



UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

CENTRO DE CIENCIAS BÁSICAS

Diseño de un Modelo de Optimización de
Colonias de Hormigas Aplicado al Problema del Clique Máximo

TESIS

QUE PARA OBTENER EL GRADO DE
DOCTOR EN CIENCIAS DE LA COMPUTACIÓN
INTELIGENCIA ARTIFICIAL

Presenta:

MCC JULIO CESAR PONCE GALLEGOS

Tutor:

DR. FELIPE PADILLA DÍAZ

Comité Tutorial:

DRA. EUNICE ESTHER PONCE DE LEÓN SENTÍ
DR. CARLOS ALBERTO OCHOA ORTIZ ZEZZATTI
DR. ALEJANDRO PADILLA DÍAZ
DRA. ELVA DÍAZ DÍAZ

Aguascalientes, Ags. México. Mayo de 2010

AGRADECIMIENTOS



Quiero aprovechar este espacio para agradecer al termino de este proyecto de mi vida el cual no habría sido alcanzado sin el apoyo incondicional, consejos y palabras de aliento de personas tan importantes en mi vida que hicieron que fuera posible alcanzar ésta meta, como son mi familia y amigos.

Agradezco a Dios por todos los logros que he alcanzado durante el tiempo que duró este proyecto, en el cual he vivido tanto momentos felices y algunos difíciles en compañía de mis seres queridos, en los cuales me ha dado la fortaleza necesaria para enfrentar todas las adversidades en mi vida sin darme nunca por vencido, quiero agradecerle por la estabilidad de la enfermedad de mi hermano Carlos.

A mi madre Ma. Felicitas Gallegos López que es la persona que más amo en el mundo y es un ejemplo a seguir, además me ha apoyado siempre.

A mi padre José Luis Ponce Muñoz a quien en esta ocasión quiero decirle que lo amo, y quiero agradecerle su apoyo incondicional.

A mis hermanos José Luis, Carlos Alberto y Daniel quienes están al pendiente de mí, y están dispuestos a apoyarme en todo lo que sea necesario, a quienes les deseo que estén bien y que puedan alcanzar sus metas y proyectos que se han puesto en la vida.

A mi hermana Cynthia a quien estimo mucho aunque no se lo demuestro, y quisiera agradecerle por tolerarme y apoyar a mi hermano y alentarle a estudiar, además quiero agradecer su paciencia y tolerancia.

A mi sobrino Ángel quien llena de alegría mi vida y a mi familia quiero que sepa que lo quiero mucho, y que deseo lo mejor para él, y espero que cuando sea capaz de entender esto vea que fue una persona amada desde niño, y espero que el ver lo que he logrado, sea un estímulo par que se realice en la parte profesional.

AGRADECIMIENTOS



Al Dr. Felipe Padilla Díaz quien ha depositado en mí una gran confianza, por lo cual le agradezco su apoyo y comprensión en general.

A la Dra. Eunice Esther Ponce de León Sentí que es la persona que hizo que me interesara en el área de la Investigación por la inteligencia artificial y que siempre me ha apoyado.

A la Dra. Elva Díaz Díaz quien es un ejemplo a seguir, ya que es una persona por la cual siento gran admiración por su forma de ser y de vivir la vida.

Al Dr. Alejandro Padilla Díaz por brindarme su amistad y su apoyo en este trabajo al colaborar como asesor, quiero aprovechar para desearle suerte en la vida y que se cumplan sus deseos, gracias por sus consejos.

Al Dr. Carlos Alberto Ochoa Ortiz Zezzatti por brindarme su amistad y apoyo, así mismo seguir fomentado en mí el gusto por la investigación, al confiar en mí y tomar todo el tiempo en cuenta mis críticas y pedirme siempre mi opinión de sus trabajos.

Al Dr. Francisco Javier Alvares Rodríguez por su apoyo brindado para poder publicar y asistir a congresos ayudando e impulsar de alguna manera el que yo realizara más publicaciones.

A Fatima Sayuri Quezada Aguilera por su amistad y su apoyo todo este tiempo que hemos convivido juntos, los años de la carrera y ahora en el posgrado, quiero decirle que es una persona especial en mi vida y aprovecho para desearle lo mejor.

A Miguel Ángel Meza de Luna por ser una persona que me ha enseñado mucho en la vida y he compartido diferentes etapas de mi vida, el tenerlo como maestro en el bachillerato y en la licenciatura, y posteriormente como compañero de trabajo y de estudio en el posgrado, gracias por brindarme su amistad.

AGRADECIMIENTOS



A todos los Maestros por brindarme sus conocimientos y haber sido parte de mi formación en esta etapa.

A mis compañeros y amigos que he conocido en este proyecto y que siempre estuvieron dispuestos a ayudarme cuando lo necesite, Fernando Flores, Francisco Ornelas, Christian Correa, Laura Rodríguez, Ricardo Mendoza, Miguel Vargas, Alberto Hernandez, Arturo Hernández, Aurora Torres, Dolores Torres.

A mis familiares Lidia Gallegos, Claudio Zamarripa, Andrea Muñoz, así como mis abuelos Ciriaco Ponce, Felicitas López y Juan Gallegos a quienes Dios los ha llamado para estar a su lado.

A mi primo Francisco Oliva ya que hemos vivido muchas cosas juntos por ser mi compañero y quien también ha demostrado su interés por aprender cosas nuevas, ahora en un área nueva como la medicina, con quien quisiera colaborar algún día para hacer alguna investigación.

A todas las personas que han estado conmigo, que me han acompañado en el proceso y que he dejado de nombrar, les pido una disculpa por la omisión, sin embargo quiero que sepan que tengo un espacio para todos ustedes en mi corazón.

TESIS TESIS TESIS TESIS TESIS

DEDICATORIA

A mis padres:

Ma. Felicitas y José Luis. Por ser las personas que me dieron el regalo más preciado que es la vida y porque además de esto, se preocupan siempre por todos sus hijos y están al pendiente de lo que nos pasa, a si mismo por ser un ejemplo de vida para mí y para mis hermanos, porque me han enseñado a luchar por lo que creo importante, los que han estado siempre allí para hacerme sentir respaldado, porque me dieron las herramientas para forjar mi propia vida.

CARTAS DE LIBERACIÓN





Centro de Ciencias Básicas

**M. en C. JULIO CÉSAR PONCE GALLEGOS
PASANTE DEL DOCTORADO EN CIENCIAS
DE LA COMPUTACIÓN
P R E S E N T E .**

Estimado (a) Alumno (a) Ponce:

Por medio de este conducto me permito comunicar a Usted que habiendo recibido los votos aprobatorios de los revisores de su trabajo de tesis titulado: **“Diseño de un Modelo de Optimización de Colonias de Hormigas Aplicado al Problema del Clique Máximo”**, hago de su conocimiento que puede imprimir dicho documento y continuar con los trámites para la presentación de su examen de grado.

Sin otro particular me permito saludarle muy afectuosamente.

ATENTAMENTE
Aguascalientes, Ags., 27 de mayo de 2010
“LUMEN PROFERRE”
EL DECANO

pa

DR. FRANCISCO JAVIER ÁLVAREZ RODRÍGUEZ



c.c.p.- Archivo





Aguascalientes, Ags. Mayo de 2010

Dr. Francisco Javier Álvarez Rodríguez
Decano del Centro de Ciencias Básicas
Universidad Autónoma de Aguascalientes
P R E S E N T E.

Por este conducto, hago de su conocimiento que el MCC. Julio Cesar Ponce Gallegos, egresado del doctorado en Ciencias de la Computación del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: “Diseño de un Modelo de Optimización de Colonias de Hormigas Aplicado al Problema del Clique Máximo”, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

Atentamente:



Dr. Felipe Padilla Díaz
Director de Tesis.



UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

Aguascalientes, Ags. Mayo de 2010

Dr. Francisco Javier Álvarez Rodríguez

Decano del Centro de Ciencias Básicas

Universidad Autónoma de Aguascalientes

PRESENTE.

Por este conducto, hago de su conocimiento que el **MCC. Julio Cesar Ponce Gallegos**, egresado del **doctorado en Ciencias de la Computación** del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: **“Diseño de un Modelo de Optimización de Colonias de Hormigas Aplicado al Problema del Clique Máximo”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

Atentamente:

A handwritten signature in black ink, appearing to read 'Eunice Ponce de León Sentí', written over a horizontal line.

Dra. Eunice Esther Ponce de León Sentí

Asesor de tesis



Aguascalientes, Ags. Mayo de 2010

Dr. Francisco Javier Álvarez Rodríguez

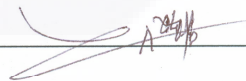
Decano del Centro de Ciencias Básicas

Universidad Autónoma de Aguascalientes

PRESENTE.

Por este conducto, hago de su conocimiento que el **MCC. Julio Cesar Ponce Gallegos**, egresado del **doctorado en Ciencias de la Computación** del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: **“Diseño de un Modelo de Optimización de Colonias de Hormigas Aplicado al Problema del Clique Máximo”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

Atentamente:



Dr. Carlos Alberto Ochoa Ortiz Zezzatti

Asesor de tesis



Aguascalientes, Ags. Mayo de 2010

Dr. Francisco Javier Álvarez Rodríguez

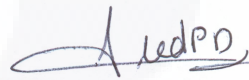
Universidad Autónoma de Aguascalientes

Decano del Centro de Ciencias Básicas

P R E S E N T E.

Por este conducto, hago de su conocimiento que el **MCC. Julio Cesar Ponce Gallegos**, egresado del **doctorado en Ciencias de la Computación** del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: **“Diseño de un Modelo de Optimización de Colonias de Hormigas Aplicado al Problema del Clique Máximo”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

Atentamente:



Dr. Alejandro Padilla Díaz

Asesor de tesis



UNIVERSIDAD AUTÓNOMA
DE AGUASCALIENTES

Aguascalientes, Ags. Mayo de 2010

Dr. Francisco Javier Álvarez Rodríguez

Decano del Centro de Ciencias Básicas

Universidad Autónoma de Aguascalientes

P R E S E N T E.

Por este conducto, hago de su conocimiento que el MCC. **Julio Cesar Ponce Gallegos**, egresado del **doctorado en Ciencias de la Computación** del Centro de Ciencias Básicas de la Universidad Autónoma de Aguascalientes, ha integrado satisfactoriamente el documento de tesis titulado: **“Diseño de un Modelo de Optimización de Colonias de Hormigas Aplicado al Problema del Clique Máximo”**, por lo que doy mi voto aprobatorio para que continúe con los trámites para presentar el examen de grado reglamentario.

Atentamente:

A handwritten signature in black ink, appearing to read 'Elva Díaz Díaz', written over a horizontal line.

Dra. Elva Díaz Díaz

Asesor de tesis

RESUMEN



Esta investigación consiste en la propuesta de un modelo basado en la metaheurística de Optimización con Colonias de Hormigas aplicado al problema del clique máximo, con el desarrollo de este modelo se pretende explorar de una manera más eficiente el espacio de búsqueda para encontrar mejores resultados, al encontrar un equilibrio entre la exploración y la explotación.

En el trabajo se describen 5 contribuciones principales:

- (1) Una tabla en la cual se muestran los principales problemas que se han resueltos con colonias de hormigas, así como los principales investigadores de cada uno de estos problemas.
- (2) El desarrollo de diversos modelos de optimización de colonias de hormigas implementados, los cuales fueron desarrollados para analizar sus características y comportamiento al trabajar y de esta manera poder diseñar un mejor algoritmo, los modelos implementados fueron todos para resolver el problema del clique máximo.
- (3) Se muestra un estudio empírico de estos algoritmos desarrollados al tratar de analizar cómo trabajan para obtener sus bondades al momento de resolver los problemas utilizando 29 de los benchmarks de la DIMACS.
- (4) Se muestra un estudio estadístico para determinar los mejores valores en los parámetros relacionados con el número de hormigas, el número de iteraciones y la forma de colocar inicialmente las hormigas.
- (5) Se realizó un análisis de la complejidad de los algoritmos.

ÍNDICE DE CONTENIDO

| CONTENIDO | PÁGINA |
|---|--------|
| Capítulo 1. INTRODUCCIÓN _____ | 1 |
| 1.1. Introducción _____ | 1 |
| 1.2 Descripción General del Problema _____ | 1 |
| 1.3 Necesidad de la Investigación _____ | 6 |
| 1.4 Planteamiento del Problema _____ | 7 |
| 1.5 Relevancia y Justificación _____ | 7 |
| 1.6 Áreas Relacionadas y Herramientas _____ | 9 |
| 1.7 Enfoque de la Investigación _____ | 10 |
| 1.8 Objetivos de la Tesis _____ | 11 |
| 1.9 Contribuciones de la Tesis _____ | 12 |
| 1.10 Descripción de capítulos _____ | 20 |
| 1.11 Conclusiones _____ | 22 |
| Capítulo 2. MARCO TEÓRICO DE COLONIAS DE HORMIGAS _____ | 23 |
| 2. 1 Introducción _____ | 23 |
| 2. 2 Descripción de Teorías Bases _____ | 23 |
| 2. 2.1 Metaheurísticas _____ | 23 |
| 2. 2.1.1 Propiedades de las meta-heurísticas _____ | 23 |

ÍNDICE DE CONTENIDO

| | |
|--|----|
| 2. 2.2 Optimización con Colonias de Hormigas (OCH) | 25 |
| 2. 2.2.1 La Estructura General de un Algoritmo de Optimización de Colonias de Hormigas | 26 |
| 2. 2.2.2 Problemas Tratados con Algoritmos de Optimización con Colonias de Hormigas | 31 |
| 2.3 Colonias de Hormigas Colectivas | 40 |
| 2.4 Migración en Colonias de Hormigas | 41 |
| 2.5 Problemas de Optimización | 42 |
| 2.6 Conclusiones | 43 |
| Capítulo 3. ESTADO DEL ARTE DEL PROBLEMA DEL CLIQUE MÁXIMO | 45 |
| 3.1 Introducción | 45 |
| 3.2 Problema del clique Máximo | 45 |
| 3.3 Formulación del Problema | 47 |
| 3.3.1 Complejidad Computacional | 47 |
| 3.3.2 Límites y Estimaciones | 48 |
| 3.4 Algoritmos Desarrollados para Resolver el Clique Máximo | 49 |
| 3.4.1 Algoritmos Enumerativos | 49 |
| 3.4.2 Algoritmos Exactos para el Caso No Pesados | 50 |
| 3.4.3 Algoritmos Exactos para el Caso Pesados | 51 |
| 3.4.4 Heurísticas | 52 |
| 3.4.4.1 Secuencial Voraz (Sequential Greedy) | 52 |

ÍNDICE DE CONTENIDO

| | | |
|----------|--|----|
| 3.4.4.2 | Búsqueda Local (Local Search) | 52 |
| 3.4.4.3 | Búsqueda Avanzada (Advanced Search) | 53 |
| 3.4.4.4 | Recosido Simulado (Simulated Annealing) | 53 |
| 3.4.4.5 | Redes Neuronales (Neural Networks) | 54 |
| 3.4.4.6 | Algoritmos Genéticos (Genetic Algorithms) | 54 |
| 3.4.4.7 | DNA Computing | 55 |
| 3.4.4.8 | Búsqueda Tabú (Tabu Search) | 55 |
| 3.4.4.9 | Búsqueda Dispersa (Scatter Search) | 56 |
| 3.4.4.10 | Optimización con Colonia de Hormigas (ACO) | 56 |
| 3.4.4.11 | Algoritmos de Estimación de la Distribución (EDA) | 57 |
| 3.4.4.12 | Heurísticas Basadas en un Dominio Continuo (Continuous-based Heuristics) | 58 |
| 3.4.4.13 | Otros | 58 |
| 3.5 | Aplicaciones | 59 |
| 3.6 | Descripción y Análisis de Algoritmos de colonias de Hormigas desarrollados para resolver el problema del clique máximo | 60 |
| 3.6.1 | Algoritmo de Fenet y Solnon | 61 |
| 3.6.2 | Algoritmo RPACOMCP | 66 |
| 3.6.3 | Algoritmo ASMC | 68 |
| 3.6.4 | ABOMC | 68 |
| 3.7 | Conclusiones | 68 |

ÍNDICE DE CONTENIDO

| | |
|--|-----|
| Capítulo 4 HORMIGAS NATURALES | 70 |
| 4.1 Introducción | 70 |
| 4.2 Hormigas Naturales | 70 |
| 4.3 Conclusiones | 81 |
| Capítulo 5 APRENDIZAJE | 83 |
| 5.1 Introducción | 83 |
| 5.2 Que es el Aprendizaje | 83 |
| 5.3 Tipos de Aprendizaje | 83 |
| 5.4 Aprendizaje por Refuerzo | 85 |
| 5.4.1 Elementos del Aprendizaje por Refuerzo | 88 |
| 5.4.2 Aprendizaje del refuerzo de Habilidades | 90 |
| 5.4.3 El Problema del aprendizaje por refuerzo | 93 |
| 5.4.4 Métodos de Aprendizaje por Refuerzo | 94 |
| 5.5 Conclusiones | 95 |
| Capítulo 6. Minería de Datos | 96 |
| 6.1 Introducción | 96 |
| 6.2 Minería de Datos | 96 |
| 6.3 Closterización | 99 |
| 6.4 Closterización | 100 |
| 6.5 Conclusiones | 100 |

ÍNDICE DE CONTENIDO

| | |
|---|-----|
| Capítulo 7. Optimización con Colonia de Hormigas Aplicado a Minería de Datos | 101 |
| 7.1 Introducción | 101 |
| 7.2 Algoritmos de Colonia de Hormigas aplicados a Minería de Datos | 101 |
| 7.2. 1 Algoritmo ANT-LGP | 102 |
| 7.2. 2 Algoritmo ANT-BASED Clustering | 102 |
| 7.2. 3 Algoritmo AntClass | 103 |
| 7.2. 4 Algoritmo Ant-Based Clustering con Neural Gas | 104 |
| 7.2. 5 Algoritmo ANT-Miner | 108 |
| 7.2. 6 Algoritmo CLICK | 111 |
| 7.3 Conclusiones | 112 |
| Capítulo 8. Desarrollo y Validación | 113 |
| 8.1 Introducción | 113 |
| 8.2 Algoritmos Implementados | 113 |
| 8. 2.1 Algoritmo del ANT CLIQUE HORMIGA | 115 |
| 8.2.2 Algoritmo de Colonia de Hormigas con Selección por Criterio de Metrópoli aplicado al Problema del Clique Máximo | 116 |
| 8. 2.3 Proceso de solución | 117 |
| 8.2.4 Algoritmo para el Problema del Clique Máximo con Expansión | 128 |
| 8.3 Complejidad de los algoritmos implementados | 131 |
| 8.3.1 Complejidad del Algoritmo ACH | 139 |

ÍNDICE DE CONTENIDO

| | | |
|---|--|-----|
| 8.3.2 | Complejidad del Algoritmo MAC | 140 |
| 8.3.3 | Complejidad del Algoritmo Exp-ACO | 140 |
| 8.4 | Conclusiones | 141 |
| Capítulo 9. Reporte y Discusión de Resultados | | 142 |
| 9.1 | Introducción | 142 |
| 9.2 | Resultados Obtenidos | 142 |
| 9.3 | Conclusiones | 153 |
| Capítulo 10. CONCLUSIONES | | 154 |
| 10.1 | Conclusiones | 154 |
| 10.2 | Trabajo Futuro | 155 |
| Anexo 1 | Publicaciones | 157 |
| Anexo 2 | Portada del Libro de Minería de Datos | 171 |
| Anexo 3 | Artículo Relevante | 172 |
| Anexo 4 | Trabajo Realizado | 181 |
| Anexo 5 | Ponencias | 183 |
| Anexo 6 | Pantallas de los Sistemas | 189 |
| Anexo 7 | Miembro de Comité Revisor en Congresos | 194 |
| Anexo 8 | Bibliografía de los problemas tratados con ACO | 196 |
| Anexo 9 | Análisis Estadístico | 206 |
| Glosario | | 222 |

TESIS TESIS TESIS TESIS TESIS

ÍNDICE DE CONTENIDO

Bibliografía _____ 224



ÍNDICE DE FIGURAS

| CONTENIDO | PÁGINA |
|---|--------|
| 1.1 Hormigas Naturales Cortando Hojas _____ | 5 |
| 1.2 Hormiga guiada por el Rastro de Feromonas _____ | 6 |
| 2.1 Marco Dorigo _____ | 26 |
| 2.2 Experimento realizado para demostrar que las hormigas siguen caminos cortos ____ | 27 |
| 3.1 Ejemplo de un clique máximo _____ | 47 |
| 4.1 Hormiga natural _____ | 71 |
| 4.2 Hormiga Jardinera _____ | 78 |
| 4.3 Libro La Inteligencia de las Hormigas _____ | 79 |
| 5.1 Componentes del aprendizaje por Refuerzo _____ | 91 |
| 5.2 Interacción del ambiente y el agente en el aprendizaje por refuerzo en A y en B un agente reactivo simple _____ | 92 |
| 6.1 Proceso de Minería de Datos _____ | 98 |
| 7.1. Hormigas escavando _____ | 102 |
| 8.1. Modelo general de un algoritmo de colonia de hormigas _____ | 115 |
| 8.2. Instancia del Grafo de prueba _____ | 118 |
| 8.3. Colocación de la hormiga en la posición inicial _____ | 122 |
| 8.4. Nodos candidatos de 3 en el grafo de prueba _____ | 123 |
| 8.5. Movimiento hacia el nodo 5 en el grafo _____ | 124 |

ÍNDICE DE FIGURAS

| | |
|--|-----|
| 8.6. Movimiento de la hormiga hacia el nodo 2 _____ | 126 |
| 8.7. Movimiento de la hormiga al nodo 8 _____ | 127 |
| 8.8. Modelo del un algoritmo de colonia de hormigas con expansión del clique _____ | 129 |



ÍNDICE DE TABLAS



| CONTENIDO | PÁGINA |
|---|--------|
| 2.1 Algoritmo de Optimización con Colonias de Hormigas General _____ | 29 |
| 2.2 Procedimiento de inicialización de las feromonas _____ | 30 |
| 2.3 Procedimiento donde se colocan las hormigas en su posición inicial _____ | 30 |
| 2.4 Procedimiento para general una solución _____ | 31 |
| 2.5 Procedimiento para mover una hormiga de un lugar a otro _____ | 31 |
| 2.6 Clasificación de los trabajos realizados con algoritmos de colonias de hormigas según el tipo de problema _____ | 33 |
| 3.1 Procedimiento general del algoritmo de optimización con colonias de hormigas para el problema del clique máximo _____ | 62 |
| 3.2 Algoritmo para construir un clique con una hormiga _____ | 63 |
| 3.3 Algoritmo de colonia de hormigas reportado por Solnon y Fenet en el 2004 _____ | 64 |
| 3.4 Algoritmo para construir un clique con una hormiga _____ | 67 |
| 3.5.- Algoritmo de optimización con colonia de hormigas basado en prohibición reactiva para el problema del clique máximo _____ | 68 |
| 4.1 Especies de hormigas naturales _____ | 76 |
| 7.1 Algoritmo AntClass _____ | 105 |
| 8.1 Algoritmo Ant Clique base _____ | 117 |

ÍNDICE DE TABLAS

| | |
|--|-----|
| 8.2 Algoritmo para construir un Clique _____ | 117 |
| 8.3 Vector de Feromonas _____ | 119 |
| 8.4 Matriz de Feromonas _____ | 120 |
| 8.5 Inicialización de la lista Tabú _____ | 120 |
| 8.6 Algoritmo para construir un Clique _____ | 121 |
| 8.7 Introducción del elemento 5 a la lista Tabú _____ | 125 |
| 8.8 Introducción del elemento 2 a la lista Tabú _____ | 126 |
| 8.9 Introducción del elemento 8 a la lista Tabú _____ | 128 |
| 8.9 Algoritmo con expansion _____ | 130 |
| 9.1 Diseño de experimento _____ | 144 |
| 9.2 Resultados del Diseño de Experimento realizado con el benchmark C500.9 _____ | 145 |
| 9.3 Resultados obtenidos en los 29 “benchmark” _____ | 146 |
| 9.4 Resultados del algoritmo en los 20 benchmarks _____ | 148 |
| 9.5 Diseño de experimento para análisis de los parámetros hormigas, iteraciones y colocación _____ | 149 |
| 9.6 Resultados del análisis de la varianza _____ | 150 |
| 9.7 Comparación con el algoritmo ABOMC _____ | 151 |

ÍNDICE DE ECUACIONES

| CONTENIDO | PÁGINA |
|-------------|--------|
| EC. 1 _____ | 28 |
| EC. 2 _____ | 28 |
| EC. 3 _____ | 65 |
| EC. 4 _____ | 65 |
| EC. 5 _____ | 66 |
| EC. 6 _____ | 66 |

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO I.- INTRODUCCIÓN

1.1 Introducción

En este capítulo se plantea el problema que se quiere resolver así como la relevancia que tiene el desarrollar este tipo de algoritmos en la resolución de problemas, al mismo tiempo se describe el enfoque de la investigación el cual es de simulación ya que se realizará una herramienta con el modelo propuesto, al cual se le harán pruebas a través de algunos juegos de datos ó benchmarks como se conoce en inglés, dichas pruebas se realizaron siguiendo una metodología para validar los resultados, por otra parte se describe el contenido de cada uno de los capítulos de este trabajo.

