se puede experimentar directamente con los modelos, los datos y el valor de h .

Sustituyendo h^* en $ECMI(\hat{f}(x))$ se obtiene la expresión con el valor mínimo para el error global del estimador $\hat{f}(x)$

$$ECMI^* \approx \frac{5}{4} n^{-\frac{4}{5}} \left(\int_R K^2(z) dz \right)^{\frac{4}{5}} (\sigma_K^2)^{\frac{2}{5}} \left(\int_R (f''(x))^2 dx \right)^{\frac{1}{5}}$$

La exactitud global del estimador depende del tamaño de la muestra en el orden $n^{-\frac{4}{5}}$ y del kernel elegido K , pero también de la densidad desconocida que se desea estimar $f(x)$, por lo que no se puede calcular directamente. La columna 4 de la Tabla I muestra la contribución de cada kernel para el valor de $ECMI(\hat{f}(x))$ y se puede constatar que el kernel que más reduce este valor es el de Epanechnikov. La diferencia con respecto a los otros kernels no es muy significativa, por lo tanto, si n es un valor fijo, la medida de $ECMI(\hat{f}(x))$ depende principalmente del factor desconocido, la densidad $f(x)$.

6.4 CÁLCULO DEL ANCHO DE BANDA ÓPTIMO

Para evaluar $f''(x)$ se supondrá que $f(x)$ tiene distribución normal. Este es el caso más común.

Sea $f(x) = N(0, \sigma^2)$ densidad normal con media cero y varianza σ^2

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

Entonces

$$\begin{aligned} (f''(x))^2 &= \frac{(\sigma^2 - x^2)^2}{2\pi\sigma^{10}} e^{-\frac{x^2}{\sigma^2}} = \\ &= \frac{1}{2\pi\sigma^{10}} (\sigma^4 e^{-\frac{x^2}{\sigma^2}} - 2\sigma^2 x^2 e^{-\frac{x^2}{\sigma^2}} + x^4 e^{-\frac{x^2}{\sigma^2}}) \end{aligned}$$

Con la sustitución $x = \sigma u$, $dx = \sigma du$

$$\begin{aligned} \int_R (f''(x))^2 dx &= \int_R (f''(\sigma u))^2 \sigma du, \\ &= \frac{\sigma}{2\pi\sigma^{10}} (\sigma^4 \int_R e^{-u^2} du - 2\sigma^4 \int_R u^2 e^{-u^2} du \\ &\quad + \sigma^4 \int_R u^4 e^{-u^2} du) \\ &= \frac{\sigma^5}{2\pi\sigma^{10}} (1.7724 - 2(0.8862) + 1.3293) \\ &= 0.2115 \sigma^{-5} \end{aligned}$$

6.5 ANCHO DE BANDA ÓPTIMO PARA EL KERNEL GAUSSIANO

Para obtenerlo se elige el kernel Gaussiano. Los valores se toman de las columnas 1 y 3 de la Tabla I y se reemplazan en la fórmula de h^*

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}, \quad x \in (-\infty, +\infty)$$

$$\int_R K^2(z) dz = 0.2821, \quad \sigma_K^2 = 1$$

Sustituyendo en h^* ,

$$\begin{aligned} h_G^* &= n^{-\frac{1}{5}} (0.2115 \sigma^{-5})^{-\frac{1}{5}} \left(\frac{0.2821}{1^2} \right)^{\frac{1}{5}} \\ &= 1.0592 \sigma n^{-\frac{1}{5}} \end{aligned}$$

Este resultado es bien conocido. Es adecuado si $f(x)$ se parece a la distribución normal. En este trabajo se usaron métodos numéricos para calcularlo.

Se han desarrollado modificaciones a esta fórmula. La siguiente se debe a Silverman y funciona bien para diferentes tipos de densidades.

$$h_G^* = 0.9 n^{-\frac{1}{5}} \min \left(\sigma, \frac{\text{rango intercuartil}}{1.349} \right)$$

En donde σ puede sustituirse con una estimación tomada de la muestra.

Ejemplo. Para entender la aplicación de la fórmula usamos el micro ejemplo anterior:

$$X: 1.1, 2.1, 2.3, 2.7, 3.8$$

$$S^2 = 0.96 \text{ (varianza muestral)}$$

$$\sigma \approx S = \sqrt{S^2} = \sqrt{0.96} = 0.9798$$

$$\text{Rango intercuartil} = 2.7 - 2.1 = 0.6$$

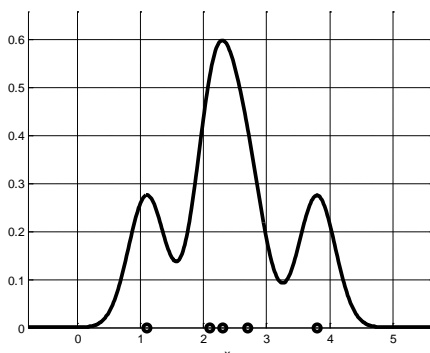
$$\begin{aligned} h_G^* &= 0.9 (5^{-\frac{1}{5}}) \min(0.9798, 0.6/1.349) \\ &= 0.2901 \end{aligned}$$

Estos datos suministrados al programa *KDEN* produjeron el siguiente gráfico con el estimador kernel Gaussiano. El resultado se muestra en la Figura 10, muy similar en forma al que se

obtuvo en la Figura 6, con otro kernel y con un ancho de banda elegido intuitivamente.

FIGURA 10

Construcción de kernels y funciones de densidad de probabilidad
Kernel Gaussiano, $h = 0.2901$



6.6 ANCHO DE BANDA ÓPTIMO PARA OTRO KERNEL

Si se requiere usar un kernel diferente, se puede usar el valor óptimo h_G^* del kernel Gaussiano para convertirlo en un valor óptimo h^* para el kernel seleccionado.

Sean

$K_G(x)$: Kernel Gaussiano con $h = 1$

$K(x)$: Kernel elegido

$K_h(x) = \frac{1}{h}K\left(\frac{x}{h}\right)$: Kernel parametrizado con h

Proponemos la siguiente expresión para encontrar el valor h_K^* para el kernel seleccionado K_h . Este es el valor de h que minimiza la diferencia global con respecto al kernel Gaussiano K_G :

$$h_K^* = \underset{h}{\operatorname{arg\,min}} \left(\int_{-h}^h |K_G(x) - K_h(x)| dx + 2 \int_h^\infty K_G(x) dx \right)$$

Su interpretación gráfica es el área sombreada en la Figura 19 en la que se muestra el caso del kernel Coseno respecto al kernel Gaussiano. Los resultados h_K^* calculados para los kernels requirieron usar métodos numéricos y están en la antepenúltima columna de la Tabla I.

