

S
M
C
C
A



BOLETÍN

Sociedad Mexicana de Computación
Científica y sus Aplicaciones

Año V - Número 5

Diciembre 2019



BOLETÍN

Sociedad Mexicana de Computación Científica y sus Aplicaciones

Consejo Editorial

Pablo Barrera Sánchez, UNAM

Pedro Flores Pérez, UNISON

Irma Delia García Calvillo, UAdeC (*editor responsable*)

José Gerardo Tinoco Ruíz, UMSNH

Editores técnicos

Vanesa Ávalos Gaytan, UAdeC

Oliver Ávalos Rosales, UAdeC

Yajaira Cardona Valdés, UAdeC

El Boletín de la Sociedad Mexicana de Computación Científica y sus Aplicaciones A. C. (SMMCA), Año V, No. 5, Diciembre 2019, es una publicación anual editada por la Sociedad Mexicana de Computación Científica y sus Aplicaciones A. C., calle Luis Horacio Salinas, 545, Col. Valle de Morelos, Saltillo, Coahuila, C.P. 25013, Tel. (844) 4 10 12 42, www.smcca.org.mx. Editor responsable: Irma Delia García Calvillo. Reserva de Derechos al Uso Exclusivo No. 04 - 2017 - 103114330600 - 203, ISSN: 2594 - 0457, ambos otorgados por el Instituto Nacional de Derechos de Autor. Responsable de la última actualización de este Número, Yajaira Cardona Valdés, calle Magisterio, s/n, Unidad Camporredondo, Edificio S, Saltillo, Coahuila, C.P. 25280, fecha de última modificación, 14 de diciembre de 2019.

Las opiniones expresadas por los autores no necesariamente reflejan la postura del editor de la publicación.

Queda prohibida la reproducción total o parcial de los contenidos e imágenes de la publicación sin previa autorización de la Sociedad Mexicana de Computación Científica y sus Aplicaciones A. C.

Suscripciones al Boletín vía correo electrónico: smcca@smcca.org.mx

Contenido

CARTA DE BIENVENIDA	4
---------------------	---

Reseña de la XXVIII Escuela Nacional de Optimización y Análisis Numérico5

Ganadores del décimo octavo Premio MIXBAAL	10
--	----

Convocatoria para el décimo noveno Premio MIXBAAL	11
---	----

ARTÍCULOS

Algoritmo memético para el problema del Particionado de Grafos en ejecuciones a largo plazo.....	13
--	----

El Porismo de Poncelet Parte I.....	26
-------------------------------------	----

El Porismo de Poncelet Parte II.....	34
--------------------------------------	----

Ruteo de vehículos escolares: caso de estudio de la Universidad Autónoma de Coahuila.....	42
---	----

PRÓXIMOS EVENTOS

IX Encuentro Cuba-México Métodos Numéricos y Optimización.....	51
--	----

XXIX ENOAN.....	52
-----------------	----

¿QUIERES PUBLICAR ARTÍCULOS, INFORMACIÓN SOBRE EVENTOS O NOTICIAS EN EL BOLETÍN?.....	53
---	----

Carta de Bienvenida

La Sociedad Mexicana de Computación Científica y sus Aplicaciones, A.C. (SMCCA) y el Comité Editorial, les dan una cordial bienvenida a la quinta edición del Boletín electrónico anual de la SMCCA, el cual tiene como objetivo mantenerlos informados de las actividades realizadas por la SMCCA y sus asociados. En el Boletín se publican noticias, eventos, artículos de divulgación y de investigación de alto nivel en el área de Cómputo Científico y sus Aplicaciones, así como resúmenes de las mejores tesis de Licenciatura en Matemáticas Aplicadas.

En esta quinta edición del boletín se presenta: una breve semblanza de la XXVIII Escuela Nacional de Optimización y Análisis Numérico, llevada a cabo este año del 26 al 30 de agosto en las instalaciones del Consejo Zacatecano de Ciencia, Tecnología e Innovación en la Ciudad de Zacatecas, México; en esta edición se presenta al ganador de la décimo octava edición del premio Mixbaal para la mejor tesis de Licenciatura en Matemáticas Aplicadas, y se presenta la convocatoria para participar en dicho premio en su décimo novena edición. Además, se presentan cuatro artículos de investigación, tres de ellos son por solicitud y uno de ellos corresponde al ganador del premio Mixbaal. Por último, presentamos convocatorias de futuros eventos a realizarse.

La SMCCA agradecerá que ante el interés que surja en los lectores en los temas que se presenten en nuestra publicación, éstos se conviertan en usuarios asiduos, así como en miembros activos de nuestra Sociedad. La información del registro de membresías a la SMCCA la pueden consultar en el Módulo de Registro de nuestra página www.smcca.org.mx.

Justino Alavez Ramírez

Presidente

Sociedad Mexicana de Computación Científica y sus Aplicaciones, A.C.



**Justino Alavez Ramírez¹, Rina Betzabeth Ojeda Castañeda² y
Leticia Adriana Ramírez Hernández³**

¹Universidad Juárez Autónoma de Tabasco

²Universidad Autónoma de Coahuila

³Universidad Autónoma de Zacatecas

Del 26 al 30 de agosto de 2019 se realizó con gran éxito la XXVIII Escuela Nacional de Optimización y Análisis Numérico (ENOAN 2019), en las instalaciones del Consejo Zacatecano de Ciencia, Tecnología e Innovación, de la Ciudad de Zacatecas, Zac. La organización de esta XXVIII edición de la ENOAN estuvo a cargo de un Comité Local formado por profesores-investigadores y alumnos de la Unidad Académica de Matemáticas de la Universidad Autónoma de Zacatecas, y del Comité Nacional conformado por profesores-investigadores miembros en activo de la Sociedad Mexicana de Computación Científica y sus Aplicaciones. En el marco de la ENOAN 2019 se realizaron los talleres “Workshop Big Data & Machine Learning”, “III Mini-symposium de Medicina y Matemáticas” y la “Sección Paralela MEX-SIAM”.

Durante esa semana, 155 asistentes en total, entre alumnos de licenciatura y posgrado, profesores, investigadores y profesionales interesados en la aplicación de las Matemáticas y el Cómputo Científico, se reunieron para participar en diversas actividades académicas: 9 cursos cortos, 5 conferencias plenarias incluyendo la Conferencia “Diego Bricio”, 4 conferencias invitadas de la ENOAN, 5 conferencias invitadas del Workshop Big Data & Machine Learning, 12 conferencias invitadas del III Mini-symposium de Medicina y Matemáticas y 12 conferencias invitadas de la Sección Paralela MEX-SIAM; así como 43 ponencias presenciales y 20 carteles de la ENOAN por solicitud.

Los conferencistas (plenarios e invitados tanto de la ENOAN como del Workshop Big Data & Machine Learning, III Mini-symposium

de Medicina y Matemáticas, y de la Sección Paralela MEX-SIAM) y los instructores de los cursos que participaron en el evento, son profesores-investigadores de diferentes Instituciones de Educación Superior y Centros de Investigación tanto extranjeros como nacionales. Son profesionales ampliamente reconocidos en sus áreas de conocimiento, con gran experiencia en el ámbito de la docencia e investigación en temas de actualidad, relevantes y en el estado del arte sobre la aplicación de las matemáticas y del cómputo científico en la Ciencia e Ingenierías. Actualmente participan en proyectos establecidos para la solución de problemas tanto de interés nacional como internacional.

Este perfil idóneo de los profesores-investigadores permitió a la ENOAN ofrecer a alumnos de licenciatura y posgrado de distintas instituciones educativas del país y del extranjero cursos especializados a diferentes niveles: básico, intermedio y avanzado, a los que generalmente ellos no

pueden acceder con regularidad en sus instituciones.

Los cursos básicos impartidos fueron:

- Introducción a la estadística espacial con R.
- Herramientas computacionales para big data.
- Introducción al cómputo científico.

Los cursos intermedios fueron:

- Introducción a métodos de machine learning orientado a problemas sociales y economía.
- Sobre el conocimiento colectivo.
- Álgebra lineal numérica con aplicaciones.

Los cursos avanzados fueron:

- Solución de ecuaciones diferenciales parciales usando machine learning.
- Discretización de un problema de control para encontrar operadores de memoria criogénica.
- Modelado y resolución de problemas de optimización con Python & Gurobi.



ALGUNOS ASISTENTES PRESENTES EN LA CLAUSURA DE LA XXVIII ENOAN

Sus contenidos, bibliografía e información de los instructores, se pueden consultar en la página de la ENOAN en la liga www.smcca.org.mx/enoan2019

La participación de conferencistas plenarios e invitados internacionales y nacionales del evento, permitió a todos los asistentes tener la oportunidad de conocer el estado del arte de la Computación Científica y sus aplicaciones tanto en México, como en otros Países. Aquí se hace mención de los nombres de algunos de ellos, pero el lector puede consultar el programa completo de conferencias impartidas en www.smcca.org.mx/enoan2019.

Conferencista Plenario Diego Bricio

Dr. Lorenzo Héctor Juárez Valencia del Depto. de Matemáticas de la UAM Iztapalapa.

CONFERENCISTAS PLENARIOS

- Dr. José Sarrate de la Universidad Politécnica de Cataluña, España.
- Dr. Jorge Nocedal of Department of Industrial Engineering and Management Sciences, Northwestern University, USA.
- Dr. Hugo Estrada Esquivel de la Dirección de Coordinación Sectorial de la Dirección Adjunta de Centros de Investigación del CONACYT.
- Dr. Pedro Eduardo Miramontes Vidal de la Facultad de Ciencias de la UNAM.

CONFERENCISTAS INVITADOS

- Dr. Pedro Flores Pérez de la Universidad de Sonora.

- Dra. Marisol Flores Garrido de la Escuela Nacional de Estudios Superiores, Unidad Morelia - UNAM.
- Dra. María de Lourdes Esteva Peralta de la Facultad de Ciencias de la UNAM.
- Dr. Joshua Mendoza Jasso de la empresa Has-IT S.C.
- Dr. Julio Waissman Vilanova de la Universidad de Sonora.
- Dr. José Alejandro Ruiz Sánchez del INEGI-Aguascalientes.
- Dra. Adriana Monroy Guzmán del Hospital General de México - UNAM .
- Dr. Heladio Verver y Vargas Ramírez del Hospital General de Zacatecas del Servicio de Pediatría y Académico de la Academia Mexicana de Pediatría.
- Dra. Edith Cárdenas Vargas de la Endocrinología Pediátrica del Hospital General "Luz González Cosío" de Zacatecas.
- Dr. Adalberto Maldonado de empresa BMSA GROUP.
- Dr. Gerardo Hernández Dueñas del Instituto de Matemáticas Unidad Juriquilla de la UNAM.
- Dr. Luis Cisneros Ake del IPN.
- Dr. Ricardo Álvarez Salas de la Universidad Autónoma de San Luis Potosí.
- Dra. Irma Delia García Calvillo de la Universidad Autónoma de Coahuila.
- Dr. José Gerardo Tinoco Ruiz de la UMSNH, entre otros.

Dentro de este marco de vinculación de academia-investigación-industria, los alumnos asistentes también tuvieron la oportunidad, un buen número de ellos (43 ponencias y 20 carteles por solicitud), de exponer de manera presencial o a través de un formato de cartel, del desarrollo y de los resultados obtenidos en investigaciones que llevan a cabo en conjunto con sus profesores, como trabajos de tesis (licenciatura, maestría, doctorado) o en proyectos de investigación establecidos en sus Instituciones, recibiendo una retroalimentación de utilidad para mejorar su trabajo, tanto de parte de investigadores especializados en los temas presentados, como de sus compañeros de otras instituciones, interesados en estos temas.

Por último es de gran importancia señalar que el gran esfuerzo de trabajo realizado tanto por el Comité Local (Unidad Académica de Matemáticas de la Universidad Autónoma de Zacatecas) como el Comité Nacional (aglutinados dentro de la SMCCA) en la organización, y contando con el importante apoyo financiero de Instituciones, Dependencias y Centros de Investigación como: CONACYT, la Sociedad Mexicana de Computación Científica y sus Aplicaciones, A.C., la Universidad Autónoma de Zacatecas, el Consejo Zacatecano de Ciencia Tecnología e Innovación (COZCyT), la Sección México - SIAM, el Sistema Zacatecano de Radio y Televisión (SIZART), el Centro de Investigación en Matemáticas Aplicadas de la Universidad Autónoma de Coahuila, la Facultad de Ciencias de la Universidad Nacional Autónoma de México, y el Departamento de Matemáticas de la Universidad Autónoma Metropolitana Unidad Iztapalapa; permitió obtener un conjunto de resultados a beneficio de una comunidad científica conformada por alumnos, profesores, investigadores y profesionales interesados en la Computación Científica y las Matemáticas Aplicadas, que incidieron en indicadores de impacto como los que se presentan en la siguiente tabla:

Indicador	Cantidad
Total de asistentes:	155
Alumnos beneficiados del evento:	82
Becas otorgadas a estudiantes de instituciones de provincia:	46
Becas inscripción a profesores y alumnos de la UAZ:	7
Becas a conferencistas plenarios, ponentes invitados e instructores de cursos:	17
Becas inscripción a miembros del comité organizador nacional:	3
Programas educativos de instituciones participantes:	33
Investigadores, docentes y profesionales beneficiados del evento:	73
Cuerpos académicos y grupos de investigación del país y del extranjero:	12
Número de Instituciones participantes:	
Nacionales: 46	49
Extranjeras: 3	

Memorias de la XXVIII ENOAN



Ganador del Premio MIXBAAL

Ganador de la *décimo octava edición* del premio Mixbaal a la
Mejor Tesis de Licenciatura en Matemáticas Aplicadas

Ganador del premio Mixbaal

Emmanuel Romero Ruíz
Lic. en Matemáticas Aplicadas
Universidad de Guanajuato

Trabajo: Algoritmo memético para el problema de particionado de grafos en ecuaciones a largo plazo.

Director: Dr. Carlos Segura González

CONVOCATORIA AL PREMIO



A LA MEJOR TESIS DE LICENCIATURA EN MATEMÁTICAS APLICADAS

La Sociedad Mexicana de Computación Científica y sus Aplicaciones A.C. (SMCCA) convoca a los profesionistas en matemáticas y ramas afines, recién titulados, a presentar su tesis o su trabajo terminal de licenciatura para el **DÉCIMO NOVENO PREMIO MIXBAAL A LA MEJOR TESIS DE LICENCIATURA EN MATEMÁTICAS APLICADAS** de acuerdo a las siguientes:

BASES

1. El trabajo, **sobre un tema de matemáticas aplicadas**, deberá haber sido presentado en alguna institución mexicana de educación superior para obtener el grado de Licenciatura.
2. La fecha de obtención del grado deberá estar comprendida en el lapso del 1 de febrero de 2019 al 31 de enero de 2020.
3. Se deberá enviar en forma impresa y por cuadruplicado:
 - a) Copia del trabajo mecanografiado o tipografiado.
 - b) Documento probatorio con fecha de obtención del título.
 - c) Documento en el que conste oficialmente que el trabajo enviado a concurso fue presentado como requerimiento para la obtención del título.
 - d) Breve currículum del candidato.
 - e) Nombre del asesor o director del trabajo.
 - f) Nombres de los sinodales del examen profesional.

La fecha límite para recibir toda la documentación es el 30 de marzo de 2020.

4. El jurado será designado por la Comisión del PREMIO MIXBAAL y estará integrado por especialistas de reconocido prestigio en diversas ramas de las ciencias relacionadas con las matemáticas aplicadas.

5. El PREMIO MIXBAAL incluye un reconocimiento y un diploma.

6. Se otorgarán menciones honoríficas a juicio del jurado.

7. Los trabajos de tesina serán evaluados previamente para su aceptación a concurso.

8. La decisión del jurado será inapelable.

Los resultados serán comunicados verbalmente a los interesados el 30 de mayo de 2020 y publicados posteriormente en la página de la sociedad: <http://www.smcca.org.mx/>

La entrega del PREMIO MIXBAAL será durante la ceremonia de inauguración de la XXIX Escuela Nacional de Optimización y Análisis Numérico.

Cualquier situación no prevista en esta convocatoria, será resuelta por la Comisión del PREMIO MIXBAAL.

Los documentos probatorios deberá enviarse a:

Universidad Nacional Autónoma de México
Facultad de Ciencias
Departamento de Matemáticas, Cubículo 226
Dr. Pablo Barrera Sánchez
Circuito Exterior s/n
Colonia Copilco Universidad
04510 México, D.F.

Informes:

Tels: 5556224836

Correos: pablobarrera@ciencias.unam.mx

guilmerg@ciencias.unam.mx

Algoritmo memético para el Problema del Particionado de Grafos en ejecuciones a largo plazo

Emmanuel Romero Ruiz¹, Carlos Segura²

¹Departamento de Matemáticas, Universidad de Guanajuato

²Área de Computación, Centro de Investigación en Matemáticas

Correos electrónicos: emmanuel.romero@cimat.mx

carlos.segura@cimat.mx (Autor de Contacto)

Resumen

El problema del Particionado de Grafos (GPP — Graph Partitioning Problem) es un problema NP-Difícil cuyo objetivo es particionar los nodos de un grafo en k conjuntos de forma que se minimice el número de aristas que unen vértices localizados en diferentes conjuntos, a la vez que se cumplen restricciones relativas al tamaño de los conjuntos. En los últimos años, el desarrollo de metaheurísticas poblacionales que incorporan un control explícito de la diversidad ha permitido realizar avances significativos al tratar diversos problemas de optimización combinatoria como el problema de coloreado de grafos o el de asignación de frecuencias. A pesar de que el GPP es un problema clásico, el desarrollo de metaheurísticas poblacionales con control de diversidad para el GPP es un tema prácticamente inexplorado. Por ello, a partir de la hipótesis de que un algoritmo memético con gestión de diversidad ayudaría a mejorar el estado del arte para el GPP, se desarrollaron nuevos algoritmos que efectivamente han permitido mejorar las soluciones encontradas hasta la fecha para una gran cantidad de casos. El trabajo presentado es un resumen de la tesis que ganó el Premio Mixbaal a la mejor tesis de licenciatura en matemáticas aplicadas, en la edición del año 2019. Este premio es otorgado por la Sociedad Mexicana de Computación Científica y Sus Aplicaciones.

Palabras clave: Particionado de Grafos; Algoritmos Meméticos; Control de Diversidad; Búsqueda Tabú.

Introducción

El Problema del Particionado de Grafos (GPP — Graph Partitioning Problem) es un problema NP-Difícil cuyo objetivo es particionar el conjunto de nodos de un grafo en k conjuntos de forma que se minimice el número de aristas que unen vértices localizados en diferentes conjuntos, a la vez que se cumplen restricciones relativas al tamaño de los conjuntos. Este problema, además de ser importante en el ámbito académico donde es común su utilización para realizar comparativas entre diferentes optimizadores, tiene también aplicaciones prácticas. Así, se ha utilizado para determinar la manera en que se agrupan componentes en un conjunto de chips cuando se busca minimizar la comunicación entre elementos de diferentes chips, así como en la paralelización de estrategias de elemento finito, donde se utiliza para distribuir las variables del problema entre los diferentes procesadores con el objetivo de minimizar la información compartida entre dichos procesadores. Bajo la hipótesis $P \neq NP$, no es posible desarrollar resolutores exactos para el GPP que tengan complejidad polinomial con respecto al número de nodos del grafo. Por ello, y dado que cada vez se quiere lidiar con grafos de mayor tamaño, hasta la fecha un gran número de investigadores han trabajado el problema de manera aproximada, desarrollándose estrategias de múltiples tipos que abarcan desde algoritmos de aproximación, hasta heurísticas de trayectoria y poblacionales.

Las metaheurísticas poblacionales han sido una de las estrategias más exitosas, siendo los métodos que han permitido alcanzar los mejores resultados conocidos hasta la fecha en gran parte de grafos de gran tamaño que se usan para validar los algoritmos propuestos en el área de GPP. En lo referente a aplicación de algoritmos poblacionales en el ámbito del GPP, a pesar de toda la literatura que hay al respecto, no se han aplicado aún algunos de los últimos avances. Por ejemplo, no hay trabajos que apliquen algoritmos poblacionales con control explícito de diversidad al GPP. Teniendo esto en cuenta,

en la tesis en la que este trabajo se basa se analizan los algoritmos poblacionales que han sido aplicados con mayor éxito al GPP y se proponen varias extensiones a los mismos incluyendo la utilización de un control explícito de diversidad. El propósito principal es desarrollar nuevos algoritmos poblacionales capaces de obtener soluciones de alta calidad a largo plazo, es decir, que sean capaces de obtener mejores resultados que cualquiera de las propuestas existentes aunque ello implique que a corto plazo no obtengan soluciones de alta calidad.

Definición matemática del GPP

En esta sección se incluye una definición formal de la versión del GPP [2] tratada en este trabajo:

- * Sea $G = (V, E, w)$ un grafo no dirigido donde V es el conjunto de nodos, E es el conjunto de aristas y $w : E \rightarrow \mathbb{R}_{>0}$ es una función de pesos.
- * En lo sucesivo $n = \|V\|$ y $m = \|E\|$.
- * Sea k un entero positivo fijo que representa el número de clases en la partición deseada, y $\epsilon > 0$ un número real relacionado con la restricción de balance del problema.
- * El espacio de soluciones admisibles para el problema es el conjunto de las particiones $\Delta = \{S = \{S_1, S_2, \dots, S_k\} \text{ tal que } S \text{ es una partición de } V \text{ en } k \text{ conjuntos y } \|S_i\| \leq L_{max} := (1 + \epsilon)[n/k] \text{ para todo } i \in \{1, 2, \dots, k\}\}$.
- * Finalmente, la función a minimizar en el GPP es $cost(S) = \sum_{i < j} w(E_{ij})$ donde $E_{ij} = \{(u, v) \in E : u \in S_i, v \in S_j\}$ y $w(C) = \sum_{e \in C} w(e)$.

Nótese que existen otras variantes más generales, que por ejemplo establecen pesos en los nodos u otras restricciones relativas al balanceo [11]. Sin embargo, los desarrollos realizados en la tesis fueron exclusivos para la variante anterior. La validación experimental se llevó a cabo con las instancias presentes en el TGPA (The Graph Partitioning Archive) de Chris Walshaw [1]. Por ello, se debe considerar que el problema tratado está más limitado que el dado por la definición anterior ya que:

- En las instancias del TGPA, todos los pesos son iguales a 1, por lo que la función objetivo es el número de aristas en conflicto.
- Se trabajó exclusivamente con el valor de $\epsilon = 0$.

Si se quisiera tratar casos para valores diferentes de ϵ o grafos que contengan pesos diferentes de 1 sería necesario realizar algunas modificaciones en el algoritmo propuesto. Sin embargo, esto queda fuera del alcance de este trabajo.

Trabajo relacionado

Esta sección está dedicada a la revisión de trabajos que están altamente relacionados con nuestro enfoque. Primero, entre el gran conjunto de Algoritmos Evolutivos (AES) que se han usado para la resolución del GPP, se discuten aquellos que han obtenido los resultados más prometedores, a la vez que aquellos que comparten varias características con nuestra propuesta. Además, puesto que nuestra propuesta está basada en el control de la diversidad, se incluye un resumen de las técnicas más populares que se han propuesto para mitigar los efectos de la convergencia prematura. Algunas de esas técnicas se usan para validar nuestra propuesta.

Algoritmos evolutivos para el GPP

Para lidiar con el GPP [3] se han diseñado un gran número de metaheurísticas poblacionales. Entre ellas, los AES son los enfoques más populares. Desde las propuestas iniciales [4], fue claro que incorporar un procedimiento para intensificar en las regiones localizadas por el AE era bastante importante. En la mayoría de los algoritmos iniciales, se toma en cuenta la heurística propuesta por Kernighan y Li. Esta heurística es un proceso barato computacionalmente que obtiene particiones competitivas. Con el incremento del poder computacional, se han propuesto formas más complejas de intensificar. En algunos casos, se integran AES con estrategias basadas en búsquedas de trayectoria [8], mientras que en otros casos, se usan búsquedas locales por escalada con grandes vecindades [21]. Estas clases de enfoques implican el uso de recursos adicionales, pero actualmente su uso es imprescindible para alcanzar las mejores soluciones conocidas al considerar grafos de gran tamaño.