En las secciones de la 1.2 a la 1.8 se describe el problema que se ha resuelto a través de la investigación, necesidad y justificación del problema de investigación, así como los objetivos, metas, hipótesis y metodología de investigación. También se describen las contribuciones científicas de la presente tesis, así como los artículos donde estas se han publicado.

1.2 Descripción General Del Problema

El diseño de algoritmos cada vez más eficientes para resolver problemas complejos de optimización y búsqueda, esto ha sido tradicionalmente uno de los aspectos más importantes de la investigación en el área de las ciencias computacionales. El objetivo perseguido en este campo es fundamentalmente el desarrollo de nuevos métodos capaces de resolver los problemas complejos con el menor esfuerzo computacional posible, mejorando así los resultados obtenidos por los algoritmos existentes. En consecuencia, esto no sólo permite afrontar problemas actuales de forma más eficiente, sino también táreas olvidadas en el pasado debido a su alto costo computacional. En este contexto, la actividad investigadora tanto en algoritmos exactos como en heurísticos *ad hoc* y metaheurísticos para resolver problemas complejos de optimización está creciendo de forma evidente en estos días.

CAPÍTULO I.- INTRODUCCIÓN

La razón es que continuamente se están afrontando nuevos problemas de optimización en diferentes ámbitos de nuestro mundo real, mientras que al mismo tiempo cada vez se dispone de mejores recursos computacionales, como nuevos tipos de equipos, redes, y entornos como Internet.

La principal ventaja de la utilización de algoritmos exactos es que garantizan encontrar el óptimo global de cualquier problema, pero tienen el grave inconveniente de que en problemas reales (que suelen ser NP-duros en la mayoría de los casos) su tiempo de ejecución y/o los requisitos de memoria crecen de forma exponencial con el tamaño del problema. En cambio, los algoritmos heurísticos *ad hoc* son normalmente bastante rápidos, pero la calidad de las soluciones encontradas está habitualmente lejos de ser óptima. Otro inconveniente de estos heurísticos *ad hoc* es que no son fáciles de definir en determinados problemas. Las metaheurísticas ofrecen un equilibrio adecuado entre ambos extremos: son métodos genéricos que ofrecen soluciones de buena calidad (el óptimo global en muchos casos) en un tiempo razonable. La resolución de estos problemas ha jugado, sin lugar a dudas un papel muy destacado en el desarrollo y utilización de este tipo de sistemas. No obstante, a medida que se han ido haciendo más populares y su penetración en el mercado es mayor, el tamaño de los sistemas de telecomunicaciones ha ido creciendo, actualmente los problemas que se plantean tienen una dimensión tan elevada que los hacen inabordables con técnicas exactas. Los algoritmos metaheurísticos son una de las mejores opciones en este contexto, ya que son capaces de encontrar soluciones de calidad en tiempos aceptables.

Los algoritmos metaheurísticos se consideran una subclase de los algoritmos heurísticos. Por ese motivo se utiliza el término “heurístico *ad hoc*” para referirse a los algoritmos aproximados pensados específicamente para un problema concreto.e

Este es el contexto donde se enmarca esta tesis doctoral. Se propuso aplicar algoritmos de colonias de hormigas a problemas de optimización, analizando distintas posibilidades para sacar el máximo partido a dicha técnica y ofrecer así soluciones al problema del clique máximo. Debido a las características de estos problemas, se han desarrollado algoritmos

CAPÍTULO I.- INTRODUCCIÓN

avanzados que utilizan mecanismos emergentes y/o sistemas híbridos. El primer caso está motivado por la naturaleza del comportamiento de las hormigas, en los que existen varios comportamientos que pueden ser simulados. El segundo caso es dado por la incorporación de concepto de otros algoritmos heurísticos, lo que permite no sólo reducir el tiempo de ejecución de los algoritmos, sino también obtener modelos de búsqueda que permitan resolver estos problemas de forma más efectiva.

En la actualidad existe una gran cantidad de algoritmos heurísticos y meta-heurísticos como son, algoritmos genéticos, recocido simulado, búsqueda tabú, búsqueda dispersa, algoritmos meméticos, cúmulo de partículas, sistema inmune artificial, algoritmos de estimación de la distribución, entre otras incluyendo a los algoritmos de colonias de hormigas (Glover y Kochenberger, 2003).

Los algoritmos de colonias de hormigas surgieron como resultado de la tesis de investigación de Marco Dorigo en 1992, aplicando esta técnica en sus orígenes al problema del agente viajero, el cual abreviaremos como PAV (en inglés Traveling Salesman Problem (TSP)) y el problema de asignación cuadrática (PAC) (en inglés Quadratic Assingment Problem (QAP)). Los Algoritmos de colonias de hormigas han demostrado obtener buenos resultados, sin embargo por ser algoritmos constructivos son más lentos que algunos de los mencionados, por lo cual es necesario buscar un punto intermedio entre la exploración y la explotación, así mismo es importante el hacer un estudio sobre sus propiedades, así como el desarrollo de estos algoritmos para resolver problemas reales, actualmente hay un gran interés en el desarrollo de algoritmos de colonias de hormigas para realizar tareas específicas en el área de la minería de datos, ya que nos permite modelar comportamientos ó hacer predicciones.

La Computación Evolutiva (CE) actualmente abarca un sistema de algoritmos llamados los “algoritmos evolutivos”, estos emulan los procesos evolutivos basados en la “supervivencia” del principio del más apto en selección natural (teoría neoDarwinista), que se aplica a una población de soluciones. La CE se implementa generalmente en los

CAPÍTULO I.- INTRODUCCIÓN

problemas donde las técnicas de optimización clásicas no pueden ser aplicadas, así como en los problemas del mundo real de la gran complejidad y de la alta dimensionalidad en los cuales probablemente ni la función objetivo ni las restricciones son diferenciables. Debido a su simplicidad conceptual y de implementación, estos algoritmos han sido ampliamente aplicados en problemas donde los métodos de investigación de operaciones no han podido proveer soluciones competitivas (Mezura, 2004).

Cada vez se exige más la solución de problemas complejos del mundo real. Ejemplo de ello son los problemas de optimización. A mediados de los años noventa, el área de la Computación Evolutiva se extendió con el surgimiento de nuevos modelos inspirados en distintos comportamientos encontrados en la naturaleza, diferentes a la evolución de las especies. Tales algoritmos han sido agrupados en el área conocida como Inteligencia Colectiva (Swarm Intelligence). En ella se distinguen dos paradigmas primordiales: La Optimización Mediante Cúmulos de Partículas (Particle Swarm Optimization ó PSO) y los algoritmos de Optimización con Colonias de Hormigas (Ant Colony Optimization ó ACO) (Kennedy y Eberhart, 2001).

Es importante mencionar que se han propuesto nuevos algoritmos basados en comportamientos sociales de grupos de seres vivos, que en conjunto pueden resolver problemáticas muy complejas. Ejemplo de ello es la Colonia Artificial de Abejas, Optimización Mediante Cúmulos de Luciérnagas, Optimización Mediante el Forrajeo de Bacterias, etc. (Mezura, 2008). Como se puede ver, el Cómputo Bio-Inspirado y los algoritmos Bio-Inspirados son un área joven dentro de la Inteligencia Artificial, la cual tiene mucho que ofrecer para la investigación básica y aplicada (Mezura, 2008).

Los algoritmos de colonias de hormigas son una meta-heurística bio-inspirada en el comportamiento estructurado de las colonias de hormigas naturales, donde individuos muy simples de una colonia (ver figura.- 1.1) se comunican entre si por medio de una sustancia química denominada feromonas, estableciendo el camino más adecuado entre el hormiguero y una fuente de alimentación, este comportamiento es muy interesante debido a

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO I.- INTRODUCCIÓN

que son capaces de realizar estos caminos a través del uso de la feromona que es la forma de comunicación entre ellas (Dorigo and Colorni, 1996), el trabajo propuesto es importante ya que el propósito de esta tesis es el desarrollar una herramienta más robusta basada en un algoritmo de optimización de colonias de hormigas, la cual sea capaz de resolver el problema del clique máximo.

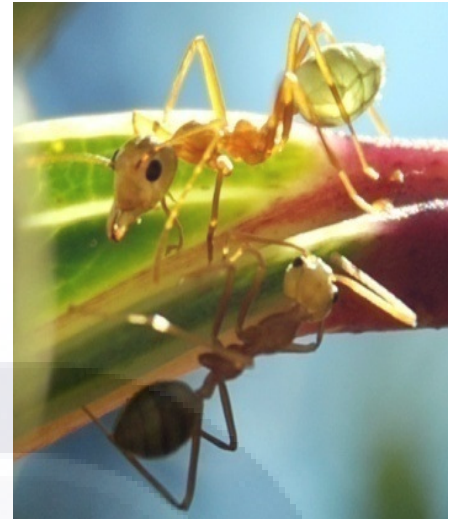


Figura 1.1. Hormigas Naturales Cortando Hojas

Los algoritmos de optimización con colonias de hormigas se han clasificado también como *algoritmos constructivos* (Pineda, 2008), debido a la forma en que trabaja, dentro del algoritmo cada hormiga construye una solución al problema recorriendo un grafo de construcción al ir de un estado a otro dentro del espacio de estados. El espacio de estados puede ser representado por un grafo en el que un vértice representa un estado y cada arista del grafo representa los posibles pasos o acciones que la hormiga puede realizar, normalmente cada arista tiene asociada información acerca del problema que guían el movimiento de la hormiga, la información asociada es el rastro de feromonas y la visibilidad (distancia), las hormigas escogen a donde ir de una manera probabilística (Dorigo and Colorni, 1996). En la figura 1.2 se muestra como es diferente la concentración de feromonas entre los dos caminos.

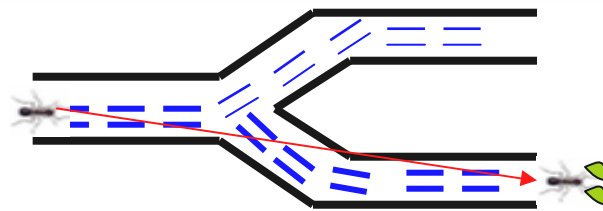


Figura 1.2 Hormiga guiada por el Rastro de Feromonas

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO I.- INTRODUCCIÓN

Es importante el desarrollo de métodos de optimización que explícitamente modelen el hallazgo de las buenas soluciones utilizando colonias de hormiga, con este tipo de algoritmos se esta trabajando con un nuevo paradigma dentro de la computación evolutiva. Los algoritmos de optimización con colonias de hormigas se basan principalmente en ir creando una solución en base a la concentración de feromonas que contienen los caminos que conducen a estados vecinos de la posición actual ya que se escogen los caminos de una manera probabilística en función de la feromona. Existen una gran cantidad de artículos que se han publicado en esta área para resolver el problema del TSP que fue una de las primeras aplicaciones, a través de los cuales se puede observar la evolución de los algoritmos de colonias de hormigas, entre ellos podemos citar los siguientes: Dorigo, Maniezzo and Colorni 1996, Gambardella and Dorigo 1996, Dorigo and Gambardella 1997 (1997a, 1997b), Stützle and Hoos 1997, Bullnheimer, Hartl and Strauss 1999 (1999a, 1999b), Cordón, de Viana, Herrera, & Morena 2000, Montgomery & Randall 2002, sin embargo todos ellos atacan una variante específica del problema del TSP.

Hoy en día hay un gran interés en el desarrollo de algoritmos de colonias de hormigas para resolver problemas que necesitan ser analizados con minería de datos, por lo cual existen algoritmos para la tárea de closterización y clasificación.

1.3 Necesidad de la Investigación

Actualmente existe la necesidad de resolver muchos problemas de optimización que forman parte de nuestra vida diaria, siempre se está en la búsqueda de cómo podemos optimizar recursos como son: tiempo, dinero, materias primas, recursos humanos, espacios, etc. Sin embargo, el resolver muchos de estos problemas no es nada sencillo ya que son problemas que se encuentran clasificados como NP-completos dentro de la teoría de la complejidad, lo que quiere decir que necesitan una gran cantidad de recursos computacionales (tiempo y espacio) para tratar de resolverlos y a veces estos recursos no son suficientes si se utilizan

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO I.- INTRODUCCIÓN

técnicas convencionales, por lo que es mejor técnicas heurísticas y/o meta-heurísticas para resolverlos, ya que estas nos permiten obtener soluciones cercanas al óptimo en un tiempo computacional razonable, es por esto que es importante el diseñar nuevos algoritmos que nos ayuden a resolver este tipo de problemas.

1.4 Planteamiento del Problema de Investigación

Existe la necesidad de diseñar algoritmos más eficientes, por lo cual se implementó un modelo de colonia de hormigas que explore de manera más eficiente el espacio de búsqueda.

En esta tesis proponemos resolver el problema de optimización del clique máximo usando un algoritmo de colonias de hormigas, al desarrollar un algoritmo el cual explore de mejor manera el espacio de búsqueda.




1.5 Relevancia y Justificación de la Investigación

Lo que se pretende con este trabajo es dar respuesta a la necesidad de crear nuevos algoritmos y /o meta-heurísticas que exploren eficientemente el espacio de búsqueda para resolver problemas de Optimización Combinatoria clasificados como problemas NP-Completos, esto a través de la generación de algoritmos que sean capaces de resolver problemas que se encuentren entre los más difíciles de resolver según la Teoría de la Complejidad Computacional por lo cual son difícil de resolver a través de técnicas exactas por su complejidad y se recomienda resolverlos por medio de técnicas heurísticas y/o meta-heurísticas. Los sistemas en los cuales se han utilizado más de una técnica inteligente son llamados sistemas inteligentes híbridos o sistemas híbridos inteligentes. ¿Por qué?, los sistemas inteligentes tienen colectivamente características como capacidad de aprendizaje, adaptación a los cambios, capacidad de la explicación y flexibilidad haciendo frente a la información imprecisa e incompleta, etc. Ningún sistema inteligente tiene todas las características y las limitaciones y las fuerzas de sistemas individuales es la fuerza



CAPÍTULO I.- INTRODUCCIÓN

impulsora central detrás de los sistemas inteligentes híbridos. Al integrar varios sistemas sus fuerzas pueden ser aumentados y las debilidades pueden ser reducidas.

Algunas razones para crear sistemas híbridos son (Wiley y Sons, 1995):


-  **Realce de la técnica:** Ésta es la integración de diversas técnicas para superar las limitaciones de cada técnica individual. Aquí lo importante es tomar una técnica que tenga debilidad en una característica particular y combinarla con una técnica que tenga fuerza en que la misma característica.
-  **Multiplicidad de tareas:** Cuando no hay técnica disponible para los muchos sub-problemas de una aplicación, dada entonces se utiliza un sistema híbrido.
-  **Multifuncionalidad:** Estos sistemas híbridos pueden exhibir capacidades múltiples de tratamiento de la información dentro de una arquitectura.

Tipos de hibridación (Wiley y Sons, 1995):

-  **Function-Replacing hybrids.** En este sistema, una función principal de la técnica dada es substituida por otra técnica de procesamiento inteligente. Se hace para aumentar la velocidad de ejecución o para realzar confiabilidad. La motivación para este acercamiento es el realce de la técnica.
-  **Intercommunicating hybrids.** Estos son independientes entre sí, autocontenidos, módulos de procesamiento inteligente que intercambian la información y se realizan separan las funciones para generar soluciones. Si un problema se puede subdividir en tareas de proceso distintas, después diversos módulos inteligentes independientes se

CAPÍTULO I.- INTRODUCCIÓN

pueden utilizar para solucionar las partes del problema, los cuales son los mejores. Estos módulos independientes, que solucionan colectivamente la tarea dada, son coordinados por un mecanismo de control. Este acercamiento es motivado por la multiplicidad de tareas del uso.

 **Polymorphic hybrids.** Estos sistemas utilizan una sola arquitectura de proceso para alcanzar la funcionalidad de diversas técnicas de procesamiento inteligente. La motivación para ampliar estos sistemas híbridos es realizando la multifuncionalidad dentro de una arquitectura de cómputo particular. Estos sistemas pueden emular funcionalmente a diversas técnicas de procesamiento. Esto es apropiado en las situaciones donde se desea cambios de la funcionalidad dinámicamente, ésta requiera la capacidad de un interruptor a partir de un estilo del procesamiento a otro.

Se estudiaron varios problemas como son el problema del agente viajero, de la mochila, clique máximo, closterización y clasificación, en este sentido se toma como tema de investigación el diseñar e implementar un modelo que sea capaz de resolver diferentes instancias de estos problemas. Por lo cual lo más importante de la propuesta es el diseño de varios modelos de colonias de hormigas que sirvieron para comprender como trabajan estos algoritmos para posteriormente tener una aplicación real en el área de minería de datos que implemente algunas ventajas de otras heurísticas y herramientas que aporten una base sólida para estos fines.

1.6 Áreas Relacionadas y Herramientas

Esta investigación relaciona áreas como son la investigación de operaciones, que es el área que estudia la forma de resolver problemas de optimización a través de métodos exhaustivos y de aproximación, el área de computación evolutiva y computación bio-inspirada que son áreas de la inteligencia artificial donde se desarrollan algoritmos como los algoritmos genéticos, cúmulos de partículas, algoritmos de colonias de hormigas, así

CAPÍTULO I.- INTRODUCCIÓN

mismo se integrará el área de minería de datos que es parte del descubrimiento del conocimiento (KD). Como se puede ver esta investigación involucra varias áreas del conocimiento lo que la hace multidisciplinaria.

1.7 Enfoque de Investigación, Métodos y Técnicas de Investigación

El enfoque de la investigación es de simulación y experimentación ya que se desarrollaron varios algoritmos en base al modelo y se realizaron las pruebas necesarias a través de los “benchmarks”.

Los “benchmarks” son instancias de los problemas a través de los cuales se demuestra si un algoritmo es capaz de obtener una solución o no, ya que se conoce la respuesta que debe de dar el algoritmo, estas pruebas son las que utilizan todas las personas a nivel mundial si es que se quiere demostrar que un algoritmo es bueno.

Esta investigación es de tipo cuantitativa debido a que si es posible medir los resultados obtenidos por el algoritmo y saber que tan bueno es, además es descriptiva, documental y de desarrollo, la cual se encuentra dentro de la línea de investigación de la Inteligencia Artificial, la cual es una rama de la ciencia considerada como de reciente creación.

El propósito de este trabajo es el proporcionar un modelo que ayude a favorecer el uso de los algoritmos de optimización para resolver problemas prácticos.

El diseño de un modelo de optimización es complicado debido a que intervienen muchos factores que deben analizarse detenidamente ya que para lograr que el modelo sea adaptativo debe tomar en cuenta que los datos de entrada pueden variar y el sistema debe ser capaz de procesar estos datos y dar la mejor solución posibles.

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO I.- INTRODUCCIÓN

Para poder realizar el modelo se debe identificar primero cuales son los datos que pueden variar que serían los datos por los cuales el sistema debe adaptarse, un ejemplo de estos datos para el problema del clique máximo sería, si el problema de que se desea resolver tiene un grafo asociado pesado ó no pesado.

En el proceso para el desarrollo del modelo, algunos de los objetivos serian:

La obtención de los datos del problema que se quiere resolver para su procesamiento posterior, estos datos pueden ser adquiridos por el sistemas a través de archivos o durante la ejecución del sistema, y este debe ser capaz de distinguir e identificar los datos.

Adaptación del sistema, en base a los datos de entradas este debe poder ser capaz de manipular los datos de la mejor manera.

Mejorar la respuesta de los componentes del modelo, los parámetros del algoritmo pueden ser ajustados en tiempo de ejecución para tratar de mejorar los resultados a través del tiempo.

1.8 Objetivos de la Tesis

Objetivo General:

1. Modelar, diseñar e implementar un algoritmo basado en colonias de hormigas aplicado al problema del clique máximo.

Objetivos Especificos:

1. Implementar un algoritmo de colonia de hormigas para resolver el problema del clique máximo.
2. Diseñar un experimento para calibrar el algoritmo
3. Ejecutar el algoritmo utilizando los benchmarks del estado de arte
4. Comparar los resultados obtenidos con los de la literatura
5. Mejorar el algoritmo implementado

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO I.- INTRODUCCIÓN

6. Aplicación en Closterización de alta dimensión

Preguntas Específicas:

1. ¿Es posible incorporar estrategias utilizadas por otras técnicas al modelo para hacerlo más eficiente?
2. ¿Es posible incorporar nuevos comportamientos emergentes de las hormigas naturales a los algoritmos para hacerlos más eficiente?

Hipótesis ó Conjetura de la Investigación

Es posible crear un modelo de colonias de hormigas aplicado a la minería de datos. Este algoritmo completaría un espacio de investigación que no ha sido explorado suficientemente, dando la posibilidad de potenciar la utilización de algoritmos eficientes, para la solución de problemas NP- completos.

1.9 Contribuciones de la Presente Tesis

En el libro “Data Mining and Discovery Knowledge in Real Life Application”, es posible ver cuatro puntos importantes, ya que está dividido en teoría de la minería de datos y el descubrimiento del conocimiento, métodos de closterización, retos y desafíos de la minería de datos y algunas aplicaciones reales, de este libro se tomo parte de la teoría base de minería de datos y sus aplicaciones.

1. Julio Ponce and Adem Karahoca, Data Mining and Discovery Knowledge in Real Life Application, Enero de 2009, Austria.

CAPÍTULO I.- INTRODUCCIÓN

En el capítulo “Artificial Societies and Social Simulation using Ant Colony, Particle Swarm Optimization and Cultural Algorithms”, se realiza una introducción a las sociedades artificiales y simulación social por medio de la aplicación de diferentes técnicas evolutivas, se muestra como sirve los modelos de negociación y la argumentación, así mismo la aplicación de colonias de hormigas, algoritmos culturales y cúmulo de partículas en problemas de logística.

1. Alberto Ochoa, Arturo Hernández, Laura Cruz, Julio Ponce, Fernando Cortes, Liang Li and Lenka Janacek. “Artificial Societies and Social Simulation using Ant Colony, Particle Swarm Optimization and Cultural Algorithms”. Aceptado en marzo 2009, Austria.

En el capítulo “Data Mining in Web Applications”, se hace una introducción a conceptos como la minería de datos y de cómo algunas aplicaciones de colonias de hormigas pueden ser aplicadas a análisis de las redes sociales, como es el caso de implementar el problema del clique máximo el cual nos permite obtener subconjunto de como se relacionan en base a un atributo.

2. Julio Ponce, Alberto Hernández, Alberto Ochoa, Felipe Padilla, Alejandro Padilla, Francisco Álvarez y Eunice Ponce De León, Data Mining in Web Applications, Aceptado Noviembre del 2008, Fecha de publicación: Enero de 2009, Austria.

Otra aportación es un algoritmo de optimización con colonias de hormigas aplicado al problema de la mochila en el cual se toman los objetos como independientes y se maneja que la probabilidad de seleccionar un objeto está en función de la feromona, el peso del objeto y la utilidad, tomados de manera independientes.

CAPÍTULO I.- INTRODUCCIÓN

3. Julio C. Ponce, Carlos A. Ochoa, Miguel A. Meza, Alejandro Padilla y Felipe Padilla
ACHPM: Algoritmo de Optimización con Colonia de Hormigas para el Problema de la Mochila, Revista Iberoamericana de Sistemas, Cibernética e Informática, Volumen 3 - Número 2, ISSN: 1690-8627, Florida, USA.