En la penúltima columna de la Tabla I está la varianza para el kernel y en la última columna está un criterio adicional para comparación de kernels según el cual es preferible el que tiene la mayor amplitud con menor varianza. Según este criterio, el mejor sería el kernel Coseno.

Se puede estimar el valor óptimo h^* para un kernel específico con la siguiente fórmula:

$$h^* = h_G^* h_K^*$$

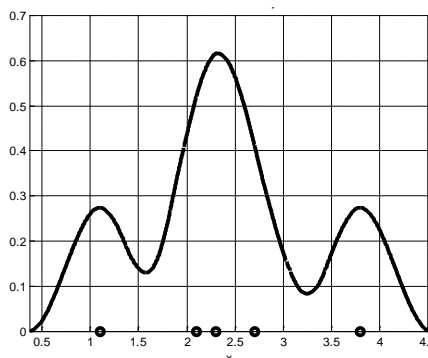
Ejemplo. Para los datos del ejemplo anterior, determine el ancho de banda óptimo si se desea usar el kernel Cúbico Sujeto.

$$h^* = h_G^* h_K^* = 0.2901(2.5150) = 0.7296$$

El gráfico obtenido con el programa *KDEN* para el kernel Cúbico Sujeto y $h = 0.7296$ es muy parecido en el nivel de detalle al que muestra el gráfico del kernel Gaussiano con $h = 0.2901$

FIGURA 11

Construcción de kernels y funciones de densidad de probabilidad
Kernel Cúbico Sujeto, $h = 0.7296$



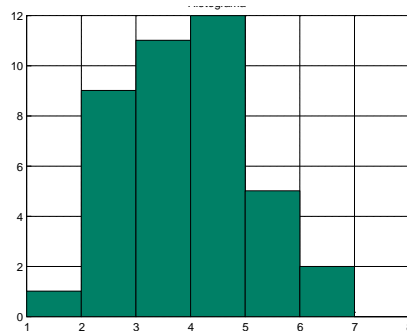
Ejemplo. La siguiente es una muestra ordenada del tiempo (minutos) que se utilizó para atender a 40 personas en una estación de servicio:

1.80 2.10 2.20 2.50 2.50 2.70 2.80 2.80
2.90 2.90 3.10 3.10 3.50 3.50 3.60 3.60
3.60 3.70 3.70 3.80 3.90 4.10 4.10 4.20
4.20 4.30 4.40 4.50 4.60 4.70 4.80 4.90
4.90 5.10 5.10 5.10 5.60 5.70 6.10 6.20

Analizar un modelo de densidad con KDEN

FIGURA 12

Construcción de kernels y funciones de densidad de probabilidad
Histograma, amplitud de clase = 1

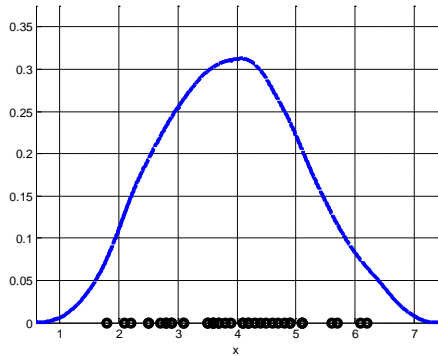


Método kernel (modelo Coseno)

Determinar el ancho de banda óptimo
 $S^2 = 1.2259$ (Varianza)
 $\sigma \approx S = \sqrt{S^2} = \sqrt{1.2259} = 1.1072$
 $Q_1 = 0.5(x_{10} + x_{11}) = 3$ (Cuartiles)
 $Q_3 = 0.5(x_{30} + x_{31}) = 4.75$
 Rango intercuartil = $4.75 - 3 = 1.75$
 $h_G^* = 0.9(40^{-\frac{1}{5}}) \min(1.1072, 1.75/1.349)$
 $= 0.4765$
 Para el modelo Coseno
 $h^* = h_G^* h_K^* = 0.4765(2.5260) = 1.2036$

FIGURA 13

Construcción de kernels y funciones de densidad de probabilidad
Kernel Coseno, $h = 1.2036$



Calcule la probabilidad que el tiempo de atención sea mayor a 5
Con el histograma $7/40 = 17.5\%$
Con el modelo kernel y el programa *KDEN* 19.01%.

El programa *KDEN* calcula y sugiere el ancho de banda óptimo, pero si se desea observar más detalles de la función de densidad se puede experimentar con el programa, y así como se puede afinar el histograma reduciendo la amplitud de clase, también se puede ensayar con el estimador kernel cambiando el ancho de banda y probando otros kernels. En los siguientes gráficos se muestran algunos resultados con los mismos 40 datos del ejemplo anterior. En estos casos el ancho de banda ya no es el valor óptimo.

FIGURA 14

Construcción de kernels y funciones de densidad de probabilidad
Histograma, amplitud de clase = 0.5

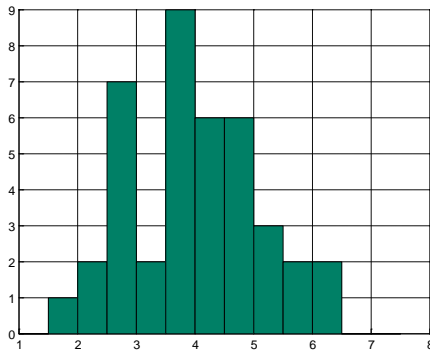


FIGURA 15

Construcción de kernels y funciones de densidad de probabilidad
Kernel Coseno, $h=0.8$

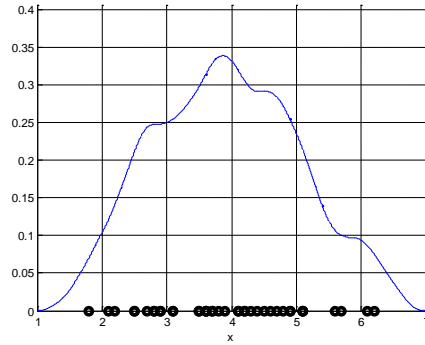


FIGURA 16

Construcción de kernels y funciones de densidad de probabilidad
Kernel Coseno, $h=0.6$