Se han dedicado también muchos esfuerzos al diseño de operadores de cruce. En los enfoques iniciales, se aplicaron operadores simples, como el uniforme y/o el cruce de n -puntos [17]. Sin embargo, rápidamente fue claro que estos operadores eran bastante destructivos. Particularmente, tenían la tendencia de crear particiones muy desbalanceadas, implicando que después de los mecanismos de balance, la descendencia pudiera compartir sólo una pequeña cantidad de las características de los padres [3]. En [5] se propone un intento de evitar este problema. En este caso, los vértices que comparten la misma clase en ambos padres mantienen su clase en los descendientes. Luego, los vértices restantes se fijan usando una heurística constructiva voraz que asegura el balance apropiado. Un elemento que no se toma en cuenta en los operadores previos es que el identificador específico de la clase de un vértice no es lo importante. Las características importantes están relacionadas con el conjunto de vértices que compartan la misma clase. Tomando esto en cuenta, se proponen operadores de cruce más complejos [8] basados en heredar estas clases de

características. El principio tras estos esquemas es transmitir grandes conjuntos de vértices que compartan la misma clase de padres a hijos. En [8] se usa un enfoque voraz basado en maximizar intersecciones y previamente en [7] se discuten propuestas no voraces. De cualquier manera, estas ideas se aplicaron al problema del coloreado de grafos y no al GPP. Puesto que esta clase de operadores han reportado resultados prometedores, nuestra propuesta toma en cuenta este principio. Particularmente, algunas de las ideas sugeridas en [7] inspiraron el desarrollo de nuestro nuevo operador de cruce. Sin embargo, nuestro operador además induce la creación de particiones relativamente balanceadas en la descendencia, lo que es una característica importante en el GPP y no necesariamente en el problema del coloreado de grafos.

Finalmente un problema importante que no ha sido estudiado en profundidad es el manejo de la diversidad. Sin embargo, algunos análisis parecen indicar que el manejo adecuado de la diversidad puede ser importante para el funcionamiento adecuado de los optimizadores en el área del GPP. Por ejemplo, en [9], el número de generaciones se limitó a 200 y los autores argumentan que después de este número de generaciones, todos los miembros de la población se encontraban en una misma región y que mejoras con ejecuciones más largas eran poco probables. Adicionalmente, en muchos casos, se incluyen algunas acciones para retrasar la convergencia. Por ejemplo, en [4] el descendiente sustituye al padre más similar, mientras que recurrir a mecanismos de reinicio cuando se detecta convergencia prematura es otra de las estrategias más utilizadas [19].

Control de diversidad en algoritmos evolutivos

Dado que en nuestro enfoque se incorpora una forma explícita de controlar la diversidad, esta subsección discute algunas de las estrategias más populares que han sido diseñadas con la intención de evitar la convergencia prematura. Particularmente, se describen las técnicas que han sido usadas en la validación experimental de nuestra propuesta. Se remite a los lectores a [26] para consultar descripciones más detalladas de estas y otras técnicas relacionadas. Nótese que se han propuesto un gran número de técnicas para tratar con convergencia prematura [20]. Los métodos se clasifican usualmente con base en la componente del AE que modifican en los siguientes grupos [26]: *basados en selección*, *basados en población*, *basados en mutación o cruce*, *basados en función objetivo* y *basados en reemplazamiento*. Los métodos basados en reemplazamiento han reportado resultados prometedores en varios problemas de optimización [25] [23]. Dado que nuestra propuesta pertenece a esta categoría, se consideran otros tres esquemas basados en reemplazamiento para este artículo:

- *Restricted Tournament Selection* (RTS) [10] es un esquema popular en el cual después de que un nuevo individuo (O) es creado, CF individuos de la población son seleccionados aleatoriamente. Luego, el mejor individuo entre (O) y el más similar a éste entre los del conjunto seleccionado sobrevive. Nótese que este esquema pertenece al grupo de estrategias de amontonamiento [18].
- En el Algoritmo Híbrido Genético con Control de Diversidad Adaptativo [27] (HGSADC - Hybrid Genetic Search with Adaptive Diversity Control), los individuos son ordenados con base en su contribución a la diversidad y a su costo original. Luego, dichas ordenaciones de los individuos son usadas para calcular una puntuación usando dos parámetros (N_{Close} y N_{Elite}) la cual es usada para determinar a los individuos sobrevivientes.
- El Algoritmo "Contribution Diversity - Replace Worst" (CD/RW) [16] es un algoritmo de estado estacionario en el que los nuevos individuos tratan de reemplazar a otros que sean peores que ellos tanto al considerar contribución a la diversidad como cali-

Algoritmo 1 Algoritmo Memético tipo Lamarck

```
1: Generar una población inicial.
2: while No se satisfagan las condiciones de paro do
3:   Evaluar todos los individuos en la población.
4:   Selecciona Padres de entre la población para el conjunto de reproducción  $\Omega$  a través de un operador  $S(\cdot)$ .
5:   for Individuos en  $\Omega$  do
6:     Evolucionar uno o varios descendientes a través de operador  $C(\cdot)$ .
7:     Mutar descendientes con operador  $M(\cdot)$ .
8:     Realizar Aprendizaje Individual en descendientes a través de un operador  $IL(\cdot)$  conforme al espíritu del aprendizaje Lamarckiano, es decir, actualizando los valores de las variables.
9:   end for
10:  Reemplazar la población actual a partir de los descendientes y la población anterior usando  $R(\cdot)$ .
11: end while
```

dad. En caso de que eso no sea posible se utiliza la técnica RW, consistente en reemplazar al peor individuo.

Nótese que, además de las técnicas con control de diversidad, incorporamos un esquema generacional con elitismo (GEN_ELIT) [6] para nuestra validación experimental. Este esquema no incorpora un mecanismo especial para retrasar la convergencia. Sin embargo, fue tomado en consideración porque es una estrategia muy aplicada en el campo de los AES. En el GEN_ELIT la nueva población contiene la descendencia y la mejor solución de la generación anterior.

Propuesta algorítmica: memético para el GPP

Esta sección está dedicada a describir nuestra propuesta para el GPP. Se presenta a detalle la nueva propuesta algorítmica, cada uno de los componentes usados y la manera de tener implementaciones eficientes de ellos. Nuestra propuesta es un AE que incorpora una búsqueda de trayectoria. El algoritmo se deriva de la estructura de un Algoritmo Memético estándar [13] (véase Algoritmo 1). Sin embargo, para satisfacer las restricciones del GPP es necesario modificar el esquema general incorporando una estrategia de balanceo, ya que tras realizar el proceso de creación de hijos, las nuevas particiones pueden estar desbalanceadas, y por tanto, no ser válidas. Con el objetivo de caracterizar completamente nuestra propuesta, se deben definir los operadores de cruce $C(\cdot)$, mutación $M(\cdot)$, el operador de balanceo, el aprendizaje individual $IL(\cdot)$, la estrategia de selección de padres $S(\cdot)$ y el operador de reemplazamiento $R(\cdot)$. Las siguientes subsecciones contienen detalles de cada una de estas componentes.

Operador de cruce

Como es habitual, el operador de cruce toma dos individuos y genera dos nuevos con características similares a los padres. Muchos de los operadores de cruce que han sido usados para el GPP están inspirados en aquellos desarrollados para el problema del coloreado de grafos. Para nuestra propuesta se ha diseñado un nuevo operador de cruce, el Operador de Cruce Basado en el Algoritmo Húngaro (HBX - Hungarian Based Crossover) — ver Algoritmo 2 — que está inspirado en uno propuesto en [7] para el problema del coloreado de grafos. Particularmente, HBX se basa en el principio de maximizar el número de nodos que comparten la misma clase en padres y descendientes.

El HBX funciona como sigue. Sean $I_1 = \{S_1^1, S_2^1, \dots, S_k^1\}$ y $I_2 = \{S_1^2, S_2^2, \dots, S_k^2\}$ los padres seleccionados como entrada para el operador de cruce. El método Húngaro es un algoritmo de optimización popular que resuelve, en tiempo polinomial, el problema de asignación en grafos bipartitos pesados, es decir, calcula el máximo o mínimo emparejamiento pesado. El algoritmo comienza construyendo un grafo completo bipartito (G) con $2 \times k$ vértices, de forma que los primeros k vértices están asociados a los subconjuntos de I_1 , mientras que los últimos corresponden a los subconjuntos de I_2 . Las aristas conectan cualquiera de los primeros k vértices con cualquiera de los últimos

Algoritmo 2 Operador de reproducción basado en el algoritmo Húngaro

```
1: Sea  $elg_r$  un vector booleano para las filas "elegibles".
2: Sea  $elg_c$  un vector booleano para las columnas "elegibles".
3: Sea  $blc_r$  un vector booleano para las filas "bloqueadas".
4: Sea  $blc_c$  un vector booleano para las columnas "bloqueadas".
5: Fijar los vectores  $elg$  a verdadero.
6: Fijar los vectores  $blc$  a falso.
7: for  $i$  en  $\{1, 2, \dots, n\}$  do
8:   if  $i \equiv 0, \text{mod}(2)$  then
9:      $L = -1$ ;  $B_L = \emptyset$ .
10:    for  $l$  tal que  $elg_c[l]$  es verdadero (de forma aleatoria) do
11:       $B_l = \{P_{i,i} \text{ no eliminadas}\} \cup \{P_{\sigma^{-1}(l),m} \text{ tal que } blc_c[m] \text{ es verdadero}\}$ .
12:      Si  $(|B_l| > |B_L|)$  fijar  $L = l$ .
13:    end for
14:     $T_j = B_L$ .
15:    Elimina todos los elementos de  $T_j$  de la matriz  $P$ .
16:     $elg_c[L] = \text{falso}$ .
17:     $elg_r[\sigma^{-1}(L)] = \text{falso}$ .
18:     $blc_r[\sigma^{-1}(L)] = \text{verdadero}$ .
19:  else
20:     $L = -1$ ;  $B_L = \emptyset$ .
21:    for  $l$  tal que  $elg_r[l]$  es verdadero (en forma aleatoria) do
22:       $B_l = \{P_{l,i} \text{ no eliminados}\} \cup \{P_{m,\sigma(l)} \text{ tal que } blc_r[m] \text{ es verdadero}\}$ .
23:      Si  $(|B_l| > |B_L|)$  fijar  $L = l$ .
24:    end for
25:     $T_j = B_L$ .
26:    Eliminar de  $P$  todos los elementos de  $T_j$ .
27:     $elg_r[L] = \text{falso}$ .
28:     $elg_c[\sigma(L)] = \text{falso}$ .
29:     $blc_c[\sigma(L)] = \text{verdadero}$ .
30:  end if
31: end for
```

k , quedando el peso de cada arista establecido como el tamaño de la intersección de los conjuntos asociados con cada vértice. La permutación σ obtenida con la aplicación del Algoritmo Húngaro a G maximiza $\sum_{i=1}^k \|S_i^1 \cap S_{\sigma(i)}^2\|$.

Principales propiedades del HBX

Previo a dar el procedimiento específico, es importante discutir algunas de las propiedades que tendrán los descendientes, que se consideraron como propiedades deseables al momento de diseñar el HBX. Sea P la matriz de intersección definida como $P_{i,j} = S_i^1 \cap S_j^2$ y sea $T = \{T_1, T_2, \dots, T_k\}$ el hijo creado, entonces T tiene las siguientes propiedades:

1. Para cada T_j , existe un único i_j tal que $S_{i_j}^1 \cap S_{\sigma(i_j)}^2 \subset T_j \subset S_{i_j}^1 \cup S_{\sigma(i_j)}^2$.
2. $P_{i,j} \subset T_l$ para todos i, j y algún l .
3. Todo T_l es la unión de exactamente k $P_{i,j}$ distintos.

La razón tras el uso de σ es que el Algoritmo Húngaro da el máximo emparejamiento entre los conjuntos de ambas particiones, lo cual está relacionado con el principio enunciado de maximizar el número de nodos que comparten la misma clase en padres y descendientes. Esto significa que S_i^1 es de alguna manera similar a $S_{\sigma(i)}^2$ y su intersección es susceptible a tener muchos elementos. La propiedad 1) establece que estas intersecciones se usan como el núcleo de los conjuntos en la nueva partición. También establece que los elementos son seleccionados exclusivamente de uniones de los subconjuntos considerados. Tomando estas propiedades en consideración se puede decir que los operadores que cumplen estas propiedades no son demasiado destructivos ya que se enfocan en heredar en el mismo conjunto grandes subconjuntos de vértices que fueron asignados al mismo subconjunto en ambos padres. Finalmente el principio tras la tercera propiedad, es evitar el exceso de desbalanceo en los tamaños de los nuevos subconjuntos generados. Sin embargo, note que usualmente aparece cierto grado de desbalanceo, por lo que es necesario un método adicional para manejar correctamente el desbalanceo de los subconjuntos.

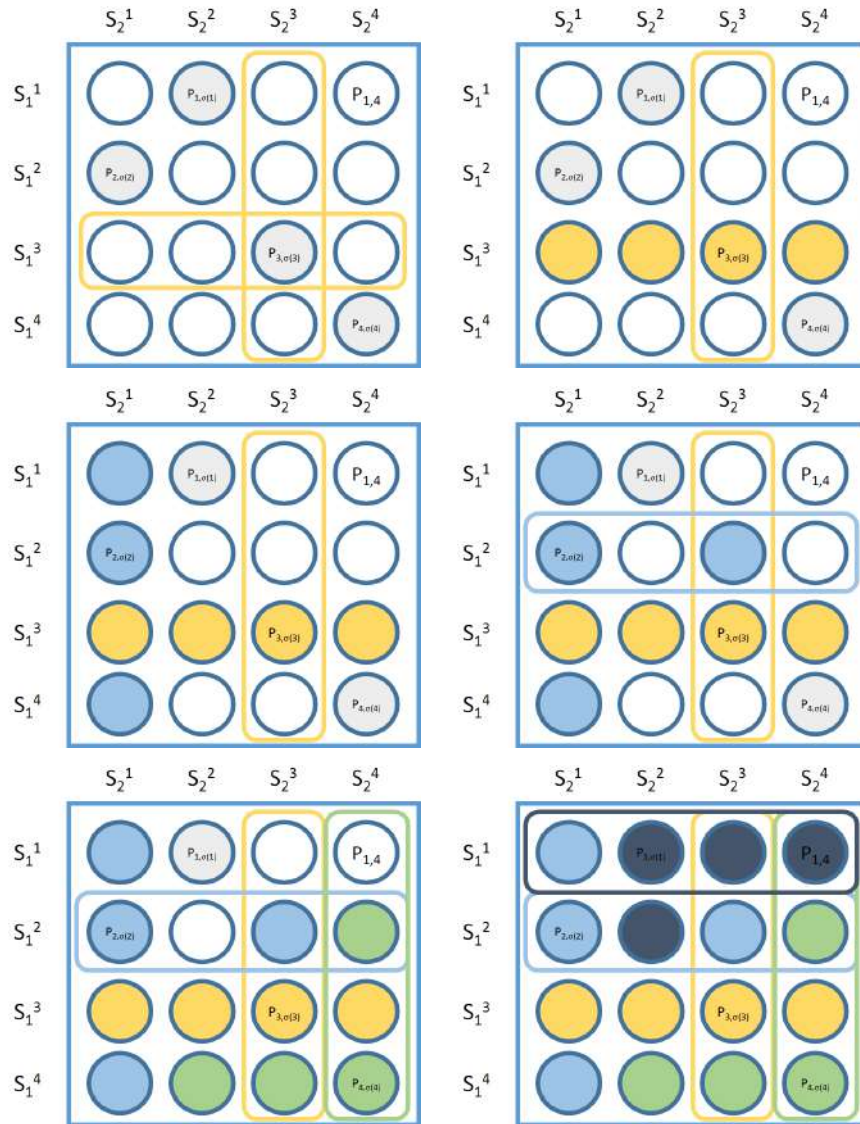


Figura 1: Ilustración del HBX

Construcción

Esta sección está dedicada a presentar el algoritmo para construir dos descendientes que cumplan las propiedades anteriormente mencionadas. Sea P la matriz de intersección previamente definida. Entonces el Algoritmo 2 es el proceso requerido para crear el primer individuo. Este proceso se ilustra en la Figura 1. La base del método es seleccionar en cada paso la posición $(i, \sigma(i))$ de la matriz, donde la unión de algunos elementos de la fila y columna correspondiente (ver detalles en línea 11 y 22) es maximizada. Nótese que se puede probar que en la línea 11 existen exactamente $k - j/2$ $P_{i,l}$ entradas no borradas de P y $j/2$ $P_{\sigma^{-1}(l),m}$ entradas tal que la columna m está “bloqueada”. Algo similar ocurre en la línea 22 y esto prueba la propiedad 3 del HBX.

Para generar el segundo individuo, se intercambia el orden en el cual las filas y las columnas son consideradas, es decir, en las iteraciones pares se toman en consideración las columnas, mientras que en las iteraciones impares se toman en cuenta las filas. Nótese que independientemente del orden usado, ambos individuos comparten las

mismas propiedades descritas anteriormente.

En la Figura 1 se muestra el funcionamiento del HBX para un individuo específico. En este caso $k = 4$, $\sigma = (2, 1, 3, 4)$. Cada una de las seis imágenes en la Figura 1 puede ser considerada como una matriz, P . Cada entrada $P_{i,j}$ de esas matrices representa la intersección de los conjuntos correspondientes en los padres S_i^1, S_j^2 . En la última imagen, la unión de los conjuntos con el mismo color representa un elemento en la nueva partición, T . La segunda imagen muestra que T_1 es escogido de manera aleatoria como una fila de P , e ilustra que la columna 3 está bloqueada. Note que la fila fue elegida aleatoriamente y la columna considera la permutación dada por el Algoritmo Húngaro. En la tercera imagen, parte de T_2 es tomada de una columna de P y el resto de los elementos se toman de los conjuntos en la fila 2 que están en una columna bloqueada ($P_{2,3}$) tal y como se aprecia en la imagen 1. Similarmente T_3 y T_4 son creados en la imagen 5 y 6, todo siguiendo el procedimiento descrito en el Algoritmo 2.

Otro hecho importante es que para el nuevo operador, no se usa ninguna información propia de la definición del GPP. Por tanto, este

Algoritmo 3 Mutación basada en mover Componentes Conexas Aleatoriamente

```
1: Sea  $u$  un nodo elegido al azar.
2: Sea  $set$  un conjunto que inicialmente sólo contiene a  $u$ .
3:  $k = n_{iter}$ 
4: while  $k > 0$  do
5:   for  $u$  en  $set$  do
6:     for  $v$  vecino de  $u$  do
7:       Tomar  $p$  aleatorio en  $[0, 1]$ .
8:       if  $p < p_m$  then
9:         Agregar  $v$  a  $set$ 
10:      end if
11:    end for
12:  end for
13:   $k = k - 1$ 
14: end while
15: Mover todos los vértices en  $set$  a una componente elegida aleatoriamente.
```

operador de cruce puede ser usado en cualquier problema de optimización en el cual compartir valores en diferentes variables sea una característica importante.

Operador de mutación

En los AES es común incluir un operador de mutación, que en algunos casos se usa para realizar pequeños cambios que permitan intensificar, mientras que en otros realiza cambios más bruscos para escapar de óptimos locales. Dado que en nuestra propuesta se incluye búsqueda tabú para intensificar, se integró con una mutación disruptiva.

La mutación propuesta se basa en que en el GPP se desea que componentes conexas estén juntas en uno de los subconjuntos de la partición. La razón es que las componentes conexas encapsulan aristas comunes, implicando que se inducirán menos aristas de corte. Considerando esta propiedad, el principio de la mutación es mover partes de componentes conexas a uno de los subconjuntos. Se aplica el enfoque presentado en [24] (ver Algoritmo 3) para la selección de esta subcomponente. Luego, los vértices seleccionados se mueven a un conjunto S_i , donde i es seleccionado de manera aleatoria. En este trabajo se usa $p_m = 0,1$ y $n_{iter} = 5$. Es importante enfatizar que la mutación propuesta en [24] fue presentada para el problema de asignación de frecuencias.

Fase de reemplazamiento

Nuestro algoritmo memético aplica una fase de reemplazamiento original llamada “Estrategia de supervivencia del Mejor No Penalizado” (“Best-Non-Penalized (BNP) survivor selection strategy”) (ver Algoritmo 4). Uno de los principios de la estrategia BNP es evitar la selección de individuos muy cercanos. Específicamente el enfoque intenta evitar la selección de pares de individuos que estén más cerca que un valor D uno del otro. El valor de D varía durante la ejecución, puesto que en las fases iniciales es importante explorar, mientras que en las últimas fases el proceso debe intensificar. En particular, la variable D es inicializada con base en el contenido de la primera población (ver línea 9) y disminuye de manera lineal hasta alcanzar el valor 0 al final de la ejecución.

En cada ejecución del operador de supervivencia se seleccionan N individuos de la población previa y los descendientes para sobrevivir. Iterativamente BNP selecciona el mejor elemento que no está penalizado. Luego, los individuos restantes con disimilitud a lo más D de los previamente seleccionados son penalizados. Si no es posible seleccionar un individuo no penalizado, se toma el individuo con mayor disimilitud a los actualmente seleccionados.

Un hecho importante que debe notarse es que el operador de supervivencia requiere una función de disimilitud entre los individuos. En este caso, se usa una función basada en el Algoritmo Húngaro. Sea τ una permutación de k elementos, entonces $d(S^1, S^2, \tau) =$

Algoritmo 4 Técnica BNP de supervivencia

```
Require: Population, Offspring
1: for all  $I \in$  Offspring do
2:    $I.cost =$  coste de corte asociado al individuo  $I$ 
3: end for
4: Penalizados =  $\emptyset$ 
5: CurrentIndividuals = Population  $\cup$  Offspring
6: Best = Individuo con el coste más bajo de corte en CurrentIndividuals
7: NewPop = { Best }
8: CurrentIndividuals = CurrentIndividuals  $\setminus$  { Best }
9: Actualiza  $D$  tomando en cuenta el valor del tiempo actual ( $T_e$ ), tiempo de paro ( $T_p$ ) y valor inicial de  $D$  ( $D_I$ ). Por ejemplo, en el caso de decrecimiento lineal, realiza  $D = D_I - D_I * \frac{T_e}{T_p}$ . Tomamos  $D_I = distInitFactor * M$ , donde  $M$  es la disimilitud media en la primera población y  $distInitFactor \in [0, 1]$ .
10: while ( $|NewPop| < N$ ) do
11:   for all  $I \in$  CurrentIndividuals do
12:      $I.DCN =$  disimilitud al individuo más cercano en NewPop
13:     if  $I.DCN < D$  then
14:       Penalized = Penalized  $\cup$  { $I$ }
15:       CurrentIndividuals = CurrentIndividuals  $\setminus$  { $I$ }
16:     end if
17:   end for
18:   for all  $I \in$  Penalized do
19:      $I.DCN =$  disimilitud al individuo más cercano en NewPop
20:   end for
21:   if CurrentIndividuals is empty then
22:     Selected =  $I$  con el  $I.DCN$  más grande en Penalized
23:     Penalized = Penalized  $\setminus$  {Selected}
24:   else
25:     Selected =  $I$  con el  $I.cost$  más bajo en CurrentIndividuals
26:     CurrentIndividuals = CurrentIndividuals  $\setminus$  {Selected}
27:   end if
28:   NewPop = NewPop  $\cup$  {Selected}
29: end while
30: Population = NewPop
```

Algoritmo 5 Búsqueda Tabú

```
Require:  $I$ 
1:  $it = 0$ 
2:  $Best_I = I$ 
3: while No se cumpla criterio de paro do
4:   ( $u, c$ ) = movimiento válido de menor coste fuera de la lista de Tabú.
5:    $I = I(u, c)$ 
6:   if  $cost(Best_I) > cost(I)$  then
7:      $Best_I = I$ 
8:   end if
9:    $it = it + 1$ 
10:  Agregar ( $u, c$ ) a lista de Tabú por  $t(it)$  iteraciones.
11: end while
12:  $I = Best_I$ .
```

$|V| - \sum_{i=1}^k \|S_i^1 \cap S_{\tau(i)}^2\|$. Nuestra disimilitud, $d(S^1, S^2)$ es definida como el mínimo de $d(S^1, S^2, \tau)$ sobre todas las permutaciones τ 's. Este valor puede calcularse eficientemente usando el Algoritmo Húngaro.

Aprendizaje individual

Con la intención de crear individuos con mejor aptitud, en los algoritmos meméticos se incorpora un operador de “aprendizaje individual”. Esta rutina se ejecuta después del balanceo y consiste de dos fases. La primera es una Búsqueda Tabú propuesta en [9] y la segunda es la estrategia denominada “búsqueda local perfectamente balanceada por medio de ciclos negativos” que se propuso en [21].

Galinier describe que esta variante de Búsqueda Tabú es una modificación del TabuCol propuesto para el coloreado de grafos [12]. Se recibe $I = \{V_1, V_2, \dots, V_k\}$, una partición balanceada, y en cada iteración se mueve un nodo de un conjunto de la partición a otro. El movimiento del nodo u al conjunto c se considera válido si y sólo si el subconjunto en el que está el nodo u no es c y además $|V_c| < lim + \delta$, estableciendo $\delta = 1$ si $n = 0 \pmod k$ y $\delta = 0$, en caso contrario. Esta elección de δ es para asegurar que siempre exista al menos un movimiento válido.