Se muestra que es posible implementar al combinar la minería de datos y una heurística a un juego de tablero en el cual se simulan sociedades y como estas interactúan por medio de la negociación de agentes, esto es importante debido a que el algoritmo de colonia de hormigas es considerado un sistema multiagente.

4. Alberto Ochoa, Julio C. Ponce, Antonio Zamarrón, Alberto Hernández, Alejandro Padilla, Francisco Álvarez, A Game Board Implementing Data Mining and Cultural Algorithms Hífen, Uruguaiana, v. 31, n. 59, pág. 191, ISSN 0103-1155, Noviembre del 2007, Uruguaiana, Brasil.

Se demostró que es posible mejorar los resultados de un algoritmo al implementar un optimizador local, en este caso se mejoro un algoritmo propuesto anteriormente en el 2006 a través del uso de un optimizador K-Opt.

5. Ponce J., Ponce de León E., Padilla F. Padilla y A. Ochoa, Algoritmo De Colonia De Hormigas Para El Problema Del Clique Máximo Con Un Optimizador Local K-Opt, Hífen, Uruguaiana, v. 30, n. 58, pág. 191, ISSN 0103-1155, Noviembre del 2006, Uruguaiana, Brasil.

CAPÍTULO I.- INTRODUCCIÓN

Se modela el comportamiento de las sociedades con el uso de minería de datos y la minería de datos para tratar de predecir como repartirán sus votos al entrar a competir un nuevo país.

6. Julio Ponce, Francisco Ornelas, Alberto Ochoa, Antonio Zamarrón, Jöns Sánchez Hybrid System for Determining a Ranking in Eurovision Song Contest, 9th Mexican International Conference on Computer Science, ENC'08, II Workshop on Data Mining , 6-10 de Octubre del 2008, Mexicali, B.C., México.

El siguiente trabajo muestra como puede ser utilizada la minería de datos para realizar predicciones de comportamientos sociales con el uso de algoritmos culturales, en este caso se hace una predicción de la asignación de sus puntos que pueden asignar a diez de los países que forma parte del concurso de Eurovisión.

7. Ochoa, Alberto Hernández, Arturo Sánchez, Jöhns Muñoz-Zavala, Angel Ponce, Julio, Determining the Ranking of a New Participant in Eurovision Using Cultural Algorithms and Data Mining, 18th International Conference on Electronics, Communications and Computers, 2008. CONIELECOMP 2008, Publication Date: 3-5 March 2008, On page(s): 47-52, Location: Puebla, Mexico, ISBN: 978-0-7695-3120-5 Digital Object Identifier: 10.1109/CONIELECOMP.2008.27, Posted online: 2008-03-12 10:18:24.0.

Se realizó un simulador en el cual se puede ver como con el uso de la minería de datos y los algoritmos culturales ayudan a un agente a la toma de decisiones para la comprensión del mejor paradigma, permitiendo comprender el concepto de negociación, a través del modelado social.

CAPÍTULO I.- INTRODUCCIÓN

8. Ochoa C., Ponce J., Baltasar R., Álvarez F, Quezada F., Torres M. y Pérez C. Baharastar- Simulador de Algoritmos Culturales Para la Minería de Datos Sociales, Memorias del 3er Congreso Internacional de Computación Evolutiva COMCEV'07, Octubre 2007, ISBN 970-728-055-7.

Se realizó un algoritmo de colonia de hormigas para la asignación de horarios, el cual fue desarrollado en base a las necesidades de CBTis 168 de Aguascalientes, esto con la finalidad de ayudar a la elaboración de horarios el cual es considerado un problema complejo el cual puede contener una gran cantidad de restricciones, y se pudo ver que es posible encontrar soluciones a problemas con uso práctico en la realidad.

9. J.F. Meza de Luna, F.J. Luna Rosas, A. Padilla Díaz, F. Padilla Díaz y J. C. Ponce Gallegos, Métodos Evolutivos para la Asignación de Horarios Docentes en el CBTis 168 de Aguascalientes México, Memorias del 3er Congreso Internacional de Computación Evolutiva COMCEV'07, Octubre 2007, ISBN 970-728-055-7.

La aportación de un modelo de relaciones al analizar las redes-sociales en Orkut, nos permite ver cómo se comportan los usuarios en relación de ciertos atributos y que es posible generar el síndrome de bipolaridad.

10. Julio Ponce, Alberto Ochoa, Wolfram Pietsch, and Zé Zolezzi-Hatsukimi, Ahankara: Identify Bipolar Síndrome in User of Orkut with Data Mining, Memorias del ENC 2007, ISBN 9780769528991, P2899, IEEE ENC 2007 29 Books Box 1 of 3 Septiembre 2007, Morelia, Michoacán.

CAPÍTULO I.- INTRODUCCIÓN

En el siguiente artículo se presento el primer algoritmo que se implemento para resolver el problema del clique máximo en el cual se resolvieron 20 de los benchmarks.

11. Ponce J., Ponce de León E., Padilla F. y Padilla A., Implementación De Un Algoritmo De Colonia De Hormigas Para El Problema Del Clique Máximo, Memorias de la 5ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2006, pág. 70-73, ISBN: 980-6560-88-4 Colección, 980-6560-89 Volumen 1, Julio del 2006, Orlando Florida, USA.

El siguiente trabajo se describe lo que es el aprendizaje y como este puede ayudar a mejorar los algoritmos al manipular la información de una mejor manera.

12. Julio Ponce, Alejandro Padilla, Felipe Padilla, Eunice Ponce de León, Alberto Ochoa, Algoritmo de Colonias de Hormigas con Aprendizaje, Memorias del Cuarto Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, Octubre del 2008, Aguascalientes, México.

El siguiente trabajo muestra un algoritmo de colonia de hormigas implementado para resolver el problema de Bin Paking, el cual es una variante del problema de la mochila que consiste en minimizar el número de contenedores necesarios para guardar n objetos.

13. Julio Ponce, Felipe Padilla, Alejandro Padilla, Eunice Ponce de León, Algoritmo de Colonia de Hormigas aplicado al problema de Bin Paking, Memorias del Tercer Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, Octubre del 2007, Aguascalientes, México

CAPÍTULO I.- INTRODUCCIÓN

Se implemento un algoritmo para resolver el problema del clique máximo el cual fue mejorado al implementar un optimizador local posteriormente en el 2007, implementando los benchmarks de la DIMACS.

14. Padilla F., Ponce J., Padilla A. y Ponce de León E., Implementación de un Algoritmo De Colonia De Hormigas para el Problema Del Clique Máximo, Memorias del 7° Seminario de Investigación, Universidad Autónoma de Aguascalientes y CONCYTEA, Noviembre del 2006, Aguascalientes, México, ISSN: 1870-4921

Se desarrollo un algoritmo de colonia de hormigas para resolver el problema de la mochila, para obtener la probabilidad de seleccionar un objeto, fue modificada la función al tomar como parámetro n el resultado de dividir la utilidad entre el peso.

15. Ponce J., Ponce de León E., Padilla F. y Padilla A., Algoritmos De Colonia De Hormigas Aplicados Al Problema De La Mochila, Memorias del Segundo Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, Noviembre del 2006, Aguascalientes, México

Se implemento un algoritmo de colonias de hormigas aplicado al problema del clique máximo, implementando los benchmarks de la DIMACS para ver su funcionamiento.

16. Ponce J., Ponce de León E., Padilla F. y Padilla A., Algoritmo De Colonia De Hormigas Aplicado Al Problema Del Clique Máximo, Memorias del Segundo

CAPÍTULO I.- INTRODUCCIÓN

Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, Noviembre del 2006, Aguascalientes, México

17. Hernández Martínez E., Padilla Díaz A. Ochoa Ortiz Zezzatti y Ponce Gallegos J., Ontologías con Optimización de Búsqueda Basada en Colonias De Hormigas Para la Solución de Problemas de Identificadores de Nombre, Memorias del Octavo Verano de la ciencia Región centro, San Luís Potosí, México, Agosto del 2006
18. Ponce J., Ponce de León E., Padilla F. y Padilla A., Diseño De Un Modelo De Optimización Con Colonias De Hormigas, Memorias del Primer Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, pág. 137, ISBN: 970-728-039-5, Noviembre del 2005, Aguascalientes, México

Este trabajo es de divulgación, permitiéndonos ver cuáles son los conceptos básicos y la teoría de donde surgieron estos algoritmos y conocer algunas de las áreas de aplicación.

19. Ponce J., Padilla F., Padilla A. y Meza M., Hormigas Atómicas, Gaceta de la Universidad Autónoma de Aguascalientes, Diciembre de 2005, Aguascalientes, México

El siguiente trabajo muestra un algoritmo de colonia de hormigas con un aprendizaje diferencial el cual consiste en obtener una diferencia entre las veces que aparece cada arista en la última y penúltima iteración, con la finalidad de premiar a los caminos en base a este indicador.

CAPÍTULO I.- INTRODUCCIÓN

20. Julio Ponce, Felipe Padilla, Eunice Ponce de León, Alejandro Padilla, Alberto Ochoa, Algoritmo De Colonias De Hormigas Con Aprendizaje Diferencial Para Resolver El Problema Del Agente Viajero, IV Seminario de Inteligencia Artificial: Un Panorama de Aplicaciones, Realizado en la Facultad de Ciencias de la Universidad Central de Venezuela, Diciembre de 2008, Caracas, Venezuela
21. Alberto Ochoa, Antonio Zamarrón, Julio Ponce, Dzoara Baydavletov & Ernesto Mancilla, Poverty in Mexico: An approach from the perspective of data mining 6° Congreso Internacional de Computó en Optimización y Software CICos 2008, Junio 2008, Cuernavaca, Morelos

1.10 Descripción de los Capítulos de la Tesis

El presente trabajo está compuesto por diez capítulos, los cuales se describen a continuación:

Capítulo 1: Introducción.- Se muestra una descripción general del contexto de la tesis, como la descripción del problema, la relevancia que tiene el hacer esta investigación, la justificación, la estructura de la investigación la cual consta de la descripción del problema de investigación, la pregunta general, preguntas específicas, la hipótesis, metas y objetivos, los cuales nos indican el contexto del problema.

Capítulo 2: Marco Teórico.- Se describe el marco teórico de optimización de colonias de hormigas (OCH's), dentro de esta sección se muestra una revisión de los antecedentes de las teorías bases que están relacionadas con los algoritmos de optimización con colonias de hormigas y algunos planteamientos multiobjetivos, así como lo que es el frente de Pareto y algunos de los algoritmos evolutivos multiobjetivos que han sido desarrollado hasta el momento.

CAPÍTULO I.- INTRODUCCIÓN

Capítulo 3: Estado del Arte del Problema Clique máximo.- dentro de esta sección se muestra una revisión de los antecedentes de los algoritmos desarrollados para resolver el problema del clique máximo, así como una descripción de los algoritmos de colonias de hormigas existentes para resolver dicho problema.

Capítulo 4: Hormigas Naturales.- Aquí se muestra un pequeño resumen sobre algunas características y estudios que se han hecho desde el punto de vista biológico de las hormigas, como son las especies que hay, como se clasifican y algunos de los comportamientos que tienen.

Capítulo 5: Modelos de Aprendizaje.- Aquí se menciona lo que es el aprendizaje, así como los tipos de aprendizaje que existe profundizando en el aprendizaje por refuerzos y a su vez se muestran varios modelos que han sido utilizados para implementar el aprendizaje por refuerzo.

Capítulo 6: Minería de Datos.- En esta sección veremos una introducción general a lo que es la Minería de Datos sus conceptos básicos y los procesos que son necesarios para llevarlo a cabo.

Capítulo 7: OCH aplicados a la Minería de Datos.- Aquí se muestra el estado del arte de los algoritmos de colonias de hormigas que se han desarrollado para la Minería de Datos, los cuales han sido enfocados a realizar las tareas de closterización y clasificación.

Capítulo 8: Desarrollo y Validación.- En este capítulo se muestra la descripción del trabajo realizado para poder comprobar lo propuesto en el presente trabajo, tanto el desarrollo de distintos algoritmos como el diseño de experimentos que se ha hecho con cada uno de ellos. Los algoritmos mostrados en esta sección son algoritmos para resolver el problema del clique máximo y el problema de la mochila, también se explica los parámetros y fórmulas utilizados en los distintos algoritmos.

Capítulo 9: Reporte y Discusión de los Resultados.- En este capítulo se muestran los resultados obtenidos por los algoritmos implementados para resolver los problemas del

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO I.- INTRODUCCIÓN

clique máximo y el problema de la mochila, ya que son los algoritmos que me ayudaron a comprender como funcionan bien los algoritmos OCH y se describen las contribuciones a los algoritmos ya existentes, y se discuten los resultados.

Capítulo 10: Conclusiones.-Se muestran las conclusiones a las que se llegaron al ver el desarrollo y los resultados obtenidos, por lo que se dan las recomendaciones como trabajo futuro.

Aquí se puede ver que los algoritmos obtienen buenos resultados pero es importante hacer más pruebas y entender más esta técnica. Así mismo la tesis cuenta con la sección de Bibliografía y Anexos.

1.11 Conclusiones

En este capítulo dio una introducción a la computación evolutiva y a los algoritmos bio-inspirados, haciendo énfasis en los algoritmos de colonias de hormigas, también se describió la importancia del uso de estas técnicas para la resolución de problemas complejos y se hace referencia a la teoría de las colonias de hormigas naturales para entender el funcionamiento del algoritmo de colonia de hormigas. Con lo cual se vio que técnicas evolutivas han comenzado a incursionar en la minería de datos debido principalmente a su poder exploratorio, ya que permiten evaluar diversas regiones de un espacio de diseño y encontrar varias soluciones a problemas complejos con relativa eficiencia. El Ant System es una técnica reciente basada en el comportamiento de forrajeo de las hormigas y con la cual se construyeron los algoritmos presentados en este trabajo.

También se describe la estructura de la tesis y se da una idea del contenido de cada uno de los capítulos. Como se pudo ver lo que se está proponiendo es crear un modelo de colonia de hormigas para aplicarlo en la minería de datos, y se muestra como se llegó a comprobar la hipótesis propuesta, y esto se logró a través del cumplimiento de las metas y respondiendo las preguntas hechas.

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

2. 1 Introducción






En este capítulo se describen las teorías bases dentro de las cuales se menciona que es una meta-heurística y sus propiedades, también se explica en qué consiste la optimización con colonias de hormigas y sus antecedentes, se muestra el algoritmo base de un algoritmo de colonia de hormigas y una tabla donde se pueden ver muchos de los problemas que han sido tratados con algoritmos de colonia de hormigas, también se puede ver la definición de algunos de los problemas de optimización con los que se ha trabajado hasta el momento que son el problema del agente viajero, el problema del clique máximo y el problema de la mochila.

2. 2 Descripción de Teorías Bases





2. 2.1 Metaheurísticas

Una meta-heurística es un conjunto de conceptos que pueden ser usados para definir métodos heurísticos que pueden ser aplicados en un conjunto amplio de problemas diferentes (Blum, 19).

2. 2.2.1 Propiedades de las metaheurísticas:

-  Las metaheurísticas son estrategias que guían el proceso de búsqueda.
-  La meta es explorar eficientemente el espacio de búsqueda con el objetivo de encontrar soluciones óptimas o cercanas al óptimo.
-  Las técnicas que constituyen algoritmos meta heurísticos abarcan desde procedimientos de búsqueda local simples hasta procesos de aprendizaje complejos.
-  Los algoritmos meta-heurísticos son aproximados y usualmente no determinísticos.
-  Incorporan mecanismos para evitar caer en un óptimo local.

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

-  Los conceptos básicos de las metaheurísticas permiten una descripción de nivel abstracto.
-  Las metaheurísticas no son específicas del problema.
-  Las metaheurísticas pueden hacer uso del conocimiento específico del dominio en la forma de heurísticas que son controladas por una estrategia de nivel superior.
-  Hoy en día las meta-heurísticas más avanzadas usan la experiencia adquirida en la búsqueda (almacenada en alguna forma de memoria) para guiarse por el espacio de soluciones.

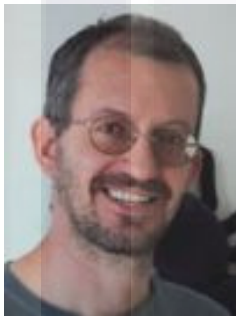
En pocas palabras podemos decir que las metaheurísticas son estrategias de alto nivel para explorar los espacios de búsqueda usando diferentes métodos. Es de gran importancia lograr en las meta heurísticas el balance dinámico de la diversificación y la intensificación en el proceso de búsqueda de las soluciones. El término diversificación generalmente se refiere a la exploración del espacio de búsqueda, mientras que el término intensificación se refiere a la explotación de la experiencia de búsqueda acumulada. El balance entre la diversificación y la intensificación (exploración y explotación) es importante por un lado para rápidamente identificar regiones en el espacio de búsqueda con soluciones de alta calidad y por otro lado no invertir mucho tiempo en regiones del espacio que ya han sido exploradas o que no tienen soluciones de alta calidad.

Existen muchas formas de clasificar las meta-heurísticas como por ejemplo: las inspiradas en la naturaleza o las que no son inspiradas por la naturaleza, las basadas en poblaciones o en un punto simple, con funciones objetivos dinámicas o estáticas, con estructuras de una vecindad o con varias vecindades, las que usan memoria o las que usan la información del estado actual, entre otras clasificaciones. Para nuestro estudio vamos a utilizar las colonias de hormigas.

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

Dentro de las meta-heurísticas basadas en poblaciones podemos citar entre otras, los algoritmos genéticos (Goldberg, 1989), los algoritmos de estimación de distribuciones (EDA) (Larrañaga y Lozano 2002, Larrañaga et al. 2003) la búsqueda dispersa (Laguna y Glover 2003, Laguna y Martí 2002, Martí y Laguna 2003), el reencadenamiento de camino (Glover 1998, Laguna y Glover 2003), las colonias de hormigas (Dorigo et al. 1991, Dorigo et al. 1996), la optimización de partículas inteligentes (Kennedy y Eberhart, 2001).

2.2.2 Optimización con Colonias de Hormigas (OCH)



Los algoritmos de optimización con colonias de hormigas son meta-heurísticas bio-inspiradas desarrolladas por Marco Dorigo (ver figura 2.1) en el año de 1992, estos algoritmos han tomado gran auge en los últimos años y han demostrado su utilidad en una gran diversidad de problemas.

Figura 2.1. Marco Dorigo

Los algoritmos de colonias de hormigas emplean estrategias inspiradas en el comportamiento de las colonias de hormigas naturales para descubrir fuentes de alimentación, al establecer el camino más corto entre el hormiguero y las fuentes de alimento, en la figura 2.2 se puede ver como la mayoría de las hormigas siguen el camino más corto, esto es debido a que las hormigas pueden transmitir información de la localización del alimento al resto de sus compañeras a través de la sustancia denominada feromona (Dorigo 1992).

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

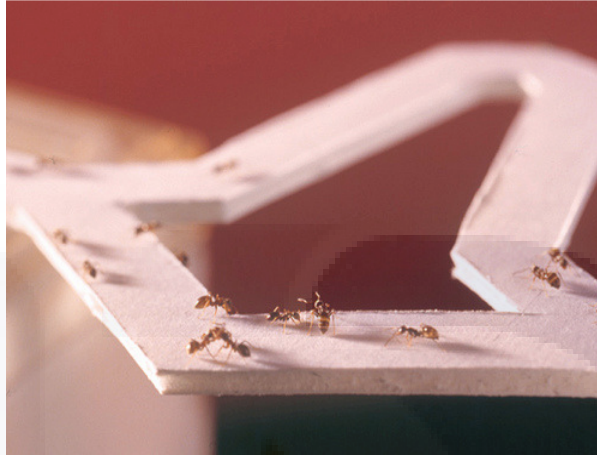


Figura 2.2. Experimento realizado para demostrar que las hormigas siguen caminos cortos

2. 2.2.1 La Estructura General de un Algoritmo de Optimización de Colonias de Hormigas

Se cuenta con M hormigas (artificiales) de la colonia, las cuales son colocadas en un estado inicial, a partir del cual estas se empiezan a mover, concurrentemente y de manera asíncrona, a través de los estados adyacentes del problema.

Este movimiento se realiza siguiendo una regla de transición que está basada en la información local disponible en las componentes, en el caso del problema del agente viajero es la distancia que existe entre un punto y otro (nodos). Esta información local incluye la información heurística y los rastros de feromona que han sido depositados por las hormigas para guiar la búsqueda, la probabilidad de que una hormiga seleccione uno de estos posibles caminos está dada por la fórmula (1).

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

$$p(v_i) = \frac{[\tau c(v_i)]^\alpha}{\sum_{j \text{ candidates}} [\tau c(v_j)]^\alpha} \quad (1)$$

La probabilidad de que la hormiga k seleccione ir al punto j a partir del punto i está dada por la concentración de feromona en el trayecto de i a j elevado a la alfa la cual representa la importancia de la feromona multiplicada por eta que es la visibilidad que existe de i a j, elevada a la beta que es la importancia de la visibilidad, esto dividido entre la sumatoria de la misma operación correspondiente a todos los caminos permitidos, donde los permitidos son todos aquellos trayectos donde la hormiga no ha pasado.

Al moverse por el grafo de construcción, las hormigas construyen incrementalmente soluciones. Opcionalmente, las hormigas pueden depositar feromona cada vez que crucen un arco (conexión) mientras que construyen la solución (*actualización en línea paso a paso de los rastros de feromona*). Una vez que cada hormiga ha generado una solución se evalúa ésta y puede depositar una cantidad de feromona que es función de la calidad de su solución y está dada por la fórmula (2). Esta información guiará la búsqueda de las otras hormigas de la colonia en el futuro.

$$\tau_{ij}(t+n) = \rho\tau_{ij}(t) + \Delta\tau_{ij} \quad (2)$$

Esta fórmula se interpreta como la concentración de la feromona en el tiempo t más n en el trayecto de i a j es igual a ρ que es el coeficiente de evaporación de la feromona a través del tiempo, multiplicada por la concentración actual de la feromona más un incremento.

Además, el modo de operación genérico de un algoritmo de optimización con colonias de hormigas incluye un procedimiento adicional que es la evaporación de los rastros de

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

feromona. La evaporación de feromona se lleva a cabo en el entorno y se usa como un mecanismo que evita el estancamiento en la búsqueda y permite que las hormigas busquen y exploren nuevas regiones del espacio de búsqueda.