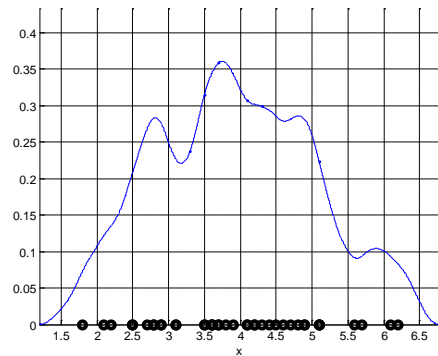
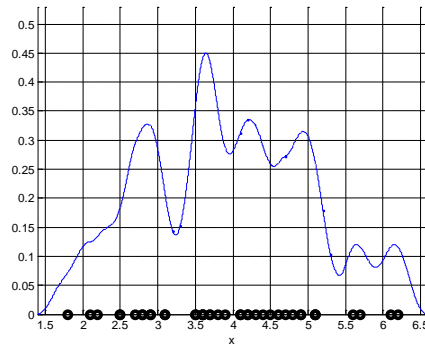


FIGURA 17

Construcción de kernels y funciones de densidad de probabilidad
Kernel Coseno, $h=0.4$



7. CONCLUSIONES

Se realizaron ensayos con muestras de diferente tamaño y se obtuvieron resultados coherentes. Sin embargo siempre será conveniente realizar pruebas con varios kernels y sus parámetros y constatar si el modelo de probabilidad muestra los detalles que uno desea. Por ello la necesidad de tener un programa para experimentar hasta llegar al modelo que nuestra intuición nos dice que es adecuado. Esto ocurre especialmente cuando los datos tienen un patrón multimodal.

El programa *KDEN* calcula y sugiere el valor óptimo del ancho de banda usando el criterio desarrollado en este artículo. La literatura contiene otros métodos más complejos, para estimar este valor crítico del método kernel.

Es importante anotar que el desarrollo de este trabajo tuvo como soporte la aplicación de métodos numéricos y de un lenguaje

computacional para diseñar e instrumentar el software para facilitar la investigación. Este software estará disponible en el sitio web del Departamento de Matemáticas de la ESPOL para que usuarios interesados puedan descargarlo y mejorarlo.

FIGURA 18

Construcción de kernels y funciones de densidad de probabilidad
Gráfico de algunos kernels

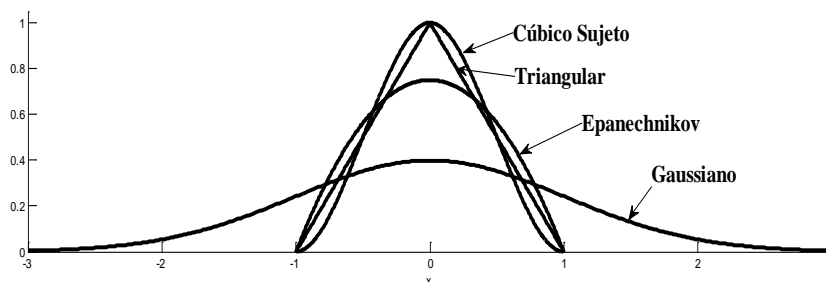


FIGURA 19

Construcción de kernels y funciones de densidad de probabilidad
Gráfico del Kernel Coseno parametrizado sobre el kernel Gaussiano

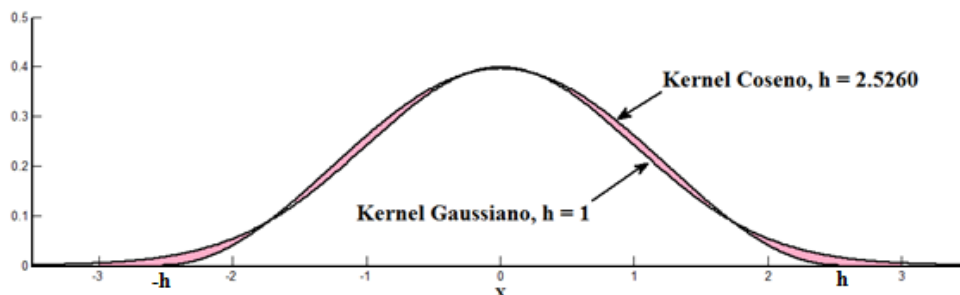


TABLA I

Construcción de kernels y funciones de densidad de probabilidad
Algunas medidas desarrolladas para comparar kernels

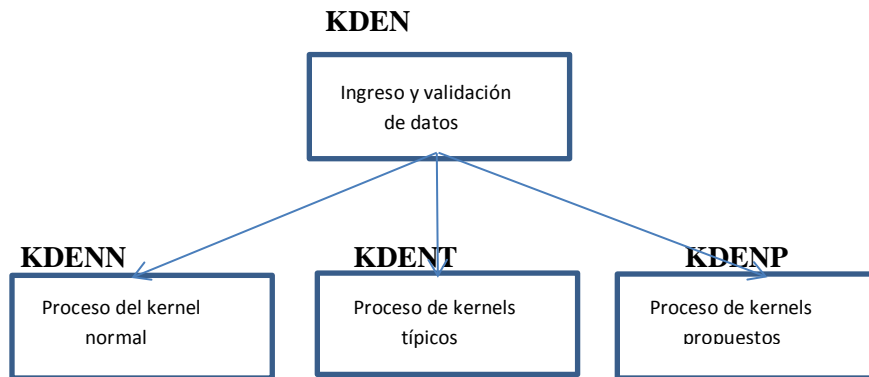
Kernel	σ_K^2	Enlace	$\int K^2(z)dz$	K_{ECMI}^*	h_K^*	$\sigma_{K_h}^2$	$h_K^*/\sigma_{K_h}^2$
Normal	1.0000	1.0000	0.2821	0.3633	1.0000	1.0000	1.0000
Rectangular	0.3333	0.0000	0.5000	0.3701	1.3800	0.6348	2.1739
Triangular	0.1666	0.3679	0.6666	0.3531	2.2120	0.8155	2.7125
Epanechnikov	0.2000	0.2231	0.6000	0.3491	1.9470	0.7582	2.5681
Arco coseno	0.1894	0.2912	0.6169	0.3492	1.9850	0.7464	2.6594
Biweight	0.1429	1.0000	0.7143	0.3508	2.3700	0.8024	2.9536
Arco circular	0.2228	0.0729	0.5709	0.3503	1.7810	0.7049	2.5267
Coseno	0.1307	1.0000	0.7500	0.3520	2.5260	0.8339	3.0292
Cúbico plano	0.2222	0.1353	0.5714	0.3502	1.9530	0.8476	2.3042
Cúbico natural	0.1867	0.3012	0.6217	0.3494	2.0180	0.7602	2.6547
Cúbico sujeto	0.1333	1.0000	0.7429	0.3521	2.5150	0.8434	2.9821

REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS

- [1]. **BERTIN, KARINE. (2012).** “*Estimación no-paramétrica de funciones*”. Universidad de Valparaiso, Chile, pp26-pp63
- [2]. **CORTES DE LA FUENTE, JORGE. (2011).** “*La información mutua como medida de asociación y su utilidad en análisis genéticos*”. Universidad Politécnica de Catalunya, pp29-pp32.
- [3]. **ZUCCHINI, WALTER. (2003).** “*Kernel Density Estimation*”, pp2-pp19
- [4]. **HANSEN, BRUCE (2009).** “*Lecture Notes on Nonparametrics*”. University of Wisconsin, pp3-pp16
- [5]. **MURAT, KAIRY. (2009).** “*Kernel Smoothing Function and choosing Bandwidth for nonparametrics Regression Methods*”. Ozean Journal of Applied Sciences, Ozean Publication, pp2-pp6
- [6]. **BRUFMAN, JUANA, URBISAIA, HERIBERTO. (2006).** “*Distribución del Ingreso Según Género: Un enfoque no paramétrico*”. Cuadernos del CIMBAGE N0. 8, pp9-pp16
- [7]. **PONOMAREVA, MARIA. (2010).** “*Nonparametrics: regresión smoothing with kernels*”. Northern Illinois University, pp2-pp13
- [8]. **HANSEN, BRUCE (2004).** “*Bandwidth Selection for Nonparametrics Distribution Estimation*”. University of Wisconsin, pp3-pp9
- [9]. **WAN, BING. (2007).** “*Bandwidth Selection for Eweighted Kernel Density Estimation*”. Electronic Journal of Statistics. ISSN 1935-7524, pp1-pp9.

APÉNDICE A**DISEÑO CONCEPTUAL DEL PROGRAMA KDEN**

Desarrollado modularmente en el lenguaje MATLAB con el soporte de su capacidad simbólica, numérica y gráfica.

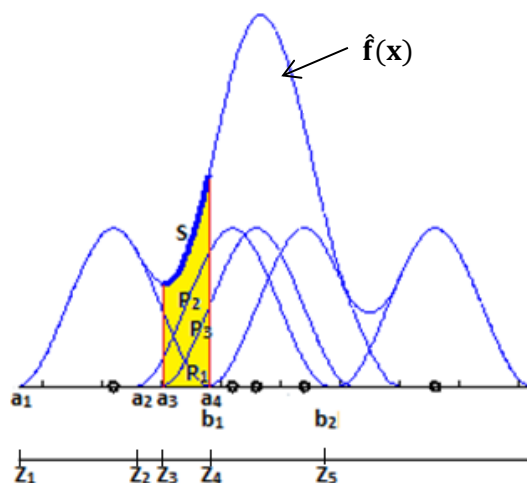
A.1 Estructura de KDEN**A.2 Módulos subyacentes**

- Graficación de puntos
- Graficación de histograma
- Graficación de kernels
- Graficación de la función de densidad
- Graficación de kernels y la función de densidad
- Cálculos estadísticos muestrales
- Cálculos estadísticos de densidad
- Cálculo del ancho de banda óptimo
- Cálculo de probabilidad
- Descripción simbólica matemática de funciones
- Función para integración numérica
- Funciones para manejo de vectores y texto

A.3 Estructuras de datos

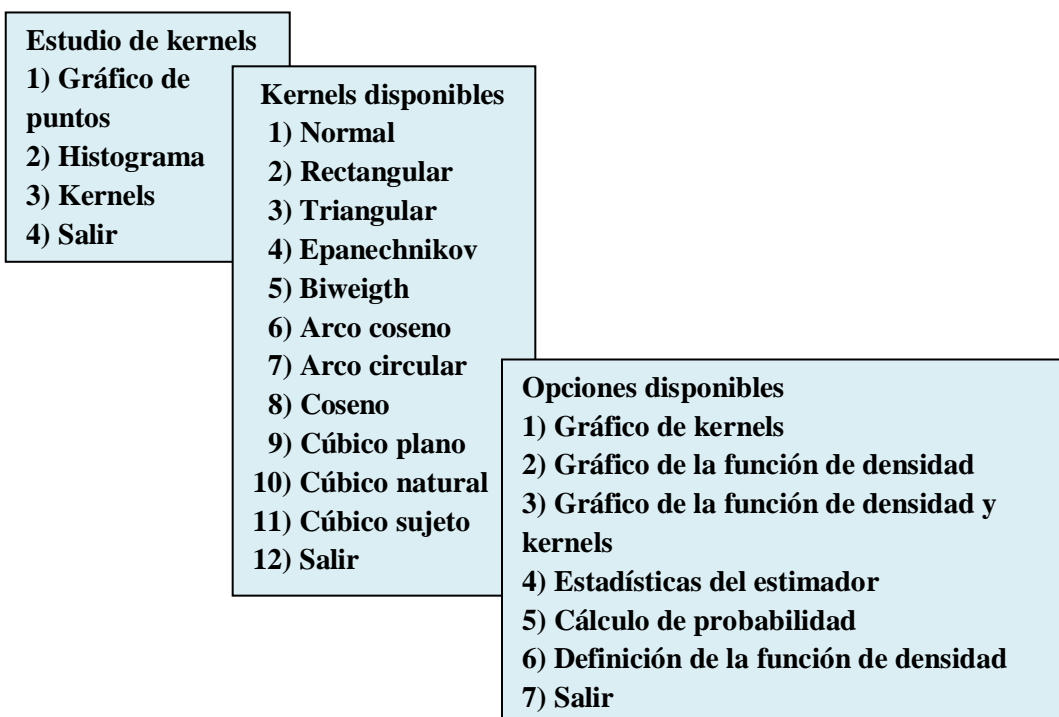
- Vector de celdas para almacenar kernels
- Vectores para almacenar bordes de kernels
- Vector para almacenar puntos de cambio de intervalo para la función de densidad
- Vector de celdas para almacenar sumas de kernels

A.4 Algoritmo para construir la función de densidad $\hat{f}(x)$



- 1) Seleccionar el kernel
- 2) Ingresar el vector con las observaciones
- 3) Ingresar el ancho de banda h
- 4) Aplicar la ponderación y colocar el kernel en cada punto muestral
- 5) Crear el vector con los bordes izquierdos y derechos a distancia h alrededor de cada punto muestral
- 6) Combinar los vectores de bordes, en un solo vector Z con los puntos de cambio de intervalo
- 7) Recorrer cada intervalo del vector Z y sumar los kernels en ese intervalo
- 8) Almacenar la suma en el vector de sumas de kernels
- 9) El vector de sumas de kernels es el estimador $\hat{f}(x)$