Llamemos $I(u, c)$ a la partición cuya diferencia con I es que el nodo u es movido al conjunto c de la partición y $M(u, c)$ a dicho movimiento. El coste de $M(u, c)$ es igual a $cost(I(u, c)) - cost(I)$. El algoritmo 5 describe la búsqueda tabú. El criterio de paro puede ser la cantidad de segundos o iteraciones que queremos que se ejecute. En

nuestro caso el algoritmo termina cuando no se ha mejorado la mejor solución, $Best_I$, en las últimas $noImprove$ iteraciones. El valor de $noImprove$ se eligió empíricamente y se muestra el proceso para hacerlo en la sección de validación experimental. La función $t(it)$, que calcula el número de iteraciones por las que un movimiento quedará en la lista tabú, es una función escalón periódica dependiente de un parámetro $maxT$. Para su construcción definiremos unos x_i que cumplirán que entre x_i y x_{i+1} la función será constante con cierto valor a_i . Formalmente la construcción de t se lleva a cabo de la siguiente forma:

- $b = \frac{1}{8}(1, 2, 1, 4, 1, 2, 1, 8, 1, 2, 1, 4, 1, 2, 1)$.
- $a_i = maxT \times b_i$.
- $x_1 = 1$ y $x_{i+1} = x_i + 4 \times maxT \times b_i$, para los i tales que b_i está definido.
- Finalmente $\forall it \in x_i, \dots, x_{i+1}, t(it) = a_i$.

Se completa t para el resto de los naturales de manera periódica. Como ejemplo Galinier y Countiho usan $maxT = 200$, valor que adoptamos en nuestro esquema.

La intención con nuestro esquema es poder resolver el GPP en grafos de tamaños muy grandes. Esto provoca que se necesite tener una implementación eficiente del algoritmo 5. Al hacer una implementación trivial del algoritmo anterior, la parte donde se pierde la eficiencia en dicho código es en la línea 4, es decir, lo más costoso es encontrar el movimiento válido de menor coste fuera de la lista de Tabú, como suele ser habitual en las implementaciones de búsqueda tabú. Se pueden proponer varias estrategias para la implementación de este proceso. En la forma trivial se realiza una búsqueda lineal sobre todos los nodos y todos los conjuntos, lo que haría que esa línea tuviera costo al menos $O(n \times k)$, lo que es muy ineficiente para nuestros propósitos. La solución incorporada en nuestro esquema hace uso de una estructura llamada montículo binario actualizable (MBA). Cabe mencionar que la optimización aquí presentada es idea propia, pues Galinier no hace explícita su implementación. En esta estructura se almacenan pares (id, v) , donde id es el identificador único de un elemento y v un valor de prioridad y la estructura permite encontrar el par (id, v) con menor v , así como actualizar $(id, v) \rightarrow (id, v')$ en tiempo logarítmico sobre el número de elementos en el montículo. En particular, nuestra implementación del MBA puede realizar las siguientes operaciones en tiempo logarítmico sobre el número de pares almacenados:

- **top()**: Devuelve el par (id, v) contenido en la estructura con menor valor v .
- **sum(id, w)**: Transforma el elemento (id, v) contenido en la estructura en el elemento $(id, v + w)$.
- **update(id, w)**: Cambia el elemento (id, v) en la estructura por (id, w) .

Además, se tiene desarrollada la función $gainValue(u, i)$ que devuelve la ganancia que habría si se moviera el nodo u al conjunto i en el individuo actual. Si se mantiene una matriz de tamaño $n \times k$ donde se indique en la entrada $[u, i]$ cuántos nodos vecinos tiene u en el conjunto i , entonces esta operación se puede responder en $O(1)$ ya que el coste de mover el nodo u del conjunto i al j se puede calcular como la diferencia de las entradas $[u, i]$ y $[u, j]$. Con esto, la nueva manera de implementar la línea 4 se detalla en el Algoritmo 6.

Esto permite que cada iteración conlleve $O(NumVecinos(u) \times k \times \log(\|V\|))$ que, al usar grafos raros, mejora el rendimiento significativamente. En este caso se trabaja con k MBAs debido a que necesitamos diferenciar entre los movimientos a diferentes conjuntos. En específico, si se usara sólo uno podría suceder que el mejor movimiento esté asociado a un conjunto ya lleno por lo que sería inválido

Algoritmo 6 Búsqueda Tabú Eficiente

```

Require: I
1: it = 0
2: Best_I = I
3: Inicializar mba[k] donde cada mba[c] es un MBA con la información
   (u, cost(I(u, c)) - cost(I)).
4: while No se cumpla criterio de paro do
5:   Despenalizar cada uno de los elementos (u, c) que salieron de la lista de tabú.
   mba[c].sum(u, -INF).
6:   (u, c) = min_{i=1,2,...,k} {mba[i].top() | \|I_i\| < lim}
7:   I = I(u, c)
8:   if cost(Best_I) > cost(I) then
9:     Best_I = I
10:  end if
11:  it = it + 1
12:  for v in {vecinos de u} union {u} y i in {1, 2, ..., k} do
13:    Actualizar valor: mba[i].update(v, gainValue(v, i)).
14:  end for
15:  Agregar (u, c) a lista de Tabú por t(it) iteraciones.
16:  Penalizar: mba[c].sum(u, INF).
17: end while
18: I = Best_I.

```

Algoritmo 7 Operador de Balanceo

```

1: Sea I el individuo de entrada.
2: // Comienzo de la primera fase.
3: t = n_iterBal
4: while t > 0 y I no esté balanceado do
5:   Seleccionar i aleatoriamente en {i tal que \|I_i\| no es el conjunto más grande de
   I}.
6:   Encontrar el nodo u que al mover a I_i de máxima ganancia en union I_j tal que
   \|I_j\| > \|I_i\| y exista arista (u, v) con v in I_i.
7:   Mover u a I_i.
8:   t = t - 1
9: end while
10: // Fin de la primera fase y comienzo de la segunda.
11: while ind no esté balanceado do
12:   Elegir u in union I_j tal que \|I_j\| > I_max.
13:   Mover u a un conjunto aleatorio I_k con \|I_k\| < I_max.
14: end while
15: // Fin de la segunda fase.

```

y en dicho caso buscar entre los movimientos válidos se degeneraría a una búsqueda lineal.

Balanceo

Esta sección está dedicada a presentar el operador de balanceo que se integró en nuestro esquema así como a detallar una implementación eficiente del mismo.

La rutina de balanceo consiste en dos fases y es muy importante debido a la restricción de balance dada por la definición del problema. Esta restricción puede no cumplirse después de la aplicación del operador de cruce y/o el operador de mutación por lo que se debe restaurar. En el Algoritmo 7 se da el procedimiento que está a cargo de esta parte.

En la primera fase, se mueven los nodos de los conjuntos más grandes a conjuntos más pequeños tomando en cuenta la implicación de moverlos en relación con el valor de la función de costo. Este proceso se repite $n_{iterBal}$ veces, el cual se fijó a $|V|$ en nuestra validación experimental.

La segunda fase, que provoca el cumplimiento de la restricción de desbalanceo, mueve de manera aleatoria nodos de conjuntos que no satisfacen la restricción dada a conjuntos que aún tienen capacidad suficiente para integrar más nodos.

Como en el caso de la Búsqueda Tabú, nos encontramos en una situación donde cierta parte del algoritmo necesita ser implementada de manera eficiente. En este caso es la línea 6 en el Algoritmo 7. Una implementación directa de esta línea puede tener costo del orden $O(\|E\|)$, lo que volvería este operador demasiado costoso computacionalmente. La idea para solucionar esto será nuevamente hacer uso del Montículo Binario Actualizable (MBA). Esta vez se mantendrán k^2 MBAs que contendrán información de la forma (e, v) donde $e \in E$ y v es un valor de prioridad. Específicamente se mantendrá una matriz de MBAs donde $(u, v) \in E$ estará en el MBA de la posición (i, j) si

Algoritmo 8 Operador de Balanceo Eficiente

```
1: Sea  $I$  el individuo de entrada.
2:  $t = n_{iterBal}$ .
3: Construir  $mba[k][k]$  con información actualizada.
4: while  $t > 0$  y  $I$  no esté balanceado do
5:   Seleccionar  $i$  aleatoriamente en  $\{i \text{ tal que } \|I_i\| \text{ no es el conjunto más grande de } I\}$ .
6:   Hallar  $(u, v) = \min_{j \in \{1, 2, \dots, k\}} \{mba[j][i, top() \mid \|I_j\| > \|I_i\|\}$ .
7:   Eliminar las aristas vecinas de  $u$  de sus respectivos  $mba$ .
8:   Mover  $u$  a  $I_i$ .
9:   Insertar las aristas vecinas de  $u$  en sus nuevos  $mba$  con el nuevo valor de reducción en aptitud.
10:   $t - -$ 
11: end while
12: while  $I$  no esté balanceado do
13:   Elegir  $u \in \cup I_j$  tal que  $\|I_j\| > I_{max}$ .
14:   Mover  $u$  a un conjunto aleatorio  $I_k$  con  $\|I_k\| < \|I_{max}\|$ .
15: end while
```

y sólo si $(I[u], I[v]) = (i, j)$ y su valor de prioridad asociado será la reducción del valor de aptitud si se moviera el nodo u al conjunto $I[v]$. Además, llamaremos una arista, (u, v) , vecina de un nodo, w , si u o v es vecino de w . Bajo estas definiciones, algo que es importante notar es que, si se mueve el nodo u al conjunto $I[v]$, las únicas aristas que cambian su valor de prioridad son las aristas vecinas de u . Si el grafo es ralo, este valor, $eVec(u)$, no será demasiado grande. Con esto en mente, el operador de balanceo queda ilustrado en el Algoritmo 8.

El nuevo algoritmo vuelve a la sección equivalente a la línea 5 en un procedimiento con costo $O((k + eVec(u)) \times \log(\|E\|))$, que es mucho mejor que el anterior costo. Lo anterior muestra una vez más que se puede aprovechar el uso de estructuras de datos más complejas para tener implementaciones eficientes de un algoritmo.

Propuesta final

Una vez revisada cada una de las componentes necesarias para la implementación de un algoritmo memético, la versión final de nuestra propuesta se ve reflejada en el Algoritmo 9. Debido a las componentes de nuestro algoritmo, lo decidimos llamar “Algoritmo Memético con Cruce basado en el Algoritmo Húngaro y Control de Diversidad” y en adelante lo abreviaremos MAHMBDCP (Memetic Algorithm with Hungarian Matching Based Crossover and Diversity Preservation). En este algoritmo se comienza con una población aleatoria con la intención de muestrear soluciones distantes y se evalúan las soluciones generadas. Recordemos que la disimilitud inicial del operador de supervivencia se fija con base en las disimilitudes iniciales de la población. Por esta razón se realiza primero una fase de aprendizaje individual, pues calcular las disimilitudes antes del aprendizaje haría que dicha disimilitud fuera demasiado grande y se forzaría el mantenimiento de soluciones de muy baja calidad. De ahí se entra a un ciclo del cual no se saldrá hasta que haya transcurrido un tiempo T . Dentro del ciclo se inicia suponiendo que todos los individuos tienen su valor de aptitud disponible durante la selección de los padres de la siguiente generación. La selección de dichos padres se hace por medio de torneo binario, un método que necesita comparar los valores de aptitud de los individuos. Una vez elegidas las parejas, si fueron seleccionadas para el cruce, se llama al operador de cruce basado en el Algoritmo Húngaro. A continuación se decide si el individuo será además mutado. Al salir de esta última rutina tendremos nuevos individuos cuyo valor de aptitud puede no estar actualizado y pueden no cumplir la restricción de balance. Por esta razón, inmediatamente se lanza el método de reparación, el cual consiste en el rebalanceo previamente explicado. Una vez balanceadas se llama nuevamente al operador de aprendizaje individual con la intención de mejorar a los individuos que competirán por continuar en la población. Esta fase incluye a la búsqueda tabú y la optimización basada en detección de ciclos negativos ya mencionada. Finalmente se llama al operador de supervivencia eligiendo así a los individuos que conformarán la siguiente generación del algoritmo.

Algoritmo 9 Algoritmo Memético con Cruce basado en el Algoritmo Húngaro y Control de Diversidad

```
1: Inicialización: Genera una población inicial  $P_0$  con  $N$  individuos. Asignar  $t = 0$ .
2: Evaluación: Evaluar todos los individuos en la población.
3: Balanceo: Los individuos se balancean inicialmente.
4: Búsqueda Tabú
5: Búsqueda perfectamente balanceada por medio de ciclos negativos
6: Inicializar  $D_I$ : Se hace  $D_I = distInitFactor * M$ , donde  $M$  es la disimilitud media de la población.
7: while Tiempo transcurrido sea menor que  $T$  do
8:   Emparejamiento: Realizar un torneo binario sobre  $P_t$  para formar  $\frac{N}{2}$  parejas de padres.
9:   Cruce Basado en el Algoritmo Húngaro: Se aplica sobre las parejas formadas con probabilidad  $p_c$  para crear el conjunto de hijos  $CP$ .
10:  Mutación: Se mutan los individuos con probabilidad  $p_m$ .
11:  Evaluación: Evaluar todos los individuos en la población.
12:  Balanceo: Aplicar la fase de balanceo para cumplir las restricciones de balance.
13:  Búsqueda Tabú
14:  Búsqueda perfectamente balanceada por medio de ciclos negativos
15:  BNP: Se aplica a la combinación de  $P_t$  y  $CP$  y guarda en  $P_{t+1}$  el resultado.
16:   $t = t + 1$ 
17: end while
```

Validación experimental

En esta sección se describe el trabajo experimental llevado a cabo para validar las propuestas algorítmicas de la tesis en la que este trabajo se basa. La sección incluye varios experimentos enfocados a analizar y comprender mejor el rendimiento y comportamiento de la técnica desarrollada y la importancia de las componentes propuestas. Además se incluyen comparativas con las mejores soluciones reportadas hasta la fecha para mostrar la contribución de la técnica en lo referente a obtención de particiones de muy alta calidad.

Descripción general de los experimentos

Con el fin de validar el rendimiento de las diferentes componentes propuestas, especialmente del mecanismo de gestión de diversidad, se implementaron varias componentes diferentes y se realizaron comparativas exhaustivas entre ellas. Los esquemas fueron implementados usando la herramienta METCO [14] y teniendo en cuenta que nuestra propuesta es un algoritmo estocástico, cada ejecución se repitió 30 veces y la comparación entre los diferentes esquemas se llevó a cabo aplicando un conjunto de tests estadísticos [22]. Para evaluar el desempeño de nuestra propuesta, se usaron instancias del TGPA de Chris Walshaw [1]. Este es un sitio que ha sido mantenido desde el año 2000 e incluye resultados sobre los paquetes de particionado más importantes. Para cada uno de estos grafos, la mejor partición conocida para $k \in \{2, 4, 8, 16, 32, 64\}$ y $\epsilon \in \{0.0, 0.01, 0.03, 0.05\}$ se encuentra disponible. Para nuestra validación se tomaron en cuenta los valores $k \in \{4, 8, 16, 32, 64\}$ y $\epsilon = 0, 0$ para probar nuestro algoritmo.

Primer experimento: selección de parámetros

En primer lugar, como es habitual en el campo de optimización con metaheurísticas, el enfoque desarrollado tiene un conjunto de parámetros para los que no es trivial fijar un valor. Considerando el caso con reemplazamiento BNP los parámetros son el tamaño de población, la probabilidad de cruce, la probabilidad de mutación, el valor de $distInitFactor$ para fijar la disimilitud inicial de penalización y $noImprove$ para la búsqueda tabú. El tamaño de población se fijó a 50, la probabilidad de cruce (p_c) a 0.85 y la probabilidad de mutación (p_m) a 0.1. Estos valores son cercanos a los estándar y estamos conscientes de que se podrían obtener algunas mejoras aplicando esquemas de calibración de parámetros [15]. Sin embargo, puesto que se realizaron ejecuciones muy largas (el criterio de paro se fijó a 48 horas) y estas mostraron resultados de gran calidad, no se llevó a cabo más experimentación con dichos parámetros.

Dado que una de las características más importantes de nuestro enfoque es que considera la diversidad explícitamente y que esta de-

Tabla 1: Resultados del MAHBCDP para $k = 16, 64$ con $noImprove = 5000$

add20	$k = 16$				$k = 64$			
	Mejor	Media	Mediana	Peor	Mejor	Media	Mediana	Peor
DInit	2044	2064.5	2062.5	2091	2953	2982	2972	3025
0.0	2040	2051.7	2052	2064	2943	2950.6	2949	2959
0.2	2040	2044.8	2043	2052	2941	2949.5	2949.5	2957
0.4	2040	2042.1	2042.5	2046	2943	2950.6	2951	2959
0.6	2040	2041.7	2042	2045	2946	2953.1	2953	2965
0.8	2040	2042.5	2043	2045	2946	2952.5	2951.5	2965
1.0	2040	2042.5	2043	2045	2946	2952.5	2951.5	2965
whitaker3	$k = 16$				$k = 64$			
	Mejor	Media	Mediana	Peor	Mejor	Media	Mediana	Peor
DInit	1088	1091.2	1091	1098	2498	2512.2	2511	2526
0.0	1085	1087	1088	1088	2486	2495.4	2494.5	2511
0.2	1085	1085.8	1085	1088	2484	2489	2489	2495
0.4	1085	1085.7	1085	1088	2484	2489.9	2487.5	2508
0.6	1085	1086.2	1085	1088	2484	2487.4	2487	2495
0.8	1085	1086.3	1085	1088	2484	2486.7	2487	2491
1.0	1085	1086.3	1085	1088	2484	2486.7	2487	2491
bcstk29	$k = 16$				$k = 64$			
	Mejor	Media	Mediana	Peor	Mejor	Media	Mediana	Peor
DInit	21941	22233.9	22204.5	22800	55591	55930.1	55890.5	56761
0.0	21900	21919.8	21900.5	22009	55591	55930.1	55890.5	56761
0.2	21900	21927.5	21920	21991	55356	55637.2	55634	55838
0.4	21900	21936.6	21927	22022	55332	55589.8	55588.5	55848
0.6	21900	21941.4	21944.5	22020	55345	55601.1	55629	55803
0.8	21900	21944.7	21944.5	22020	55345	55601.1	55629	55803
1.0	21902	21935.1	21918	22004	55338	55625.6	55608	55917

Tabla 2: Resultados del MAHBCDP para $k = 16, 64$ con $distInitFactor = 0,4$

add20	$k = 16$				$k = 64$			
	Mejor	Media	Mediana	Peor	Mejor	Media	Mediana	Peor
noImprove	2040	2047.5	2046	2056	2947	2953.9	2953	2964
200	2040	2046	2045.5	2053	2943	2950.8	2950	2962
2000	2040	2044.1	2042	2054	2944	2950.2	2950	2962
5000	2040	2043.7	2042	2052	2944	2949.7	2950	2956
10000	2040	2043.2	2041.5	2052	2944	2949.6	2949	2955
15000	2040	2044.5	2043	2052	2943	2950.1	2951	2955
25000	2040	2043.8	2042	2052	2944	2950.2	2950	2961
35000	2040	2045.3	2043	2052	2944	2949.7	2950	2955
50000	2040	2044.5	2042	2053	2946	2950.7	2950	2960
60000	2040	2044.6	2043	2053	2942	2949.6	2949	2957
75000	2040	2044.6	2043	2053	2942	2949.6	2949	2957
whitaker3	$k = 16$				$k = 64$			
	Mejor	Media	Mediana	Peor	Mejor	Media	Mediana	Peor
noImprove	1085	1087.2	1088	1089	2485	2492.8	2492	2504
200	1085	1085.8	1085	1088	2484	2489.1	2489	2498
2000	1085	1085.5	1085	1088	2484	2488.6	2487	2496
5000	1085	1085.5	1085	1088	2484	2488.4	2487.5	2499
10000	1085	1085.5	1085	1088	2483	2487.1	2487	2493
15000	1085	1086.1	1085	1088	2484	2486.9	2487	2493
25000	1085	1086.6	1087	1088	2484	2488.3	2488.5	2493
35000	1085	1086.7	1086.5	1088	2484	2487.7	2488	2492
50000	1085	1086.7	1087.5	1088	2484	2488.5	2489	2493
60000	1085	1087	1088	1088	2484	2488.6	2489	2493
75000	1085	1087	1088	1088	2484	2488.6	2489	2493
bcstk29	$k = 16$				$k = 64$			
	Mejor	Media	Mediana	Peor	Mejor	Media	Mediana	Peor
noImprove	21900	21941.2	21946.5	22032	55500	55714.8	55695	56043
200	21900	21920.3	21904	21989	55442	55657.4	55629.5	56006
2000	21900	21917.7	21909.5	21958	55407	55639.8	55618	55879
5000	21900	21927.8	21904	22059	55408	55631.5	55610.5	55905
10000	21900	21916	21903	21988	55314	55653.2	55651	55908
15000	21900	21929.6	21925	22017	55416	55708	55672.5	56010
25000	21900	21930.3	21917	21998	55367	55655.1	55671.5	55967
35000	21900	21920	21904	22019	55396	55636.6	55646.5	55905
50000	21900	21944.7	21933	22066	55371	55642.7	55625	55925
60000	21900	21919.3	21903	21984	55500	55671.3	55651	56011
75000	21900	21919.3	21903	21984	55500	55671.3	55651	56011

pende de $distInitFactor \in [0, 1]$, se realizó un esfuerzo por buscar un valor adecuado para este parámetro. De la misma forma, como el valor $noImprove$ influye en gran medida sobre el número de generaciones evolucionadas, también se hizo el esfuerzo de realizar pruebas para fijar dicho valor. Para $distInitFactor$ se consideró que podía tomar uno de seis valores diferentes equidistribuidos en el intervalo $[0, 1]$, mientras que para $noImprove$ se consideraron los siguientes valores: 200, 2,000, 5,000, 10,000, 15,000, 25,000, 35,000, 50,000, 60,000, 75,000. Hacer experimentos con todas las posibles combinaciones es muy costoso computacionalmente, por lo que se optó por optimizar fijando uno de los parámetros y reiterando hasta convergencia. Nótese que en este proceso se está optimizando cada parámetro de forma independiente al otro, a pesar de que existen dependencias entre ellos.

Sin embargo, debido al alto costo computacional asociado se decidió realizar el proceso de esta forma. Además, estos experimentos solo se realizaron con tres grafos, add20, whitaker3 y bcstk29 con $k = 16, 64$. Para elegir el parámetro en cada paso, se compararon todas las configuraciones ejecutadas, y se eligió aquella que maximizaba el número de veces que fue mejor estadísticamente menos el número de veces que fue peor estadísticamente.

Los parámetros iniciales se fijaron estableciendo $distInitFactor$ a 0,6 y $noImprove$ a 5,000. En la Tabla 1 se muestran los resultados con $noImprove$ fijo para los diferentes valores de $distInitFactor$. Con estos experimentos se concluyó que el mejor valor para $distInitFactor$ en este caso era 0,4. Análogamente en la Tabla 2 se varía ahora $noImprove$ y los test estadísticos arrojaron que el mejor valor pa-

Tabla 3: Resultados del MAHMBCDP para $k = 16, 64$ con $noImprove = 15000$

add20	$k = 16$				$k = 64$			
	Mejor	Media	Mediana	Peor	Mejor	Media	Mediana	Peor
DInit	2053	2065	2060.5	2094	2950	2984.2	2974.5	3026
0.0	2041	2050.8	2052	2059	2943	2949.8	2949.5	2958
0.2	2040	2043.8	2043	2053	2944	2949.5	2949	2958
0.4	2040	2041.2	2041	2043	2945	2951.7	2951.5	2959
0.6	2040	2041.5	2041	2044	2943	2950.9	2950	2961
0.8	2040	2041.9	2043	2045	2946	2952.9	2952.5	2959
1.0	2040	2041.9	2043	2045	2946	2952.9	2952.5	2959
whitaker3	$k = 16$				$k = 64$			
	Mejor	Media	Mediana	Peor	Mejor	Media	Mediana	Peor
DInit	1085	1090.3	1090	1094	2492	2501.8	2502	2514
0.0	1085	1087.2	1088	1088	2484	2491	2490.5	2499
0.2	1085	1086	1085	1088	2484	2488	2488	2494
0.4	1085	1086.2	1085.5	1088	2484	2487.8	2488	2491
0.6	1085	1086.6	1086.5	1088	2484	2493.7	2488	2527
0.8	1085	1087	1088	1088	2484	2487.2	2488	2491
1.0	1085	1087	1088	1088	2484	2487.2	2488	2491
bcsttk29	$k = 16$				$k = 64$			
	Mejor	Media	Mediana	Peor	Mejor	Media	Mediana	Peor
DInit	21973	22287.3	22265	22865	56052	56563.6	56499	57540
0.0	21900	21919.1	21901.5	22007	55582	55873.3	55861.5	56127
0.2	21900	21924.2	21904	21980	55434	55693.8	55699	55934
0.4	21900	21933.5	21932	22066	55405	55597.4	55585	55800
0.6	21900	21928.1	21907	21988	55343	55607.3	55608.5	55831
0.8	21900	21938.3	21936	22006	55424	55624.6	55627	56009
1.0	21900	21938.3	21936	22006	55424	55624.6	55627	56009

ra este parámetro en este caso era 15,000. Finalmente, en la última iteración al variar $distInitFactor$ este queda fijo otra vez en 0,4, por lo que el proceso finaliza. Los resultados se presentan en la Tabla 3. Con esto se terminó de decidir los parámetros que usamos para el resto de las ejecuciones. Así, $distInitFactor$ se fijó a 0,4 y $noImprove$ se fijó a 15,000.