La estructura general de un algoritmo de optimización con colonias de hormigas es:

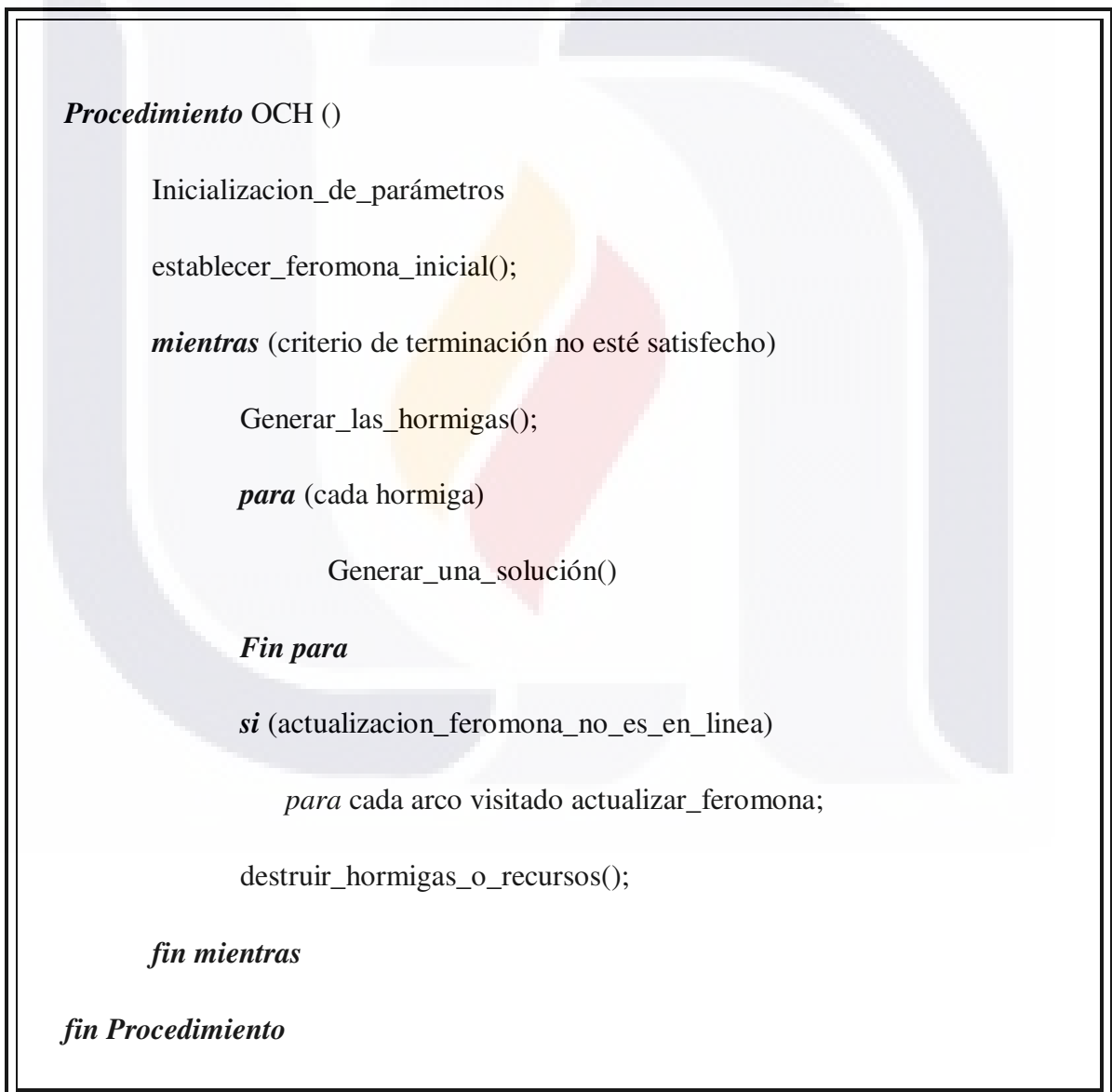


Tabla 2.1.- Algoritmo de Optimización con Colonias de Hormigas General

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

| |
|---|
| <p>Procedimiento establecer_feromona_inicial ()</p> <p>Para cada arco del grafo existente colocar un concentración inicial (C) de feromona</p> <p>fin Procedimiento</p> |
|---|

Tabla 2.2.- Procedimiento de inicialización de las feromonas

| |
|---|
| <p>Procedimiento Generación_de_Hormigas()</p> <p>repetir desde k=1 hasta m (número_hormigas)</p> <p>crear hormiga y colocar en su nodo inicial (actualizar lista tabú)</p> <p>fin repetir</p> <p>fin Procedimiento</p> |
|---|

Tabla 2.3.- Procedimiento donde se colocan las hormigas en su posición inicial

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

| |
|--|
| <p>Procedimiento Generar_una_solución()</p> <p>mientras (el criterio para obtener una solución no esté satisfecho)</p> <p style="padding-left: 40px;">mover_hormiga()</p> <p>fin mientras</p> <p>fin procedimiento</p> |
|--|

Tabla 2.4.-Procedimiento para general una solución

| |
|---|
| <p>Procedimiento mover_hormiga()</p> <p>para (todo el vecindario factible)</p> <p style="padding-left: 40px;">calcular_probabilidades_de_movimiento con la fórmula</p> <p>fin para</p> <p>estado_seleccionado:=seleccionar_el_movimiento();</p> <p>llevar_hormiga_a_ciudad(ciudad_seleccionada);</p> <p>si (actualizacion_feromona_en_linea_paso_a_paso)</p> <p style="padding-left: 40px;">depositar_feromona_en_el_arco_vistado()</p> <p>fin Procedimiento</p> |
|---|

Tabla 2.5.- Procedimiento para mover una hormiga de un lugar a otro

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

Los resultados obtenidos por la aplicación de algoritmos de Optimización con Colonias de Hormigas son muy alentadores, algunos de los resultados son mejores que los obtenidos por otras heurísticas (Dorigo et al., 2000).

2. 2.2.2 Problemas Tratados con Algoritmos de Optimización con Colonias de Hormigas

Debido a los resultados que se han obtenido a través de los algoritmos de colonia de hormigas en la última década, muchos investigadores se han interesado por la originalidad y el rendimiento de los algoritmos de optimización con colonias de hormigas, hasta el momento se han aplicado estos algoritmos con excelentes resultados a problemas tan diversos como son los mostrados en (Dorigo et al., 2000), también podemos ver algunos problemas que se han resuelto con esta técnica en el libro Ant Colony Optimization (Dorigo y Stützle, 2004) en las páginas 39 y 40, las tablas en ambas referencias solo muestran trabajos realizados hasta el año 2003. Por lo cual se realizó una investigación sobre los trabajos realizados hasta el momento con colonias de hormigas, para esto la tabla cuenta con los siguientes campos: trabajos, problemas y/o tipo de problemas y autores, como resultado de dicha investigación, fueron incorporados a la tabla anterior después del estudio bibliográfico diversos problemas y aplicaciones, y se presento una primera versión en la tesis Ponce en el 2007, mientras que en la presente tesis se retoma para actualizarla hasta el año 2010, en la tabla se indican con un * (asterisco) aquellos tipos de problemas, problemas y autores que se han agregado a esta como producto de la revisión bibliográfica de alrededor de 400 artículos, quedando la tabla de la siguiente manera:

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

| Tipo de Problema | Nombre del Problema | Autores | Año |
|------------------------------------|---|--|-----------------|
| Problemas de Subconjuntos (Subset) | *Subset Selection | *Solnon and Bridge | 2006 |
| | Problema de la Mochila (Múltiple knapsack) | Leguizamón and Michalewicz | 1999 |
| | | *Fidanova | 1999,2000, 2004 |
| | | *Alaya, Solnon, & Ghédira | 2004,2005 |
| | | *Ponce, Ponce de León, Padilla and Padilla | 2006b |
| | Max Independent Set | Leguizamón and Michalewicz | 2000 |
| | Redundancy Allocation | Liang and Smith | 1999 |
| | Set Covering | Leguizamón & Michalewicz | 2000 |
| | | Hadji, Rahoual, Talbi, & Bachelet | 2000 |
| | | *Lessing, Dumitrescu & Stützle | 2004 |
| | | *Ren, Feng, Ke and Zhang | 2010 |
| | Weight Constrained Graph Tree Partition | Cordone & Maffioli | 2001 |
| | Arc-Weighted K-Cardinality Tree | *Blum | 2002a |
| | | Blum & Blesa | 2003 |
| | Maximun Clique | Fenet & Solnon | 2003 |

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

| | | |
|---|---|--------------|
| | *Solnon & Fenet | 2004,2005 |
| | *Elliman and Youssef | 2004 |
| | *Bui and Rizzo | 2004 |
| | *Ponce, Ponce de León, Padilla and Padilla | 2006a, 2006c |
| | *Ponce, Ponce de León, Padilla and Padilla and Zezzatti | 2006 |
| | *Bui, Nguyen, and Rizzo Jr. | 2009 |
| Shortest common supersequence | Michel and Middendorf | 1998,1999 |
| Constraint satisfaction | Solnon | 2000,2002 |
| | *I. van Hemert and Solnon | 2004 |
| | *Tarrant and Bridge | 2004,2005 |
| 2D-HP Protein Folding | Shmygelska, Aguirre-Hernández, & Hoos | 2002 |
| | *Shmygelska and Hoos | 2003 |
| *2D and 3D Hydrophobic Polar Protein Folding | *Shmygelska and Hoos | 2005 |
| Bin Packing | Levine & Ducatelle | 2003 |
| | *Brugger, Doerner, Hartl and Reimann | 2004 |
| *Feature Subset Selection | *Al-Ani | 2004 |
| *Set Parking Problem | *Gandibleux, Delorme & T'Kindt | 2004 |

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

| | | | |
|---------------------------|--|--|-------------------|
| | | *Gandibleux, Jorge, Angibaud, Delorme & Rodríguez | 2005 |
| | *Estimating phylogenetic | *Daniele Catanzaro, Raffaele Pesenti y Michel C Milinkovitch | 2007 |
| | Tres | | |
| Problemas de Ruteo | Problema del Agente Viajero (Traveling Salesman) | Dorigo, Maniezzo and Colorni | 1991 ^a |
| | | | 1991b |
| | | | 1996 |
| | | Dorigo | 1992 |
| | | Gambardella and Dorigo | 1995,1996 |
| | | Dorigo and Gambardella | 1997a,1997b |
| | | Stützle and Hoos | 1997,2000 |
| | | Bullnheimer, Hartl and Strauss | 1999c |
| | | Cordón, de Viana, Herrera, & Morena | 2000 |
| | | *Montgomery & Randall | 2003 |
| | Problema de Ruteo de Vehiculos (Vehicle Routing) | Bullnheimer, Hartl and Strauss | 1999a |
| | | | 1999b |
| | | Gambardella, Taillard and Agazzi | 1999 |
| | | Reimann, Stummer, & Doerner | 2002 |
| | | *Reimann, Doerner, Hartl | 2002,2003 |
| | | *Lopes, Dalle and Erig | 2005 |

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS



| | | | | |
|---|--|---|------------------------|-----------|
| | | *Solnon | 2008 | |
| | Sequential ordering | Gambardella and Dorigo | 1997,2000 | |
| Problemas de Asignación | Problema de Asignación Cuadrática (Quadratic assignment) | Maniezzo, Colorni and Dorigo | 1994 | |
| | | *Gambardella, Taillard and Dorigo | 1997,1999 | |
| | | Stützle | 1997 | |
| | | Maniezzo | 1999 | |
| | | Maniezzo and Colorni | 1999 | |
| | | Stützle and Hoos | 2000 | |
| | | *Oliver van Kaick, Ghassan Hamarneh, Hao Zhang y Paul Wighton | 2007 | |
| | | Problema de Coloreo de Grafos (Graph coloring) | Costa and Hertz | 1997 |
| | | Problema de Asignación de Frecuencia (Frequency assignment) | Maniezzo and Carbonaro | 2000 |
| | | Problema de Asignación Generalizada (Generalized assignment) | Lourenço and Serra | 1998,2002 |
| Problema de Asignación de Horarios | Socha, Knowles, & Sampels | 2002 | | |

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

| | | | |
|---|---------------------------------|--|------------|
| | | Socha, Sampels, & Manfrihn | 2003 |
| Problemas de planificación de tareas (Scheduling problems) | Job Shop | Colorni, Dorigo and Maniezzo & Trubian | 1994 |
| | | *Ventresca and Ombuki | 2004 |
| | Open Shop | Pfahringner | 1996 |
| | Flow Shop | Stützle | 1998 |
| | Total Tardiness | Bauer, Bullheimer, Hartl, & Strauss | 2000 |
| | Total weighted Tardiness | den Besten, Stützle and Dorigo | 2000 |
| | | Merkle & Middendorf | 2000,2003 |
| | | Gagné, Gravel & Price | 2000 |
| | | Gagné, Price, & Gravel | 2002 |
| | Project Scheduling | Merkle, Middendorf, & Schmeck | 2000,2002 |
| | Group Shop | blue | 2002b,2003 |
| | *Grid Scheduling | *Fidanova and Durchova | 2006 |
| | *Shop | *blue and Sampels | 2004 |
| | | *Blum | 2005 |
| | *Permutation Scheduling | *Montgomery, Randall, & Hendtlass | 2005 |
| | *Operation scheduling | *Gang Wang, Wenrui Gong, Brian DeRenzi, y Ryan Kastner | 2007 |
| Machina Learning | Classification Rules | Parpinelli, Lopes, & Freitas | 2002,2005 |
| | | *P. Jaganathan, K.Thangavel, A. | 2007 |

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

| | | | |
|-----------------------------------|---|---|-------------------|
| | | Pethalakshmi y M. Karnan | |
| | Bayesian Network | De Campos, Gámez, & Puerta | 2002 |
| | Fuzzy Systems | Casillas, Cordon, & Herrera | 2000 ^a |
| | | | *2000b |
| | | | *2000c |
| Problema de Ruteo de Redes | Ruteo de Redes | *Sim K. M. and Sun W.H. | 2002, 2003 |
| | Problema de Ruteo de Redes Orientadas-Conexión (Connection-oriented network routing) | Schoonderwoerd, Holland, Bruten, & Rothkrantz | 1996 |
| | | Schoonderwoerd, Holland, & Bruten | 1997 |
| | | White, Pagurek and Oppacher | 1997,1998 |
| | | Di Caro and Dorigo | 1998b |
| | | Bonabeau, Henaux, Guérin, Snyers, Kuntz & Theraulaz | 1998 |
| | Problema de Ruteo de Redes (Connection-less network routing) | Di Caro and Dorigo | 1997,1998c |
| | | | 1998d |
| | | Subramanian, Druschel and Chen | 1997 |
| | | Heusse, Snyers, Guérin & Kuntz | 1998 |
| | Van der Put | 1998 | |
| | Problema Ruteo de Redes | Navarro Varela and Sinclair | 1999 |
| | Ópticas (Optical | | |

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS




| | | | |
|---------------------------------------|--------------------|--|---------------------|
| | networks routing) | | |
| | | *Katangur, Akkaladevi , Pan Y. and Fraser | 2005 |
| *Problemas de Minería de Datos | *Clustering | *Monmarché, Slimane and Venturini | 1997 |
| | | *Monmarché | 1999 |
| | | *Ramos and Almeida | 2000 |
| | | *Ramos and Merelo | 2002 |
| | | *Ramos, Muge and Pina | 2002 |
| | | *Parag M. Kanade and Lawrence O. Hall | 2003 |
| | | *Abraham A y Ramos Victor | 2003 |
| | | *Handl J., J. Knowles and M. Dorigo | 2003a , 2003b, |
| | | | 2006 |
| | | *Ling Chen, Xiao-Hua Xu, and Cyi-Xin Chen | 2004 |
| | | *Montes de Oca , Garrido and Aguirre | 2005 ^a , |
| | | | 2005b |
| | | *Claus Aranha y Hitoshi Iba | 2005 |
| | | *André L. Vizine, Leandro N. de Castro, Eduardo R. Hruschka, y Ricardo R. Gudwin | 2005 |
| | | *James Smaldon y Alex A. Freitas | 2005 |
| | | *Aranha, Claus de Castro | 2006 |
| | | *Parpinelli, Lopes, and Freitas | 2001 |

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

| | | | |
|--------------|---|--|------|
| | | *Ponce, Padilla, Zezzatti, Padilla and Ponce de León | 2009 |
| | *Image Clustering | *Salima Ouadfel y Mohamed Batouche | 2007 |
| Otros | *Capacitated Minimum Spanning Tree Problem | *Reimann, & Laumanns | 2004 |
| | | *Kopinitsch. | 2006 |
| | *Graph Matching Problem | *Sammoud, Solnon, and Ghédira | 2005 |
| | | *Sammoud, Sorlin, Solnon, and Ghédira | 2006 |
| | *Óptimal power flow | * Linda Slimani y Tarek Bouktir | 2005 |

Tabla 2.6.-Clasificación de los trabajos realizados con algoritmos de colonias de hormigas según el tipo de problema


A continuación se mencionan algunos de los algoritmos de colonias de hormigas junto con sus creadores que pueden ser encontrados al revisar la literatura mencionada en la tabla anterior, estos son:


-  El AS que fue el primer algoritmo de colonia de hormigas desarrollado por Marco Dorigo.
-  El algoritmo Ant-Q (Gambardella y Dorigo, 1995): Es un sistema híbrido del algoritmo AS con aprendizaje-Q (*Q-learning*), un modelo de aprendizaje por reforzamiento muy conocido.
-  Rank-based (Bullnheimer et al., 1997). Este algoritmo es una variación del AS en el cual la forma de calcular la actualización de las feromonas depositadas por las hormigas entre las ciudades se realiza sumando la cantidad de la feromona depositada entre


CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

dichas ciudades por las n mejores hormigas. Además, se realiza un incremento adicional si la mejor hormiga viajó por esa ruta.

 Ant Colony System: ACS (Dorigo y Gambardella, 1997), Es un algoritmo el cual es una extensión del Ant-Q.

 MAX-MIN Ant System (Stützle y Hoos, 1998): Este algoritmo se caracteriza por sólo actualizar los rastros de feromona de la mejor hormiga de cada ciclo y se establecen valores máximos y mínimos como límites en la acumulación de feromona.

 ASGA (White et al., 1997): Es un algoritmo híbrido entre AS y un GA en el cual se utiliza el AG para adaptar los parámetros del AS, de tal forma que la búsqueda se va adaptando en base a los resultados.

 ACSGA-TSP (Pilat y White, 2002): Es un algoritmo el cual incorpora una hibridación entre un algoritmo genético y el algoritmo ACS-TSP.

Estos algoritmos y algunos otros, son analizados un poco más afondo dentro de la tesis.

2.3 Colonias de Hormigas Colectivas / Optimización con Colonias de Hormigas Múltiple

Un acercamiento a la optimización de la colonia de la hormiga múltiple (Múltiple Ant Colony Optimization, MACO) es el algoritmo propuesto por Sim aplicado al balanceo de cargas en redes con conmutador de circuito. Este algoritmo utiliza colonias múltiples de hormigas para buscar las alternativas a una trayectoria óptima. Uno de los objetivos de este algoritmo es optimizar el funcionamiento de una red congestionada encaminando llamadas por medio de varias trayectorias alternativas para prevenir el congestionamiento. En ésta

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

estrategia cada grupo de agentes móviles corresponde a una colonia de hormigas, y la tabla de encaminamiento de cada grupo corresponde a una tabla de la feromona de cada colonia Sim y (2002 y 2003).

El algorítmico contiene tres mecanismos que se utilizan para organizar el trabajo de los individuos en cada colonia y el trabajo de todas las colonias. Es decir hay dos niveles de interacción. El primer es el nivel de la colonia y el segundo es el nivel de la población.

La interacción a nivel de la colonia se puede alcanzar con el proceso de depósito de la feromona dentro de la misma colonia. El mecanismo de actualización de la feromona es responsable de la implementación de esta clase de interacción. La interacción a nivel de la población es alcanzada evaluando las feromonas de diversas colonias usando una cierta función de evaluación. El aspecto importante aquí es el mecanismo de evaluación de la feromona. El grado de interacción de las diversas colonias es el rol del mecanismo de exploración/explotación. Este acercamiento algorítmico será llamado la optimización múltiple de las colonias de la hormiga. La interacción entre las colonias de la hormiga usando feromona se puede organizar en diversos términos (Aljanaby et al., 2010).

2.4 Migración en Colonias de Hormigas

El comportamiento de la migración probablemente para ser de importancia en el ajuste natural de las hormigas, donde habitan pequeñas cavidades preformadas cuya fragilidad puede obligar movimientos frecuentes. En el laboratorio, una colonia cuya jerarquía se ha dañado no sólo eficientemente se traslada a un nuevo sitio dentro de algunas horas, pero también elige confiablemente el mejor sitio de varias alternativas, discriminando entre sitios según área y altura de la cavidad, tamaño de la entrada y nivel de altura (Pratt et al., 2005).

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

Existen estudios anteriores que han descrito el comportamiento del trabajador que es la base de las capacidades de una colonia (Mallon et al., 2001; Franks et al. 2002). Las migraciones son organizadas por un subconjunto de trabajadores activos, cerca de la mitad de la población, busca nuevos hogares potenciales, determinan su calidad, y reclutan compañeros del mismo nivel para los hallazgos prometedores. Estas hormigas siguen una estrategia con transiciones a niveles más altos dependiendo de calidad de sitio y las interacciones con los compañeros de mismo nivel. En el nivel más bajo, una hormiga visita en varias ocasiones un sitio pero no la hacen recluta.

Un conjunto independiente de migración en el cual las colonias son requeridas para elegir entre dos lugares de calidad diferente.

2.5 Problemas de Optimización

A continuación se describen brevemente los tres problemas con los cuales se han trabajado para poder entender la implementación de este tipo de algoritmos, los cuales son: el problema del Agente Viajero y el problema de la Mochila, con los cuales se pudo ver que este tipo de algoritmo realmente funcionan y son capaces de resolver distintos problemas.

Problema del Agente Viajero (TSP)

El problema del agente viajero es el problema al que se enfrenta un vendedor que, saliendo de su ciudad natal, desea encontrar el camino más corto a través de un sistema de ciudades dado de los clientes, visitando cada ciudad una vez antes de finalmente volver a casa. El problema del TSP puede ser representado a través de un grafo completo con peso $G=(N,A)$ donde N es el conjunto de nodos (ciudades) y A es el conjunto de arcos(caminos) los cuales conectan a los nodos. Cada arco $(i,j) \in A$ tiene asignado un peso d_{ij} que representa la distancia que existe entre las ciudades i y j . El problema del TSP consiste en buscar el camino hamiltoniano de longitud mínima del grafo, donde un camino hamiltoniano es un

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

camino cerrado llamado tour, en el cual se visita cada nodo del grafo solamente una vez (Dorigo y Stützle, 2004).

Es importante mencionar que el TSP ha servido y continúa sirviendo como problema de prueba para distintos algoritmos. Actualmente existen varios trabajos que manejan algoritmos de colonias de hormigas para resolver el TSP o algunas de sus variantes.

Problema de la Mochila Múltiple (KP)

Este problema consiste en: Dado un conjunto de artículos $i \in I$ que están asociados a un vector de espacio requerido r_i y de un vector de beneficio b_i , el problema de la mochila es el problema de seleccionar un subconjunto de artículos de I de tal manera que quepan en una mochila con capacidad limitada y maximice la suma de beneficios de los artículos elegidos (Dorigo y Stützle, 2004).

2.6 Conclusiones

En esta sección la principal aportación es la revisión de la literatura existente, con la cual se realizó la ampliación de la tabla clasificatoria propuesta en la literatura por Marco Dorigo y Tomas Stützle incorporando los trabajos relacionados desde el año 2003 hasta el 2006, debido a que la tabla reportada en la literatura solo llegaba al año 2003, como resultado de esta revisión la tabla se amplió con la siguiente información se agregó 1 tipo de problema, 14 problemas nuevos que fueron tratados con ACOS y un total de 70 referencias nuevas que tratan también varios de los problemas ya existentes en la tabla, se mencionan varios de los algoritmos existentes, así mismo se puede ver algunas de las estrategias que han sido incorporadas recientemente a los algoritmos de colonias de hormigas como es el caso de las

CAPÍTULO II.- MARCO TEÓRICO DE COLONIAS DE HORMIGAS

colonias de homigas colectivas y la migración, por otra parte se describen brevemente en qué consisten los problemas del agente viajero y el problema de la mochila.



CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

3. 1 Introducción

El problema de Optimización Combinatoria conocido como problema del clique máximo (PCM) es un problema clásico que se clasifica dentro de los problemas NP- duros que son los más difíciles dentro de la clasificación de la teoría de la NP-Complejidad. El problema del clique máximo tiene importantes aplicaciones en diferentes dominios como son: teoría de código, diagnóstico de fallas, visión computacional entre otras. Por lo cual en este capítulo se muestra el problema y una visión global de donde algunos trabajos relacionados con el, así como la descripción de algunas de las técnicas con las que se ha tratado de resolverlo, así mismo se describen y analizan los trabajos que utilizan un algoritmo de colonia de hormigas para resolver el problema del clique máximo.

3. 2 Problema del clique Máximo

Dado $G=(V,E)$ donde G es un grafo cualquiera no dirigido y puede o no ser pesado al menos que se especifique, y $V=\{1,2,\dots,n\}$ es el conjunto de los vértices (nodo) de G , y $E \subseteq V \times V$ es el conjunto de arcos de G . Para cada vértice $i \in V$, se le asocia un peso positivo w_i con i , tomado de un vector de pesos $w \in \mathfrak{R}^n$. Se cuenta con una matriz simétrica de $n \times n$ esta es $A_G = (a_{ij})$ donde $(i, j) \in V \times V$ (i, j) y $a_{ij} = 1$ si $(i, j) \in E$ es una arista de G , y $a_{ij} = 0$ si $(i, j) \notin E$ (i, j), esta matriz es llamada matriz de adyacencia de G . Donde Para cualquier nodo v , contiene la expresión $N(v) = \{j \in V : a_{vj} = 1\}$ que denota el vecindario de v en G , es decir el conjunto de todos los nodos adyacentes a v .