A.5 Interacción con el programa KDEN



ALGORITMOS FACTIBLES, PROBLEMAS TRATABLES Y LA COMPLEJIDAD COMPUTACIONAL DE UNA VARIANTE DEL PROBLEMA DE LA DIVERSIDAD MÁXIMA

Sandoya Fernando¹

Resumen. La intención de este artículo es dar una demostración formal del carácter NP-duro de una nueva variante del conocido problema de optimización combinatoria de la diversidad máxima, esta nueva variante es denominada problema del máximo promedio. Además se presenta una breve revisión de las nociones y conceptos relacionados con la NP-dureza, y la conjetura P vs.NP, con el fin de comprender la naturaleza de los problemas de optimización, y por qué algunos de ellos se pueden considerar fáciles y otros pueden llamarse difíciles.

Palabras Claves: complejidad computacional, optimización combinatoria, problema de la diversidad máxima.

Abstract. The purpose of this article is to give a formal proof of the NP-hard nature of a new variant of the classical maximum diversity problem, this new variant is called average maximum problem. It also presents a brief review of the notions and concepts related to the NP-hardness, and conjecture P vs.NP, in order to understand the nature of the optimization problems, and why some of them can be considered easy and others may call difficult.

Keywords: Computational complexity, combinatorial optimization, maximum diversity problem.

Recibido: Junio 2013

Aceptado: Agosto 2013

1. INTRODUCCIÓN

Es algo evidente, validado empíricamente, que los problemas de toma de decisiones o de optimización que se abordan en la realidad tienen diferentes grados de “dificultad”, hay problemas que son fáciles de resolver, en el sentido de que podemos llegar a su solución óptima con un grado moderado de esfuerzo analítico y computacional, mientras que otros son difíciles y aparentemente imposibles de resolver. ¿Pero esto es formalmente cierto?, es decir podríamos afirmar que ¿existe realmente una diferencia cualitativa entre estos problemas, que hace que los primeros puedan resolverse y los otros no, y que además este hecho nunca va a cambiar por más que evolucionen las matemáticas y la tecnología? La teoría de la complejidad computacional trata de dar una respuesta a esta discusión, en la primera parte de este artículo se abordan los aspectos más importantes. Para un detalle más profundo sobre esta teoría, el lector puede remitirse al influyente libro de texto de M. Garey y D. Johnson “Computers and Intractability: A Guide to the Theory of NP-Completeness” (1). En la segunda parte se da una demostración formal de la NP-dureza de un nuevo modelo de optimización combinatoria.

2. ALGORITMOS FACTIBLES Y PROBLEMAS TRATABLES

De manera informal, podemos denominar a los problemas “difíciles” como los problemas duros. Entonces a un problema \mathcal{P}_0 se le denomina NP – duro si es al menos tan duro como otros problemas de cierta clase razonable de problemas. En lo que sigue se trata de formalizar esta definición, que se basa en las nociones de algoritmos factibles y problemas tratables.

2.1 ALGORITMOS FACTIBLES

La noción de NP – dureza está relacionada al hecho empíricamente conocido de que algunos algoritmos son factibles y otros no lo son. Donde la factibilidad de un algoritmo está relacionada íntimamente con el tiempo computacional requerido para su ejecución. Por ejemplo, si para alguna entrada x de longitud $len(x) = n$, un algoritmo requiere 2^n pasos, entonces para una entrada de tamaño mediano, digamos $n = 200$, se necesitarán 2^{200} pasos, trabajo que no podría realizarse en un tiempo humanamente razonable, por más que evolucione la velocidad de los computadores. Formalmente, denominamos *algoritmo de tiempo exponencial* a cualquier algoritmo cuyo tiempo de ejecución $t(n)$ con entradas de longitud n crece al menos como una función exponencial, es decir, para el que:

$$\exists c > 0, N \in \mathbb{N}, \forall n \geq N, t(n) \geq e^{cn}$$

¹ Sandoya Fernando, Ph.D. (c), Profesor de la Escuela Superior Politécnica del Litoral (ESPOL).
(e_mail: fsandoya@espol.edu.ec).

Como resultado de esta definición, los algoritmos de tiempo exponencial son usualmente considerados no factibles. Aunque hay que aclarar que el hecho de que un algoritmo sea no factible no significa que nunca puede ser aplicado, simplemente indica que hay casos para los que el tiempo de ejecución será tan grande que no es práctico utilizarlos, aunque para ciertas entradas (de talla pequeña) el algoritmo si podría ser útil.

Por otro lado, el algoritmo se denomina de *tiempo polinomial* si el tiempo de ejecución crece sólo como un polinomio de la talla de las entradas n ; es decir, si y solo si:

“Existe un polinomio $P(n)$ tal que para toda entrada x de longitud $len(x)$, el tiempo computacional $t_u(x)$ del algoritmo \mathcal{U} con la entrada x está acotado por $P(len(x))$: $t_u(x) \leq P(len(x))$ ”.

Como resultado de la definición, estos algoritmos de tiempo polinomial suelen ser bastante factibles.

A partir de estas dos posibilidades, en la literatura sobre complejidad computacional se ha consensuado sobre la siguiente definición de factibilidad: “Un algoritmo \mathcal{U} se denomina factible si y solo si es de tiempo polinomial”. En muchos casos prácticos, esta definición describe adecuadamente la idea intuitiva de factibilidad: los algoritmos de tiempo polinomial son usualmente factibles, y los algoritmos de tiempo no polinomial son usualmente no factibles. Sin embargo habrán casos en que esta intuición no funcione, por ejemplo si un algoritmo tiene un tiempo de ejecución $2^{500}n$, es claramente de tiempo polinomial, pero obviamente infactible. Pero a pesar de estos aparentes vacíos en esta formalización, por ahora “tiempo polinomial” es la mejor descripción conocida de factibilidad, por tanto establecemos la siguiente definición:

Definición 1: Un algoritmo \mathcal{U} se denomina factible si y solo si:

“Existe un polinomio $P(n)$ tal que para toda entrada x de longitud $len(x)$, el tiempo computacional de ejecución $t_u(x)$ del algoritmo \mathcal{U} está acotado por $P(len(x))$ ”.

Donde $len(x)$ representa la longitud de la entrada x , es decir el número de bits que forma esta entrada.