Segundo experimento: análisis de diversidad

Una de las principales novedades de nuestro esquema es el control explícito de la diversidad. Esto se lleva a cabo en la fase de supervivencia con el operador de remplazamiento BNP. La Figura 2 muestra la evolución del fitness y la diversidad a lo largo de las ejecuciones y ofrecen una buena ilustración del tipo de casos que encontramos. El grafo que se está visualizando en este caso es el (*vibrobox*) para $k = 8, 64$. Notamos que en ambos casos se tiene un decremento paulatino de la diversidad, que es lo que se esperaba. Además, en las figuras relativas al valor de aptitud se indica también cuál es el mejor valor que se conocía antes de comenzar la tesis. Podemos ver que en el caso de $k = 8$ se supera dicho valor mientras que en el caso $k = 64$ no se alcanza. En el siguiente experimento se ofrecen resultados adicionales que comparan los resultados alcanzados en la tesis con los mejores conocidos antes de empezar la tesis.

Para validar el impacto que se tiene sobre la evolución de la diversidad y confirmar los beneficios del BNP se comparó nuestro operador con otros esquemas que han ganado gran popularidad, habiendo sido propuestos algunos de ellos como mecanismos para lidiar con la convergencia prematura. Específicamente, se obtuvieron resultados con MAHMBCDP, RTS, CD/RW, HGSADC y GEN_ELIT. La Figura 3 muestra la evolución de la diversidad y la aptitud de cada uno de los diferentes algoritmos al ser aplicados al grafo *whitaker3*. Para entender a fondo las razones de la superioridad del operador BNP hay que poner especial atención a este tipo de gráficas de diversidad. En las gráficas, la diversidad se toma como la media de la disimilitud entre cada par de individuos de la población. Es notorio que MAHMBCDP decrece la diversidad de manera paulatina durante toda la ejecución del algoritmo. Algo muy diferente ocurre en los otros esquemas que en algunos casos pierden la diversidad muy tempranamente y en otros mantienen alta diversidad en toda la ejecución. Al comenzar con una diversidad alta, en MAHMBCDP el espacio de soluciones es explorado de manera general. Esto también explica por qué, al principio de las ejecuciones, MAHMBCDP tiene una calidad en la aptitud inferior a los

demás esquemas, pues no intensifica en cada zona sino que sólo trata de detectar zonas promisorias. Una vez que se han identificado zonas promisorias para el algoritmo, con el paso del tiempo y la reducción de la diversidad, se comienza a intensificar a mayor profundidad. Esto se ve reflejado en la calidad de las soluciones al superar a cada uno de los otros esquemas en las últimas horas.

Tercer experimento: comparativa con el estado del arte

Para ilustrar la efectividad del algoritmo memético propuesto, se realizaron pruebas con grafos adicionales del TGPA y se compararon con los mejores resultados encontrados hasta la fecha. Se probó el MAHMBCDP con los valores $k = \{4, 8, 16, 32, 64\}$. Las Tablas 4, 5, 6 resumen los resultados obtenidos con MAHMBCDP. Adicionalmente al mejor, la media y el peor resultado obtenido, la mejor solución conocida encontrada por cualquier esquema previo a nuestra tesis se muestra en la columna TGPA. En la tablas se muestra en negrita los casos en los que MAHMBCDP alcanzó el mejor resultado conocido. Además, aquellos casos en los que se superó al mejor conocido se marcan con un asterisco. Cabe destacar que de los 115 casos probados, la mejor solución conocida fue alcanzada o superada en 80 casos de ellos, mientras que se pudo superar en 35 casos.

Conclusiones y trabajos futuros

El problema del particionado de grafos es un problema clásico y muy estudiado en la comunidad de optimización combinatoria. Se ha estudiado usando una gran cantidad de técnicas de las cuales se realizó una revisión en nuestra tesis. En este trabajo se propuso un algoritmo memético con gestión de diversidad para el GPP cuyas novedades son un operador de cruce diseñado especialmente para particiones y una estrategia de reemplazamiento que ayuda a mitigar la convergencia prematura y toma en cuenta el criterio de paro fijado por el usuario. Se consideró que esta clase de algoritmos podrían mejorar el estado del arte del problema puesto que se ha visto su efectividad en problemas como el Problema del Coloreado de Grafos y hasta la fecha no se habían probado con el GPP. Además, dado que se pretendía ejecutar la propuesta durante miles de generaciones, se cuidó particularmente el aspecto de la eficiencia en las subrutinas implementadas. En lo referente a contribuciones específicas relativas a los resultados obtenidos, cabe destacar que una vez elegidos los operadores de manera ade-

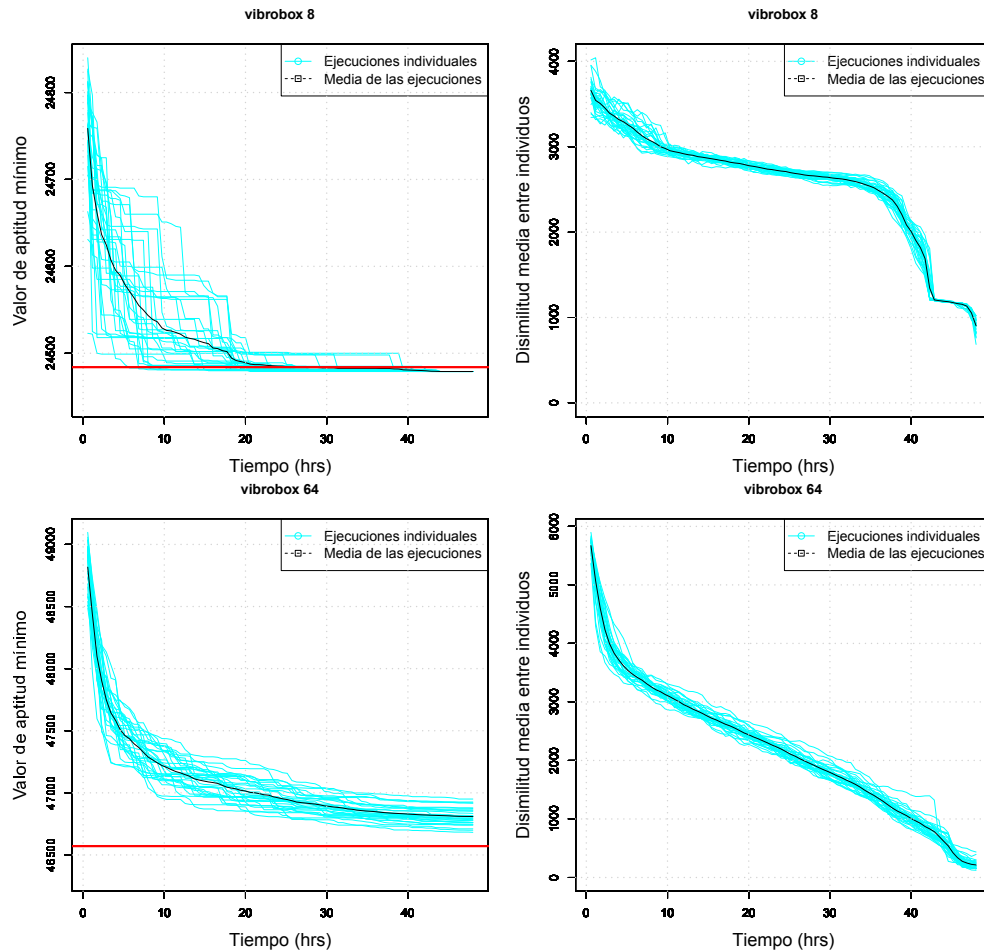


Figura 2: Evolución de la aptitud y la diversidad para instancias *vibrobox*, $k = 8, 64$ en el método MAHMBCDP

Tabla 4: Resultados del MAHMBCDP para $k = 4, 8$ con la parametrización final ($distInitFactor = 0,4$ y $noImprove = 15,000$)

Grafo	$k = 4$					$k = 8$				
	TGPA	Mejor	Media	Mediana	Peor	TGPA	Mejor	Media	Mediana	Peor
add20	1152	1151*	1151	1151	1151	1681	1678*	1680.5	1680.5	1684
data	382	382	382	382	382	668	668	668	668	668
3elt	201	201	201	201	201	345	345	345	345	345
uk	41	42	42.6	43	43	83	84	85.5	85	88
add32	34	34	34	34	34	67	67	67.6	68	68
bcsttk33	21717	21717	21717	21717	21717	34437	34437	34437	34437	34437
whitaker3	381	381	381	381	381	656	656	656	656	656
crack	366	366	366	366	366	679	679	679	679	679
wing_nodal	3575	3575	3575	3575	3575	5435	5435	5435.2	5435	5436
fe_4elt2	349	349	349	349	349	607	606*	606.6	606.5	608
vibrobox	18976	18976	18976.5	18976	18978	24484	24479*	24479	24479	24479
bcsttk29	8035	8035	8035	8035	8035	13975	13958*	13958	13958	13959
4elt	326	326	326	326	326	545	545	545	545	545
fe_sphere	768	768	768	768	768	1156	1156	1156	1156	1156
cti	954	954	954	954	954	1788	1788	1788	1788	1788
memplus	9448	10646	10922.5	10918.5	11007	11712	12409	12914.2	12999.5	13590
cs4	932	972	1575.1	1610	2545	1440	1919	2615.8	2664	2974
bcsttk30	16651	16651	16686.6	16651	16972	34846	34846	34854.3	34846	34893
bcsttk31	7351	7342*	7566	7531.5	7927	13283	13272*	13583.1	13498	14217
fe_pwt	705	705	705	705	705	1447	1447	1447	1447	1447
bcsttk32	9311	9311	9433.3	9430	10447	20009	19874*	20498.3	20232.5	22359
fe_body	599	619	1702.7	679	6150	1033	1061	1323	1153.5	2775
t60k	209	2224	5669.1	6876.5	8547	456	8114	10467	10888.5	11594

cuada se procedió a la validación experimental con las instancias del TGPA de Chris Walshaw y en las mismas se logró superar las mejores soluciones que había hasta la fecha en 35 instancias. Estos resulta-

dos fueron validados por Chris Walshaw y ahora nuestro algoritmo se encuentra listado en el TGPA como uno de los métodos a batir. Esto es un gran logro tomando en cuenta la cantidad de trabajo y

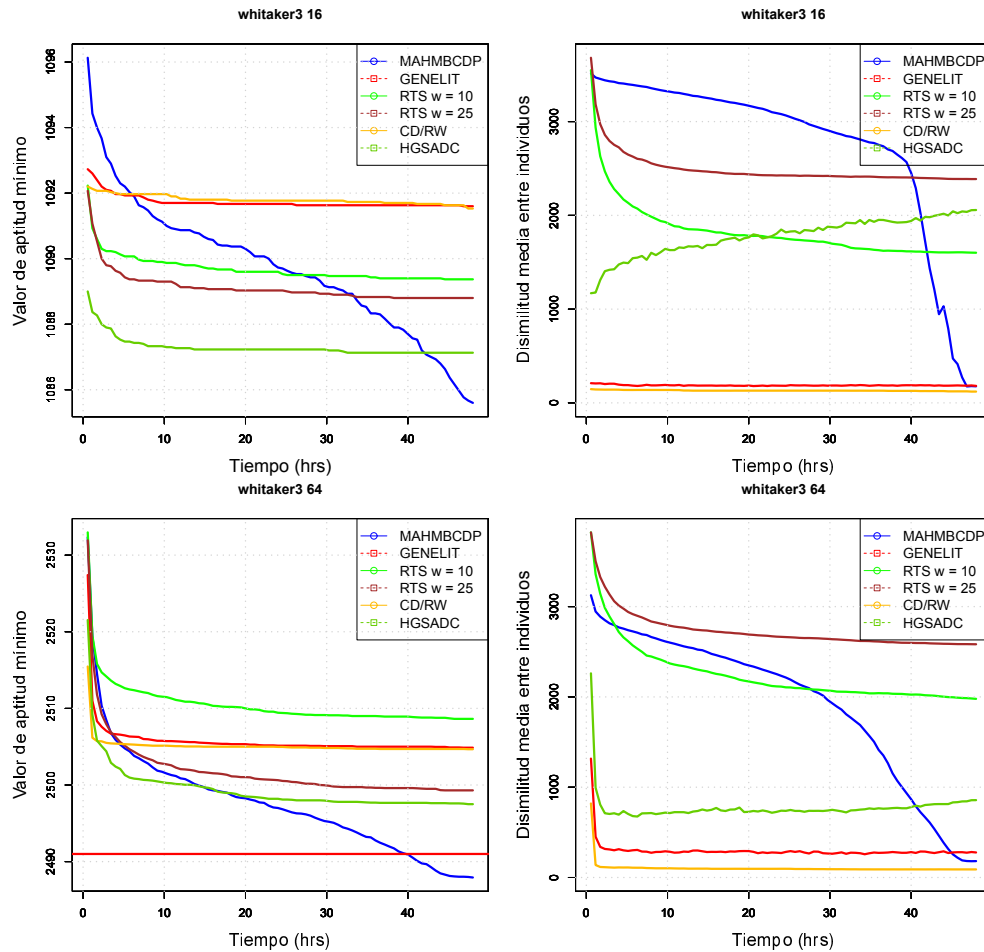


Figura 3: Evolución de la media de la aptitud mínima de treinta ejecuciones y la media de la diversidad para instancias *whitaker3*, $k = 16, 64$ y los diferentes operadores de supervivencia.

Tabla 5: Resultados del MAHMBCDP para $k = 16, 32$ con la parametrización final (*distInitFactor* = 0,4 y *noImprove* = 15,000)

Grafo	$k = 16$					$k = 32$				
	TGPA	Mejor	Media	Mediana	Peor	TGPA	Mejor	Media	Mediana	Peor
add20	2041	2040*	2043.7	2043	2052	2361	2357*	2359.8	2359	2365
data	1127	1127	1127	1127	1127	1799	1799	1799.1	1799	1800
3elt	573	573	573	573	573	960	960	960.2	960	961
uk	145	148	150.4	150	155	247	255	259.3	259.5	265
add32	118	119	125.9	126	132	213	213	223.5	224.5	234
bcsstk33	54680	54680	54680.5	54680	54685	77414	77410*	77414.5	77415	77417
whitaker3	1085	1085	1085.9	1085	1088	1668	1668	1669.2	1669	1672
crack	1088	1088	1088	1088	1088	1679	1678*	1680.4	1681	1682
wing_nodal	8334	8331*	8333.2	8332	8342	11768	11760*	11766.6	11765	11782
fe_delt2	1007	1008	1008.1	1008	1009	1614	1613*	1615	1615	1621
vibrobox	31850	32324	32759.7	32786.5	32898	39477	39397*	39634.1	39630	40102
bcsstk29	21905	21900*	21922.5	21910.5	21984	34737	34721*	34837.9	34802	35051
4elt	934	934	934.9	934	940	1551	1551	1552.1	1552	1553
fe_sphere	1714	1714	1714	1714	1714	2488	2488	2489.4	2489	2490
cti	2793	2792*	2792.8	2793	2794	4046	4043*	4047.3	4045	4066
memplus	12895	14125	14271.3	14266.5	14582	13953	15338	15362.6	15362.5	15385
cs4	2075	2718	3065	3044	3435	2928	3472	3776.7	3778	4145
bcsstk30	70408	70404*	70576.9	70443	71118	113336	113013*	113697.6	113642	114619
bcsstk31	23869	23605*	23910.1	23914	24262	37158	37363	37838.4	37887	38695
fe_pwt	2830	2835	2836.4	2836.5	2838	5575	5575	5583.4	5579	5605
bcsstk32	36250	36141*	36744.9	36698	37658	60038	59701*	60426.1	60389.5	61646
fe_body	1736	1823	2074.1	1962.5	3287	2846	2929	2996.1	2971	3472
t60k	813	9684	11317.1	11463	12379	1323	3693	8952.7	9315	9845

publicaciones que se han usado para lidiar con el GPP y en particular con las instancias en el TGPA. Como principal desventaja de nuestra propuesta podemos mencionar que incorpora operadores muy costosos. Esto provoca que al aumentar el número de nodos o el número

Tabla 6: Resultados del MAHMBCDP para $k = 64$ con la parametrización final ($distInitFactor = 0,4$ y $noImprove = 15,000$)

Grafo	$k = 64$				
	TGPA	Mejor	Media	Mediana	Peor
add20	2949	2944*	2949.4	2949	2955
data	2839	2835*	2838.7	2838.5	2847
3elt	1532	1532	1533.5	1533.5	1535
uk	408	416	420.9	420.5	427
add32	485	485	495	493	517
bcsstk33	107185	107179*	107268.1	107272	107353
whitaker3	2491	2484*	2487.7	2488	2492
crack	2535	2538	2541.2	2541	2547
wing_nodal	15775	15769*	15773.4	15772.5	15780
fe_4elt2	2478	2471*	2478.2	2478.5	2485
vibrobox	46571	46682	46808.9	46807.5	46950
bcsstk29	55241	55364	55644.7	55661.5	55856
4elt	2565	2562*	2565.8	2566	2569
fe_sphere	3543	3543	3545.2	3545.5	3547
cti	5629	5618*	5646.4	5648	5667
memplus	16223	17204	17295	17281	17420
cs4	4027	4332	4490.9	4481	4702
bcsstk30	171153	171088*	171801.7	171772	172624
bcsstk31	57402	57347*	57539.9	57500	57996
fe_pwt	8180	8275	8317.7	8313.5	8359
bcsstk32	90895	91109	92042	92111	92756
fe_body	4730	4798	4885.5	4895	4981
t60k	2077	2311	2856.8	2841.5	3450

de conjuntos en las particiones no se evolucionen demasiadas generaciones, algo muy importante en los esquemas basados en control de diversidad y en particular en el nuestro, por lo que en este aspecto hay que seguir mejorando el esquema.

Existen muchas líneas de investigación que aún pueden ser exploradas. En primer lugar, consideramos que una línea promisoría de trabajo es rediseñar el operador de mutación ya que algunas pruebas preliminares realizadas al final de la tesis mostraron que al incorporar una mutación de mayor intensidad, la calidad de las soluciones se puede mejorar significativamente en algunos tipos de grafos. También, para continuar con la validación de nuestro esquema, se podría comparar contra otros operadores de cruce más sofisticados. Otra línea podría consistir en rediseñar el operador de reemplazamiento utilizando control adaptativo, pues a pesar de que se logró que la diversidad decreciera poco a poco durante las ejecuciones, no se logró inducir un decremento lineal tan sólo por decrementar D de esta manera. Para solucionar esto se podría tratar de incorporar un mecanismo que controle cómo evolucionar el tamaño de las hiperesferas para mantener un decremento lineal más preciso. Una rama de investigación más tiene que ver con la escalabilidad. Con los experimentos realizados se pudo observar que conforme la k aumenta, el número de generaciones evolucionadas se reduce drásticamente. Presentar un algoritmo escalable tanto en las k como en el número de nodos es deseable y esto se puede abordar trabajando en las estructuras de datos usadas o cambiando algunas de las componentes más costosas. Finalmente, se ve razonable pensar que se puede paralelizar la propuesta para reducir el tiempo necesario para obtener soluciones de alta calidad como las encontradas y esto podría además ayudar a mejorar la escalabilidad.

Referencias

- [1] <http://chriswalshaw.co.uk/partition/> Last visited 03/03/17, 2000.
- [2] David A. Bader, Henning Meyerhenke, Peter Sanders, and Dorothea Wagner, editors. *10th DIMACS Implementation Challenge Workshop on Graph Partitioning and Graph Clustering Proceedings*, volume 588 of *Contemporary Mathematics*, Atlanta, GA, USA, 2013. American Mathematical Society.
- [3] Charles-Edmond Bichot. *Population-Based Metaheuristics, Fusion-Fission and Graph Partitioning Optimization*, pages 163–199. John Wiley & Sons, Inc., 2013.
- [4] Thang Nguyen Bui and Byung Ro Moon. Genetic algorithm and graph partitioning. *IEEE Transactions on Computers*, 45(7):841–855, July 1996. ISSN 0018-9340.
- [5] P. Chardaire, M. Barake, and G. P. McKeown. A probe-based heuristic for graph partitioning. *IEEE Transactions on Computers*, 56(12):1707–1720, Dec 2007.
- [6] A. E. Eiben and J. E. Smith. *Hybridisation with Other Techniques: Memetic Algorithms*, pages 173–188. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.
- [7] Philippe Galinier and Jin-Kao Hao. Hybrid evolutionary algorithms for graph coloring. *Journal of Combinatorial Optimization*, 3(4):379–397, 1999. ISSN 1573-2886.
- [8] Philippe Galinier, Zied Boujbel, and Michael CoutinhoFernandes. An efficient memetic algorithm for the graph partitioning problem. *Annals of Operations Research*, 191(1):1–22, 2011.
- [9] Philippe Galinier, Zied Boujbel, and Michael Coutinho Fernandes. An efficient memetic algorithm for the graph partitioning problem. *Annals of Operations Research*, 191(1):1–22, Nov 2011. ISSN 1572-9338.
- [10] Georges R. Harik. Finding multimodal solutions using restricted tournament selection. In *Proceedings of the 6th International Conference on Genetic Algorithms*, pages 24–31, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1-55860-370-0.

- [11] Matthias Hein and Simon Setzer. Beyond spectral clustering - tight relaxations of balanced graph cuts. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 2366–2374, USA, 2011. Curran Associates Inc. ISBN 978-1-61839-599-3.
- [12] A. Hertz and D. de Werra. Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351, Dec 1987. ISSN 1436-5057.
- [13] Minh Nghia Le, Yew-Soon Ong, Yaochu Jin, and Bernhard Sendhoff. Lamarckian memetic algorithms: local optimum and connectivity structure analysis. *Memetic Computing*, 1(3):175, Sep 2009. ISSN 1865-9292.
- [14] C. León, G. Miranda, and C. Segura. METCO: A Parallel Plugin-Based Framework for Multi-Objective Optimization. *International Journal on Artificial Intelligence Tools*, 18(4):569–588, 2009.
- [15] Fernando G. Lobo, Cláudio F. Lima, and Zbigniew Michalewicz, editors. *Parameter Setting in Evolutionary Algorithms*, volume 54 of *Studies in Computational Intelligence*. Springer, 2007.
- [16] Manuel Lozano, Francisco Herrera, and José Ramón Cano. Replacement strategies to preserve useful diversity in steady-state genetic algorithms. *Inf. Sci.*, 178(23):4421–4433, December 2008. ISSN 0020-0255.
- [17] H. Maini, K. Mehrotra, C. Mohan, and S. Ranka. Genetic algorithms for graph partitioning and incremental graph partitioning. In *Proceedings of Supercomputing '94*, pages 449–457, Nov 1994.
- [18] Ole J. Mengshoel and David E. Goldberg. The crowding approach to niching in genetic algorithms. *Evol. Comput.*, 16(3): 315–354, September 2008.
- [19] Peter Merz and Bernd Freisleben. Memetic algorithms and the fitness landscape of the graph bi-partitioning problem. In *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature - PPSN*, pages 765–774. Springer, 1998.
- [20] Hari Mohan Pandey, Ankit Chaudhary, and Deepti Mehrotra. A comparative review of approaches to prevent premature convergence in GA. *Applied Soft Computing*, 24:1047 – 1077, 2014.
- [21] Peter Sanders and Christian Schulz. *Think Locally, Act Globally: Highly Balanced Graph Partitioning*, pages 164–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [22] C. Segura, C. A. Coello Coello, E. Segredo, and A. H. Aguirre. A novel diversity-based replacement strategy for evolutionary algorithms. *IEEE Transactions on Cybernetics*, 46(12):3233–3246, Dec 2016.
- [23] C. Segura, S. I. V. Peña, S. B. Rionda, and A. H. Aguirre. The importance of diversity in the application of evolutionary algorithms to the sudoku problem. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 919–926, July 2016.
- [24] C. Segura, A. Hernandez, F. Luna, and E. Alba. Improving diversity in evolutionary algorithms: New best solutions for frequency assignment. *IEEE Transactions on Evolutionary Computation*, In Press(99), 2017.
- [25] Carlos Segura, Arturo Hernández Aguirre, Sergio Iván Valdez Peña, and Salvador Botello Rionda. *The Importance of Proper Diversity Management in Evolutionary Algorithms for Combinatorial Optimization*, pages 121–148. Springer International Publishing, 2017.
- [26] Matej Črepinšek, Shih-Hsi Liu, and Marjan Mernik. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Computing Surveys*, 45(3):35:1–35:33, July 2013.
- [27] Thibaut Vidal, Teodor Gabriel Crainic, Michel Gendreau, and Christian Prins. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & Operations Research*, 40(1): 475 – 489, 2013.