Un grafo $G=(V,E)$ es completo si todos sus vértices en parejas son adyacentes, es decir $\forall i, j \in V$ con $i \neq j$ existe un arco $(i, j) \in E$.

Un clique es un conjunto C de vértices ($C \subseteq V$) donde todo par de vértices de C esta conectado con un arco en G , es decir C es un subgrafo completo. Un clique es parcial si

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

este forma parte de otro clique, de otra forma es maximal. El problema del clique máximo consiste en encontrar el clique máximo, es decir el clique maximal del grafo G con mayor cardinalidad.

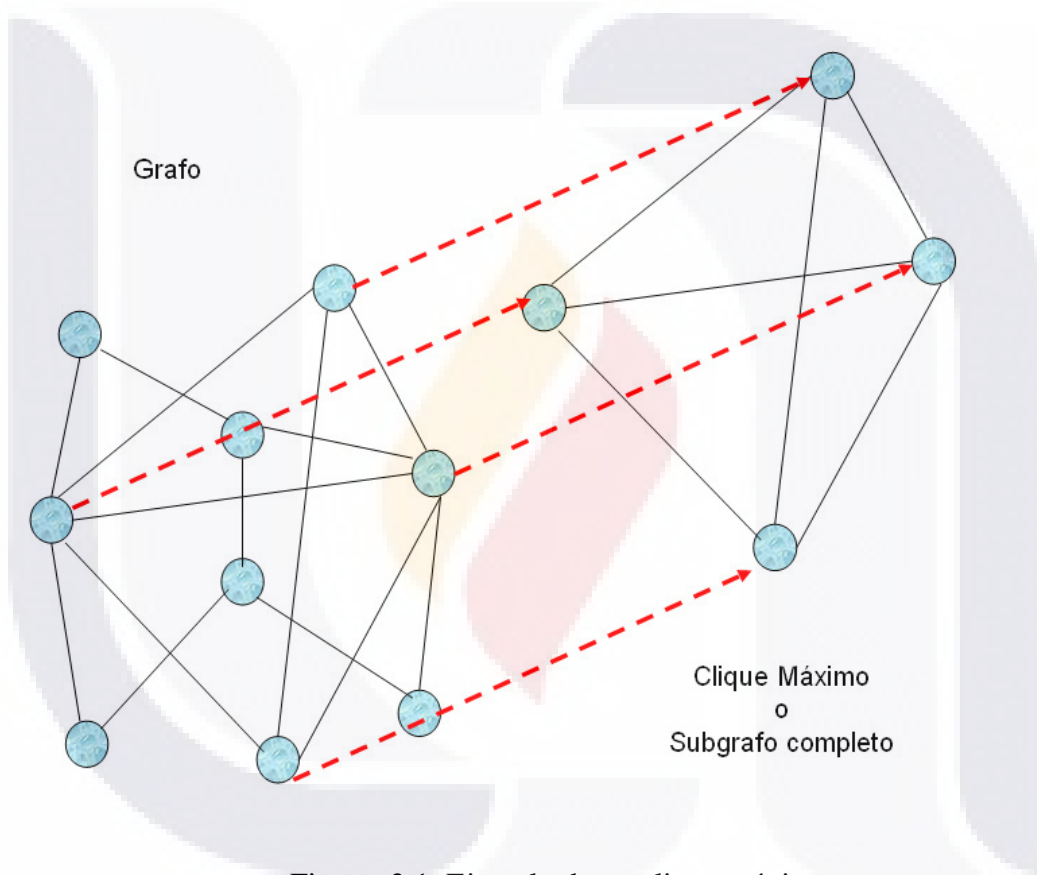


Figura 3.1 Ejemplo de un clique máximo

El problema del clique de peso máximo busca los cliques de peso máximo. Dado el vector de peso $w \in \mathbb{R}^n$, el peso de un clique es el total del peso de un clique máximo.

Un subconjunto independiente, es un subconjunto de V , con elementos en parejas no adyacentes. El problema de conjunto máximo independiente busca un conjunto independiente de cardinalidad máxima. El tamaño de un conjunto máximo independiente es

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

el número de la estabilidad de G (denotada $\alpha(G)$). Por lo cual podemos decir que el problema del clique máximo y el problema de conjunto máximo independiente son equivalentes.

3.3 Fórmulación del Problema

La fórmulación de los problemas es uno de los puntos clave en todas las áreas ya que esta parte es de crucial importancia en la solución de estos. Si usamos diferentes fórmulaciones, podemos adquirir más idea sobre la complejidad de los problemas y así podemos proveer resultados más interesantes.

3.3.1 Complejidad Computacional

El problema del clique máximo es uno de los primeros problemas que se demostró que era NP-Completo (Karp R.M. 1972), que quiere decir que, al menos que $P = NP$, un factor que es muy creíble hasta ser falsa, los algoritmos exactos garantizan el regresar una solución en un tiempo que se incrementa exponencialmente con el número de vértices en el grafo (Garey M. and Jonson D. 1979).

El interés por lo tanto ha ido cambiando hacia caracterización de propiedades de este problema. Los primeros trabajos en esta área empiezan a mediados de los 70's cuando Garey y Johnson (Garey M. and Jonson D. 1976). prueban que si el problema del clique máximo permite un algoritmo de aproximación en tiempo polinomial(es decir, este es aproximable a un factor constante), entonces este tiene un esquema de aproximación en tiempo polinomial(es decir, éste es aproximable a cualquier factor pequeño arbitrariamente). Esto es consistentemente expresado por lo dicho que si el problema del clique máximo debe

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

estar en la clase APX(constant-factor approximable problems), entonces debe estar en PTAS(polynomial-time approximation scheme)





Johnson et al (Johnson D.S, Yannakakis M. and Papadimitriou C.H. 1988): demostraron en 1988, basado en uno de sus resultados, la no existencia de un algoritmo polynomial-delay para la enumeración de cliques maximales en un orden lexicográfico inverso (si $P \neq NP$).

Se ha hecho una clasificación de los algoritmos utilizados para resolver este tipo de problema la cual consiste en dos grupos en base a su complejidad sintáctica y computacional que son:

1. La Clase de los Algoritmos de aproximación sintáctica que permite obtener resultados estructurados y tiene problemas completos naturales.
2. La Clase de los Algoritmos de aproximación computacional que permite trabajar con las clases de los problemas que resuelven bien con una aproximación.

3.3.2 Límites y Estimaciones

Las formas de estimar el tamaño del clique máximo y la complejidad de un grafo están basadas en su mayoría en las propiedades de la matriz A_G . Tomando en cuenta algunas restricciones y propiedades como:

-  La matriz A_G no puede ser reducida.
-  Se toma en cuenta el número de arcos del grafo (m).
-  La se calcula la densidad (δ) del grafo
-  La matriz A_G es simétrica y sus elementos son positivos

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

Existen muchos trabajos relacionados con la estimación del tamaño del clique máximo y de la complejidad de un grafo, algunos de ellos son estimaciones para grafos con características particulares varias formas de calcular los límites de los cliques y su complejidad, se encuentran en las siguientes referencias: Amin A.T. and Hakimi S.L. (1972), Wilf H.S. (1967), Knuth D.E. (1994), Helmberg C., Rendl F., Vanderbei R.J. and Wolkowicz H.(1996), Wilf H.S. (1986), Budinich M. and Matula D.W.(1976), Horn R.A. and Johnson C.R.(1985).

3.4 Algoritmos Desarrollados para Resolver el Clique Máximo

Actualmente existen una gran cantidad de algoritmos que han sido desarrollados para resolver el problema del clique máximo, estos han sido clasificados de diferente manera, a continuación se mencionan la clase de algoritmos y se mencionan algunos de los trabajos realizados en ese tipo de algoritmos.

3.4.1 Algoritmos Enumerativos

El primer algoritmo para la enumeración de todos los cliques de un grafo arbitrario en la literatura es probablemente el de Harary F. and Ross I.C.(1957). Ellos proponen un método inductivo que primero identifica todos los cliques de un grafo especial con no más de tres cliques. Entonces el problema dentro de un grafo general es reducido a este caso especial.

Algunos trabajos que siguieron a Harary and Ross pueden ser los realizados por Paull M.C. and Unger S.H. (1959) , y Marcus P.M.(1964) quienes propusieron algoritmos para minimizar el número de filas en el flujo de la tabla para una función cambiando la secuencia. Bednarek A.R. and Taulbee O.E.(1966) propusieron algoritmos para generalizar todas las cadenas maximales de un conjunto con una relación binaria definida en esta.

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

En Auguston J.G. and Minker J. (1970), investigaron severamente la teoría de grafos y el grupo de técnicas usadas en sistemas de información.

En 1973, se utilizaron dos nuevos algoritmos usando el método de backtracking que fueron propuestos por Akkoyunlu E.A. (1973) y por Bron C. and Kerbosch J.(1973).

En la década de los setentas fueron propuestos más algoritmos enumerativos algunos de estos están reportados en Osteen R. E. and Tou J.T. (1973), Osteen R. E. (1974), Meeusen W. and Cuyvers L. (1975), Johnston H.C.(1976), Johnson L.F.(1975), Leifman L.J.(1976), Tsukiyama S., Ide M., Aviyoshi H. and Shirakawa I.(1977), Gerhads L. and Lindenberg W.(1979).

En la década de los ochentas otros algoritmos propuestos están reportados en Loukakis E.. (1983), Loukakis E. and Tsouros C. (1981), Chiba N. and Nishizeki T. (1985), Tomita E., Tanaka A. and Takahashi H.(1988), Johnson D.S, Yannakakis M. and Papadimitriou C.H. (1988).

3.4.2 Algoritmos Exactos para el Caso No Pesados

Si nuestra meta es hallar un clique máximo o apenas el tamaño de un clique máximo, existe mucho trabajo que puede ahorrarnos el uso de los algoritmos de enumeración. Porque de los cliques encontrados, nosotros solo necesitamos enumerar los cliques mejores al mejor clique actual. Modificando los algoritmos de enumeración basado en este argumento resultan nuevos algoritmos de enumeración implícita.

El más conocido y utilizado comúnmente es el método de enumeración implícita para el problema del clique máximo es el método “branch and bound”.

Podemos ver un background del método de “branch and bound” en Balas E. and Toth P.(1985) , Nemhauser G.L. and Wolsey L.A.(1988).

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

Algunos trabajos que siguieron al “branch and bound” como algoritmos de enumeración implícita son:

En la década de los setentas, Desler J.F. and Hakimi S.L. (1970), Tarjan R.E.(1972), Houck D.J.(1974), Tarjan R.E and Trojanowski A.E. (1977), Chvátal V. (1977) y Houck D.J. and Vemuganti R.R. (1977).

En la década de los ochentas otros algoritmos propuestos estan reportados en Loukakis E. and Tsouros C. (1982), Ebenegger C., Hammer P.L. and de Werra D. (1984), Balas E. and Yu C.S. (1986), Kikusts P. (1986), Robson J.M. (1986), Tomita E., Kohata Y. and Takahashi H.(1988), Gendreau M., Picard J.C. and Zubieta L. (1988), Gendreau A., Salvail L. and Soriano P. (1993) y Pardalos P.M. and Rodgers (1992).

En la década de los noventas otros algoritmos propuestos estan reportados en Pardalos P.M. and Phillips A.T. (1990), Friden C., Hertz A. and de Werra M. (1990), Carraghan R. and Pardalos P.M. (1990), Babel L and Tinhofer G. (1990), Babel L. (1991), Xue J. (1991), Della Croce F. and Tadei R.(1994).

3.4.3 Algoritmos Exactos para el Caso Pesados

Los algoritmos para encontrar el conjunto maximal independiente pesado de un gráfico arbitrario comenzaron en 1975 con Nemhauser G.L. and Trotter L.E.(1975). Ellos consideraron las relaciones del poliedro entre sus bordes en la formulación del problema del conjunto maximal dependiente pesado y el problema de relajación lineal

Algunos trabajos que siguieron a este como algoritmos exactos son:

En la década de los setentas, Balas E. and Samuelsson H. (1977). En la década de los ochentas el algoritmo propuesto está reportado en Loukakis E. and Tsouros C. (1983). En la

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

década de los noventa otros algoritmos propuestos están reportados en Pardalos P.M. and Desai N. (1991), Balas E. and Xue J.(1991,1992), Nemhauser G.L. and Sigismondi G. (1992), Babel L. (1994) y Bomze I.M., Pelillo M. and Stix V. (1998).

3.4.4 Heurísticas

Debido a la complejidad computacional del problema del clique máximo, según lo visto de que es difícil de aproximar, mucho de los esfuerzos recientes se han enfocado hacia desarrollar una heurística eficiente. En la cuál no puede ser proporcionada ninguna garantía formal del funcionamiento, pero de todos modos hay interés en usos de aplicaciones prácticas.

Existen muchas heurísticas en la literatura diseñadas específicamente para aproximarse a una solución en el problema del clique máximo. Describiré algunas de estas heurísticas.

3.4.4.1 Secuencial Voraz (Sequential Greedy)

Muchos de los algoritmos de aproximación para el problema del clique máximo son llamadas Heurísticas Secuencial Voraz. Estas heurísticas generan un clique maximal repitiendo la acción de agregar un vértice dentro de un clique parcial, y borrando este vértice del conjunto que no forma parte del clique. Uno ejemplo de estos algoritmos puede verse en Kopf R. and Ruhe G. (1987).

3.4.4.2 Búsqueda Local (Local Search)

Una característica común en las heurísticas secuenciales es que todas ellas solo encuentran un clique maximal. Una vez que encuentren un clique maximal, la búsqueda se detiene. Se puede ver este tipo de heurísticas desde un diferente punto de vista. Se muestra un

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

algoritmo de este tipo en Aarts E. and Lenstra J.K.(1997). Uno de los trabajos más recientes que utiliza un algoritmo de este tipo es el realizado por Katayama K., Hamamoto A. y Narihisa H. en el 2004.

3.4.4.3 Búsqueda Avanzada (Advanced Search)

Los algoritmos de búsqueda local solamente son capaces de encontrar soluciones locales de un problema de optimización. Pocos años atrás, se realizaron muchas variaciones al algoritmo básico de la búsqueda local, que intentan evitar este problema, muchos de los cuales están inspirados en varios fenómenos que ocurren en naturaleza. Por ejemplo: recocido simulado (simulated annealing), redes neuronales (neural networks), algoritmos genéticos (genetic algorithms), optimización con colonias de hormigas (ant colony optimization), etc. Muchas de estas técnicas han sido utilizadas para tratar de resolver el problema del clique máximo.

3.4.4.4 Recocido Simulado (Simulated Annealing)

Está basado en la condensación de la materia dentro del área de la física. Cuando un sólido primero se calienta elevando su temperatura hasta que se derrite, y después se enfría lentamente hasta que se solidifica llegando a un estado bajo de la energía. Durante este proceso, la energía liberada del sistema es minimizada. Simulate Annealing fue desarrollado por Kirkpatrick S., Gelatt C.D. and Vecchi M.P. (1983), es un algoritmo de búsqueda en vecindarios aleatoriamente basado en el proceso físico annealing. Aquí la solución de un problema de optimización combinatoria corresponde a l estado del sistema físico, y del costo de una solución es equivalente al estado de energía. Uno de los trabajos para el problema del clique máximo se muestra en Jerrum M. (1992).

3.4.4.5 Redes Neuronales (Neural Networks)

Redes neuronales artificiales, están inspiradas en la anatomía y fisiología del cerebro, su objetivo esta puesto en simular varias características útiles tales como aprender y adaptación, aproximación universal, y reconocimiento de patrones. Las redes neuronales se utilizan en sistemas paralelos y distribuidos, aunque ya las redes neuronales artificiales ya existían, estas fueron utilizadas por primera vez en problemas de optimización en 1980 por Hopfield J.J and Tank D.W. (1985) .algunos trabajos con redes neuronales para el problema del clique máximo son: Lin F. anf Lee K. (1992), Grossman T. (1996), Jagota A., Sanchis L. and Ganesan R. (1996), Funabiki N., Takefuji Y. and Lee K.C. (1992), Bertoni A., Campadelli P. and Grossi G.(1997), Pelillo M.(1995).

3.4.4.6 Algoritmos Genéticos (Genetic Algorithms)

Los algoritmos genéticos son un proceso de búsqueda en paralelo inspirado en el mecanismo de evolución de los sistemas naturales [Holland J.H. (1975), Goldberg D.E. (1989)]. En contraste con muchas de las técnicas de optimización tradicionales, esta trabaja en una población, en la cual se tiene la siguiente terminología, se tiene a los cromosomas o individuos. Un cromosoma es una cadena larga de bits, cada individuo tiene asociado un “fitness” (valor que determina la probabilidad de sobre vivencia en la siguiente generación). El algoritmo genético empieza con una población inicial generalmente seleccionada aleatoriamente. El algoritmo genético cuenta con tres operadores básicos que son: reproducción, cruzamiento y mutación. La reproducción usualmente consiste en seleccionar a los cromosomas que van hacer copiados en la siguiente generación de acuerdo a una probabilidad que es proporcional a su fitness. Una vez seleccionados es aplicado el operador de cruzamiento entre pares de individuos seleccionadas, los cuales son divididos

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

en dos o más segmentos los cuales son combinados tomando segmentos intercalados para formar los nuevos individuos. Finalmente el operador de mutación es aplicado, este consiste en seleccionar un bit aleatoriamente y cambiarlo de estado, esto se hace con una probabilidad fija. El procedimiento descrito es el implementado para un algoritmo genético simple. Algunos de los trabajos realizados con algoritmos genéticos para resolver el problema del clique máximo y el problema del conjunto máximo independiente se muestran en: Murthy A.S., Parthasarathy G. and Sastry V.U.K. (1994), Bui T.N. and Eppley P.H. (1995), Foster J.A. and Soule T. (1995), Carte B. and Park K. (1993), Marchiori E. (1998).

3.4.4. 7 DNA Computing

El DNA Computing es una forma de representar los datos que utiliza la forma del DNA y la biología molecular, la cual en ves de representar la información como una cadena de datos binarios la cual se codifica con unos y los ceros, un filamento de la DNA se codifican con cuatro bases, representadas por las letras A, T, C, y G estas bases son conocidas como nucleotides. Se puede ver un algoritmo para resolver el problema del clique máximo en (Ouyang, Q., Kaplan P.D., Liu S. and Libchaber A. 1997)

3.4.4. 8 Búsqueda Tabú (Tabú Search)

La búsqueda tabú es una modificación del algoritmo de búsqueda local, la cuál esta basada en una estrategia de prohibición que se emplea para evitar ciclos en la trayectoria de búsqueda y para explorar nuevas regiones en el espacio de búsqueda (Glover F. 1989,1990). En cada paso del algoritmo, la solución siguiente visitada se elige siempre para ser el mejor vecino del estado actual, incluso si su costo es peor que la solución actual. El conjunto de vecinos es restringido por una o más listas tabú las cuáles previenen al algoritmo para ir de nuevo a soluciones recientemente visitadas. Éstas listas son utilizadas

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

para almacenar la información histórica de las trayectorias seguidas por el procedimiento en la búsqueda. Algunas veces las restricciones tabú se relajan, y algunas soluciones tabú son aceptadas si satisfacen una cierta nivel o condición de aspiración. Pueden existir otras listas que contengan las últimas modificaciones realizadas, es decir cambios realizados cuando movimos una solución a otra. Este tipo de listas son llamadas memorias a corto plazo; otras formas de listas son utilizadas para intensificar la búsqueda en una región promisorias o para diversificar la búsquedas a regiones inexploradas.

Algunos de los trabajos realizados con tabú para resolver el problema del clique máximo y el problema del conjunto máximo independiente se muestran en: Gendreau A., Salvail L. and Soriano P. (1993) y Soriano P., and Gendreau M. (1996).

3.4.4. 9 Búsqueda Dispersa (Scatter Search)

La búsqueda dispersa es un método evolutivo que combina vectores de la solución por combinaciones lineales para producir nuevas generaciones sucesivas. El método también incluye el uso de una heurística con un proceso rounding para colocar componentes fraccionarios dentro de los discretos, como se requiere en problemas enteros, mixtos y puros. La búsqueda dispersa confía mucho en una estrategia de decisión para combinar soluciones. Puesto que la búsqueda del tabú (TS) es la base de la misma filosofía (Cavique L., Rego C., Themido I. 2001).

3.4.4. 10 Optimización con Colonia de Hormigas

Los algoritmos de optimización de colonias de hormigas están basados en el comportamiento de las colonias de hormigas naturales, en la forma en que estas buscan un camino desde el hormiguero a las fuentes de alimentación. Es un algoritmo que trabaja con

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

poblaciones donde cada individuo es una hormiga la cual es un agente, cada hormiga es colocada en un estado inicial de donde empezará a explorar su espacio construyendo una solución con una probabilidad de escoger una ruta que depende de la cantidad de feromona almacenada a través del tiempo, y depositará en cada arco una cantidad de feromona dependiendo de que tan buena fue su solución. La concentración de feromona en una ruta se va incrementando cuando por este pasan más hormigas, de lo contrario este va perdiendo feromonas a través del tiempo. Estas es una de las técnicas más recientes que ha sido utilizada para tratar de resolver el problema del clique máximo y el problema del conjunto máximo independiente se pueden ver el trabajo realizado al respecto en: Fenet S. and Solnon C. (2003), Solnon C. y Fenet S. (2004, 2005), Elliman D.G. y Youssef S.M. (2004), Bui T.N. y Rizzo J.R.(2004) y Bui T. N., T. H. Nguyen, and J. R. Rizzo Jr. (2009).

3.4.4. 11 Algoritmos de Estimación de la Distribución (EDA)

Los algoritmos de estimación de la distribución están basados en poblaciones y genera la siguiente población de una manera absolutamente diversa. Mantienen un modelo de la probabilidad que caracterice la distribución de las soluciones prometedoras en cada iteración. El modelo de la probabilidad es actualizado, basado en la información estadística global extraída de la población actual. Muestreando de la población actual para generar la población descendiente. La información global sobre el espacio de búsqueda se utiliza hasta ahora para producir la nueva población en los algoritmos de estimación de la distribución. Sin embargo, la información sobre las localizaciones de las soluciones encontradas hasta ahora no se utiliza directamente para dirigir la búsqueda. Podemos ver un algoritmo para el problema del clique máximo en (Zhang Q., Sun J. y Tsang E. 2005).

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

3.4.4. 12 Heurísticas Basadas en un Dominio Continuo (Continuous-based Heuristics)

Recientemente, ha habido mucho interés alrededor del Motzkin-Straus en relación con fórmulaciones continuas del problema del clique máximo. Ellos sugieren una nueva manera fundamental de resolver el problema del clique máximo, permitiendo que cambiemos del dominio discreto al un dominio continuo de una manera elegante. Mientras que recientemente se describieron las fórmulaciones continuas de los problemas de optimización discretos estas resultan ser particularmente atractivas. Ellas no solamente permiten que explotemos el arsenal lleno de las técnicas de optimización continua, de tal modo que esto conduce al desarrollo de nuevos algoritmos, pero puede también revelar propiedades teóricas inesperadas. Podemos ver algunos de los trabajos realizados en el dominio continuo en: (Pardalos P.M. and Phillips A.T. 1990, Pelillo M. 1995, Bomze I.M., Pelillo M. and Giacomini R. 1997).

3.4.4. 13 Otros

Existen otros tipos de heurísticas implementadas para tratar de resolver el problema del clique máximo como son:

Aproximación de Subgrafos (subgraph approach). Este esta basado en el hecho de que un clique máximo es un subgrafo de G , por lo que el algoritmo primero obtiene un subgrafo G' el cual es usado para aproximarse a la solución (Balas E. and Yu C.S. 1986, Xue J. 1994).

También fueron presentadas nuevas heurísticas en el challenge de la DIMACS como son:

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

Un algoritmo basado en la observación que busca un clique máximo en la unión de dos cliques en los cuales se puede usar la técnica de emparejamiento bipartito propuesto por Balas E. and Niehaus W. (1996).

Goldberg D.E. and Rivenburgh R.D. (1996) usan el backtracking restringido para proporcionar una compensación entre el tamaño del clique y lo completo de la búsqueda.