2.2 PROBLEMAS TRATABLES

Una vez que se ha formalizado la definición de algoritmo factible, se puede describir qué es un problema manejable (o tratable) y que es un problema inmanejable (o intratable):

Definición 2: Si existe un algoritmo de tiempo polinomial que resuelve todas las instancias de un problema, este problema se dice que es tratable,

caso contrario se dice que es intratable. A estos problemas tratables también se les denomina “fáciles de resolver”, o “problemas de la clase P ”

Es decir, para decidir sobre el carácter del problema, o bien tenemos explícitamente un algoritmo de tiempo polinomial para resolver todas las instancias del problema, o tenemos una demostración de que no existe un algoritmo de ese tipo.

Desafortunadamente, en muchos casos, no se conoce ni lo uno ni lo otro, o es sumamente complicado llegar a establecer una de las dos cosas. Esto no significa que no haya solución al problema, pues en lugar de la falta de información ideal, tenemos otra información que es casi tan buena. Es decir, en algunos casos, no sabemos si el problema se puede resolver en tiempo polinomial o no, pero por ejemplo, sí podríamos saber que este problema es tan duro como pueden ser algunos problemas prácticos: entonces si pudiéramos solucionar este problema fácilmente, habría un algoritmo que resuelve todos los problemas con facilidad, pero la existencia de tal algoritmo es muy improbable, pues nuestra intuición, y sobretodo experiencia empírica, nos indica que en la realidad hay problemas muy difíciles. A estos problemas “duros” se les denomina intratables.

Para formular esta definición en términos más formales, debemos primero describir lo que entendemos por un *problema práctico* y que significa *reducir* un problema a otro.

Para describir que es un problema práctico, supongamos que:

- Tenemos alguna información, representada por x, y
- Conocemos la relación $R(x, y)$ entre la información conocida x y el objeto deseado y

Suponemos que tanto la información x como el objeto deseado y se han representado como sucesiones binarias, esto siempre se puede hacer ya que en la computadora cualquier cosa puede ser representada como una sucesión binaria.

Por ejemplo si se tiene una proposición matemática x , y el objeto deseado de estudio y es o bien una demostración de que x es verdadero, o una refutación de x ; es decir una demostración de que x es falso. Entonces $R(x, y)$ significa que y es una demostración cualquiera de x , o de “no x ”.

Para un problema que es práctico, debemos tener una manera de comprobar si la solución propuesta es correcta. En otras palabras, debemos suponer que existe un algoritmo factible que chequee la veracidad de $R(x, y)$ dados x e y . Si no existe tal algoritmo factible, entonces no existe un criterio para decidir si hemos logrado encontrar una solución o no al problema.

Otro requerimiento para un problema de la vida real es que en tales problemas, usualmente conocemos una cota superior para la longitud $len(y)$ de la descripción de y . En el ejemplo anterior: una demostración no debería ser demasiado grande, sino sería imposible comprobar si se trata de una prueba o no.

En todos los casos, es necesario que un usuario sea capaz de leer la solución deseada símbolo tras símbolo, y el tiempo requerido para la lectura debe ser factible. Anteriormente se formalizó "tiempo factible" como un tiempo que está acotado por algún polinomio de $len(x)$. El tiempo de lectura es proporcional a la longitud $len(y)$ de la respuesta y . Por lo tanto, el hecho que el tiempo de lectura es acotado por un polinomio de $len(x)$ significa que la longitud de la salida y está también acotada por algún polinomio de $len(x)$. Es decir: $len(y) \leq P_L(len(x))$ para algún polinomio P_L . Así, se llega a la siguiente formulación de un problema práctico:

Definición 3: Se denomina problema práctico (o simplemente problema) a la pareja $\langle R(x,y), P_L \rangle$, donde $R(x,y)$ es un algoritmo factible que transforma dos sucesiones binarias en un valor Booleano, verdadero o falso, y P_L es un polinomio.

Definición 4: Se denomina instancia de un problema $\langle R, P_L \rangle$ al siguiente problema:

"Dada una sucesión binaria x , generar o bien y tal que $R(x,y)$ es verdadero y $len(y) \leq P_L(len(x))$, o, si tal y no existe, un mensaje diciendo que no hay solución".

Por ejemplo, para el problema matemático descrito antes, una instancia puede ser: hallar una demostración de que x es verdadero o una refutación (una demostración de que $no\ x$ es verdadero).

2.3 LA CONJETURA DE P VS NP

En la literatura de optimización matemática o de complejidad computacional es común oír hablar de los "problemas prácticos generales", usualmente también descritos como "problemas de la clase NP", esta es una clasificación de los problemas de optimización para separarlos de aquellos problemas más complicados en los cuales la solución no es fácilmente verificable. Recordemos que los problemas tratables, o fáciles de resolver, son aquellos para los cuales existe un algoritmo factible que resuelve todas sus instancias.

La opinión generalizada es que no todos los problemas (prácticos generales) deben tener una solución fácil, de ahí que la conjetura que se maneja es que $NP \neq P$, pero nunca ha sido probada. Justamente esta conjetura es uno de los denominados "problemas del milenio" (1),

establecidos por el Clay Mathematics Institute en el año 2000, y por cuya resolución se ofrece un premio de un millón de dólares por cada uno. Estos problemas del milenio son siete, y hasta el momento de escribir este artículo únicamente ha sido resuelta la denominada hipótesis de Poincaré, por el matemático ruso Grigori Perelman, logro que fue reconocido en el año 2010 (3) por lo que todavía seis de ellos permanecen abiertos, entre ellos la conjetura $NP \neq P$.

A modo de ejemplo, una manera de resolver un problema NP es chequear $R(x,y)$ para todas las sucesiones binarias y que satisfacen $len(y) \leq P_L(len(x))$. Así, este algoritmo, denominado algoritmo del Museo Británico, requeriría un total de $2^{P_L(len(x))}$ chequeos, es decir es de tiempo exponencial, y por tanto no factible.

2.4 REDUCCIÓN DE UN PROBLEMA A OTRO

En términos informales, la manera como se demuestra que un problema pertenece a la clase NP es "reduciéndolo" a alguno de los que se conoce que pertenecen a esta clase. Una reducción aquí debe ser entendida como una transformación que no cambia la naturaleza de la complejidad del problema, y esto solo puede ser posible si se utiliza, de manera transitiva, una cadena de algoritmos factibles.