El Porismo de Poncelet

Parte I

Juan Salvador Garza Ledesma^{1*}

¹Facultad de Ciencias, UNAM

* Correo electrónico: juan.garza@ciencias.unam.mx

Resumen

En esta primer parte de una serie de artículos sobre el Porismo de Poncelet, se presentan algunas pruebas elementales de casos particulares del resultado y se propone el problema de hallar cuadriláteros bicéntricos racionales. El material está dirigido a estudiantes de geometría y cálculo elementales. Se construyó un sitio web de libre acceso:

<https://sites.google.com/view/juan-garza/nigromante/poncelet>
para ilustrar algunos ejemplos interesantes.

Palabras clave: Porismo de Poncelet; Problema de Fuss; Polígonos Bicéntricos.

Introducción

El *Porismo de Poncelet* o *Gran Teorema de Poncelet*, fue descubierto por el ingeniero y matemático francés Jean-Victor Poncelet durante sus años de encarcelamiento en la prisión de Saratoff (1813-1814). Poncelet estuvo bajo el mando de Napoleón en su campaña contra Rusia.

Para enunciar el resultado es conveniente introducir la siguiente definición (véase Figura 1):

Definición 1. Sean C, D dos cónicas no singulares en \mathbb{R}^2 . Un n -ágono de Poncelet respecto al par ordenado (C, D) es un polígono (de n lados/vértices, $n \in \mathbb{N}$, $n \geq 3$) cuyos vértices son puntos de C y cuyos lados son segmentos de rectas tangentes a D . De existir, decimos que (C, D) admite un n -ágono de Poncelet.

En su versión más sencilla, el Porismo de Poncelet dice lo siguiente (ver la Figura 2):

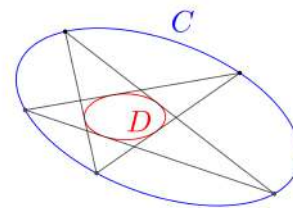


Figura 1: Un pentágono de Poncelet respecto a (C, D) .

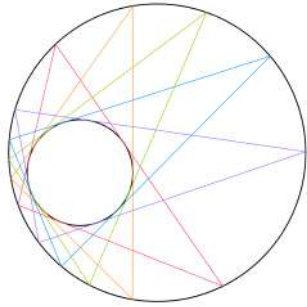


Figura 2: El Porismo de Poncelet para $n = 3$.

Teorema 1.1. *Sea (C_1, C_2) un par de circunferencias en \mathbb{R}^2 . Si existe un n -ágono de Poncelet respecto a (C_1, C_2) , entonces existe una infinidad. Más todavía, si C_2 está en el interior del disco definido por C_1 , entonces todo punto en C_1 es vértice de un n -ágono de Poncelet respecto a (C_1, C_2) ¹.*

El Porismo de Poncelet para circunferencias en \mathbb{R}^2

Comenzaré dando una prueba del Teorema 1.1 en el caso en que tenemos un par de circunferencias (C_1, C_2) en \mathbb{R}^2 , con C_2 estrictamente contenida en el interior del disco cuya frontera es C_1 . La geometría de la configuración que nos interesa estudiar está completamente determinada por los radios de las circunferencias y la distancia entre sus centros. Por lo tanto no hay pérdida de generalidad si suponemos que las ecuaciones de C_1 y C_2 son las siguientes:

$$C_1 : x^2 + y^2 = 1, \quad (1a)$$

$$C_2 : (x - d)^2 + y^2 = r^2, \quad (1b)$$

donde $d \geq 0$ y $r > 0$ son un par de números reales que satisfacen además $d + r < 1$ (ver Figura 3).

Los pasos de la prueba son los siguientes:

¹Dualmente, toda tangente a C_2 contiene un lado de un n -ágono de Poncelet respecto a (C_1, C_2) .

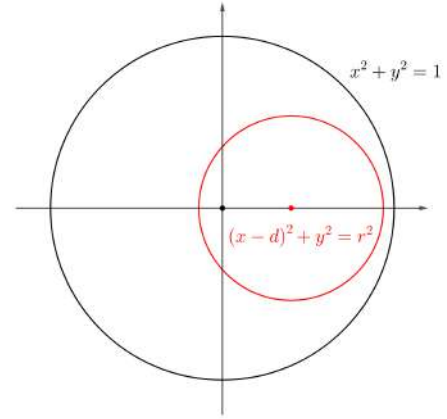


Figura 3: El par de circunferencias (C_1, C_2)

1. Se asociará una función $f : [0, \infty) \rightarrow \mathbb{R}$ al par de circunferencias (C_1, C_2) que medirá qué tan rápido recorremos la circunferencia C_1 en sentido positivo (esto es, en sentido contrario de las manecillas del reloj) al trazar tangentes a C_2 .
2. Se impondrán condiciones a la n -ésima iteración de f para garantizar la existencia de un n -ágono de Poncelet respecto a (C_1, C_2) .
3. Se mostrará que si un punto satisface las condiciones del inciso anterior, entonces lo mismo es válido para cualquier punto de C_1 .

Paso 1: La función f

Pensemos en los puntos $P \in C_1$ como imágenes del mapeo $\varphi : [0, \infty) \rightarrow \mathbb{R}^2$ dado por

$$\varphi(t) := (\cos t, \sin t), \quad (2)$$

es decir, estamos recorriendo la circunferencia C_1 en sentido opuesto a las manecillas del reloj comenzando en el punto $(0, 1) = \varphi(0)$. Por cualquier punto $P = \varphi(t)$ pasan dos cuerdas tangentes a C_2 (ver segmentos PQ_1 y PQ_2 en la Figura 4).

Sea Q_1 el punto más cercano a P recorriendo C_1 en el sentido antes mencionado. Definimos

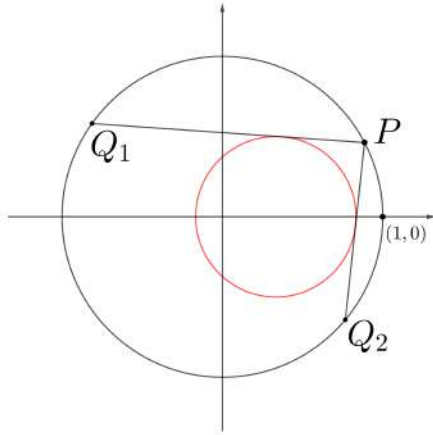


Figura 4: Construyendo la función f .

$f(t)$ como el número mínimo que satisface:

$$\varphi(f(t)) = Q_1. \quad (3)$$

Obsérvese que cualquier número de la forma $f(t) + 2m\pi$ con $m \in \mathbb{N}$ es mapeado bajo φ en Q_1 , de aquí el requerimiento de minimalidad para la definición de $f(t)$.

Paso 2: Construyendo n -ágonos de Poncelet

Por construcción, la existencia de un n -ágono ($n \geq 3$) de Poncelet respecto a (C_1, C_2) es equivalente a que exista un $t_0 \geq 0$ tal que

$$f^{on}(t_0) = t_0 + 2\pi m, \quad (4)$$

para algún $m \in \mathbb{N}$. Geométricamente, la ecuación (4) dice que después de trazar n tangentes manteniendo la orientación, volvemos al punto inicial $P = \varphi(t_0)$ (ver Figura 5).

Determinar explícitamente la regla de correspondencia que define a la función f es complicado. Como veremos enseguida, resulta más conveniente trabajar con su derivada $\frac{df}{dt}$. El hecho de que f sea derivable se sigue de que las circunferencias son curvas no singulares. Calcularemos

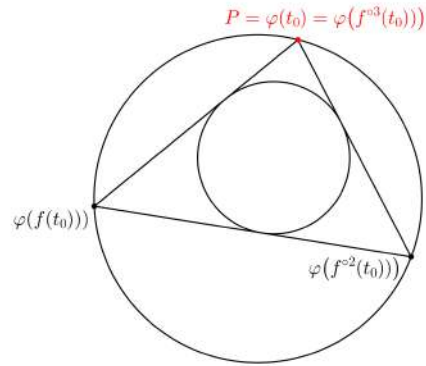


Figura 5: Un caso en que $f^{o3}(t_0) = t_0 + 2\pi$.

$\frac{df}{dt}$ directamente, tomando el límite

$$\lim_{h \rightarrow 0} \frac{f(t+h) - f(t)}{h}. \quad (5)$$

Sean $P_1 := \varphi(t)$, $P_2 := \varphi(t+h)$; $Q_1 := \varphi(f(t))$, $Q_2 := \varphi(f(t+h))$. Por construcción, los números $f(t+h) - f(t)$ y h son las medidas de los arcos de circunferencia comprendidos entre los puntos Q_1, Q_2 y P_1, P_2 respectivamente. Denotaré tales medidas con $\overline{Q_1Q_2}$ y $\overline{P_1P_2}$.

Sea $O := \overline{P_1Q_1} \cap \overline{P_2Q_2}$. Para calcular el límite (5), observemos que los triángulos ΔP_1P_2O y ΔQ_2Q_1O son semejantes (ver Figura 6).

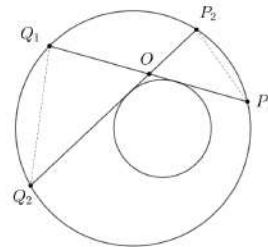


Figura 6: $\Delta P_1P_2O \sim \Delta Q_2Q_1O$.

Por lo tanto

$$\frac{|Q_1Q_2|}{|P_1P_2|} = \frac{|Q_1O|}{|OP_2|}, \quad (6)$$

donde $|\cdot, \cdot|$ denota la longitud del segmento correspondiente. Cuando h tiende a 0, o equivalentemente P_2 tiende a P_1 , el cociente $\frac{Q_1Q_2}{P_1P_2}$ tiende a (6), ambas cuerdas tangentes se sobreponen y el punto O tiende al punto de tangencia con C_2 . Por lo tanto si tomamos una cuerda tangente PQ con punto de tangencia T , tenemos $\frac{df}{dt} = \frac{|TQ|}{|PT|}$ (ver Figura 7).

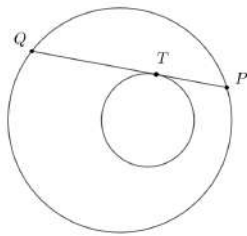


Figura 7: El límite $h \rightarrow 0$.

Las longitudes $|PT|$ y $|TQ|$ pueden calcularse en función de los radios de la circunferencias, 1 y r y la distancia entre sus centros, d de la siguiente manera (ver Figura 8):

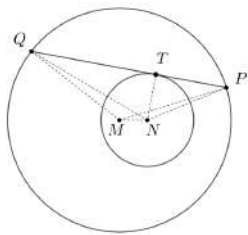


Figura 8: Construcciones auxiliares.

Denotaré con M y N los centros de C_1 y C_2 respectivamente. Por construcción $|MQ| = 1$, $|MN| = d$ y si $P = \varphi(t)$, el ángulo $\angle QMN$ mide $f(t)$. Por la ley de cosenos tenemos:

$$|NQ|^2 = 1 + d^2 - 2d \cos(f(t)) \quad (7)$$

y por el Teorema de Pitágoras aplicado al triángulo ΔTQN tenemos:

$$|TQ| = \sqrt{1 + d^2 - r^2 - 2d \cos(f(t))}. \quad (8)$$

Análogamente uno calcula:

$$|PT| = \sqrt{1 + d^2 - r^2 - 2d \cos(t)}. \quad (9)$$

Por lo tanto:

$$\frac{df}{dt} = \frac{\sqrt{1 + d^2 - r^2 - 2d \cos(f(t))}}{\sqrt{1 + d^2 - r^2 - 2d \cos(t)}}, \quad (10)$$

o bien:

$$\begin{aligned} \frac{1}{\sqrt{1 + d^2 - r^2 - 2d \cos(f(t))}} \frac{df}{dt} &= \\ &= \frac{1}{\sqrt{1 + d^2 - r^2 - 2d \cos(t)}}. \end{aligned} \quad (11)$$

Consideremos la función

$$g(x) := \frac{1}{\sqrt{1 + d^2 - r^2 - 2d \cos(x)}}. \quad (12)$$

Por construcción g es estrictamente positiva y continua. Por lo tanto existe una función inyectiva G tal que

$$\frac{dG}{dx} = g(x). \quad (13)$$

Afirmo que existe una constante ω tal que para todo $t \geq 0$:

$$G(f(t)) - G(t) = \omega. \quad (14)$$

En efecto, por (13) y la regla de la cadena:

$$\frac{d}{dt} (G(f(t))) = g(f(t)) \frac{df}{dt} \quad (15)$$

combinando esta última ecuación con (13) y (11) tenemos:

$$\frac{d}{dt} (G(f(t)) - G(t)) = 0. \quad (16)$$

De aquí se sigue (14).

Aplicando (14) para $t = f^{oj}(t)$ con j desde 1 hasta $n - 1$ y sumando las n ecuaciones resultantes, tenemos:

$$G(f^{on}(t)) - G(t) = n\omega. \quad (17)$$

Por lo tanto la existencia de un n -ágono de Poncelet respecto a (C_1, C_2) equivale a la existencia de un $t_0 \geq 0$ y un natural m tales que:

$$G(t_0 + 2\pi m) - G(t_0) = n\omega. \quad (18)$$

Paso 3: Independencia respecto al vértice inicial

El paso final de la prueba consiste en observar que la condición impuesta por la ecuación (18) no depende de t_0 . Esto es consecuencia de que G es una antiderivada de una función 2π -periódica. La condición impuesta por (18) para la existencia de un n -ágono de Poncelet consiste precisamente en que la integral en un intervalo formado por m periodos concatenados de una función periódica sea igual a $n\omega$. Es bien sabido que el valor de dicha integral no depende de la elección de los m periodos concatenados. En otras palabras, si (18) se cumple para t_0 , entonces se cumple para todo $t \geq 0$. Esto demuestra el resultado de Poncelet.

¿Por qué *porismo*?

El resultado de Poncelet es llamado *porismo* porque no resuelve el problema de hallar polígonos de Poncelet respecto a un par de circunferencias. Poncelet dedujo la existencia de una infinidad de polígonos interinscritos siempre que exista al menos uno.

El Problema de Fuss para triángulos

El problema de Fuss consiste en dar una solución explícita al Porismo de Poncelet para circunferencias en un plano. Como veremos a continuación, tal solución está dada por una ecuación

algebraica en los parámetros que determinan la posición relativa de las circunferencias: sus radios y la distancia entre sus centros. La complejidad de estas ecuaciones crece considerablemente junto con el número n de lados de los polígonos buscados. En esta primera parte, estudiaremos los casos $n = 3, 4$ de forma elemental.

Proposición 1. Consideremos un par de circunferencias coplanares C_1, C_2 de radios R y r respectivamente tales que la distancia entre sus centros d satisface $d < R - r$ (en otras palabras, C_2 está contenida en el interior del disco cuya frontera es C_1). Supongamos que existe un triángulo ΔABC de Poncelet respecto a (C_1, C_2) . Entonces:

$$R^2 - d^2 = 2Rr. \quad (19)$$

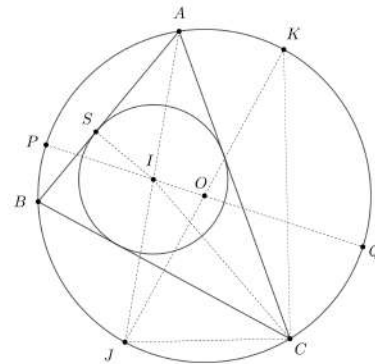


Figura 9: Probando la Proposición 1.

Prueba: En la Figura 9, O e I son los centros de C_1 y C_2 respectivamente. Prolongamos \overline{AI} hasta intersectar de nuevo C_1 en J . \overline{PQ} y \overline{JK} son los diámetros de C_1 que contienen a los segmentos \overline{IO} y \overline{JO} respectivamente. S es el punto de tangencia de C_2 con \overline{AB} .

Es claro que los triángulos ΔAPI y ΔIJQ son semejantes. Luego $|AI| \cdot |IJ| = |PI| \cdot |IQ|$. Es decir:

$$|AI| \cdot |IJ| = (R - d)(R + d). \quad (20)$$

Falta probar que $|AI| \cdot |IJ| = 2Rr$. Para ello observamos que los triángulos ΔASI y ΔKJC son

semejantes: $\angle ASI$ es recto porque su vértice es la intersección de un radio y una tangente de C_2 . $\angle JCK$ es recto por subtender un diámetro de C_1 . Por otro lado $\angle IAS = \angle CAJ$ porque \overline{AJ} es bisectriz de $\angle CAB$ y $\angle CAJ = \angle CKJ$ por subtender la misma cuerda. De aquí se sigue que:

$$|AI| \cdot |JC| = |KJ| \cdot |SI| = 2Rr. \quad (21)$$

Finalmente $|JC| = |IJ|$, pues $\angle CIJ = \angle JCI$ (ambos ángulos son iguales a $\frac{1}{2}(\angle CAB + \angle BCA)$). De aquí y (20) obtenemos el resultado. *Q.E.D.*

El recíproco de la Proposición 1 también es válido:

Proposición 2. Sean C_1 y C_2 dos circunferencias coplanares de radios R y r respectivamente. Sea d la distancia entre sus centros. Supongamos que $R^2 - d^2 = 2Rr$. Entonces existe un triángulo de Poncelet respecto a (C_1, C_2) (y por el Porismo de Poncelet, una infinidad).

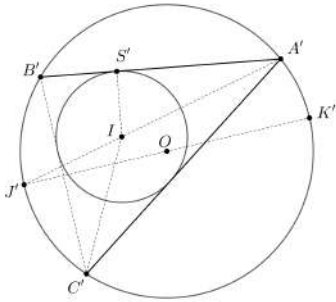


Figura 10: Recuperando un triángulo de Poncelet.

Prueba: Tomemos un punto A' en C_1 . Sean $\overline{A'B'}$ y $\overline{A'C'}$ las cuerdas tangentes a C_2 . Prolonguemos $\overline{A'I}$ hasta intersectar a C_1 de nuevo en J' . Prolonguemos también $\overline{J'O}$ hasta obtener el diámetro $\overline{J'K'}$. Sea S' el punto de tangencia de $\overline{A'B'}$ con C_2 . Al igual que en la prueba de la Proposición 1, tenemos

$$|A'I| \cdot |IJ'| = R^2 - d^2 \quad (22)$$

pero $R^2 - d^2 = 2Rr$. Por lo tanto

$$|A'I| \cdot |IJ'| = 2Rr. \quad (23)$$

Como los triángulos $\Delta A'S'I$ y $\Delta K'J'C'$ son semejantes, tenemos

$$|A'I| \cdot |J'C'| = 2Rr. \quad (24)$$

De (23) y (24) se sigue que $|IJ'| = |J'C'|$. Por lo tanto

$$\angle C'IJ' = \angle J'C'I. \quad (25)$$

Ahora $\angle C'IJ' = \angle IC'A' + \angle C'A'I$ y $\angle J'C'I = \angle B'C'I + \angle J'C'B'$. Pero $\angle J'C'B' = \angle J'A'B'$ por subtender la misma cuerda y $\angle J'A'B' = \angle C'A'I$ por ser $\overline{A'I}$ bisectriz del ángulo en A' . Por lo tanto

$$\angle IC'A' = \angle B'C'I. \quad (26)$$

De aquí se sigue que I es incentro de $\Delta A'B'C'$. En particular $\overline{B'C'}$ es tangente a C_2 . *Q.E.D.*

Cuadriláteros bicéntricos

Recordemos que un cuadrilátero es llamado **bicéntrico** si es de Poncelet respecto a un par de circunferencias (C_1, C_2) . A continuación encontraremos una condición necesaria para la existencia de cuadriláteros bicéntricos análoga a la obtenida en la Proposición 1.

Proposición 3. Sean C_1 y C_2 dos circunferencias coplanares de radios R y r respectivamente tales que la distancia entre sus centros d satisfice $d < R - r$. Supongamos que existe un cuadrilátero $ABCD$ de Poncelet respecto a (C_1, C_2) . Entonces:

$$(R^2 - d^2)^2 = 2r^2(R^2 + d^2). \quad (27)$$

Prueba: Sean O, I los centros de C_1 y C_2 respectivamente. Prolonguemos \overline{AI} y \overline{CI} hasta intersectar a C_1 en los puntos L y M . Sean J y K los puntos de tangencia de C_2 con los lados \overline{AB} y \overline{BC} respectivamente (ver Figura 11).

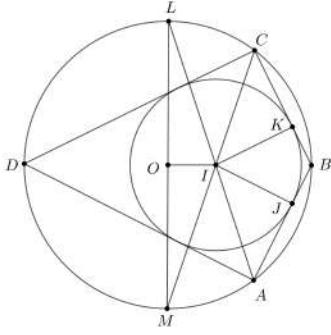


Figura 11: Problema de Fuss para cuadriláteros.

Como $ABCD$ es cíclico, $\angle DAB + \angle BCD = \pi$. De aquí se sigue que

$$\angle IAB + \angle BCI = \frac{\pi}{2} \quad (28)$$

y como $|IK| = |IJ| = r$, existe un triángulo rectángulo con catetos de longitudes $|CI|$ y $|IA|$ e hipotenusa de longitud $|AJ| + |KC|$. El doble del área de este triángulo puede calcularse de dos maneras obvias, dando lugar a la siguiente igualdad:

$$|CI| \cdot |IA| = r(|AJ| + |KC|). \quad (29)$$

Por el Teorema de Pitágoras:

$$(|AJ| + |KC|)^2 = |CI|^2 + |IA|^2. \quad (30)$$

De (29) y (30) se sigue que:

$$|CI|^2 \cdot |IA|^2 = r^2(|CI|^2 + |IA|^2), \quad (31)$$

o equivalentemente:

$$\frac{1}{|IA|^2} + \frac{1}{|CI|^2} = \frac{1}{r^2}. \quad (32)$$

Observemos ahora que los puntos L , O y M son colineales, porque $\angle MOD = 2\angle MCD$, $\angle DOL = 2\angle DAL$ y $\angle MCD + \angle DAL = \frac{\pi}{2}$. De aquí se sigue que \overline{OI} es mediana del triángulo $\triangle ILM$. Por lo tanto:

$$|LI|^2 + |IM|^2 = 2|OI|^2 + \frac{|LM|^2}{2} = 2(R^2 + d^2). \quad (33)$$

\overline{AL} , \overline{CM} y el diámetro de C_1 que contiene a \overline{OI} son cuerdas con un punto en común. Entonces:

$$|LI| \cdot |IA| = |CI| \cdot |IM| = R^2 - d^2. \quad (34)$$

Usando (33) y (34) tenemos:

$$\frac{1}{|IA|^2} + \frac{1}{|CI|^2} = \frac{2(R^2 + d^2)}{(R^2 - d^2)^2}. \quad (35)$$

De (32) y (35) concluimos que:

$$(R^2 - d^2)^2 = 2r^2(R^2 + d^2). \quad (36)$$

Q.E.D.

El lector interesado puede probar como ejercicio que la ecuación (36) es también una condición suficiente para la existencia de cuadriláteros bicéntricos.

Proyecto numérico

En esta sección se propone un proyecto con dos objetivos: que el lector interesado reafirme el material expuesto e introducir algunas de las ideas que se utilizarán en las siguientes partes de este trabajo.

Un problema numérico interesante es encontrar configuraciones de circunferencias para las cuales se cumpla el Porismo de Poncelet y tales que los parámetros que las definen (los radios y la distancia entre sus centros) sean números racionales. El problema no es trivial incluso para un número de lados relativamente chico de los polígonos. Por ejemplo, si uno considera cuadrados regulares, digamos de lado de longitud 2, tendremos que el radio del circuncírculo R será $R = \sqrt{2}$.