Mannino C. and Sassano A. (1996) propusieron una técnica de proyección del arco para obtener un nuevo súper heurística bound para el problema del conjunto máximo independiente.

Otras heurísticas recientes son:

El algoritmo basado en la Rentabilidad-Monotonica de juegos dinámicos el cual usa la evolución del comportamiento vía procesos de imitación, y para estudiar la estabilidad de sus equilibrios cuando un parámetro de la regularización se le permite adquirir valores negativos (Pelillo and Torsillo, 2006).

3.5 Aplicaciones

En muchas aplicaciones, el problema subyacente se puede formular como el problema del clique máximo mientras que el otro subproblema del procedimiento de la solución consiste en el encontrar un clique máximo. Esto hace necesario el desarrollo rápido de algoritmos exactos y de aproximación para el problema.

Las áreas de aplicación consideradas son diversas por ejemplo, presentaremos una clase de grafos de los cuales podemos probar o refutar la conjetura de Keller's; un problema famoso en geometría, una parte la cual todavía está abierta (Corradi K. and SzaboS., 1990).

Otro ejemplo se presenta en teoría de código de la forma donde uno desea encontrar en grandes códigos binarios si es posible que se pueda corregir un número especificado de errores. El problema puede ser resuelto al encontrar una solución al problema del clique máximo en un gráfico correspondiente.

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

También puede utilizarse el problema del clique máximo en modelos de la diagnóstico de fallas en el cual la tarea principal es identificar todas las fallas en los procesadores de un sistema (Preparata F.P., Metze G. and Chien R.T., 1967).

Otra área de aplicación importante del problema del clique máximo es visión computacional y reconocimiento de patrones en la cual muchos problemas fundamentales pueden ser formulados como el problema de emparejamiento de estructuras relacionadas (Ambler A.P., Barrow H.G., Brown C.M., Burstall R.M. and Popplestone R.J., 1973).

La enumeración eficiente de cliques máximos tiene usos en análisis de datos microarray y un gran número de problemas en el área biológica (Abu-Khzam1 F. N. , Baldwin N.E., Langston M. A. y Samatova N. F., 2005).

Otra aplicación es el agrupamiento de datos (clusters) que se ocupan de datos basados o predispuestos que conducen a los modelos con clusters no-disjuntos. Allí, los objetos que están en un cluster pueden pertenecer a varios en el mismo tiempo, lo que da lugar agrupamiento difuso. Por lo que esto es equivalente a buscar todos los cliques maximales en un grafo dinámico (Volker Stix 2001, Peters M. y Zaki M.J., 2004).

3.6 Descripción y Análisis de Algoritmos de colonias de Hormigas desarrollados para resolver el problema del clique máximo

Actualmente existen varios trabajos realizados con algoritmos de colonias de hormigas para tratar de resolver el problema del clique máximo los cuales pueden ser consultados en Fenet S. y Solnon C.(2003), Solnon C. y Fenet S. (2004, 2005), Elliman D.G. y Youssef S.M. (2004) y Bui T.N. y Rizzo J.R.(2004), Bui, T. N., T. H. Nguyen, and J. R. Rizzo Jr. (2009).

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

3.6.1 Algoritmo de Fenet y Solnon

Es el primer trabajo realizado para resolver el problema del clique máximo a través de un algoritmo de optimización con colonias de hormigas. Este trabajo es titulado “Searching for Maximum Cliques with Ant Colony Optimization” y puede verse en Fenet y Solnon en el 2003. En este artículo Fenet y Solnon proponen el primer algoritmo de optimización con colonias de hormigas para el problema del clique máximo al cual llamaron Ant-Clique y es el que se muestra en la tabla 3.1.

| |
|--|
| <p>Procedimiento Ant-Clique</p> <p>Inicializar los rastros de feromonas</p> <p>Repetir</p> <p> Para k en $1..nb$ Hacer: Construir un clique C_k</p> <p> Actualizar los rastros de feromonas w.r.t. $\{C_1, \dots, C_{nb}\}$</p> <p> Fin Para</p> <p>Hasta Alcanzar el número de ciclos o encontrar una solución óptima</p> |
|--|

Tabla 3.1.- Procedimiento general del algoritmo de optimización con colonias de hormigas para el problema del clique máximo

En este algoritmo nb es el número de hormigas que se tienen, k es el índice la k -ésima hormiga, C_k es el clique construido por la hormiga k y C_1, \dots, C_{nb} son todos los cliques construidos por las hormigas.

Inicialización de los rastros de feromona:

Las hormigas se comunican a través de la feromona depositada en los arcos del grafo. La concentración de feromona en el arco (v_i, v_j) es denotada por $\tau(v_i, v_j)$, y representa la importancia aprendida para que v_i y v_j se encuentren en un mismo clique. Ellos

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

explícitamente imponen un límite inferior y superior τ_{min} y τ_{max} en los rastros de feromonas con ($0 < \tau_{min} < \tau_{max}$).

Construcción del Clique por las Hormigas

El procedimiento para construir un clique se puede ver en la Tabla 3.2. En este algoritmo las hormigas primero seleccionan aleatoriamente un vértice inicial, y después eligen iterativamente los vértices que se agregarán al clique, estos vértices se encuentran dentro de un conjunto de candidatos el cual es denotado con la letra “*Candidatos*” el cual contiene todos los vértices que están conectados con cada vértice del clique pero no forma parte de él todavía. Se tiene un vértice inicial el cual es denotado por v_f , la probabilidad de escoger uno de los vértices candidatos se encuentra dada por $P(v_i)$ y C contiene el clique que se está construyendo.

Algoritmo para construir un clique

Escoger aleatoriamente el primer vértice $v_f \in V$

$C \leftarrow \{v_f\}$

$Candidatos \leftarrow \{v_i / (v_f, v_i) \in E\}$

Mientras $Candidatos \neq \emptyset$ **Hacer**

Escoger un vértice $v_i \in Candidatos$ con una probabilidad

$$p(v_i) = \frac{[\tau_{c(v_i)}]^\alpha}{\sum_{v_j \in candidatos} [\tau_{c(v_j)}]^\alpha}$$

$C \leftarrow C \cup \{v_i\}$

$Candidatos \leftarrow Candidatos \cap \{v_j / (v_i, v_j) \in E\}$

Fin Mientras

Tabla 3.2.- Algoritmo para construir un clique con una hormiga

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

Los algoritmos de Solnon y Fenet (Vertex-AC Y Edge-AC).

Estos algoritmos fueron reportados por Solnon y Fenet en el 2004 en su artículo titulado “Investigating ACO capabilities for solving the Maximum Clique Problem” donde proponen dos algoritmos de colonias de hormigas para el problema del clique máximo a los cuales llamaron Vertex-AC y Edge-AC los cuales se describen a continuación.

| |
|---|
| <p>Búsqueda de una aproximación de un clique máximo en un grafo $G = (V,E)$:</p> <p>Inicializar rastros de feromonas en τ_{max}</p> <p>Repetir el siguiente ciclo:</p> <p> Para cada hormiga K en $1..nb$ hormigas, construir in clique maximal C_k de la siguiente manera:</p> <p> Escoger aleatoriamente el primer vértice $v_f \in V$</p> <p> $C_k \leftarrow \{v_i\}$</p> <p> $Candidatos \leftarrow \{v_j \in V \mid (v_f, v_i) \in E\}$</p> <p> Mientras $Candidatos \neq \emptyset$ Hacer</p> <p> Escoger un vértice $v_i \in Candidatos$ con una probabilidad $P(v_i)$</p> <p> $C_k \leftarrow C_k \cup \{v_i\}$</p> <p> $Candidatos \leftarrow Candidatos \cap \{v_j \mid (v_i, v_j) \in E\}$</p> <p> Fin Mientras</p> <p> Fin Para</p> <p> Actualizar los rastros de feromonas w.r.t. $\{C_1, \dots, C_{nbAnts}\}$</p> <p> Si un rastro de feromona es menor a τ_{min} entonces colocarlo en τ_{min}</p> <p> Si un rastro de feromona es mayor a τ_{max} entonces colocarlo en τ_{max}</p> <p>Hasta Alcanzar el números de ciclos \bullet encontrar una solución óptima</p> <p>Regresar le clique más grande construido en todo el proceso</p> |
|---|

Tabla 3.3.- Algoritmo de colonia de hormigas reportado por Solnon y Fenet en el 2004

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

El algoritmo anterior es una estructura generalizada para el Vertex-AC y Edge-AC

La fórmula de la probabilidad que ellos reportan esta dada por:

$$p(v_i) = \frac{[\tau \text{ fact}(v_i)]^\alpha}{\sum_{v_j \text{ candidates}} [\tau \text{ fact}(v_j)]^\alpha} \quad (3)$$

Donde α es un parámetro que representa la importancia del factor de la feromona, y el $\text{fact}(v_i)$ es el factor de la feromona del vértice v_i . Este factor de la feromona depende de la cantidad de feromona depositada en los componentes del grafo, es decir:

En Vertex-AC, las hormigas depositan los rastros de feromonas en los vértices V del grafo. Por lo cual el factor de la feromona depende de la cantidad de feromona depositada en el vértice candidato.

$$\text{fact}(v_i) = \tau_i \quad (4)$$

En Edge-AC, las hormigas depositan los rastros de feromonas en los arcos E del grafo. Por lo cual el factor de la feromona depende de la cantidad de feromona depositada en los arcos de elemento candidato con todos los elementos del clique.

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

$$\tau_{fact}(v_i) = \sum_{v_j \in C_k} \tau_{ij} \quad (5)$$

Para la actualización del rastro de feromona utiliza la fórmula 2

En la cual ρ es porcentaje de evaporación de la feromona y $\Delta\tau_{ij}$ es el incremento por haber pasado una hormiga por ese lugar.

El incremento se obtiene de la siguiente manera:

$$\Delta\tau_{ij} = \frac{1}{1 + |C_{best}| - |C_k|} \quad (6)$$

Donde C_{best} es el tamaño del mejor clique encontrado hasta el momento y C_k es el tamaño del clique que encontró en ese momento la hormiga k .

Los Algoritmos Vertex-AC+LS Y Edge-AC+LS

En el tercer trabajo realizado por Solnon y Fenet en el 2005 fue titulado “A study of ACO capabilities for solving the Maximum Clique Problem” donde proponen dos nuevos algoritmos ACO para el problema del clique máximo a los cuales llamaron Vertex-AC+LS y Edge-AC+LS.

Las siglas AC de estos algoritmos significan Ant Clique, los algoritmos Vertex-AC, Vertex-AC+LS tiene la palabra Vertex debido a que la feromona es depositada en sus vértices y el segundo cuenta con las siglas LS porque al algoritmo Vertex-AC se le agregó una búsqueda local (Local Search), los algoritmos Edge-AC y Edge-AC+LS tiene la palabra Edge debido a que la feromona es depositada en los arcos.

Por lo cual toman como base el algoritmo propuesto por ellos en su artículo del 2004 y solo le agregan un optimizador local el cual es descrito en Solnon C. y Fenet S. (2005).

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

3.6.2 El Algoritmo RPACOMCP

En el 2004 Elliman D.G. y Youssef S.M. realizaron el trabajo titulado “Reactive Prohibition-based Ant Colony Optimisation (RPACO): A New Parallel Architecture for Constrained Clique Sub-Graphs” donde proponen un algoritmos ACO para el problema del clique máximo con peso al cual llamaron RPACOMCP. En su artículo muestran los siguientes dos algoritmos:

| |
|--|
| <p>Algoritmo RPACO</p> <p>Escoger aleatoriamente el primer vértice $s\text{-nodo} \in V$</p> <p>$CC \leftarrow \{s\text{-nodo}\}$</p> <p>$Candidatos \leftarrow \{vi / (s\text{-nodo}, vi) \in E\}$</p> <p>Mientras $Candidatos \neq \emptyset$ Hacer</p> <p> Escoger un vértice $vi \in Candidatos$ con una probabilidad</p> $p(v_i) = \frac{[\tau(v_i)]^\alpha [\omega(v_i)]^\beta}{\sum_{j \text{ candidates}} [\tau(v_j)]^\alpha [\omega(v_j)]^\beta}$ <p> $CC \leftarrow CC \cup \{vi\}$</p> <p> $Candidatos \leftarrow Candidatos - \{vi\} + \{vk / (vi, vk) \in E, \forall v_k \notin CC\}$</p> <p>Fin Mientras</p> |
|--|

Tabla 3.4.- Algoritmo para construir un clique con una hormiga

En el cual CC es el clique en construcción, s-nodo es el nodo inicial y ω el peso del vértice.

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

Algoritmo RPACPMCP

Inicializar rastros de feromonas

Repetir

Generar una población de hormigas de tamaño PopSize

Para cada hormiga (k) en la población

Tamaño-clique = 0

Seleccionar aleatoriamente el inicio de CC (s-nodo)

Repetir

Construir una lista de candidatos factibles PN con los vecinos de CC

Evaluar la función de probabilidad de selección $p(v_i)$ para cada $V_i \in PN$

Siguiente-nodo = seleccionar un vecino factible usando rand-proportional-rule

Agrega el siguiente-nodo a CC

Incrementar la longitud del clique

Hasta que un clique es construido

Aplicar una regla local de la evaporación de la feromona.

Fin Para

Agregar-a-lista-de-prohibición()

mejor-recorrido = clique de longitud máxima en la población actual

$\Delta\tau = 1 / (1 + |G-mejor| - |mejor-recorrido|)$

SI (mejor-recorrido > G-mejor)

G-mejor = mejor-recorrido

Incrementar los rastros de feromonas en el mejor recorrido con $\Delta\tau$

Comprobar-mejora (actual_mejor_clique, previo_mejor_clique, SI)

Actualizar-prohibición (T)

Hasta Alcanzar el números de ciclos o encontrar una solución óptima

Tabla 3.5.- Algoritmo de optimización con colonia de hormigas basado en prohibición reactiva para el problema del clique máximo

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

3.6.3 Algoritmo ASMC

Este algoritmo fue propuesto por Bui T.N. y Rizzo J.R. en el 2004 en su artículo titulado “Finding maximum cliques with distributed ants”. En este algoritmo cada hormiga tiene solamente conocimiento local del grafo. Trabajando juntas las hormigas inducen un conjunto de vértices candidatos los cuales pueden formar parte del clique en construcción. El algoritmo fue diseñado para poderlo poner en ejecución fácilmente en un sistema distribuido.

3.6.4 Algoritmo ABOMC

Este algoritmo fue propuesto por Bui, T. N., T. H. Nguyen, and J. R. Rizzo Jr. (2009). en su artículo titulado “Parallel Shared Memory Strategies for Ant-Based Optimization Algorithms”. En este algoritmo se maneja un procesamiento en paralelo a través de una topología OpenMP en una maquina con Linux, con 8-Core Intel Xeon a 2.8 GHz y 16 Gb de memoria, este algoritmo fue probado con 121 benchmarks, la mitad de ellos son lo de la DIMACS. El algoritmo fue diseñado para poderlo poner en ejecución fácilmente en un sistema distribuido con diferentes topologías mencionadas en el artículo.

3.7 Conclusiones

Como se pudo ver actualmente existe una gran cantidad de algoritmos de colonias de hormigas que han sido desarrollados para resolver el problema del clique máximo, actualmente el mejor algoritmo que existe para resolver este problema es el “Marchiori”, y al mismo tiempo han ido evolucionando tanto los algoritmos como los benchmarks, siendo uno de los últimos utilizados los de la librería DIMACS. Es importante conocer las ventajas que ofrece cada uno de estos en relación con los demás y sus características específicas para poder desarrollar algoritmos mejores que, ya que puedes ser posible el utilizar

CAPÍTULO III.- ESTADO DEL ARTE DEL CLIQUE MÁXIMO

conjuntamente dos o más características de los algoritmos existentes con la finalidad de desarrollar nuevo algoritmos que puedan ser más eficientes.



CAPÍTULO IV.- HORMIGAS NATURALES

4.1. Introducción

En este capítulo se muestran algunos de los estudios realizados en las hormigas naturales, para esto se mencionan algunas de las especies que existen en el continente Americano y su importancia como parte de los ecosistemas, por otra parte se mencionan algunas preguntas sobre los comportamientos que se están estudiando o se desean estudiar, por ejemplo el cómo se comunican y colaboran entre sí, esto con la finalidad de poder entender más a fondo las bases de los algoritmos de colonias de hormigas.

4.2. Hormigas Naturales

Las hormigas son un grupo de insectos sociales pertenecientes a la familia *Formicidae* del orden *Hymenoptera*.



Este grupo ha tenido un notable éxito biológico, el cual se manifiesta en el gran número de especies, la diversidad de ambientes que ocupan y, sobre todo, en su extraordinaria abundancia en los ecosistemas terrestres.

Su éxito biológico se debe a que fue el primer grupo de insectos sociales con hábitos depredadores que vivió y forrajeó en el suelo y la hojarasca.

Figura. 4.1 Hormiga natural.

CAPÍTULO IV.- HORMIGAS NATURALES

Las hormigas viven en colonias y gracias a su mutua colaboración son capaces de mostrar comportamientos complejos y realizar tareas complicadas desde la óptica de una hormiga individual.

Un comportamiento característico de las colonias de hormigas es su capacidad de encontrar los caminos más cortos entre las fuentes de comida y el hormiguero. Este hecho no sería muy interesante si no tenemos en cuenta que la mayoría de las especies de hormigas son prácticamente ciegas, y no pueden apoyarse en pistas visuales para encontrar los caminos.

Algunas especies de hormigas depositan una sustancia química (feromona) mientras recorren el camino entre el hormiguero y la fuente de alimento. Si una hormiga no encuentra rastro de feromona a su paso, deambula al azar. Si existe un rastro de feromona, sigue el rastro como una tendencia probabilística. Si el rastro se divide en dos, una hormiga sigue elige con probabilidad más alta el rastro que contiene mayor concentración de feromona. Como a su vez, las hormigas depositan feromona, el camino se va reforzando consiguiendo caminos marcados con una cantidad elevada de feromona.

Las hormigas son consideradas un elemento central en la naturaleza debido principalmente a su papel como estructuradoras de las comunidades de plantas y por las diversas interacciones que establecen con otros organismos. La mirmecofauna es considerada como el estudio de las especies que representan las especies de hormigas en una zona determinada.

Darwin y las Hormigas

La documentación conservada de Darwin es fabulosa. Lo guardaba todo. Miles de cartas, notas, diarios, manuscritos, apuntes, dibujos, fotografías... La bibliografía darwiniana es inmensa. Las hormigas le interesaron de sobremanera, desde su arribada a Brasil en el viaje de exploración a bordo del Beagle, hasta sus observaciones y experimentos en los alrededores de su casa de Down, en Inglaterra. Allí mismo tenía de vecino a su amigo J. Lubbock, autor de una de las primeras obras de psicología experimental de los

CAPÍTULO IV.- HORMIGAS NATURALES

himenópteros (*Ants, bees and wasps*, 1882). Allí leyó, impresionado, el libro de P. Huber *Historia de las hormigas* (1810), que le llevaría a meditar largamente sobre el origen evolutivo del comportamiento esclavista en las hormigas, meditación plasmada en el capítulo dedicado al instinto del *Origen de las especies*. Las hormigas, además, le plantearon un serio problema a la hora de comprender, desde el punto de vista de la selección natural, la aparición y el polimorfismo de la casta obrera no reproductiva.

Para Darwin los insectos sociales eran un “abominable problema” para hacer de la selección natural una respuesta satisfactoria para la evolución de las especies. Si las obreras de una colonia son estériles y no dejan descendencia ¿cómo puede actuar la selección sobre ellas?. Este “abominable” problema no tuvo una aparente solución sino hasta mediados del siglo pasado, cuando el biólogo Hamilton en 1964 propuso la selección parental, como un compromiso entre genética, parentesco de obreras y selección para explicar la evolución y mantenimiento del comportamiento altruista en las hormigas y otros himenópteros sociales.

En la escasa tradición mirmecológica mucho antes del comienzo de la investigación académica en la década de los años setenta del siglo XX– destacan dos figuras cuyos estudios sobre las hormigas apenas son conocidos. El primero en el siglo XVIII, José Celestino Mutis (1732-1808), quien realizó numerosas observaciones sobre las hormigas legionarias y las cortadoras de hojas de Colombia durante más de 20 años. Se trata de un proyecto único para su época, que sitúa a Mutis como fundador de la mirmecología junto al francés Réaumur y al inglés Gould (estudiosos de las hormigas europeas). El segundo es Santiago Ramón y Cajal (1852-1934), que entre 1917 y 1921 llevó a cabo numerosos experimentos y observaciones acerca del comportamiento y las capacidades sensoriales de las hormigas. El manuscrito inédito, con más de 200 hojas –muchas de ellas ilustradas con dibujos del autor– se conserva en el Legado Cajal de Madrid.

Las hormigas están consideradas entre los organismos más importantes de los ecosistemas terrestres ya que constituyen una alta proporción de la biomasa animal llegando a constituir hasta el 10% del total en las selvas tropicales y los pastizales (Wilson 2000).





CAPÍTULO IV.- HORMIGAS NATURALES

Parece que las hormigas están en todas partes –en nuestras cocinas y hogares, en los árboles, aceras, céspedes y campos, troncos podridos, hojas caídas y siempre en los días de campo. No es de extrañar, que hay muchas. Aunque se han descrito más de 8.000 especies, se estima que podría haber hasta 20.000 especies, ¿Y cuál es el número de hormigas? Según E. O. Wilson, especialista en hormigas de la Universidad de Harvard, en un momento cualquiera hay en el planeta cerca de 10^{23} ó 100,000,000,000,000,000,000,000 de hormigas ocupadas en ganarse la vida. Juntas representan entre 10 y 15 por ciento del total de biomasa animal del mundo –más que el peso de todos los humanos juntos.












De la misma manera, las hormigas son un componente faunístico importante de las zonas áridas y semiáridas ya que presentan una alta riqueza de especies (Polis 1991, MacKay 1991). Por ejemplo, en el desierto Australiano se han registrado hasta 150 especies en una área menor a una hectárea (Morton 1982 en MacKay 1991), mientras que en Deep Canyon, California se han registrado hasta 59 especies de hormigas siendo la zona árida de Norteamérica con la mayor riqueza de especies (Polis 1991).

El comportamiento es un tipo de respuesta que el organismo puede dar a un estímulo interno o ambiental. Está determinado en parte por la herencia y el instinto y en parte por la experiencia. Una respuesta conductual requiere coordinación y comunicación a diferentes niveles, desde la célula hasta todo el organismo, por lo que es un área rica para una gran variedad de investigaciones.

Algunas Preguntas sobre el comportamiento de las hormigas son:

-  ¿Qué comen las hormigas?
-  ¿Qué tan lejos del nido van las hormigas a comer?
-  ¿Cuán activo es el nido a lo largo del día?
-  ¿Cuán activo es el nido en determinado momento?

CAPÍTULO IV.- HORMIGAS NATURALES

-  ¿Tienen las hormigas un tipo de comida preferido?
-  ¿Cuánto tiempo les toma a las hormigas encontrar comida?
-  ¿Qué tan rápido viajan las hormigas?
-  ¿Es la velocidad de las hormigas cuando salen de la colonia para ir a un lugar conocido donde hay comida diferente de cuando salen a buscar comida?
-  ¿Qué pasa si hay un cambio en el camino que siguen las hormigas?
-  ¿Qué pasa si hay una hormiga u otro insecto muerto en el camino de las hormigas?
-  ¿Hay alguna diferencia si la hormiga muerta en el camino es de la misma especie o especie diferente?
-  ¿Qué pasa si se toma una hormiga del camino y se le coloca en un nuevo lugar?
-  ¿Qué pasa si la especie de una colonia es colocada en otra?
-  ¿Qué hacen las hormigas si se les rocía con un tipo de polvo?
-  ¿Qué pasa cuando se molesta una colonia?

En la tabla siguiente se puede ver algunas de las especies de hormigas que existen en América así como su alimentación y distribución. Cabe mencionar que los algoritmos de colonias de hormigas están basados en el comportamiento de las hormigas forrajeras.