Como ejemplo, supongamos que se tiene un algoritmo que chequea cuando un sistema dado de desigualdades lineales es consistente. Y Consideremos adicionalmente otro problema, el de chequear cuando un sistema dado de desigualdades e igualdades es consistente. Es fácil concluir que éste último puede ser reducido al problema de la consistencia de un sistema de desigualdades, pues para esto basta reemplazar cada igualdad por dos desigualdades.

Definición 5: Dados dos problemas $\mathcal{P} = \langle R, P_L \rangle$ y $\mathcal{P}' = \langle R', P_L' \rangle$, se dice que \mathcal{P} puede reducirse a \mathcal{P}' , si existen tres algoritmos factibles U_1 , U_2 y U_3 con las siguientes propiedades:

- El algoritmo factible U_1 transforma cada entrada x del primer problema en una entrada del segundo problema.
- El algoritmo factible U_2 transforma cada solución y del primer problema en la solución del caso correspondiente del segundo problema; es decir, si $R(x,y)$ es verdadero, entonces $R'(U_1(x), U_2(y))$ es también verdadero.
- El algoritmo factible U_3 transforma cada solución y' de la correspondiente instancia del segundo problema en la solución del primer problema; es

decir, si $R'(U_1(x), y')$ es verdadera, entonces $R(x, U_3(y'))$ también es verdadera.

Si existe una reducción de \mathcal{P} a \mathcal{P}' , entonces una instancia x del primer problema es resoluble si y solo si la correspondiente instancia $U_1(x)$ del segundo problema es resoluble también. Por otra parte, si podemos resolver la segunda instancia (y encontrar una solución y'), entonces seremos capaces de encontrar una solución de la instancia original x del primer problema (como $U_3(y')$). Por lo tanto, si tenemos un algoritmo factible para resolver el segundo problema, sería posible diseñar un algoritmo para resolver el primer problema también.

fácilmente se puede ver como corolario que la relación de reducción satisface la propiedad transitiva; es decir, si un problema \mathcal{P} puede reducirse a un problema \mathcal{P}_0 , y el problema \mathcal{P}_0 se puede reducir a un problema \mathcal{P}_1 , entonces, combinando estas dos reducciones, podemos concluir que \mathcal{P} se puede reducir a \mathcal{P}_1 .

A partir de este nuevo concepto se puede definir finalmente la NP-dureza de los problemas.

Definición 6: Un problema (no necesariamente de la clase NP) se denomina *NP-duro* si cada problema de la clase NP puede ser reducida a él.

Definición 7: Si un problema de la clase NP es *NP-duro*, es llamado *NP-completo*.

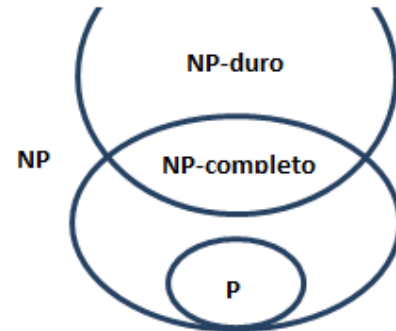
Según estas definiciones, si un problema \mathcal{P} es *NP-duro*, entonces cada algoritmo factible para resolver este problema \mathcal{P} daría lugar a algoritmos factibles para resolver todos los problemas de la clase NP; es decir, $NP = P$, lo cual se conjetura que es imposible. En vista de esta convicción, los problemas *NP-duros* suelen ser denominados también intratables.

En resumen, aunque suele confundirse la terminología, los problemas de NP-completos son aquellos problemas NP-duros que están contenidos en NP, siendo los problemas NP-duros aquellos que tienen la propiedad que cualquier problema en NP puede ser transformado por medio de una reducción a uno NP-completo, es decir, si la conjetura $NP \neq P$ fuera cierta, los problemas estarían clasificados por su complejidad de acuerdo al gráfico, denominado diagrama de Euler, mostrado en la figura 1.

FIGURA 1

Algoritmos factibles, problemas tratables y la complejidad computacional de una variante del problema de la diversidad máxima

Diagrama de Euler con la clasificación de los problemas si $NP \neq P$



2.5 DEMOSTRACIÓN DEL CARÁCTER NP-DURO DE UN PROBLEMA

La demostración original del carácter NP-duro de cierto problema P_0 es bastante compleja, ya que se basa en demostrar explícitamente que todos los problemas de la clase NP se pueden reducir al problema P_0 . Sin embargo, una vez que se ha demostrado la NP-dureza de un problema P_0 , la demostración de que otro problema P_1 es NP-duro es mucho más fácil a partir del concepto de reducción, ya que para demostrar que P_1 es NP-duro, es suficiente demostrar que alguno de aquellos problemas que se conoce son NP-duros puede ser reducido a este problema P_1 .

Históricamente, el primer problema que se demostró que es NP-completo fue el problema de satisfacibilidad booleana, lo que se conoce en la literatura como el teorema de Cook. Luego, en 1972, Richard Karp demostró que otros 21 problemas eran también NP-completos. Posteriormente se han descubierto que cientos de problemas pertenecen también a la clase de los NP-completo por reducciones desde otros problemas que previamente se habían demostrado NP-completos, muchos de estos problemas constan en el libro de Garey y Johnson (1). Entre los más conocidos de estos problemas están el problema de la mochila, el problema del agente viajero, el problema de la jorga máxima, etc. En particular, en este artículo, se utiliza el problema de la jorga máxima para demostrar que un nuevo problema combinatorio es NP-duro.

Teorema: Si un problema P_0 es NP-duro, cualquier problema más general P_1 también es NP-duro.

Demostración: El teorema es demostrado en (1) de la siguiente manera:

El hecho de que P_0 es NP-duro implica que cada instancia p de cualquier problema P puede ser

reducida a alguna instancia p_0 del problema P_0 . Ya que el problema P_1 es más general que el problema P_0 , cada instancia p_0 del problema P_0 es también una instancia del problema más general P_1 . Así, cada instancia p de cualquier problema P puede ser reducida a alguna instancia p_0 del problema P_1 , es decir el problema general P_1 es NP-duro.

El Problema de la Máxima Jorga: El problema de la máxima jorga es uno de los problemas que se conocen son NP-completos, y se define de la siguiente manera:

Sea un grafo $G=(V,E)$, donde V es el conjunto de nodos y E es el conjunto de aristas, un subconjunto de nodos $C \subseteq V$ se denomina una jorga de G si $\forall v_1, v_2 \in C \subseteq V$, existe una arista $(v_1, v_2) \in E$

La versión de decisión del problema de la máxima jorga consiste en chequear si:

“Dado un número $k \in \mathbb{N}$ ¿es posible encontrar una jorga C de tamaño al menos k en G ?