Una solución en el caso de cuadriláteros, es la siguiente:

1. Considere un triángulo rectángulo $\triangle ABC$ con catetos de longitudes $\ell_1 := |AB|$ y ℓ_2

$:= |BC|$. Supongamos $\ell_1 < \ell_2$. Sea D la reflexión de B respecto a la hipotenusa \overline{CA} . Es fácil probar que $ABCD$ es bicéntrico. Uno calcula los radios de las circunferencias circunscrita e inscrita, R y r respectivamente, así como la distancia d entre sus centros en términos de ℓ_1 y ℓ_2 :

$$\begin{aligned} R &= \frac{1}{2}\sqrt{\ell_1^2 + \ell_2^2}, & r &= \frac{\ell_1\ell_2}{\ell_1 + \ell_2}, \\ d &= R - \frac{\sqrt{\ell_1^2 + \ell_2^2}}{1 + \frac{\ell_2}{\ell_1}} \end{aligned} \quad (37)$$

2. Del punto anterior, se sigue que si $\ell_1, \ell_2 \in \mathbb{N}$, entonces $R, r, d \in \mathbb{Q}$. Por lo tanto, basta resolver el problema numérico clásico de encontrar todas las ternas pitagóricas enteras.
3. Muestra que, realizando los pasos anteriores, uno obtiene una infinidad de configuraciones esencialmente distintas de pares de circunferencias que cumplen el Porismo de Poncelet para $n = 4$ (observa que podría pasar que todos los parámetros obtenidos de esta manera fueran proporcionales entre sí respectivamente, de forma que uno obtuviera una sola configuración y copias homotéticas de la misma).

En la Parte II de esta serie, explicaré la relación entre el proyecto anterior y un famoso teorema de Faltings, (anteriormente conjetura de Mordell).

Agradecimientos

Le agradezco al Profesor Pablo Barrera Sánchez por haberme dado a conocer el Porismo de Poncelet durante mi tiempo como estudiante de licenciatura en la UNAM y por sus comentarios y sugerencias para mejorar este trabajo. Muchas gracias también al grupo de trabajo del Laboratorio de Cómputo Científico del Departamento de Matemáticas de la UNAM por brindarme todas las facilidades para la elaboración de este material.

Referencias

- [1] Garza Ledesma, J. S., *Curvas de Poncelet*. Tesis de Licenciatura, UNAM. 2011.
- [2] Garza Ledesma, J. S., Sitio web: <https://sites.google.com/view/juan-garza/nigromante/poncelet>
- [3] IGI, GeoGebra. Software interactivo disponible en: <http://www.geogebra.org>

El Porismo de Poncelet

Parte II

Juan Salvador Garza Ledesma^{1*}

¹Facultad de Ciencias, UNAM

* Correo electrónico: juan.garza@ciencias.unam.mx

Resumen

En esta segunda parte de una serie de artículos sobre el Porismo de Poncelet se discute una prueba general del resultado utilizando herramientas de geometría algebraica. Se construyó un sitio web de libre acceso:

<https://sites.google.com/view/juan-garza/nigromante/poncelet>

para complementar el material. Se busca mostrar cómo las herramientas computacionales pueden motivar el estudio de muchas áreas de investigación de gran actualidad en matemáticas que a primera vista podrían aparentar no tener relación entre sí.

Palabras clave: Porismo de Poncelet; Espacio Proyectivo; Curvas Racionales; Funciones y Curvas Elípticas.

Hacia una versión general del Porismo de Poncelet

La experimentación computacional permite conjeturar que los resultados discutidos en [2] son consecuencia de un teorema más general. Utilizando programas como **GeoGebra** (disponible en [7]) podemos construir configuraciones que involucren cónicas en \mathbb{R}^2 que no sean circunferencias y para las cuales sin embargo, se cumpla el Porismo de Poncelet. La Figura 1 ilustra un ejemplo, con pentágonos no convexos, una hipérbola y una elipse. Este y más ejemplos interactivos pueden verse en el sitio web:

<https://sites.google.com/view/juan-garza/nigromante/poncelet>

Ejemplos como los mencionados, sugieren que el Porismo de Poncelet depende de propiedades de las cónicas que difícilmente son intuitivas li-

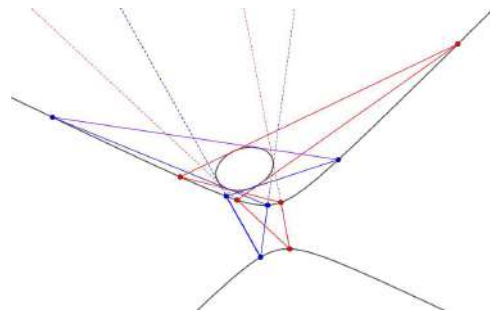


Figura 1: Una elipse y una hipérbola de Poncelet.

mitándonos a estudiarlas en \mathbb{R}^2 .

A lo largo de la historia, varios matemáticos descubrieron que las cónicas no singulares en \mathbb{R}^2 son distintas visualizaciones de un mismo objeto: una **cónica proyectiva** no singular. También se descubrieron razones técnicas por las que era mucho más sencillo estudiar objetos definidos sobre \mathbb{C} y no sobre \mathbb{R} . Una vez entendido este par de puntos

fue posible dar una prueba de una versión general del Porismo de Poncelet, que nos lleva a estudiar muchos temas de suma importancia en geometría y que discutiremos en este trabajo.

En resumen, los pasos de la prueba general que mencionamos son los siguientes:

1. Se construye el espacio proyectivo complejo \mathbb{P}^2 y se muestra que las figuras (cónicas, polígonos, etc.) que hemos estudiado hasta ahora en \mathbb{R}^2 , son visualizaciones de ciertos subconjuntos de puntos de objetos en \mathbb{P}^2 que resultan técnicamente mucho más sencillos de manipular.
2. Dadas dos cónicas $C, D \subset \mathbb{P}^2$, se prueba que los pares punto-recta (p, ℓ) con $p \in C$ y ℓ tangente a D con $p \in \ell$ (esto es, los objetos con los que construimos los posibles polígonos de Poncelet respecto (C, D)), están en correspondencia uno-a-uno con los puntos de una **curva elíptica** E . Las curvas elípticas son **variedades abelianas**. Esto quiere decir que dada una curva elíptica E , existe una operación binaria $+: E \times E \rightarrow E$ de modo que $(E, +)$ es un grupo abeliano.
3. Sean (p_i, ℓ_i) , $i = 1, 2$ dos pares tales que $p_i \in C$, ℓ_i es tangente a D , $p_1 \in \ell_1$ y $p_2 = \ell_1 \cap \ell_2$. Sean P_i , $i = 1, 2$ los puntos en E que corresponden a los pares (p_i, ℓ_i) respectivamente. Se prueba que la función definida mediante $(p_1, \ell_1) \mapsto (p_2, \ell_2)$ corresponde a la función algebraica $f(P_1) = P_1 + P_0$ para cierto $P_0 \in E$ fijo, donde $+$ es la operación binaria que da estructura de grupo a E . En otras palabras, existe $P_0 \in E$ tal que $P_2 = P_1 + P_0$.
4. Se observa que la existencia de un n -ágono de Poncelet equivale a que exista un P_1 tal que $f^{on}(P_1) = P_1$, esto es, a que f^{on} tenga un punto fijo. Pero esto equivale a su vez a:

$$P_1 = P_1 + nP_0, \quad (1)$$

lo cual implica que nP_0 es el elemento neutro de $(E, +)$. Es decir, $f^{on} = id_E$. De aquí

uno concluye que la existencia de un n -ágono de Poncelet implica la existencia de una infinidad.

En las siguientes secciones, discutiremos algunos aspectos geométricos y computacionales de los pasos descritos anteriormente. Al final se incluye una serie de ejercicios que ayudarán al lector interesado a profundizar en las ideas que expon-dremos.

El espacio proyectivo

Notación: Dado un espacio vectorial V , escribiré V^\times para denotar a $V \setminus \{0\}$.

La recta proyectiva $\mathbb{P}^1_{\mathbb{C}}$ puede definirse de las siguientes maneras equivalentes:

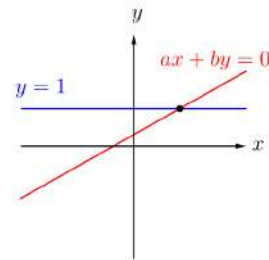


Figura 2: Construyendo la recta proyectiva.

1. La recta afín compactificada con un punto al infinito, esto es: $\mathbb{P}^1_{\mathbb{C}} := \mathbb{C} \cup \{\infty\}$.
2. El conjunto de subespacios vectoriales de dimensión 1 de un espacio de dimensión 2. Geométricamente, el conjunto de rectas en el plano que pasan por su origen.
3. Las órbitas de la acción de \mathbb{C}^\times en $(\mathbb{C}^2)^\times$ dada por:

$$\lambda \cdot (a, b) := (\lambda a, \lambda b).$$

La motivación geométrica de estas 3 definiciones viene de la construcción para el caso real (ver la Figura 2). Dentro del plano \mathbb{R}^2 con coordenadas (x, y) , tomemos una copia de la recta afín \mathbb{R} , digamos la recta con ecuación $y = 1$. Los subespacios de \mathbb{R}^2 de dimensión 1 son las rectas que pasan por el origen, que son los ceros de polinomios de la forma $ax + by$ para algún par $(a, b) \in (\mathbb{R}^2)^\times$. Estas rectas están determinadas precisamente por las órbitas de la acción del inciso 3 anterior; el par (a, b) define la misma recta que el par (c, d) si y sólo si existe un $\lambda \in \mathbb{R}^\times$ tal que $(a, b) = \lambda \cdot (c, d)$. De aquí se sigue que $\mathbb{P}_{\mathbb{R}}^1$ contiene una copia de la recta afín, pues cada una de las rectas $ax + by = 0$, a excepción de $y = 0$, interseca a $y = 1$ precisamente en un punto y viceversa, por cada punto de $y = 1$ pasa exactamente una recta que contiene al origen. La recta $y = 0$, o bien la órbita de los pares de la forma $(a, 0)$ con $a \in \mathbb{R}^\times$, es asociada al punto al infinito ∞ . Si tomamos un punto en la recta afín $y = 1$ y nos alejamos indefinidamente hacia alguna de las direcciones, por ejemplo hacia la derecha, la recta correspondiente que pasa por el origen se aproximará tanto como se desee a la recta horizontal $y = 0$. Si bien alcanzar el hipotético punto en la recta $y = 1$ es imposible, desde el punto de vista de las rectas por el origen simplemente estamos rotando en el sentido de las manecillas del reloj y no solo es posible hacerlo hasta alcanzar la horizontal, sino que uno puede visualizar fácilmente cómo dicha rotación puede continuar y la intersección de la recta correspondiente con $y = 1$ *vuelve súbitamente desde el infinito* pero esta vez desde el lado izquierdo. Es natural por lo tanto, anexar un solo punto al infinito para obtener la recta $\mathbb{P}_{\mathbb{R}}^{1,1}$.

El lector puede pensar de manera análoga en las construcciones de los espacios proyectivos \mathbb{P}^n para $n = 2, 3$, etc. Como es usual, denotaré con $(a_0 : a_1 : \dots : a_n)$ al punto de \mathbb{P}^n correspondiente a la órbita de $(a_0, \dots, a_n) \in (\mathbb{C}^n)^\times$ bajo la

¹Si escribo un espacio proyectivo sin un subíndice que indique el campo sobre el que se realiza la construcción, supondré que dicho campo es \mathbb{C} .

\mathbb{C}^\times -acción dada por

$$\lambda(a_0, \dots, a_n) := (\lambda a_0, \dots, \lambda a_n).$$

Los números de la $(n + 1)$ -ada $(a_0 : a_1 : \dots : a_n)$ o bien cualesquiera en la misma órbita son llamados **coordenadas homogéneas** del punto correspondiente. Es fácil ver que un objeto geométrico en \mathbb{P}^n puede definirse en términos de coordenadas homogéneas como los ceros de un conjunto de polinomios si y sólo si dichos polinomios son homogéneos.

Utilizando coordenadas homogéneas uno puede pensar a \mathbb{P}^n como una variedad topológica, tomando como cartas, copias del espacio afín de la dimensión correspondiente. Por ejemplo, en \mathbb{P}^2 con coordenadas $(x : y : z)$ podemos considerar la carta dada por

$$\{(x : y : z) \in \mathbb{P}^2 \mid z \neq 0\}. \quad (2)$$

Los puntos de la forma (2) son, por definición, exactamente los puntos con coordenadas $(a : b : 1)$ con $a, b \in \mathbb{C}$. Evidentemente estos puntos forman una copia del plano afín \mathbb{C}^2 . Podemos definir cartas similares utilizando la primer o segunda coordenadas, obteniendo una cubierta del plano proyectivo con *parches afines*.

Pensaré en las cónicas, rectas, etc. usuales como restricciones a puntos reales de imágenes en parches afines de un objeto geométrico en un espacio proyectivo. Por ejemplo, la circunferencia

$$\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 = 1\} \quad (3)$$

es la restricción a puntos reales de la cónica en \mathbb{P}^2 dada por:

$$\{(x : y : z) \in \mathbb{P}^2 \mid x^2 + y^2 = z^2\} \quad (4)$$

en el parche afín definido por (2).

En este contexto, una cónica no singular es un objeto muy sencillo; se trata simplemente de una copia de \mathbb{P}^1 encajada en \mathbb{P}^2 mediante:

$$(s : t) \mapsto (s^2 : st : t^2). \quad (5)$$

El mapeo $\mathbb{P}^1 \rightarrow \mathbb{P}^2$ definido por (5) es llamado **segundo encaje de Veronese** de la recta proyectiva. Si tomamos coordenadas $(x : y : z)$ en el plano \mathbb{P}^2 , los puntos de la forma $(s^2 : st : t^2)$ son ceros del siguiente polinomio homogéneo de grado 2:

$$xz - y^2. \quad (6)$$

Uno puede verificar que toda cónica no singular en \mathbb{R}^2 es la restricción real de la imagen en un parche afín de una cónica en \mathbb{P}^2 cuya ecuación puede llevarse a la forma (6) mediante un automorfismo del plano proyectivo, esto es, mediante una función definida en coordenadas multiplicando por una matriz no singular $A \in \text{Mat}_{3 \times 3}(\mathbb{C})$. Por ejemplo, la cónica (4) puede llevarse a la forma (6) utilizando la matriz

$$A = \begin{pmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & i & 0 \\ -\frac{1}{2} & 0 & \frac{1}{2} \end{pmatrix},$$

esto es, mediante la transformación

$$(x : y : z) \mapsto \left(\frac{x+z}{2} : iy : \frac{z-x}{2} \right).$$

El plano dual

Una recta en \mathbb{P}^2 es el conjunto de puntos cuyas coordenadas homogéneas son los ceros de un polinomio homogéneo de grado 1:

$$\{(x : y : z) \in \mathbb{P}^2 \mid ax + by + cz = 0\} \quad (7)$$

donde $(a, b, c) \in (\mathbb{C}^3)^\times$. Es claro que las rectas en \mathbb{R}^2 son restricciones reales en parches afines de una recta de la forma (7). Más aún, también de (7) se sigue que dos tripletas (a, b, c) y (a', b', c') determinan una misma recta si y sólo si existe un $\lambda \in \mathbb{C}^\times$ tal que $(a, b, c) = \lambda \cdot (a', b', c')$. Es decir, el conjunto de rectas en el plano proyectivo tiene estructura a la vez de plano proyectivo, mismo que suele denotarse con \mathbb{P}^{2^\vee} .

Sea C una cónica en \mathbb{P}^2 :

$$C := \{(x : y : z) \in \mathbb{P}^2 \mid xz - y^2 = 0\}. \quad (8)$$

La condición para que una recta en \mathbb{P}^2 sea tangente a C puede calcularse utilizando el jacobiano de la función $(x, y, z) \mapsto xz - y^2$:

$$(z, -2y, x). \quad (9)$$

Así, las rectas en \mathbb{P}^2 , o equivalentemente los puntos en \mathbb{P}^{2^\vee} tangentes a C son:

$$C^\vee := \{(z : -2y : x) \in \mathbb{P}^{2^\vee} \mid xz - y^2 = 0\}. \quad (10)$$

Claramente (10) es también una cónica y por lo tanto una recta proyectiva encajada en el plano dual. Esta cónica es llamada la **cónica dual** de C .

La curva elíptica de Poncelet

² Los ejemplos que estudiamos en [2] involucraron un par de circunferencias ajenas en \mathbb{R}^2 . Si pensamos en dichas circunferencias como restricciones afines y reales del siguiente par de cónicas en \mathbb{P}^2 :

$$C_1 := \{(x : y : z) \in \mathbb{P}^2 \mid x^2 + y^2 - z^2 = 0\}, \quad (11)$$

$$C_2 := \{(x : y : z) \in \mathbb{P}^2 \mid x^2 - 2dxz + y^2 + (d^2 - r^2)z^2 = 0\}. \quad (12)$$

uno puede ver que:

$$C_1 \cap C_2 = \{(\pm 1 : i : 0), (1 + d^2 - r^2 : \pm \sqrt{4d^2 - (1 + d^2 - r^2)^2} : 2d)\}. \quad (13)$$

Cuando ocurre, como en este caso, que dos cónicas en \mathbb{P}^2 se intersecan en 4 puntos distintos, se dice que están en **posición general**. Como veremos enseguida, la posición general tiene una consecuencia geométrica muy importante para el problema de hallar n -ágonos de Poncelet. Abusando de la notación, dados $p \in \mathbb{P}^2$ y $t \in \mathbb{P}^{2^\vee}$ escribiré $p \in t$ para indicar que el punto p está en la recta en \mathbb{P}^2 definida por t .

²Esta sección supone conocimientos básicos sobre curvas algebraicas. El capítulo IV de [5] cubre todo lo que requerimos.

Proposición 1 Sean $C, D \subset \mathbb{P}^2$ un par de cónicas en posición general. Entonces

$$E := \{(p, t) \in C \times D^\vee \mid p \in t\} \quad (14)$$

es una **curva elíptica**.³

Prueba: Como se discutió previamente, tanto C como D^\vee son copias de \mathbb{P}^1 . Por lo tanto puedo pensar en los puntos de E como puntos en la superficie $S := \mathbb{P}^1 \times \mathbb{P}^1$. Uno verifica que la restricción $p \in t$ de (14) define una curva no singular en S (es decir, E es una curva algebraica no singular). Tenemos un par de mapeos $E \rightarrow \mathbb{P}^1$ inducidos por las proyecciones a la primer y segunda coordenadas. Consideremos el mapeo $\pi_1 : E \rightarrow \mathbb{P}^1$ inducido al proyectar sobre C .

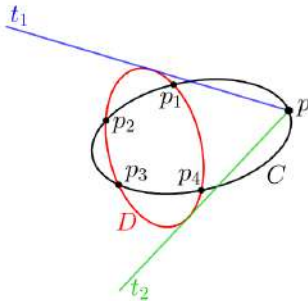


Figura 3: Una cubierta doble de \mathbb{P}^1 ramificada en 4 puntos.

Sean p_1, \dots, p_4 los puntos de intersección de C y D y denotemos con \mathcal{R} al conjunto que los contiene. Entonces π_1 es 2-a-1 en $C \setminus \mathcal{R}$. Esto es consecuencia del hecho de que estos puntos son los únicos puntos de C desde donde no es posible trazar dos tangentes a D , sino una sola. Por abuso de notación para el mapeo π_1 , tenemos que

³La definición formal de curva elíptica puede consultarse en [5], Capítulo IV, §4. Para nuestros propósitos la caracterización dada en la prueba de la Proposición 1 es suficiente.

para un punto $p \in C \setminus \mathcal{R}$ existen $t_1, t_2 \in D^\vee$ tales que $\pi_1(p, t_1) = \pi_1(p, t_2) = p$ (ver Figura 3). Los puntos p_1, p_2, p_3, p_4 son pues, los **puntos de ramificación** de π_1 . E es entonces, una curva no singular que admite un mapeo 2-a-1 a \mathbb{P}^1 con cuatro puntos de ramificación. Por un resultado de Hurwitz (ver Corolario 2.4 en el capítulo IV de [5]) se sigue de inmediato que E es una curva de género 1. Q.E.D.

Dado que estamos considerando curvas algebraicas proyectivas sobre \mathbb{C} , podemos interpretar el género como el número de agujeros de la superficie de Riemann compacta correspondiente. Topológicamente pues, E es un toro.

El modelo de Weierstrass

Puede probarse (ver por ejemplo, la versión extendida de este trabajo en [4]) que toda curva elíptica E es de la forma:

$$E \cong \text{Proj } \mathbb{C}[x_1, x_2, x_3] / (x_2^3 + ax_1^2x_2 + bx_1^3 - x_1x_3^2). \quad (15)$$

Esto es, una cúbica no singular en \mathbb{P}^2 ,⁴

El modelo dado por (15) suele llamarse **modelo de Weierstrass** de una curva elíptica. En el parche afín $\mathbb{P}^2 \setminus (x_1 = 0)$, podemos tomar coordenadas $x := x_2/x_1$, $y := x_3/x_1$ y la ecuación que define a E , $x_1x_3^2 = x_2^3 + ax_1^2x_2 + bx_1^3$ se convierte en:

$$y^2 = x^3 + ax + b. \quad (16)$$

El lector familiarizado con la teoría de funciones elípticas notará su relación con el Porismo de Poncelet a partir de la ecuación (16). Una buena introducción a dicha teoría está dada en el primer capítulo de [9]. Es bien sabido que la función elíptica de Weierstrass denotada por \wp y su derivada \wp' satisfacen una ecuación diferencial de la forma:

$$[\wp']^2 = 4[\wp]^3 - g_2\wp - g_3, \quad g_2, g_3 \in \mathbb{C}. \quad (17)$$

⁴La no singularidad, implica en particular que las constantes a, b no pueden ser simultáneamente 0, pues de ser así, $(1 : 0 : 0)$ sería un punto singular.

De aquí se sigue que el mapeo $t \mapsto (1 : \wp(t) : \wp'(t))$ define una parametrización ⁵ de una curva elíptica y viceversa, a partir de (16) es claro que toda curva elíptica admite una parametrización de este tipo.

La prueba del Porismo

Utilizaré la notación de la Proposición 1. Dado un par de cónicas en \mathbb{P}^2 en posición general, consideremos la curva elíptica dada por

$$E := \{(p, t) \in C \times D^\vee \mid p \in t\}. \quad (18)$$

C y D se intersecan en 4 puntos, p_1, \dots, p_4 y dualmente, existen 4 tangentes comunes a ambas cónicas, ℓ_1, \dots, ℓ_4 . Definamos un par de funciones, $\eta, \iota : E \rightarrow E$, como sigue:

$$\eta(p, t) = (p', t), \quad (19)$$

donde $\{p, p'\} = C \cap t$.

$$\iota(p, t) = (p, t'), \quad (20)$$

donde $\{t, t'\} = D^\vee \cap p$.

Tenemos $\eta(p, t) = (p, t)$ si y sólo si $t \in \{\ell_1, \dots, \ell_4\}$. $\iota(p, t) = (p, t)$ si y sólo si $p \in \{p_1, \dots, p_4\}$. Es claro además que $\eta \circ \eta = \iota \circ \iota = Id_E$. La conclusión es que η y ι son automorfismos respecto a la estructura de grupo de E , mismos que por las observaciones anteriores, tienen exactamente 4 puntos fijos y son involuciones. También por construcción, si definimos $f : E \rightarrow E$ como

$$f = \iota \circ \eta. \quad (21)$$

Entonces (C, D) admite un n -ágono interinscrito si y sólo si $f^{on}(p_0, t_0) = (p_0, t_0)$ para algún $(p_0, t_0) \in E$, es decir, si y sólo si f^{on} tiene un punto fijo.

⁵En principio, esta parametrización es local, pero puede extenderse a una parametrización global si consideramos como dominio de \wp y \wp' la superficie de Riemann \mathbb{C}/Λ , donde Λ es la latiz de periodos de \wp y \wp' .

Ahora bien, dada una latiz generada por dos puntos no colineales del plano complejo, v_1, v_2 :

$$\Lambda := \{av_1 + bv_2 \in \mathbb{C} \mid a, b \in \mathbb{Z}\}, \quad (22)$$

denotaré por \bar{z} a la clase de equivalencia del número $z \in \mathbb{C}$ en el toro cociente \mathbb{C}/Λ . Sea $\alpha \in \mathbb{C}$ tal que $\alpha\Lambda = \Lambda$. Es sabido que los automorfismos del grupo \mathbb{C}/Λ son de la forma

$$\bar{z} \mapsto \alpha\bar{z} + \bar{\beta}, \quad \text{para algún } \beta \in \mathbb{C} \text{ fijo.} \quad (23)$$

Como ejercicio, el lector puede probar que un automorfismo dado por (23) es una involución con al menos un punto fijo si y sólo si $\alpha = -1$. Se sigue que, bajo la parametrización de E dada por las funciones elípticas \wp y \wp' , la función $f = \iota \circ \eta$ corresponde a un automorfismo de \mathbb{C}/Λ de la forma

$$\bar{f}(\bar{z}) = \bar{z} + \bar{\gamma}, \quad \text{para algún } \gamma \in \mathbb{C} \text{ fijo.} \quad (24)$$

Por lo tanto f^{on} tiene un punto fijo si y sólo si existe algún $\bar{z}_0 \in \mathbb{C}/\Lambda$ tal que

$$\bar{z}_0 = \bar{z}_0 + n\bar{\gamma}. \quad (25)$$

Lo cual ocurre si y sólo si $\bar{\gamma} = \bar{0}$. Es decir, si y sólo si f^{on} es el mapeo identidad. De aquí se sigue el Porismo de Poncelet.