CAPÍTULO IV.- HORMIGAS NATURALES


| Especie | Gremio Alimentario | Distribución |
|------------------------------------|-------------------------|--|
| Subfamilia Ponerinae | | |
| Odontomachus Clarus Roger | Depredadoras | SO de Estados Unidos y México |
| Subfamilia Myrmicinae | | |
| Pogonomyrmex barbatus (Smith) | Granívoras | Centro y SO de Estados Unidos hasta S de México |
| Atta mexicana (Smith) | Micófagas | S de región Neártica y región Neotropical |
| Solenopsis xyloni McCook | Forrageras generalistas | S de Estados Unidos y N de México |
| Pheidole sp.1 | Granívoras | Trópicos y zonas de clima cálido |
| Pheidole sp.2 | Granívoras | Trópicos y zonas de clima cálido |
| Pheidole sp.3 | Granívoras | Trópicos y zonas de clima cálido |
| Pheidole sp.4 | Granívoras | Trópicos y zonas de clima cálido |
| Pheidole sp.5 | Granívoras | Trópicos y zonas de clima cálido |
| Pheidole sp.6 | Granívoras | Trópicos y zonas de clima cálido |
| Leptothorax (Macromischa) sp. | Forrageras generalistas | Zonas de clima templado |
| Leptothorax (Myrafant) sp. | Forrageras generalistas | Zonas de clima templado |
| Leptothorax sp.1 | Forrageras generalistas | Zonas de clima templado |
| Tetramorium spinosum Pergande | Forrageras generalistas | Zonas de clima templado |
| Strumigenys sp. | Depredadoras | Trópicos y zonas de clima cálido |
| Subfamilia Dolichoderinae | | |
| Dorymyrmex cf. flavus (McCook) | Forrageras generalistas | Florida, Texas y California |
| Dorymyrmex cf. insanus (Buckley) | Forrageras generalistas | Centro y SE de Estados Unidos y N de México |
| Dorymyrmex sp.1 | Forrageras generalistas | Región Neártica y Neotropical |
| Forelius sp. grupo pruinosus | Forrageras generalistas | S de Estados, México, Guatemala, Nicaqragua, Panamá y Cuba |
| Subfamilia Formicinae | | |
| Forelius mccooki McCook | Forrageras generalistas | S y Centro de Estados Unidos, México y Jamaica |
| Brachymyrmex sp. | Forrageras generalistas | Región Neártica y Neotropical |
| Camponotus (Myrmobrachys) | Forrageras generalistas | Cosmopolita |
| Rubrithorax Forel | | |
| Camponotus (Tanaemyrmex) | Forrageras generalistas | S de Texas y México |
| Atriceps (Smith) | | |
| Camponotus (T.) flavopubens | Forrageras generalistas | Cosmopolita |
| Emery | | |
| Camponotus (T.) festinates | Forrageras generalistas | Centro y S de Estados Unidos, México |
| Paratrechina mexicana (Forel) | Forrageras generalistas | S de Estados Unidos y N de México |
| Subfamilia Pseudomyrmecinae | | |
| Pseudomyrmex major Forel | Depredadoras | Región Neotropical y S de Neártica |
| Pseudomyrmex pallidus Smith | Depredadoras | S de Estados Unidos hasta Sudamerica |


Tabla. 4.1 Especies de hormigas naturales en America


CAPÍTULO IV.- HORMIGAS NATURALES


Algunas de estas especies viven organizadas en sociedades compuestas por castas o clases sociales (Weber, 1972).

La función específica de cada casta se detalla a continuación:

 **La reina o hembra fértil:** Su función es exclusivamente reproductiva. La reina copula y pone los huevos.

 **Los machos o zánganos:** Son los que se encargan de copular y preñar a las hembras fértiles, después de esto mueren.

 **Los soldados:** Son los que se encargan de la seguridad de la reina y de cuidar el camino por el cual las obreras transportan el material. Estas son hembras estériles que se diferencian porque son de mayor tamaño que los demás, poseen una cabeza más pronunciada y mandíbulas fuertes.

 **Las obreras:** Estas son estériles y no pueden reproducirse; sin embargo, son las que tienen la mayor carga de trabajo en el nido; se dividen a su vez en:

- **Cortadoras o recolectoras:** Son de menor tamaño que los soldados y se encargan de cortar y llevar el material hasta el nido, además de elaborar las cámaras dentro de los nidos.

CAPÍTULO IV.- HORMIGAS NATURALES

- **Jardineras o niñeras:** Son las más pequeñas, están encargadas de preparar la comida del hongo cortando finamente el material llevado por las recolectoras, cuidar huevos, gusanos y pupas, además de la limpieza del nido.



Figura 4.2 Hormiga Jardinera

Algunas especies de hormigas poseen sistemas de comunicaciones complejos; pero el mecanismo principal de comunicación es el contacto físico directo que realizan al tocarse con las antenas. Por lo cual no son hormigas “primitivas” o “simples” desde el punto de vista evolutivo. Poseen una organización social muy desarrollada y un repertorio muy amplio de conductas individuales y colectivas (Miramontes, 2000).

Las hormigas utilizan las antenas como medio de comunicación, éstas perciben olores y sabores, de esta manera pueden identificar los miembros de su colonia o a extraños. El método general de comunicación entre los individuos de una colonia es en forma química, ya que la sustancia utilizada para comunicarse es una feromona, propia de cada hormiga. Las hormigas regresan a su colonia por el rastro de feromona, que se desprende del extremo del abdomen cuando camina; cada colonia tiene un olor distinto con el que marcan su territorio de trabajo defendiéndolo de invasiones de otras colonias (Cherrett, 1976).

CAPÍTULO IV.- HORMIGAS NATURALES

Una colonia de hormigas, es una estructura altamente integrada y con orden, a este nivel, proviene de las interacciones masivas y coordinadas que son facilitadas por la existencia de canales efectivos de comunicación. La conducta social puede verse como consecuencia inevitable de las interacciones y debe considerarse como una propiedad robusta y genérica de los sistemas complejos ya sean naturales o artificiales (Miramontes, 2000).

Actualmente existe mucha bibliografía relacionada con el estudio del comportamiento de las hormigas un ejemplo de esto es el libro *La inteligencia de las hormigas* publicado en el 2007 el cual comprende dos largos textos mirmecológicos de la segunda mitad del siglo XIX, de G. Romanes y L. Büchner, el cual muestra una fotografía en la portada que representa una conducta elaborada. Hormigas legionarias (*Eciton hamatum*) forman un puente con sus propios cuerpos, ver figura 4.3.

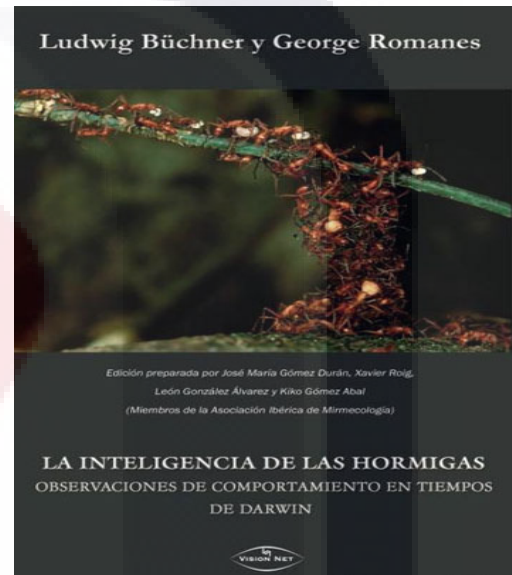


Figura 4.3 Libro La Inteligencia de las Hormigas

Las estrategias de comportamiento social en las hormigas son mucho más variadas, ricas y sorprendentes de lo creído hasta hace poco. Estos nuevos estudios comprometen la propuesta realizada en los 70 para explicar el alto grado de “sacrificio” o altruismo en estos insectos. Si la genética (bajo la mira neodarwinista) no puede explicar todas las manifestaciones “anómalas” en el comportamiento social de las hormigas, ¿dónde puede estar la luz en el túnel? Algunos investigadores piensan que la teoría y principios de la autorganización pueden ayudar a entender (Fernández, 2003).

CAPÍTULO IV.- HORMIGAS NATURALES

Nacida dentro de las ramas de física y química para “describir la emergencia de patrones macroscópicos más allá de procesos e interacciones definidas en el nivel microscópico”, la teoría de la autorganización se ha extendido al campo de la biología, en el estudio de los insectos sociales. La teoría se ha aplicado a diversos patrones de comportamiento en hormigas, abejas, avispas sociales y termitas (Bonabeau et al., 1997).

La autorganización posee cuatro “ingredientes” en las colonias de hormigas (Fernández, 2003).



Reacción positiva (amplificación): acciones simples que promueven la creación de estructuras (p.e. el reclutamiento de hormigas a una fuente de alimentos)



Reacción negativa: contrapeso a la reacción positiva, p.e. ayudando a la estabilización de un patrón colectivo (saturación, agotamiento, competencia) (p.e. cuando el número de obreras disponibles es insuficiente para explotar una fuente de alimento, o llegan hormigas de otra colonia o especie a esta fuente de alimentación)



Amplificación de fluctuaciones: por ejemplo, las acciones al azar (hormigas buscando alimento caminando sin rumbo fijo) pueden descubrir soluciones espontáneamente (toparse con una rica e inexplorada fuente de alimento)



Interacciones múltiples: Bajo una “mínima densidad de individuos mutuamente tolerantes” un individuo de una colonia puede generar una estructura autorganizada.

Consecuentemente, hay tres características de la autorganización (Fernández, 2003):



La creación de estructuras espaciotemporales en un medio inicialmente no homogéneo.



La posible coexistencia de diferentes estados estables (“multiestabilidad”).



La existencia de bifurcaciones en frente de parámetros variables.

CAPÍTULO IV.- HORMIGAS NATURALES

La aplicación de las ideas arriba expuestas a varios aspectos en el comportamiento de los insectos sociales ha mostrado algunas consecuencias interesantes. Sus investigadores se apresuran a aclarar que la autorganización no es “mecanismo universal” y que variadas acciones de comportamiento en insectos sociales pueden tener otra explicación. Existen dos aspectos atractivos en la autorganización:



Se puede aplicar a todos los insectos sociales. La selección parental descansa en el mecanismo de la haplodiploidía, universal en Hymenoptera, pero ausente en Isoptera (termitas).



Supera el problema de los bajos grados de cercanía genética entre individuos de una misma colonia, el cual ha sido un problema importante para la selección parental.

Acerca del origen y evolución de la sociabilidad, la autorganización no ofrece respuestas completas, aunque “no contradice sino que complementa las teorías en evolución”. De hecho, sus propagadores piensan que puede hablarse de selección operando sobre sistemas autorganizados, y que algunos de éstos puede ser “favorecidos por la evolución” (Fernández, 2003).

Es posible observar en el campo a las hormigas *Aphaenogaster senilis* haciendo emigraciones desde un nido a otro. Estas emigraciones tienen dos funciones principales totalmente diferentes que son (Cerdá, 2009):



Puede tratarse de la fundación de una nueva colonia por fisión colonial, en la que parte de las obreras marchan con una reina a un nuevo nido, mientras que el resto de la colonia permanece en el nido original con otra reina.



Puede ser una relocalización del hormiguero, en busca de mejores condiciones ambientales.

Un estudio de la Universidad británica de Leeds publicado en la revista PNAS acaba de echar por tierra la fama de cooperadoras y ecuanímes que tenían las hormigas.

CAPÍTULO IV.- HORMIGAS NATURALES

Hasta ahora se pensaba que la nutrición era el secreto de la "realeza" en estos insectos, y que el hecho de que algunas larvas consumieran ciertos alimentos permitía su desarrollo hasta convertirse en reinas. Sin embargo, Bill Hughes, investigador de la Universidad de Leeds (Reino Unido) acaba de demostrar que es la genética la que manda. Según el investigador, algunos machos pasan uno o varios genes que permiten a sus descendientes convertirse en reinas reproductoras, y no en "simples obreras" estériles. De esta forma, estas nuevas hormigas obtienen una ventaja y "estafan a sus hermanas altruistas, que nunca tendrán la oportunidad de convertirse en reinas", puntualiza Hughes.

El problema es que si demasiadas larvas se convierten en reinas, el desequilibrio podría ser descubierto por las hormigas trabajadoras, que podrían ponerse en contra de sus dirigentes. Por eso, los machos que portan el "ADN real" lo diseminan por diferentes colonias, para pasar más desapercibidos.

Según el doctor Hughes, cuando uno estudia insectos sociales como las hormigas y las abejas "es el aspecto cooperativo de su sociedad lo que primero destaca". Sin embargo, un análisis más profundo pone de manifiesto que la corrupción y el egoísmo están también muy presentes en sus colonias.

4.3. Conclusiones

Como se puede ver las hormigas tienen sociedades muy elaboradas y los algoritmos de colonias de hormigas están basadas en el comportamiento que tienen en la búsqueda de fuentes de alimentación de aquellas que están clasificadas según su gremio alimentario como hormigas forrajeras, sin embargo existen una gran cantidad de especies que entran en esta clasificación, actualmente se siguen realizando investigaciones sobre el comportamiento de las hormigas naturales sobre diversos comportamientos sociales de las diferentes especies y los resultados obtenidos de estas investigaciones podrían ser tomados en cuenta para modelar nuevos comportamientos o estrategias de búsqueda para modificar los algoritmos existentes de colonias de hormigas y ver si es posible mejorar su

CAPÍTULO IV.- HORMIGAS NATURALES

funcionamiento un punto importante que debe tomarse en cuenta por su importancia en el estudio del comportamiento de las hormigas es la teoría de la autorganización.



CAPÍTULO V.- APRENDIZAJE

5.1. Introducción

En esta sección hablaremos como una manera de inducción del aprendizaje en general y de los tipos de aprendizaje que existen desde el punto de vista de la psicología, para poder entender más a fondo algunos modelos que han sido desarrollados en el área de la Inteligencia Artificial para elaborar sistemas que sean capaces de aprender en base a sus experiencias, la clasificación de estos sistemas más común es en base a los aprendizajes supervisado y no supervisado, así mismo se describe de manera más amplia el aprendizaje por refuerzo, ya que es uno de los tipos de aprendizaje más utilizado en el desarrollo de sistemas inteligentes, por lo que se describen varios de los métodos que han sido desarrollados para tratar de simular este tipo de aprendizaje.

5.2. Aprendizaje


Se dice que una maquina aprende, si el sistema que transforma un mensaje de Entrada en una Salida, está sujeto a cierto Criterio de validez, y el sistema se ajusta a fin de que tienda a mejorar el funcionamiento; en este caso se dice que el Sistema Aprende.


Cuando los sistemas se ajustan o adaptan al conjunto de entradas que provienen del entorno; reciben la información y la almacena con el fin de reutilizarla en situaciones o patrones de estímulos semejantes. Todas las entradas a los que un sistema responde en un cierto contexto pueden no ser efectivas para producir una conducta de aprendizaje en otros contextos.


5.3. Tipos de Aprendizaje


Existen diferentes tipos de aprendizaje desde el punto de vista de los humanos y animales como son:

CAPÍTULO V.- APRENDIZAJE


 **Habitación.** Una respuesta que decae ante un conjunto de estímulos repetidos (o Continuos) no asociados a ningún tipo de recompensa o refuerzo. Implica tendencia a borrar todo tipo de respuesta ante un estímulo que no tiene importancia para la supervivencia. Sirve como filtro a un conjunto de estímulos no relevantes.

 **Asociativo.** Un evento permite predecir, con cierta confianza, la ocurrencia (o no) de otro. Alguien que conoce estas relaciones puede sacar provecho anticipándose a esos eventos y comportarse apropiadamente.


 **Condicionamiento.** Consiste en adquirir la Capacidad de Responder a un Estímulo Determinado con la misma Acción Refleja con que Respondería a otro Estímulo Condicionante (Refuerzo o Recompensa) cuando ambos Estímulos se presentan Concurrentemente (o sobre puestos en una Secuencia) un cierto número de veces.

 **Prueba y Error.** Cuando alguien permanece siempre Activo y su Atención se fija Primero aquí y luego allá probando todas las Posibilidades inimaginables hasta que de manera más o menos Accidental resuelve con Éxito la Tarea y Obtiene Recompensa. Requiere la Existencia del Refuerzo (o Recompensa) para animar la Selección de la Respuesta Deseada.

 **Latente.** Tiene lugar en Ausencia de Recompensa. Se aprende algo que permanece Latente hasta que es Necesario.

 **Imitación.** Implica Copiar una Conducta, Acción o Expresión Nueva o que resulta Imprescindible de Aprender si no es Copiada de otro Individuo.

CAPÍTULO V.- APRENDIZAJE

 **Impronta.** La Manera en que un Rango Específico de Estímulos es Capaz de Elicitar una Respuesta pudiendo ser Limitado y Refinado mediante la Experiencia. *Su Duración se Restringe a Periodos Sensitivos.*

En este caso se desea realizar un modelo de colonia de hormigas el cual este basado en el aprendizaje por refuerzo, por lo cual en base a las definiciones anteriores se pueden modelar distintos tipos de aprendizaje al utilizar recompensas como son: la habituación, el condicionamiento, prueba y error e impronta.

5.4. Aprendizaje por Refuerzo (Reinforcement Learning)

El aprendizaje por refuerzo es una aproximación computacional para comprender y automatizar el aprendizaje basado en objetivos y procesos de toma de decisiones. Un *agente* interacciona con un *entorno*, y realiza *acciones* en determinados *estados* del sistema, por lo cual recibe *recompensas*, positivas o negativas en función de los resultados de dichas acciones. De esta forma, evaluando las recompensas, aprende a elegir las mejores acciones en cada estado.


Uno de los mayores problemas a la hora de implementar sistemas de agentes es la dificultad del diseñador para prever todas las situaciones que un agente puede encontrar ni, por lo tanto, especificar la conducta óptima del agente por adelantado. Esto es especialmente grave en entornos de múltiples agentes donde la fuente de incertidumbre no es sólo la Naturaleza sino la presencia de otros agentes con creencias, objetivos e intenciones diferentes e incluso antagónicas. Es por lo tanto ampliamente reconocido en la comunidad de agentes que una de las características más importantes de los sistemas inteligentes es su capacidad para adaptarse y aprender (por ejemplo, Russell y Norvig, 1995). Este campo de aprendizaje en agentes ha sido especialmente fructífero en las últimas dos décadas, en


CAPÍTULO V.- APRENDIZAJE

particular en lo que se refiere al aprendizaje en sistemas de múltiples agentes (ver Sen, 1998; Weiss, 1997).

La mejor forma de aprender es mediante la interacción con el entorno en el que se está inmerso. Esta idea es bastante natural: un niño que juega solo, se mueve, toca y coge objetos, al mismo tiempo observa los resultados de estos actos. Simplemente recibe estímulos del ambiente, una gran cantidad de información de tipo *causa-efecto*, acerca de las consecuencias de sus actos. Así se aprende cómo actuar para alcanzar un objetivo. El aprendizaje mediante la interacción es un concepto fundamental en gran parte de las teorías acerca del aprendizaje y la inteligencia.

El aprendizaje por refuerzo es un fenómeno que cubre muchas facetas (adquisición de nuevo conocimiento, desarrollo de habilidades, organización del conocimiento adquirido, etc.), y que puede organizarse alrededor de dos métodos de aprendizaje (Mitchalski84):

 La Adquisición del conocimiento, o el aprendizaje de información simbólica organizada mediante leyes, modelos, teorías, reglas, etc... junto con la capacidad de aplicar esa información de una manera efectiva.

 El Refinamiento ó perfeccionamiento de Habilidades motoras o cognitivas a través de la práctica. El aprendizaje por refuerzo (AR) es una técnica que se encuentra a mitad de camino entre ambos métodos; si bien requiere un conocimiento estructurado del dominio para trabajar, se basa en la práctica para adquirir habilidades y aprender comportamientos.

El aprendizaje por refuerzo es diferente al aprendizaje supervisado, el cual es el tipo de aprendizaje más estudiado en las investigaciones en el área de aprendizaje máquina

CAPÍTULO V.- APRENDIZAJE

(machine learning), reconocimiento de patrones estadísticos y redes neuronales artificiales. El aprendizaje supervisado es donde los ejemplos son proporcionados por un supervisor externo. Este es un tipo importante de aprendizaje, pero solo no es adecuado para aprender a partir de iteraciones.

Uno de los desafíos que se presentan en aprendizaje por refuerzo y no en otras clases de aprendizaje es el equilibrar entre la explotación y exploración. Para obtener una gran recompensa, un agente con aprendizaje por refuerzo debe preferir las acciones que ha intentado en el pasado y ha encontrado que son efectivas para producir una recompensa. Pero para descubrir tales acciones, tiene que intentar las acciones que no ha seleccionado antes. El agente tiene que explotar lo que sabe ya para obtener la recompensa, pero también tiene que explorar para hacer mejores selecciones sobre las acciones en el futuro. El dilema es que ni la exploración ni la explotación pueden ser perseguidas exclusivamente sin fallar en la tarea. El agente debe intentar una variedad de acciones y favorecer progresivamente las que parezcan ser las mejores. En una tarea estocástica, cada acción se debe intentar muchas veces otra vez a una estimación confiable de su recompensa prevista. El dilema de la exploración-explotación ha sido intensamente estudiado por los matemáticos por muchas décadas. Por ahora, se hace notar que la aplicación entera del equilibrio entre explotación y explotación no se presenta en el aprendizaje supervisado como es definido generalmente.

Otra característica clave del aprendizaje por refuerzo es que considera explícitamente el problema entero de una meta-dirigida por el agente que interactúa con un ambiente incierto.

Esto está en contraste con muchos acercamientos que consideren sub-problemas sin la dirección de cómo puede ser que quepan en un cuadro más grande. Por ejemplo, hemos mencionado que mucha de investigación del aprendizaje máquina está relacionada al aprendizaje supervisado sin especificar explícitamente cómo tal capacidad finalmente sería útil.

CAPÍTULO V.- APRENDIZAJE

Otras investigaciones han desarrollado teorías de la planificación con metas generales, pero sin la consideración del rol de planeación en la toma de decisión en tiempo real, o la cuestión de donde los modelos de predicción necesarios. Aunque estos acercamientos hayan rendido muchos resultados útiles, su foco en sub-problemas aislados es una limitación significativa.

El aprendizaje por refuerzo toma la parte opuesta, comenzando con un agente completo, interactivo. Todos los agentes con aprendizaje por refuerzo tienen metas explícitas, pueden detectar aspectos de sus ambientes, y pueden escoger las acciones que influyen sus ambientes.


Por otra parte, se asume generalmente desde el principio que el agente tiene que funcionar a pesar de incertidumbre significativa sobre los ambientes a los que se enfrenta. Cuando el aprendizaje por refuerzo implica planeación, tiene que tratar la interacción entre el planeamiento y la selección de una acción en tiempo real, así como el cuestionarse sobre cómo los modelos ambientales se adquiere y mejoran.


Cuando el aprendizaje por refuerzo implica el aprendizaje supervisado, hace que por las razones específicas que determinan cuales capacidades son críticas y cuáles no. Para aprender la investigación para hacer progreso, los sub-problemas importantes tienen que ser aislados y ser estudiados, pero deben ser los sub-problemas que desempeñan papeles claros en agentes completos, interactivos, incluso si todos los detalles del agente completo no pueden todavía completados.


5.4.1. Elementos del Aprendizaje por Refuerzo

Más allá del agente y de los ambientes, uno puede identificar cuatro subelementos principales de los sistemas de aprendizaje por refuerzo: una política, una función de la recompensa, una función de valor, y, opcionalmente, un modelo del ambiente.

CAPÍTULO V.- APRENDIZAJE

 Una política define la manera en que los agentes aprenden un comportamiento en un momento dado. Por lo general, una política es un trazado de los estados percibidos del ambiente a las acciones que se tomarán cuando se esté en estos estados. Esta corresponde a qué en psicología sería llamada un sistema de reglas o de asociaciones de estímulo-respuesta. En algunos casos la política puede ser una función simple o una tabla de operaciones de búsqueda, mientras que en otros puede implicar el cómputo extenso tal como un proceso de la búsqueda. La política es la base de un agente de aprendizaje por refuerzo en el sentido que solamente es suficiente determinar el comportamiento. Generalmente las políticas pueden ser estocásticas.

 Una función de la recompensa define la meta en un problema de aprendizaje por refuerzo. En general, traza cada estado percibido (o pares del estado-acción) del ambiente a un número, una recompensa, indicando la deseabilidad intrínseca de ese estado.