3. EL PROBLEMA DEL MÁXIMO PROMEDIO Y SU CARÁCTER NP-DURO

Este problema del máximo promedio es un nuevo modelo del clásico problema de la diversidad máxima (4), que surge de la optimización de la medida de la diversidad promedio, y al contrario de otros modelos del problema de la diversidad máxima. En el problema del máximo promedio la cardinalidad del subconjunto seleccionado, M , es también una variable de decisión. Según puede observarse en (5) el problema del máximo promedio puede formularse como:

$$\max_{M \subseteq V, |M| \geq 2} \frac{\sum_{i < j, i, j \in M} d_{ij}}{|M|}$$

Donde V es un conjunto de elementos (población), donde se quiere seleccionar el subconjunto M más diverso.

Genéricamente hablando, este problema trata de maximizar la diversidad promedio. Una formulación de programación matemática con variables binarias es entonces la establecida por la formulación (3.1)–(3.3):

$$\max \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n d_{ij} x_i x_j}{\sum_{i=1}^n x_i} \quad (3.1)$$

$$s. t. \sum_{i=1}^n x_i \geq 2 \quad (3.2)$$

$$x_i \in \{0,1\}, 1 \leq i \leq n \quad (3.3)$$

En este problema la función objetivo 0 es el promedio de la suma de las distancias entre los elementos seleccionados, la restricción 0 indica que

por lo menos dos elementos deben seleccionarse. Tal como se presenta en (5), es un problema de optimización binaria fraccional

Propiedad :

Si los coeficientes d_{ij} no tienen restricciones en el signo, entonces el problema del máximo promedio es NP-duro.

Demostración:

Para la demostración consideramos como nuestro problema \mathcal{P}_0 , al problema de la Máxima Jorga, el cual es conocido que es *NP-completo*, según puede consultarse en (1).

Sea $G = (V, E)$ un grafo, donde V es el conjunto de nodos y E es el conjunto de aristas, con $|V| = n$ nodos.

Consideremos ahora la siguiente instancia del problema del máximo promedio en su formulación con variables binarias:

$$\max_{x \in \{0,1\}^n, x \neq 0} g(x) = \frac{\sum_{i < j} d_{ij} x_i x_j}{\sum_{i=1}^n x_i}$$

Donde las distancias inter-elemento son tomadas

$$\text{como: } d_{ij} = \begin{cases} 1 & \text{si } (i, j) \in E \\ -n^2 & \text{si } (i, j) \notin E \end{cases}$$

Entonces se tiene:

$$\begin{aligned} & \max_{x \in \{0,1\}^n, x \neq 0} g(x) \\ &= \frac{-n^2 \sum_{(i,j) \notin E, i < j} x_i x_j + \sum_{(i,j) \in E, i < j} x_i x_j}{\sum_{i=1}^n x_i} \end{aligned}$$

Sea $x^* \in \{0,1\}^n, x^* \neq 0$. Definimos un subgrafo inducido $\mathcal{C}(V_C, E_C) \subseteq G$, donde:

$$V_C = \{i \in \{1, \dots, n\} : x_i^* = 1\}$$

Si C no es una jorga: $x_i^* = 1, x_j^* = 1$ implica que $(i, j) \notin E$, entonces $g(x^*) < 0$.

Por el contrario si C es una jorga: $\forall (i, j)$ tal que $x_i^* = 1, x_j^* = 1$ se tiene $(i, j) \in E$, y entonces:

$$g(x^*) = \frac{|C|(|C| - 1)}{2|C|} = \frac{|C| - 1}{2} \geq 0$$

Por lo tanto, el grafo G contiene una jorga de tamaño al menos k si y sólo si:

$$\max_{x \in \{0,1\}^n, x \neq 0} g(x) \geq \frac{k-1}{2}$$

Con lo que se demuestra que para esta instancia del problema de la diversidad máxima el problema se reduce a un problema de la máxima jorga, el cual se conoce, como se demuestra en (1), que es *NP-completo*.

REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS

- [1]. **GAREY, M., JOHNSON, D.** (1979). *“Computers and Intractability: A Guide to the Theory of NP-Completeness”*. s.l. : W.H. Freeman y Cía.
- [2]. **CLAY MATHEMATICS INSTITUTE.** [En línea] [Citado el: 20 de 05 de 2013.] <http://www.claymath.org/millennium/>.
- [3]. **CARLSON, J.** (2013). First Clay Mathematics Institute Millennium Prize Announced Today. [En línea] [Citado el: 22 de 05 de 2013.] <http://www.claymath.org/poincare/millenniumPrizeFull.pdf>.
- [4]. **GHOSH.** (1996). *“Computational aspects of the maximum diversity problem”*. 4, s.l.: Elsevier, Operations Research Letters, Vol. 19, págs. 175-181.
- [5]. **MARTÍ, R., SANDOYA, F.** *GRASP and Path Relinking for the Equitable Dispersion Problem*. s.l. : Elsevier, 2012, Computers & Operations Research, págs. 1-18. <http://dx.doi.org/10.1016/j.cor.2012.04.005>.
- [6]. **PROKOPYEV, O., KONG, N., MARTÍNEZ-TORRES, D.** *“The equitable dispersion problem”*. 197, 2009, European Journal of Operational Research, págs. 59 - 67.

CONTENIDO

EDITORIAL.....	5
DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO GRASP PARA EL PROBLEMA DE COLORACIÓN DE GRAFOS Delgado Erwin.....	7
IMPLEMENTACIÓN DE UN ALGORITMO GRASP PARA EL PROBLEMA DE COLORACIÓN DE GRAFOS APLICADO A LA CALENDARIZACIÓN DE EXAMENES EN UNA INSTITUCIÓN EDUCATIVA Delgado Erwin.....	15
ATENUACIÓN DE RIZADO EN LA DENSIDAD ESPECTRAL DE POTENCIA CALCULADA EN UNA SEÑAL DE RITMO CARDIACO González Javier, Forero Edwin, Jiménez Fabián, Mariño Iván.....	22
CONSTRUCCIÓN DE KERNELS Y FUNCIONES DE DENSIDAD DE PROBABILIDAD Rodríguez Luís.....	27
ALGORITMOS FACTIBLES, PROBLEMAS TRATABLES Y LA COMPLEJIDAD COMPUTACIONAL DE UNA VARIANTE DEL PROBLEMA DE LA DIVERSIDAD MÁXIMA Sandoya Ferenando.....	41