Sobre el problema de Fuss

Explotando aún más la teoría de funciones elípticas, es posible dar una solución definitiva al problema de Fuss para circunferencias discutido en [2]. Tal solución fue publicada por D. Hulin en 2007 en su artículo [6]. Puede probarse que la relación necesaria y suficiente entre los radios de dos circunferencias en el plano afín, R y r y la distancia entre sus centros d para que satisfagan el Porismo de Poncelet para polígonos de n lados, es una ecuación polinomial homogénea en (R, r, d) con coeficientes enteros que puede calcularse recursivamente respecto a n . En particular, el lugar geométrico de soluciones al problema de Fuss es una curva proyectiva plana. Con ayuda

de un programa de álgebra computacional, puede probarse que a partir de polígonos de 6 lados estas curvas tienen género mayor a 1. La geometría de estas curvas, así como los problemas aritméticos relacionados no están del todo resueltos. Sería interesante investigar, por ejemplo, la existencia de circunferencias de Poncelet con radios y distancia entre sus centros racionales que admitan polígonos interinscritos de 6 o más lados (ver el proyecto numérico sugerido en [2], §3).

Ejercicios

El objetivo de estos ejercicios es auxiliar al lector interesado para profundizar en los conceptos discutidos en las secciones anteriores.

Considera el triángulo Δ con vértices en $P_1 := (0, 0)$, $P_2 := (4, 0)$ y $P_3 := (0, 2)$.

1. Calcula la ecuación de la circunferencia \mathcal{C} circunscrita a Δ .
2. Calcula la ecuación de la elipse \mathcal{E} inscrita en Δ que pasa por los puntos medios de sus lados,⁶ siguiendo alguna de estas estrategias:

- a) **Utilizando el Teorema de Marden:** Puedes consultar este resultado en [8]. Piensa en los vértices de Δ como puntos en \mathbb{C} . Considera el polinomio de tercer grado $p(z) \in \mathbb{C}[z]$ cuyas raíces son P_1, P_2 y P_3 :

$$p(z) = z(z - 4)(z - 2i).$$

El Teorema de Marden establece que los focos de la elipse de Steiner son los ceros de $p'(z)$. Calculando dichos ceros, puedes calcular la ecuación de la elipse.

- b) **Utilizando cónicas duales:** Conocemos 3 rectas tangentes a \mathcal{E} (las rectas determinadas por los lados de

Δ). Dos de ellas son los ejes horizontal y vertical del plano, que son tangentes a \mathcal{E} en los puntos $M_1 := (2, 0)$ y $M_2 := (0, 1)$ respectivamente. Es fácil probar que el baricentro de Δ , denotado con O , es precisamente el centro de \mathcal{E} . Sean $M'_i, i = 1, 2$, las reflexiones de los puntos M_i respecto a O . Por simetría, la recta paralela al eje horizontal que pasa por M'_1 y la recta paralela al eje vertical que pasa por M'_2 son tangentes a \mathcal{E} . Conociendo 5 rectas tangentes a \mathcal{E} , puedes calcular su ecuación utilizando el principio de dualidad. Puedes encontrar más detalles al respecto y un código computacional para realizar el cálculo automáticamente en [3].

3. Escribe las ecuaciones de las completaciones proyectivas complejas de \mathcal{C} y \mathcal{E} y encuentra $\mathcal{C} \cap \mathcal{E} := \{R_i\}_{i=1}^4$.
4. Sea $F(x, y, z)$ el polinomio de grado 2 que obtuviste homogeneizando la ecuación de \mathcal{C} en el inciso anterior. Calcula la matriz $A \in \text{Mat}_{3 \times 3}(\mathbb{C})$ que transforma a $F(x, y, z)$ en el polinomio $xz - y^2$ (consulta como guía, el ejemplo al final de la sección §2). Recuerda que la transformación definida por A es conocida como **proyectividad**; denotemos dicha proyectividad con $\varphi_A : \mathbb{P}^2 \rightarrow \mathbb{P}^2$.
5. Calcula una biyección entre los puntos de la cónica $xz - y^2 = 0$ y los puntos de la recta proyectiva \mathbb{P}^1 . Recuerda que esta cónica es exactamente la imagen de \mathbb{P}^1 bajo el segundo encaje de Veronese $V_2 : \mathbb{P}^1 \rightarrow \mathbb{P}^2$, definido mediante $V_2(s : t) = (s^2 : st : t^2)$. En otras palabras, debes calcular una inversa para V_2 . Denotemos con ψ a dicha inversa.
6. Calcula $B_i := \psi \circ \varphi_A(R_i)$, las imágenes en la recta proyectiva de los puntos de intersección $\mathcal{C} \cap \mathcal{E}$ que obtuviste en el inciso 3. Revisa la Proposición 1 y explica por qué la curva elíptica de Poncelet asociada al par

⁶Esta elipse es conocida como **inelipse de Steiner** de Δ .

de cónicas $(\mathcal{C}, \mathcal{E})$ está determinada por los puntos B_i .

Referencias

- [1] W. Bosma, J. Cannon, C. Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput., 24, 235–265, 1997. Calculator en línea de libre acceso: <http://magma.maths.usyd.edu.au/calc/>
- [2] Garza Ledesma, J. S., *El Porismo de Poncelet Parte I*. Por aparecer en el Boletín de la Sociedad Matemática de Computación Científica y sus Aplicaciones. 2019.
- [3] Garza Ledesma, J. S., *La cónica dual*. 2019. Material de libre acceso disponible en: <https://sites.google.com/view/juan-garza/nigromante/poncelet/dualidad>
- [4] Garza Ledesma, J. S., Sitio web: <https://sites.google.com/view/juan-garza/nigromante/poncelet>
- [5] Hartshorne, R., *Algebraic geometry*. Graduate Texts in Mathematics, Vol. 52, Springer-Verlag, New York. 1977.
- [6] Hulin, D., *Factorisation des polynômes de Poncelet*, Geom. Dedicata 130, pp. 109-136. 2007.
- [7] IGI, GeoGebra. Software interactivo disponible en: <http://www.geogebra.org>
- [8] Kalman, D., *An Elementary Proof of Marden's Theorem*. The Mathematical Association of America. Monthly 115. pp. 330-338. 2008. Disponible en línea en la liga: <http://dankalman.net/AUhome/pdffiles/mardenAMM.pdf>
- [9] Lang, S., *Elliptic Functions*. Graduate Texts in Mathematics, Vol. 112, Springer-Verlag, New York. 1987.

Ruteo de vehículos escolares: caso de estudio de la Universidad Autónoma de Coahuila

O. Avalos-Rosales¹, M. Juárez-Peláez², Y. Cardona-Valdés^{1*}

¹Centro de Investigación en Matemáticas Aplicadas
Universidad Autónoma de Coahuila

² Facultad de Sistemas, Universidad Autónoma de Coahuila

* Correo electrónico: y.cardona@uadec.edu.mx

Resumen

En este trabajo se proponen un conjunto de nuevas rutas de transporte para dar servicio a la comunidad estudiantil actual del Campus Arteaga de la Universidad Autónoma de Coahuila atendiendo a la dispersidad de los estudiantes en la zona metropolitana de Saltillo. Se analiza un caso de estudio con la información de una Facultad, considerando 380 estudiantes y 46 paradas correspondientes a las colonias de Saltillo. El problema se formula matemáticamente como un modelo lineal entero mixto, el cual es resuelto a través de un optimizador comercial. Con el modelo sólo es posible resolver a optimalidad instancias de 10 paradas y 50 estudiantes, por lo que se diseñó un algoritmo heurístico constructivo híbrido basado en el algoritmo de Clarke y Wright, en donde la factibilidad de una solución se verifica al resolver una versión simplificada del modelo matemático. Se implementan dos estrategias para reducir el tiempo de cómputo del algoritmo heurístico y se reportan las nuevas rutas propuestas, así como algunas recomendaciones para facilitar la construcción de las instancias del problema.

Palabras clave: Ruteo de vehículos escolares; Caso de estudio; Programación entera mixta; Heurísticos.

Introducción

La Universidad Autónoma de Coahuila en el año 2010 inauguró un nuevo campus en el municipio de Arteaga Coahuila, el cual se encuentra a 15.7 kilómetros de Saltillo, Ciudad capital del Estado de Coahuila. Ya que el transporte público era muy limitado, en el año 2011, la Universidad implementó el servicio de transporte universitario (Lobús) para trasladar a los usuarios del campus Arteaga con el municipio de Saltillo. Dicho servicio constaba de sólo dos autobuses y era suficiente para cubrir la demanda de los usuarios, ya que hasta ese momento solo se había instalado la Facultad de Sistemas. Sin embargo, el servicio

original se componía de una sola ruta con una cobertura muy limitada.

En los últimos años, la población estudiantil y el número de usuarios se han incrementado al establecerse en el campus Arteaga además de la Facultad de Sistemas otras tres Facultades. Actualmente, la cantidad de estudiantes es de 3072 y se estima que 2500 son usuarios del Lobús. Sin embargo, a la fecha la Universidad sólo ha incrementado el número de autobuses (de dos a cuatro) para dar el servicio, pero manteniendo la misma ruta y paradas que en el diseño original.

Cabe mencionar que los estudiantes viven en zonas alejadas de las paradas oficiales por las que pasa el transporte universitario, lo que hace no-

tar que la ruta y las paradas existentes sean insuficientes para la creciente demanda y la dispersidad de la comunidad Universitaria en la zona metropolitana de Saltillo, por lo que urge un plan en el rediseño de las rutas escolares que tome en cuenta ambas problemáticas (ver Figura 1).



Figura 1: Dispersidad de las colonias de los estudiantes de la Facultad de Sistemas de la base de datos del 2014.

La problemática planteada se ha estudiado en la literatura científica como un problema de ruteo de vehículos escolares (SBRP por sus siglas en inglés) el cual involucra el transporte de estudiantes desde las paradas de autobús hacia sus escuelas. El objetivo es diseñar rutas optimizando una función objetivo de interés. El SBRP al ser una variante del bien conocido problema de ruteo de vehículos (VRP) se clasifica como un problema NP-hard [4].

En la literatura científica se han propuesto formulaciones matemáticas y algoritmos heurísticos y metaheurísticos para resolver el problema [5, 6] y [3].

El SBRP se descompone en 5 subproblemas:

1. Preparación de los datos
2. Selección de paradas de autobús
3. Generación de rutas
4. Ajuste a los horarios de la escuela

5. Programación de las rutas

La propuesta de solución que consideramos en este trabajo se enfoca en los tres primeros subproblemas. Donde, además se considera el problema de selección de paradas y generación de rutas de manera simultánea.

Descripción del problema

El SBRP considera un conjunto potencial de paradas y un conjunto de estudiantes, donde cada estudiante puede acceder a una o más paradas, expresado a través de lo que denominaremos matriz de accesibilidad. Se dispone de una flota homogénea de autobuses con capacidad conocida. Todas las rutas comienzan en una estación de autobuses central, recorren un conjunto de paradas y culminan en la escuela. Se debe determinar cuáles paradas serán visitadas por los autobuses escolares, así como en qué parada debe recogerse cada estudiante y el conjunto de rutas que minimicen el costo total de recorrido de todos los autobuses. Dicho costo podría ser la distancia de recorrido, el tiempo de recorrido o algún otro.

Metodología de solución

Para abordar el problema se implementan dos metodologías tomando como base las propuestas por Schittekat et al. (2013) [5], una es un modelo matemático de programación lineal entera mixta (MILP por sus siglas en inglés) y la otra un algoritmo heurístico. Para ambas metodologías se consideran las siguientes suposiciones: 1) se dispone de suficientes autobuses para cubrir la demanda, 2) una parada sólo será visitada por un autobús, esto significa que el número de estudiantes por parada no puede exceder la capacidad del autobús, 3) todos los autobuses tienen la misma capacidad, 4) un autobús sólo puede realizar una ruta, y 5) cada estudiante cuenta como una unidad de demanda que nos sirve para validar que no se exceda la capacidad del autobús.

Dada la complejidad del problema, utilizando el modelo matemático sólo fue posible resolver a optimalidad instancias de 10 paradas con 50 estudiantes. Para más detalles sobre el modelo y su desempeño ver [2]. Para instancias de mayor tamaño se implementó un algoritmo heurístico del tipo constructivo basado en el método de los ahorros de Clarke-Wright, que se describe en la siguiente sección.

Algoritmo constructivo

La idea central del algoritmo consiste construir una solución donde cada parada se visita en una ruta independiente, luego ir uniendo pares de rutas al conectar la última parada de una ruta con la primer parada de la otra ruta, verificar si para dicha unión es posible asignar los estudiantes de tal forma que no se exceda la capacidad de ningún autobús y seguir uniendo rutas hasta que ya no sea posible unir ningún par de ellas. Para ello se crean matrices auxiliares que permitirán seleccionar el par de paradas a unir. El algoritmo se esquematiza diagrama de flujo de la Figura 2.

De manera inicial, el algoritmo construye una **solución tipo estrella** en donde se establece una ruta independiente por cada parada (ver Figura 3). En caso de que la solución no sea factible el algoritmo termina su ejecución.

Si la solución tipo estrella resulta factible, se procede al proceso de unión de rutas. Para ello se calcula por única ocasión la **matriz de ahorros** aplicando la fórmula $\sigma_{ij} = c_{ij} - c_{i(np+1)} - c_{0j}$, la cual es una modificación de la fórmula propuesta por Clarke y Wright [1] (ver Figura 4). Donde np representa el número de paradas, c_{ij} representa el costo de ir de la parada i a la parada j y, σ_{ij} representa el ahorro que resultaría al unir dos rutas, tal que i es la última parada de la primer ruta y j es la primer parada de la segunda ruta.

Luego calculamos la **matriz de conectables** que define cuáles son los pares de paradas posibles que se podrían conectar. Ésta es una matriz binaria donde un elemento toma el valor de 1

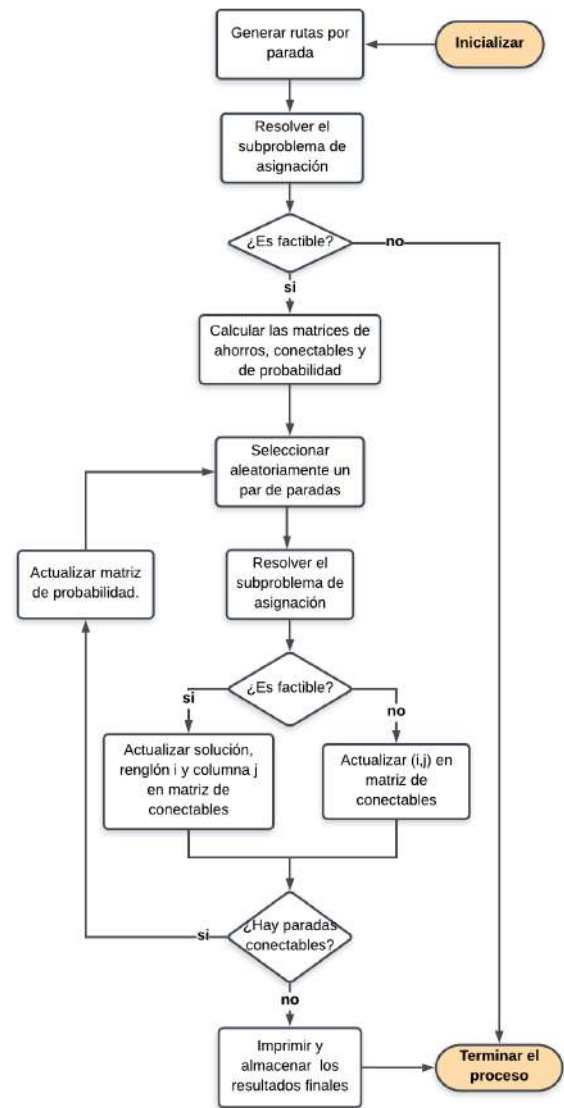


Figura 2: Diagrama de flujo del algoritmo heurístico

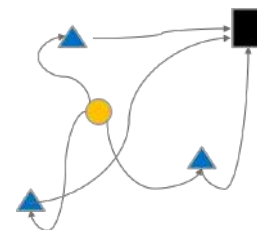


Figura 3: Solución tipo estrella. Depósito (amarillo), paradas (azul), escuela (negro)

si un par de paradas son conectables y 0 en caso contrario. En la solución inicial tipo estrella todas las paradas son conectables con todas las

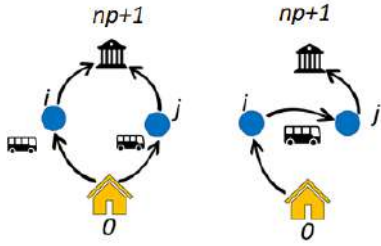


Figura 4: Modificación algoritmo Clarke y Wright. Depósito en color amarillo (nodo 0), paradas en color azul (i y j), escuela en color negro ($np + 1$).

demás paradas.

Enseguida calculamos la **matriz de probabilidad** que nos permite establecer prioridades en cuanto a la selección de paradas a conectar, donde la probabilidad de conectar cada par de paradas i y j es proporcional a su ahorro. Esta probabilidad se calcula como el cociente del costo de conectar dichas paradas, sobre la suma de los costos de todas las paradas que aún son conectables (de acuerdo a la matriz de conectables).

Entonces llevamos a cabo el proceso de **selección de ruleta**, el cual consiste en generar un número aleatorio entre 0 y 1, e ir sumando las probabilidades de cada elemento en la matriz de probabilidades, hasta que la suma iguale o exceda al número aleatorio generado, identificando así las paradas seleccionadas para conectarse y, en consecuencia, las rutas a que estas paradas pertenecen.

Obtenemos la **solución candidata** al modificar las dos rutas seleccionadas: pasamos todas las paradas de la segunda ruta al final de la primera ruta, y dejamos vacía la segunda ruta. Entonces se valida la factibilidad de dicha solución candidata al resolver el **subproblema de asignación** de estudiantes a autobuses (siguiente sección).

Si la unión de rutas genera una solución factible, se actualizan la solución, la asignación de estudiantes a autobuses, la función objetivo y la matriz de conectables. En la matriz de conectables se ponen como cero todos los elementos del

renglón i y de la columna j . Si la unión de rutas no da solución factible sólo se pone como cero el elemento (i, j) de la matriz de conectables y se desecha la solución candidata.

Si aún hay pares de paradas por conectar se recalcula la matriz de probabilidades y se regresa al proceso de selección de ruleta. En caso contrario, se guarda la información relevante y el algoritmo termina.

Subproblema de asignación de estudiantes a autobuses

Dado que cada estudiante puede acceder a una o más paradas, la demanda de una parada no se determina de forma directa y es necesario resolver un subproblema de asignación de estudiantes a autobuses para verificar si para un conjunto de rutas dado, existe una asignación factible al problema, es decir, que respete la capacidad de los autobuses.

Para ello, se calcula la matriz de **costo de asignación** de estudiantes a autobuses y se resuelve el subproblema que minimice el costo asignar de estudiantes a autobuses.

El modelo del subproblema de asignación se establece como sigue:

$$\text{minimizar } \sum_{l \in S} \sum_{k \in K} t_{kl} z'_{kl} \quad (1)$$

sujeto a:

$$\sum_{k \in K} z'_{kl} = 1; \quad \forall i \in S \quad (2)$$

$$\sum_{l \in S} z'_{kl} \leq Q; \quad \forall k \in K \quad (3)$$

$$z'_{kl} \in \{0, 1\} \quad \forall k \in K, l \in S \quad (4)$$

Las restricciones (2) aseguran que cada estudiante l es asignado a exactamente una ruta k . Las restricciones (3) garantizan que la capacidad de los autobuses no se exceda y, por último, las restricciones (4) establecen la naturaleza binaria de las variables, sin embargo éstas se pueden relajar ya que al resolver la asignación de estudiantes

a las rutas, se está resolviendo un problema de transporte y es sabido que en este tipo de problemas si el suministro y la demanda son enteros, la solución será entera.

El modelo anterior también fue planteado por Schittekat et al. (2013) [5], sin embargo ellos optaron por usar el bien conocido *primal-dual labeling method*. En nuestro heurístico sí se resuelve de forma exacta con el modelo matemático, al llamar internamente al optimizador CPLEX.

Asignación final de estudiantes a paradas

De la heurística anterior se obtienen las rutas de los autobuses, es decir, la asignación y orden en que cada autobús debe visitar las paradas que le fueron asignadas. Sin embargo, no se sabe con exactitud cuál es la asignación de estudiantes a las paradas; y si es necesario visitarlas todas.

La asignación de estudiantes a paradas la hacemos a través de una matriz binaria P , el valor 1 indica que un estudiante l es asignado a una parada i y cero en otro caso. La matriz se inicializa en ceros y asignaremos un estudiante a la vez, primero el estudiante 1, 2, 3, etc.

- Para cada estudiante l buscaremos cual fue la ruta k^* a la que fue asignado.
- Usando la ruta k^* y la matriz de accesibilidad (de los parámetros del problema) determinamos cuál es la primer parada i^* en la ruta k^* a la que puede llegar caminando y lo asignamos a ella.

Con esto hemos determinado las rutas de los autobuses, es decir, la asignación de paradas a autobuses y el orden en que se visitan. También se ha determinado la asignación de estudiantes a paradas.

Validación de la metodología

Adicional a la instancia generada para el caso de estudio, para determinar el alcance del modelo matemático y del heurístico presentado en las

No.	Instancia	MILP		Heurístico	
		FO	Tiempo CPU	Desv	Tiem.
1	5p25e25q5w	332.63	0.06	20.9	0.92
2	5p25e50q5w	332.63	0.05	20.9	1.86
3	5p25e25q10w	374.91	0.06	12.1	2.35
4	5p25e50q10w	374.91	0.06	9.8	4.91
5	5p25e25q20w	362.36	0.09	26.1	4.71
6	5p25e50q20w	362.36	0.09	30.9	5.84
7	5p25e25q40w	361.17	0.09	17.1	5.99
8	5p25e50q40w	361.17	0.11	30.7	6.64
9	5p50e25q5w	638.29	0.07	17.3	5.35
10	5p50e50q5w	326.6	0.09	3.0	1.51
11	5p50e25q10w	611.3	0.06	7.3	5.44
12	5p50e50q10w	402.88	0.08	17.2	4.31
13	5p50e25q20w	622.68	0.10	0.0	3.99
14	5p50e50q20w	293.56	0.28	32.1	2.04
15	5p50e25q40w	433.00	0.25	8.9	5.09
16	5p50e50q40w	274.39	0.15	22.8	1.24
17	5p100e25q5w	1306.41	0.03	0.0	8
18	5p100e50q5w	831.03	0.07	8.6	3.6
19	5p100e25q10w	1245.28	0.02	0.0	8.26
20	5p100e50q10w	616.69	0.06	0.0	2.92
21	5p100e25q20w	998.82	0.09	0.0	6.79
22	5p100e50q20w	710.10	0.21	13.9	4.17
23	5p100e25q40w	711.73	0.86	6.7	7.42
24	5p100e50q40w	457.26	0.73	12.6	3.06
25	10p50e25q5w	733.41	20.33	19.0	9.52
26	10p50e50q5w	555.15	11.93	30.9	2.58
27	10p50e25q10w	747.57	70.30	34.1	10.12
28	10p50e50q10w	595.00	1.75	39.0	7.15

Tabla 1: Resultados comparativos entre el MILP y el algoritmo heurístico para las primeras 20 instancias

secciones anteriores también se construyeron un conjunto de instancias pseudoaleatorias.

Se generaron 14 grupos de instancias con diferente número de paradas (p) y estudiantes (e), cada grupo se conforma de 8 instancias en las que la capacidad del autobús (q) varía en 25 ó 50, y la distancia que el estudiante puede caminar a una parada (w) varía entre 5 y 40 unidades. Para más detalles sobre la construcción de las instancias ver [2].

La nomenclatura utilizada se ejemplifica a continuación con la instancia 5p25e25q5w, con 5 paradas potenciales, 25 estudiantes, la capacidad del autobús en 25 unidades, y finalmente, la distancia que el estudiante puede caminar hacia una parada es de 5 unidades.

La experimentación se llevó a cabo en una estación de trabajo WorkStation HP Z440, Intel Xeon, RAM 8GB, 1 TB. El modelo matemático

se implementó en Visual C++ usando CPLEX 12.6 con Concert Technology y estableciendo un tiempo límite de ejecución de 7200 segundos.

En la Tabla 1 se muestran los resultados para las primeras 28 instancias, donde el MILP fue capaz de obtener soluciones óptimas. En la columna 2 se muestran los resultados correspondientes al modelo, indicando el valor de la función objetivo (FO) y el tiempo (Tiempo) en segundos. En la columna 3 se muestran los resultados del algoritmo heurístico, indicando el valor de la desviación con respecto al óptimo (Desv) y el tiempo (Tiempo CPU) en segundos.

Cabe destacar que el heurístico es capaz de obtener soluciones que coinciden con el óptimo del modelo, sin embargo en otros casos obtienen soluciones que se desvían hasta un 34.1 % del costo de las soluciones óptimas.

Reducción de tiempos

En la experimentación computacional realizada con el heurístico se notó que el tiempo computacional resultó ser muy elevado, por lo que se diseñaron dos estrategias para reducirlo. Ambas estrategias modifican el criterio de paro, el cual actualmente consiste detener el proceso de unión de rutas si no hay más pares de paradas por conectar. Las estrategias son las siguientes:

1. Detener el proceso de unión de rutas si la cantidad de veces que de forma consecutiva ésta ha resultado infactible, es menor que una predeterminada cantidad de veces (umbral).
2. Detener el proceso de unión de rutas si la cantidad de rutas actuales es igual a la cantidad mínima de autobuses requeridos.

La Figura 5 muestra los tiempos de ejecución CPU en segundos, para las instancias de la 49 a la 112, para distintos valores umbrales (10, 20, 30 y 40) comparados con los tiempos sin Umbral.

Se encontró que con todos los umbrales se pierde muy poco en la calidad de las soluciones

obtenidas respecto a la versión sin umbral (menos de 2.66 %) y además se obtiene una reducción de tiempos importante. Se eligió un umbral de 30 por ser el que ofrece los mejores valores (0.6 % de pérdida en calidad, y tiempo promedio de ejecución de 27.77seg). La segunda estrategia incrementa ligeramente el tiempo computacional respecto a la primera, por lo que al final no fue considerada en la versión final del algoritmo.

Caso de estudio

Para el caso de estudio de la UAdeC, los datos de entrada a considerar son: el número de estudiantes, el número de paradas, el número de autobuses, la capacidad de los autobuses, las coordenadas de ubicación de los estudiantes, de las paradas, del depósito de autobuses, y de la escuela, así como la máxima distancia que puede caminar un estudiante a una parada. A partir de dichos datos se determinan los siguientes parámetros para construir una instancia del SBRP en estudio:

- Matriz de accesibilidad de estudiantes a paradas
- Matriz de costos entre paradas

Para el caso de estudio se consideró la base de datos (BD) del año 2014, proporcionada por la Facultad de Sistemas, dicha base consta de 885 alumnos de los cuales 736 indicaron una dirección de la ciudad de Saltillo y 149 se registraron como foráneos.

Para determinar las paradas potenciales se tomó en cuenta a todas aquellas colonias que tienen registrado en la BD al menos 4 estudiantes, ya que se busca cubrir el mayor número de demanda de los estudiantes que se encuentran dispersos geográficamente por la ciudad de Saltillo y sean trasladados al Campus Arteaga (ver Tabla 2).

Con todas estas consideraciones la instancia a trabajar considera: 46 paradas (una por cada

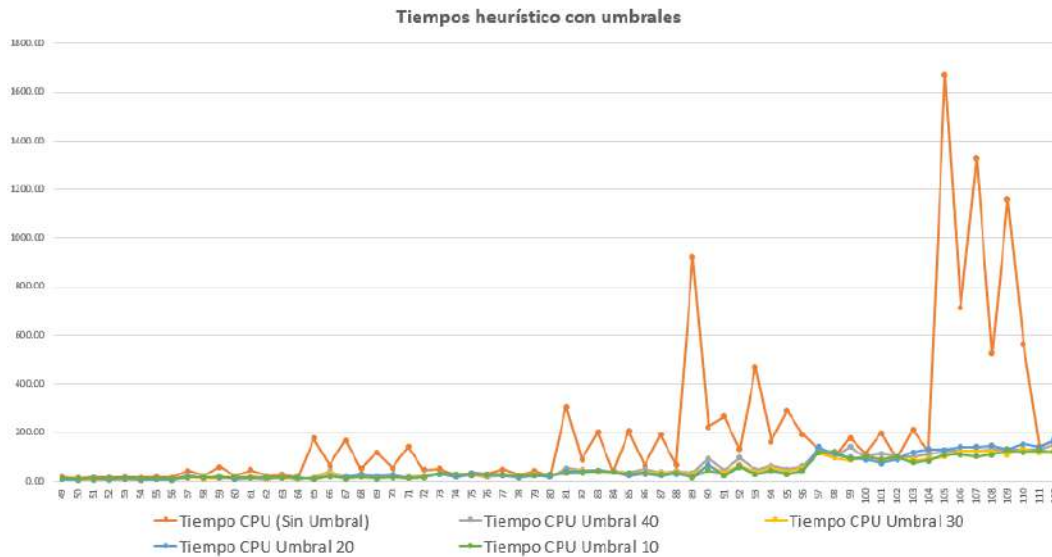


Figura 5: Comparativo del tiempo de ejecución para algunas las instancias 49 a 112 para distintos valores umbrales (Fuente: Elaboración Propia)

colonia), 380 estudiantes y la capacidad del autobús de 50. Para determinar la matriz de accesibilidad que representa la distancia caminando de los estudiantes a las paradas, no se consideraron las distancias reales caminando de los estudiantes a cada parada debido al tiempo que conlleva el cálculo de las mismas; en su lugar, se consideró que un estudiante únicamente puede llegar caminando a la parada que se encuentra en su colonia. Las coordenadas de una parada (colonia) se obtuvieron al introducir el nombre de la colonia en GoogleMaps y registrar la latitud y longitud que éste reporta por defecto.

A partir de dichos datos, se construyeron tres instancias variando la matriz de costos entre paradas. La primera instancia considera la matriz de costos como la matriz de distancia euclidiana, la segunda considera la matriz de costos como la matriz de distancias más corta entre calles entre cada par de paradas, y la tercera considera la matriz de costos como la matriz de menor tiempo de traslado entre cada par de paradas. Las dos últimas matrices fue posible obtenerlas mediante el uso del Software NEVA.¹ Al variar en cada

instancia la matriz de costos se varía la función objetivo, para las primeras dos se minimiza la distancia y para la tercera el tiempo de recorrido.

Rutas propuestas

La Figura 6 muestra para cada una de las tres instancias consideradas para el caso de estudio, el valor de la función objetivo (FO), el tiempo de ejecución del algoritmo (CPU en segundos) y las rutas generadas. Las primeras dos instancias minimizan la distancia en kilómetros (km) y la tercera minimiza el tiempo en minutos (min). Las rutas generadas se componen por un conjunto de paradas, el cero que denota la salida del depósito de autobuses y el 47 denota el destino, es decir, el Campus Arteaga.

Las Figuras 7 y 8 muestran las nueve rutas generadas para la instancia cuya matriz de costos es la matriz de tiempos obtenida a través de NEVA.

¹NEVA es un Sistema de Información Geográfica desarrollado por el Dr. Joaquín A. Pacheco (Universidad de Burgos, España), que emplea ficheros shape para cargar cartografía digitalizada.

No.	Colonia	#Est.	No.	Colonia	#Est.	No.	Colonia	#Est.
1	Zona Centro	42	17	Brisas Poniente	7	33	Privadas la Torre	5
2	Mirasierra	21	18	Nuevo Mirasierra	7	34	San Ramón	5
3	Morelos	17	19	Valle de las Torres	7	35	Satélite Norte	5
4	Zaragoza	17	20	Bonanza	6	36	Topochico	5
5	Amp. Morelos	15	21	Cerro del Pueblo	6	37	Buitres	4
6	Bella vista	13	22	Chamizal	6	38	Europa	4
7	Satélite Sur	13	23	La Fragua	6	39	Girasol	4
8	Ignacio Zaragoza	12	24	Landín	6	40	Jardines Coloniales	4
9	Vista Hermosa	12	25	Lomas de Lourdes	6	41	La Aurora	4
10	Fundadores	11	26	Oceanía Boulevares	6	42	La Minita	4
11	Vicente Guerrero	11	27	Chapultepec	5	43	Pueblo Insurgentes	4
12	Saltillo 2000	10	28	Colinas Del Sur	5	44	República Oriente	4
13	Amp. 23 de Noviembre	9	29	Guayulera	5	45	Santa Cristina	4
14	Las Teresitas	9	30	Miguel Hidalgo	5	46	Zapaliname	4
15	Amp. 26 de Marzo	8	31	Oceanía	5			
16	Amistad	7	32	Provivienda	5			

Tabla 2: Colonias consideradas en la instancia del caso de estudio

Figura 6: Análisis de la instancia real 46p380e50q40w (Fuente: Elaboración Propia)

Instancia	FO	Tiempo CPU (s)	Ruta
46p380e50q40w Matriz euclidiana Google Maps	511.469 km	68.199	R1: 0 1 47 R6: 0 7 18 43 30 26 47 R7: 0 8 27 9 35 47 R10: 0 11 36 6 25 5 47 R13: 0 14 2 39 38 47 R14: 0 15 29 10 45 16 28 19 47 R30: 0 31 46 21 33 37 3 22 47 R39: 0 40 12 32 23 4 44 47 R40: 0 41 13 17 20 24 42 34 47
46p380e50q40w Matriz de distancias NEVA	425.610 km	361.806	R2: 0 3 27 40 4 44 47 R13: 0 14 8 47 R16: 0 17 5 30 33 34 19 37 47 R19: 0 20 36 10 39 18 47 R21: 0 22 1 47 R30: 0 31 38 26 12 11 28 35 47 R31: 0 32 15 13 7 21 23 47 R42: 0 43 6 41 2 47 R45: 0 46 16 24 25 9 42 29 45 47
46p380e50q40w Matriz de tiempos NEVA	697.47 min	69.501	R0: 0 1 15 47 R2: 0 3 44 12 40 18 47 R19: 0 20 43 2 47 R23: 0 24 27 7 16 31 23 30 47 R24: 0 25 19 17 5 45 47 R27: 0 28 14 6 29 38 26 47 R34: 0 35 36 32 10 11 34 22 47 R36: 0 37 8 33 4 41 47 R45: 0 42 39 9 21 13 47

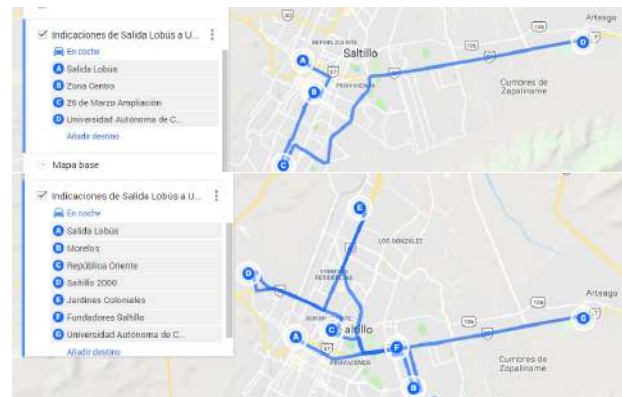


Figura 7: Rutas R0 y R2 generadas con la matriz de tiempos del software Neva

generan en GoogleMaps se observó que éstas tienen una distancia de recorrido muy larga, y el orden de visita de las paradas no resulta muy práctico, incluso si se reordenan a posteriori la distancia del recorrido sigue siendo larga.

Se concluye que el heurístico debe adaptarse para proporcionar soluciones de mejor calidad. Actualmente se emplea un proceso de selección de ruleta en el que a mayor ahorro, mayor probabilidad de ser elegido; sin embargo, aunque la probabilidad sea baja, por la aleatoriedad del proceso también es elegible una unión con baja probabilidad.

Como trabajo a futuro, se proponen cambiar la forma de seleccionar las paradas, restringir la longitud de las rutas, plantear alternativas para

Conclusiones

A diferencia de la ruta actual seguida por el Lobús, al establecer que cada estudiante sólo puede acceder a la parada ubicada en su colonia, obliga a que las rutas visiten dichas paradas, con ello se atiende la dispersidad de los estudiantes en el área metropolitana de Saltillo.

Al visualizar la secuencia de las rutas que se

generar la matriz de accesibilidad y desarrollar un sistema que permita la actualización y manipulación de los datos que conforman la instancia de una manera más eficiente.

Referencias

- [1] Clarke, G. y Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581.
- [2] Juárez-Peláez M. (2019). Implementación de un sistema heurístico para la generación de rutas para el servicio de transporte de Lobús de la UAdeC. Tesis de maestría. Universidad Autónoma de Coahuila, México.
- [3] Pacheco, J., Caballero, R., Laguna, M., y Molina, J. (2013). Bi-objective bus routing: An application to school buses in rural areas. *JORS*, 47:397–411.
- [4] Park, J. y Kim, B.-I. (2010). The school bus routing problem: A review. *European Journal of Operational Research*, 202(2):311–319.
- [5] Schittkat, P., Kinable, J., Sörensen, K., Sevaux, M., Spieksma, F., y Springael, J. (2013). A metaheuristic for the school bus routing problem with bus stop selection. *European Journal of Operational Research*, 229(2):518 – 528.
- [6] Soares de Siqueira, V., Jorge Ebrahim Lima e Silva, F., Nascimento da Silva, E., Vieira Santos da Silva, R., y Lisboa, M. (2016). Implementation of the metaheuristic GRASP applied to the school bus routing problem. *International Journal of e-Education, e-Business, e-Management and e-Learning*, 6:137–145.

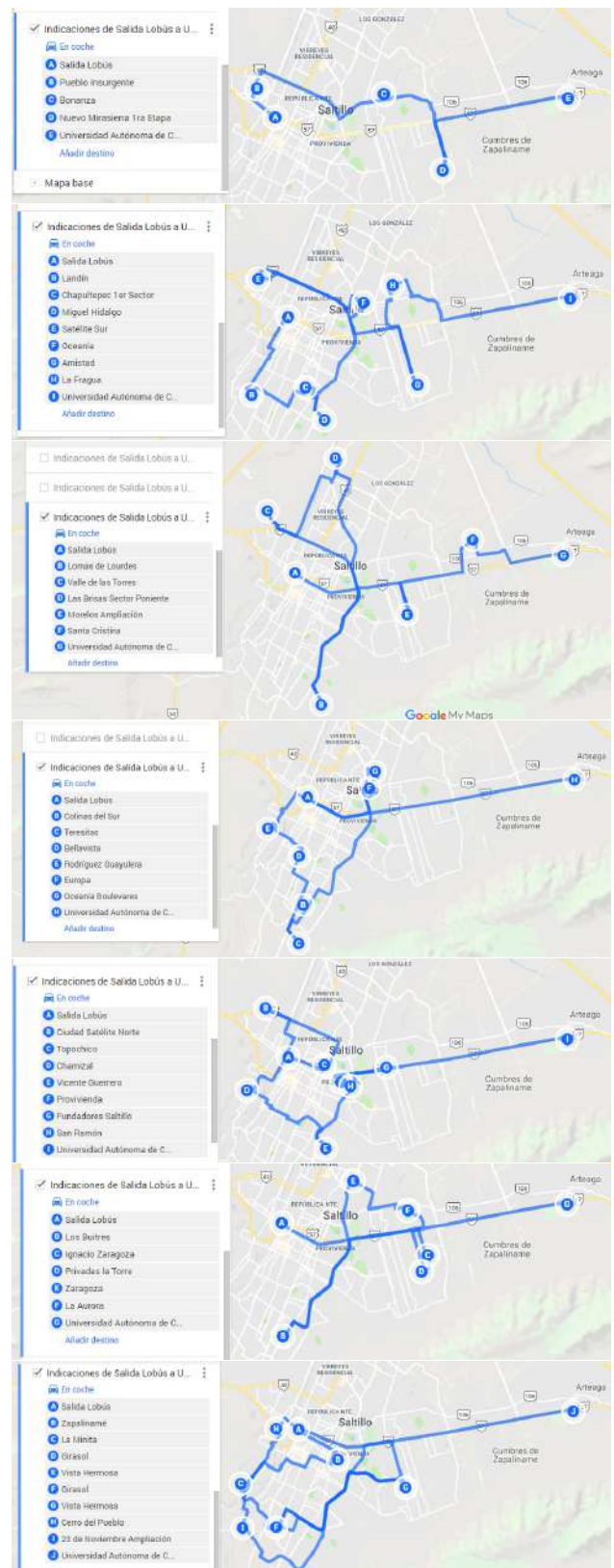


Figura 8: Rutas R19 a R45 generadas con la matriz de tiempos del software Neva

Próximos eventos

IX Encuentro Cuba – México Métodos Numéricos y Optimización

Del 16 al 20 de marzo del 2020 en La Habana, Cuba

E 2
M 0
N 2
O 0



Convocatoria

Teniendo en cuenta el éxito de los ocho encuentros celebrados en La Habana desde el año 2012, el Comité Organizador de EMNO 2020 ha decidido extender una cordial invitación a todos los científicos interesados en colaborar con instituciones cubanas y mexicanas en las disciplinas de Análisis Numérico y Optimización, con independencia de su país de procedencia.

El Encuentro está dirigido a investigadores, profesores y estudiantes interesados en los temas más actuales de investigación en las áreas ya mencionadas. Se realizarán dos tipos de actividades:

- Cursos introductorios
- Presentación de ponencias

Se dará prioridad a las ponencias que presenten el estado del arte en algún tema de investigación de las siguientes temáticas:

- Optimización
- Álgebra Lineal Numérica
- Interpolación y Aproximación
- Software para Cómputo Científico
- Solución Numérica de Ecuaciones Diferenciales
- Geometría Computacional

Los interesados en participar con una ponencia o impartir un curso deben elaborar un resumen de una página que contenga el nombre de los autores, instituciones, título de la ponencia o curso y palabras claves. Las memorias del evento serán publicadas en un número especial de la Revista *Ciencias Matemáticas*. El Comité Científico seleccionará los trabajos a publicar entre los presentados en el evento.

Fechas Importantes

- Entrega de resúmenes:
Hasta el 21 de Enero 2020
- Aceptación de trabajos:
A partir del 3 de Febrero 2020

Cuota de Inscripción

- Extranjeros: 80 CUC
- Cubanos: 150 CUP

Más información en

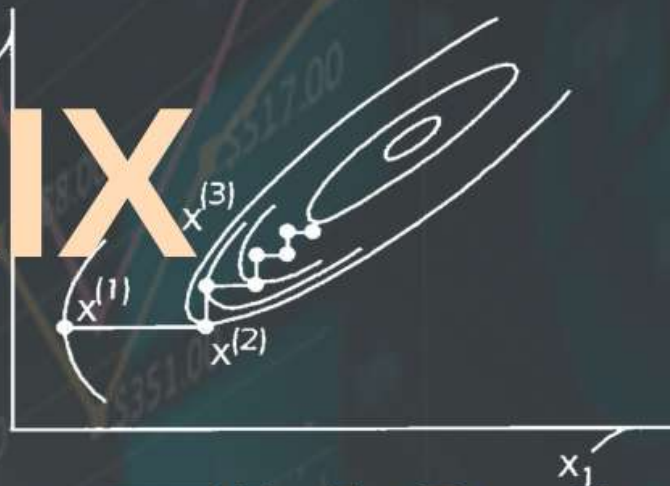
<http://tikhonov.fciencias.unam.mx/emno2020>

Comité Organizador

- Pablo Barrera Sánchez, UNAM, México
- Guilmer F. González Flores, UNAM, México
- Francisco Domínguez Mota, UMSNH, México
- Victoria Hernández Mederos, ICIMAF, Cuba
- Marta Lourdes Baguer, UH, Cuba



XXIX



ENOAN

Escuela Nacional de Optimización y Análisis Numérico

La sociedad Mexicana de Computación Científica y sus Aplicaciones, A. C. y la Facultad de Ciencias de la Universidad Nacional Autónoma de México, informan que, del 22 al 26 de junio de 2020, se llevará a cabo en las instalaciones de la Facultad de Ciencias, la XXIX Escuela Nacional de Optimización y Análisis Numérico.

En este evento se llevarán a cabo:

Cursos, Talleres, Conferencias y Presentación de Trabajos relacionados con la Ciencia de Datos, Cómputo Científico, Análisis Numérico, Optimización, Modelación y Simulación, Algoritmos y Aplicaciones.

La temática principal será El riesgo desde la perspectiva de la Matemática.

Informes

informes@smcca.org.mx

¿QUIERES PUBLICAR "ARTÍCULOS", "INFORMACIÓN SOBRE EVENTOS" O "NOTICIAS" EN EL BOLETÍN?

La **Sociedad Mexicana de Computación Científica y sus Aplicaciones**, A. C. (SMCCA), convoca a toda la comunidad interesada en el área de la Computación Científica y sus Aplicaciones, a presentar noticias, información sobre eventos, artículos de divulgación e investigación de alta calidad en el área, así como reportes de trabajos de tesis de nivel licenciatura y posgrado en Matemáticas Aplicadas.

Requisitos para las colaboraciones en el Boletín

I.- Artículos de Divulgación e Investigación

- a) Los artículos que se envíen para ser publicados deberán ser inéditos y no haber sido ni ser sometidos simultáneamente a la consideración en otras publicaciones.
- b) Los artículos **deben** presentarse en el formato de la plantilla LaTeX disponible en www.smcca.org.mx

II.- Información sobre eventos

- a) Los eventos cuya información quiera ser publicada para promocionarlos, deberán estar relacionados con el área de las Matemáticas Aplicadas y la Computación Científica.
- b) La información debe enviarse en un archivo de imagen: PDF, JPG, PNG.
- c) La información no deberá exceder a una cuartilla.
- d) Enviar la información con al menos 6 meses de anticipación a la fecha en que se llevaría a cabo.

III.- Noticias

- a) Las noticias a ser publicadas en el Boletín deben ser noticias relevantes de actividades de la SMCCA, Socios, Comunidad Científica interesada en las Matemáticas y Computación Aplicada
- b) La información de las noticias debe enviarse en un archivo de imagen: PDF, JPG, PNG.
- c) La información no deberá exceder a una cuartilla.

El material de colaboración: noticias, información eventos, artículos deberán ser dirigidos a la Dra. Irma Delia García Calvillo al correo electrónico de la SMCCA smcca@smcca.org.mx

Todos los artículos son sometidos a evaluación por especialistas de instituciones nacionales e internacionales y su publicación estará sujeta a la disponibilidad de espacio en cada número. Las demás colaboraciones se someterán a corrección de estilo y su publicación estará sujeta a la disponibilidad de espacio en cada número. Sólo se aceptará el material enviado que satisfaga todos los requisitos aquí señalados.

El envío de cualquier colaboración al Boletín implica no sólo la aceptación de lo establecido en este documento, sino también la autorización al Comité Editorial del Boletín de la SMCCA para incluirlo en su página electrónica, reimpresiones, colecciones y en cualquier otro medio que permita lograr una mayor y mejor difusión.

Sociedad Mexicana de Computación Científica y sus Aplicaciones

Consejo Directivo de la Sociedad Mexicana de Computación Científica y sus Aplicaciones 2018-2020

Dr. Justino Alavez Ramírez.	Presidente
Dra. Rina Betzabeth Ojeda Castañeda.	Vicepresidenta
Dra. Ma. Luisa Sandoval Solís.	Secretaria de actas y acuerdos
Dr. Jorge López López.	Tesorero
Dr. Pedro Flores Pérez.	Secretario general
Dr. Miguel Ángel Uh Zapata.	Vocal
Lic. Gerardo Tinoco Guerrero.	Vocal

La Sociedad Mexicana de Computación Científica y sus Aplicaciones fue fundada el 16 de Mayo de 2013, para realizar actividades de investigación científica o tecnológica inscritas en el RENIECyT (Registro Nacional de Instituciones y Empresas Científicas y Tecnológicas), prestadas únicamente a los socios y asociados. Es una Asociación sin fines de lucro. Entre sus tareas fundamentales destacan: Conjuntar acciones e intereses comunes en los investigadores, profesores y estudiantes interesados en la Computación Científica y sus Aplicaciones, con el fin de fomentar la investigación de calidad, promover la actualización y el perfeccionamiento para el desarrollo científico, tecnológico y social; promover la creación, organización, acumulación y difusión de conocimientos referidos a la Computación Científica y sus Aplicaciones; promover la formación e interacción de redes y grupos de trabajo orientados hacia el desarrollo disciplinar, interdisciplinar y temático de la investigación; fomentar el desarrollo de la investigación sobre la Computación Científica y sus Aplicaciones en la República Mexicana; contribuir al mejoramiento de la enseñanza de la Computación Científica y sus Aplicaciones en la República Mexicana; promover y organizar toda clase de encuentros y eventos académicos orientados a la comunicación y discusión entre investigadores y profesores, así como también a la difusión del conocimiento hacia sectores interesados en la integración de la Computación Científica y sus Aplicaciones en los problemas de su sector.

Correo electrónico: smcca@smcca.org.mx
<http://www.smcca.org.mx>