 Un agente con aprendizaje por refuerzo tiene el único objetivo de maximizar la recompensa total que recibe a largo plazo. La función de la recompensa define cuáles son los buenos y malos acontecimientos para el agente. En un sistema biológico, no sería inadecuado identificar recompensas con placer y dolor. Las características inmediatas y de definiciones de los problemas hechos frente por el agente. Como tal, la función de la recompensa debe necesariamente ser inalterable por el agente. Puede, sin embargo, servir como base para alterar la política. Por ejemplo, si una acción seleccionada por la política es seguida por una recompensa baja, después la política puede ser cambiada para seleccionar alguna otra acción en esa situación en el futuro. Generalmente las funciones de recompensa pueden ser estocásticas.

CAPÍTULO V.- APRENDIZAJE

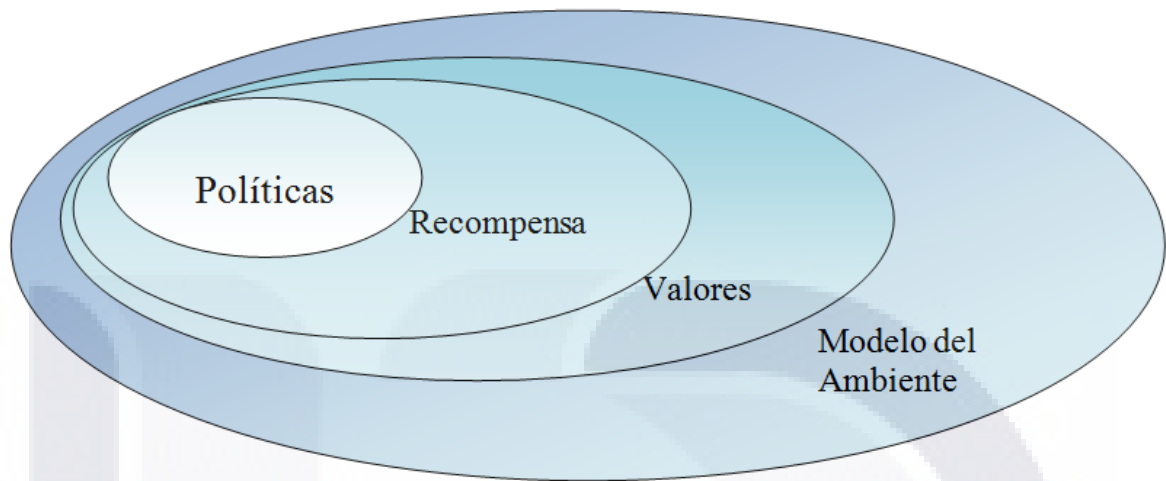


Figura. 5.1 Componentes del aprendizaje por Refuerzo.

Actualmente el aprendizaje por refuerzo se ha utilizado en combinación con diferentes técnicas evolutivas como Algoritmos Genéticos en (Pettinger J. E. and Everson R. M. 2003), Colonias de Hormigas (Gambardella, L.M. y Dorigo, M. 1995).

5.4.2. Aprendizaje del refuerzo de habilidades

Según la vista “estándar” de Aprendizaje por Refuerzo (e.g., [11]) la interacción del agente-ambiente se prevé como la interacción entre un regulador (el agente) y el sistema controlado (el ambiente), con una señal especializada de la recompensa viniendo de un “crítico” en el ambiente que evalúa (generalmente con un valor en una escala de recompensa) el comportamiento del agente (fig. 1A). El agente aprende mejorar su habilidad en controlar el ambiente en el sentido de aprender cómo aumentar la cantidad total de recompensa que recibe en un cierto plazo del crítico.

CAPÍTULO V.- APRENDIZAJE

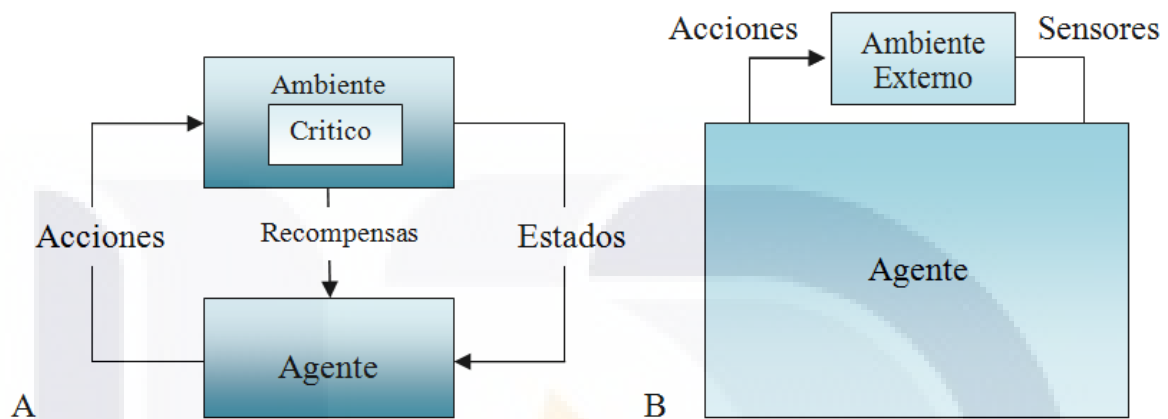


Figura 5.2 Interacción del ambiente y el agente en el aprendizaje por refuerzo en A y en B un agente reactivo simple

Sutton y Barto precisan que uno no debe identificar este agente de aprendizaje por refuerzo con un animal o robot. Las señales de la recompensa de un animal son determinadas por procesos dentro de su cerebro y no solo del estado externo sino también el estado interno del animal. El crítico está en la cabeza del animal. La fig. 1B hace esto más explícito “descomponiendo en factores” el ambiente de fig. 1A en un ambiente externo y un ambiente interno, este último contiene al crítico que determina la recompensa primaria. Este esquema todavía incluye los casos en los cuales la recompensa es esencialmente un estímulo externo (e.g., una palmadita en la cabeza o una palabra de la alabanza). Éstos son simplemente estímulos traducidos por el ambiente interno para generar el nivel apropiado de recompensa primaria.

La práctica generalmente en la aplicación de algoritmos de aprendizaje por refuerzo es formular el problema que uno quiere que el agente aprenda cómo solucionar y definir una

CAPÍTULO V.- APRENDIZAJE

función de la recompensa adaptada especialmente para este problema. El ingenio es necesario a veces para hacer una función apropiada de la recompensa. El punto de partida para nuestro acercamiento es observar que contiene el ambiente interno, entre otras cosas, el sistema de motivación del organismo, que necesita ser un sistema sofisticado que no debe tener que ser reajustado para diversos problemas. Creando un sistema diferente de motivación con un propósito especial.

El Aprendizaje por Refuerzo (RL) (Kaelbling, Littman, y Moore, 1996) abarca a una familia de algoritmos incrementales que construyen una política de control con la experimentación del mundo real. Una clave para el problema del escalamiento en aprendizaje por refuerzo es que en dominios grandes un número enorme de decisiones deben ser hechas. Por lo tanto, en vez del aprendizaje usando acciones primitivas individuales, un agente podría potencialmente aprender mucho más rápidamente si puede abstraer las innumerables microdecisiones, y se centra en lugar de otro en un pequeño sistema de la decisión importante. Esto plantea inmediatamente la cuestión de cómo reconocer las estructuras jerárquicas dentro de sistemas de aprendizaje y de cómo aprender las estrategias para la toma de decisión jerárquica. Dentro del paradigma de aprendizaje por refuerzo, un camino para hacer esto es introducir sub-metas con sus propias funciones de la recompensa, aprender las políticas para alcanzar estas sub-metas, y después incluir estas políticas como acciones. Esta estrategia puede facilitar transferencia de la habilidad a otras tareas y acelerar el aprendizaje. Es deseable que el agente con aprendizaje por refuerzo descubra las sub-metas automáticamente.

Varios investigadores han propuesto métodos los cuales aprenden políticas para un conjunto de tareas que son examinadas para las concordancias (Thrun y Schwartz, 1995) o son probabilísticamente combinadas para formar las nuevas políticas (Bernstein, 1999). Sin

CAPÍTULO V.- APRENDIZAJE

embargo, ninguno de los métodos de aprendizaje por refuerzo introduce sub-metas. En otro trabajo, se eligen las sub-metas en base en la información sobre la frecuencia en que un estado fue visitado durante la adquisición de la política o basado en la recompensa obtenida. Digney (Digney 1996, 1998) elige los estados que se visitan con frecuencia o los estados donde el gradiente de la recompensa está alto como sub-metas. Similarmente, McGovern (McGovern y Barto, 2001) utiliza la densidad diversa para descubrir sub-metas útiles automáticamente. Sin embargo, en el caso de ambientes y de recompensas más complicados puede ser difícil acumular y clasificar los conjuntos de trayectorias acertada y fracasada necesarias para calcular la medida de densidad o contar la frecuencia. Además, estos métodos no permiten que el agente descubra las sub-metas que no son explícitamente parte de las tareas usadas en el proceso de descubrimiento de ellas. El enfocarse en el descubrimiento de sub-metas para buscar un modelo para el aprendizaje de la política para ciertas características estructurales. Este método puede descubrir sub-metas incluso si no son una parte de la trayectoria acertada de la política. Si el agente puede descubrir estos estados sub-metas y aprender políticas para alcanzarlas, puede incluir estas políticas como acciones y utilizarlas para la exploración eficaz, así como para acelerar el aprendizaje en otras tareas en las cuales las mismas sub-metas sean útiles.

5.4.3. El Problema del Aprendizaje por Refuerzo

El problema del aprendizaje por refuerzo se considera un modelo sencillo del problema más general de aprender a alcanzar un objetivo o meta. El aprendiz, que elige las acciones, se denomina agente. Todo con lo que interactúa con el agente se conoce como ambiente ó entorno.

El agente es capaz de percibir el ambiente a través de sensores y selecciona acciones en base a lo que percibe de este, y el entorno responde ante esas acciones y presenta nuevas situaciones al agente.

CAPÍTULO V.- APRENDIZAJE

El entorno también da lugar a la aparición de recompensas, una señal especial cuyo valor intenta maximizar el agente a lo largo del tiempo. Una especificación completa de un entorno define una tarea o acción, con la cual se evalúa la recompensa de dicha acción para obtener el aprendizaje por refuerzo.

5.4.4. Métodos de aprendizaje por refuerzo

Existen métodos que se basan en funciones de estimación para los valores que representan la recompensa o castigo de una solución, sin embargo no es estrictamente necesario utilizarlas para resolver problemas de aprendizaje con refuerzo. Algunos métodos como los algoritmos genéticos, la programación genética, recocido simulado, colonias de hormigas y otros métodos de optimización de funciones se han utilizado para resolver este tipo de problemas.

Si el espacio de políticas es suficientemente pequeño, o puede estructurarse de tal modo que las políticas adecuadas sean fáciles de encontrar, entonces los métodos evolutivos suelen ser efectivos. Además, si el agente no puede percibir con suficiente precisión el estado de su entorno, los métodos evolutivos presentan más ventajas.

No obstante, hay que considerar que el aprendizaje por refuerzo, tal y como se ha definido, implica aprender a la vez que se interacciona con el entorno, algo que los métodos evolutivos no hacen por sí solos. Los métodos capaces de aprovechar los detalles de la interacción con el comportamiento del entorno son mucho más eficientes que los algoritmos evolutivos sin aprendizaje en un gran número de casos. Los métodos evolutivos ignoran gran parte de la estructura del problema de aprendizaje por refuerzo, no se percatan del hecho de que la política que se busca es una función que relaciona estados con acciones; no tienen en consideración por qué estados pasa un individuo a lo largo de su vida, o qué acciones elige.

CAPÍTULO V.- APRENDIZAJE

A pesar de que la evolución y el aprendizaje comparten muchas características y pueden trabajar juntos, como de hecho ocurre en la naturaleza, los métodos evolutivos no se consideran por sí solos especialmente adecuados para problemas de aprendizaje. Por ello, en este trabajo se trata de implementar aprendizaje por refuerzo dentro de las colonias de hormigas.

Existen varios tipos de métodos para la resolución del problema de aprendizaje por refuerzo. Se trata métodos de estimación de intervalos, teorema de Bayes, la programación dinámica, los métodos de Monte Carlo y el aprendizaje por diferencia temporal.

Algunos de estos métodos son:



Aprendizaje por Diferencia Temporal



Sarsa



Actor-crítico.



Método H-learning



Método Q-Learning



Método R-Learning



Métodos de estimación de intervalos



Métodos con Teorema de Bayes

5.5. Conclusiones

El aprendizaje es una característica de los sistemas para que estos sean más autónomos y tengan la capacidad de aprender de su actuación, al ser capaces de aprender de sus acciones tomadas en el pasado, actualmente existen diversos estilos de aprendizaje entre los que se encuentra el aprendizaje supervisado, no supervisado y por refuerzo, los cuales han sido utilizados para el desarrollo de sistemas inteligentes (agentes que aprende), generándose de esta manera diferentes métodos que simulan estos tipos de aprendizaje, sin embargo se le dió más importancia a conocer los distintos métodos del aprendizaje por refuerzo con el que se desarrolló un algoritmo de colonia de hormigas con un aprendizaje diferencial.

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO VI.- MINERÍA DE DATOS

6.1 Introducción

En este capítulo se da una introducción a la minería de datos la cual es una de las herramientas más importantes para el descubrimiento del conocimiento, se describe la importancia que tiene dicha herramienta actualmente como un medio para realizar predicciones, tomar de decisiones, analizar comportamientos sociales, entre otras aplicaciones, así mismo se describe el proceso que se debe seguir para llevarla a cabo y se habla un poco sobre su situación actual.

6.2 Minería de Datos

El descubrimiento del conocimiento (Knowledge Discovery) y la minería de datos (Data Mining) son poderosas herramientas de análisis de datos de gran alcance y se predice que se pueden convertir en las herramientas analíticas más usadas frecuentemente en un futuro. La rápida difusión de estas tecnologías pide una investigación urgente de su impacto social.

Los términos “descubrimiento del conocimiento” y “minería de datos” se utilizan para describir la extracción no trivial o implícito”, previamente desconocido y potencialmente útil de la información de los datos (Wahlstrom y Roddick).

El descubrimiento del conocimiento es un concepto que describe el proceso de la búsqueda en grandes volúmenes de datos de patrones que se pueden considerar conocimiento sobre los datos. La rama más conocida del descubrimiento del conocimiento es la minería de datos.

CAPÍTULO VI.- MINERÍA DE DATOS

La minería de datos, consiste en la extracción de la información oculta en grandes bases de datos, es una nueva tecnología de gran alcance y potencial. La minería de datos es un proceso de descubrimiento del conocimiento en grandes y complejos conjuntos de datos, refieren al proceso de extracción o “minería” de grandes cantidades de datos. Por otra parte, la minería de datos se puede utilizar para predecir un resultado para una entidad dada (Hernández te al., 2006). Los algoritmos de closterización en la minería de datos son equivalentes a la tárea de identificar grupos de expedientes que sean similares entre sí mismos pero diferente del resto. (Varan, 2006). La minería de datos es un campo multidisciplinario con muchas técnicas. Con estas técnicas usted puede crear un modelo de minería que describa los datos que usted utilizará. (ver fig. 6.1).

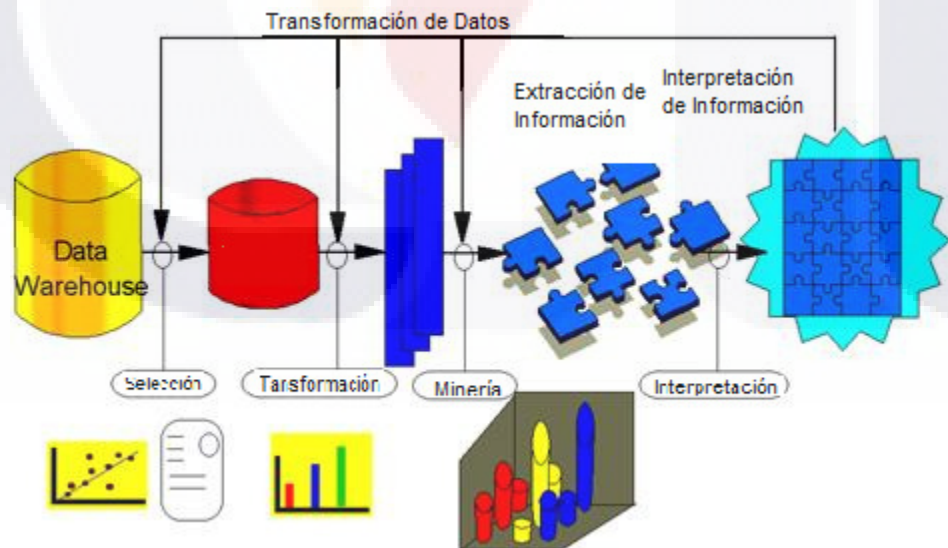





Figura 6.1. Proceso de minería de Datos


CAPÍTULO VI.- MINERÍA DE DATOS

Algunos elementos en proceso de la minería de datos son el conjunto de datos, pre-procesamiento, resultados, árboles de decisión y reglas de la asociación.

 Conjunto de datos. Un conjunto de datos es una colección de datos, presentada generalmente de forma tabular. Cada columna representa una variable particular. Cada fila corresponde a un miembro dado del conjunto de datos en la pregunta. Enumera los valores para cada uno de las variables, tales como altura y peso de un objeto o valores de números al azar. Cada valor se conoce como dato. El conjunto de datos puede abarcar los datos para uno o más miembros, correspondiente al número de filas.

 Pre-procesamiento. La minería de datos requiere el proceso de pre-procesamiento substancial de los datos. Este es especialmente en el caso para hacer que los datos sean comparables, todos los datos necesitan ser normalizados.

 Resultados generales. Esta actividad se relaciona con el resultado final del esfuerzo completo para descubrir si algunas relaciones importantes pudieron haber sido pasadas por alto. Este es el paso donde una decisión se toma en base a las otras decisiones que se han hecho. Si todos los pasos anteriores eran satisfactorios y los resultados satisfacen objetivos del problema, el proyecto puede moverse a su fase concluyente.

 Árboles de decisión. Los árboles de decisión son herramientas de gran alcance y populares para la clasificación y la predicción. La atracción de los árboles de decisión

CAPÍTULO VI.- MINERÍA DE DATOS

es debido al hecho que estos árboles de decisión representa reglas. La inducción del árbol de decisión es un acercamiento inductivo típico para aprender conocimiento en la clasificación. Los requerimientos claves para hacer la minería con los árboles de decisión son: descripción del Atributo-valor, clases predefinidas, clases discretas y suficientes datos.

Reglas de la asociación. Las reglas de la asociación describen los acontecimientos que tienden a ocurrir juntos. Son declaraciones formales bajo la forma de $X \Rightarrow Y$, donde si sucede X, entonces probablemente Y sucederá.

La minería de datos actual mente tiene diversas aplicaciones y ha sido utilizada en: detección de intrusos (Bloedorn, 2001, Dokas et al., 2002), redes sociales (Cai et al., 2005, Ponce et al., 2007), minería de datos en la web (Garofalakisn et al.1999, Ponce et al., 2009), determinación de secuencias de ADN (Ponce et al., 2009).

6.3 Closterización




La agrupación o closterización consiste en agrupar un conjunto de elementos ó datos, sin tener clases predefinidas, basándose en la similitud de los valores de los atributos de cada dato. Esta agrupación, se realiza de forma no supervisada a diferencia de la clasificación, ya que no se conocen de manera apriori las clases del conjunto de datos de entrenamiento. La costerización identifica closters ó regiones densamente pobladas, de acuerdo a una medida de distancia, en grandes conjuntos de datos (Chen et al., 1996). Por otra parte la closterización se basa en maximizar la similitud de las instancias en cada closter y minimizar la similitud entre closters (Han & Kamber, 2001).

6.4 Clasificación

La clasificación es una agrupación supervisada que se utiliza para ordenar y categorizar un conjunto de datos basado en los valores de sus atributos en distintas clases. La clasificación encuentra las propiedades comunes entre un conjunto de objetos y los clasifica en diferentes clases, de acuerdo a un modelo de clasificación.

Para construir este modelo, se utiliza un conjunto de entrenamiento, en el que cada instancia consiste en un conjunto de atributos y el valor de la clase a la cual pertenece (Chen et al., 1996).

Para realizar la clasificación es necesario:

-  Crear un modelo base en la distribución de los datos.
-  El modelo es usado para clasificar nuevos datos.
-  Dado el modelo, una clase puede ser predicha para nuevos datos.

6.5 Conclusiones

Como se pudo ver la minería de datos tiene una gran cantidad de aplicaciones como son la predicción, modelado de sociedades, closterización, toma de decisiones, etc.

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

7.1 Introducción

En este capítulo se muestran los algoritmos de colonias de hormigas que han sido desarrollados para realizar tareas de minería de datos como son la closterización y la clasificación, por lo cual se describe las características básicas de estos algoritmos, algunos de los algoritmos analizados son: ANT-LGP, ANT-BASED Clustering, AntClass, Ant-Miner, entre otros.

7.2 Algoritmos de Colonias de Hormigas aplicados a Minería de Datos

El estudio de el comportamiento de las colonias de la hormiga y sus capacidades de auto-organización es un área de interés de la recuperación/administración del conocimiento y de las ciencias de los sistemas de apoyo a la decisiones, ya que este proporciona los modelos de la organización adaptativa distribuida, que son útiles para solucionar la problemas de optimización difícil, clasificación, y los problemas de control distribuido, entre otros (Ramos et al. 2002, Ramos y Merelo, 2002, Ramos y Almeida 2000).



Figura 7.1. Hormigas excavando

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

7.2.1 Algoritmo ANT-LGP

Uno de los algoritmos de closterización con colonias de hormigas para descubrir patrones es el ANT-LGP. Este algoritmo utiliza una closterización y una aproximación de programación genético lineal para analizar la tendencia de los visitantes a sitios web. Los resultados demuestran claramente que el algoritmo realiza la closterización cuando está comparado a un mapa de auto-organización, aunque la exactitud del funcionamiento no es que eficiente cuando está comparada con una closterización evolutiva-difusa (i-miner) (Abraham A., 2003).

7.2.2 Algoritmo ANT-BASED Clustering

El Algoritmo ANT-BASED Clustering es un heurístico bio-inspirado para las tareas generales de closterización. Se ha aplicado varios de los problemas que se presentaban en comercio como son: diseño de circuitos, el text-mining. Este propone un esquema que permite la interpretación imparcial de las soluciones de las closterizaciones obtenidas, el cual es utilizado para conducir una evaluación completa del algoritmo. Ellos utilizan aplicaciones para el análisis de los datos de los cuales uno es de datos verdaderos y otro con datos artificiales. Sus resultados los comparan con técnicas de closterización establecidas como son: K-means, Average link e One Dimensional Self-Organising Maps (1D-SOM), y dan como conclusión que el algoritmo ANT-BASED Clustering es una alternativa robusta y viable (Handl et al. 2003a, Handl et al. 2003b).

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

7.2.3 Algoritmo AntClass

El Algoritmo AntClass es un algoritmo de colonia de hormigas híbrido para la closterización de datos, este algoritmo descubre automáticamente clusters en datos numéricos sin un conocimiento aprioris sobre el número de clases, sin una partición inicial y sin parámetros complejos. Este utiliza principios exploratorios y estocásticos de una colonia de hormigas con principios determinísticos y heurísticos del algoritmo K-means, las hormigas trabajan en un tablero de dos dimensiones en el cual pueden tomar y depositar objetos, la hormiga puede levantar los objetos y colocarlos en una pila de objetos ya existente, dependiendo de la similaridad del objeto y la pila, el algoritmo K-means provee la convergencia para la closterización con colonia de hormiga, este es un algoritmo de closterización no supervisado (Monmarché et al. 1997, Monmarché 1999).

La integración de K-means en el AntClass, el K-means es un algoritmo iterativo el cual requiere de una partición inicial de los datos. En el algoritmo AntClass esa partición está dada por las hormigas.

A continuación se muestra el algoritmo base del AntClass

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

1. Initialize randomly the ants positions,
2. Repeat
3. For each ant ant_i Do
 - (a) Move ant_i ,
 - (b) If ant_i does not carry any object Then look at the 8 cells in the neighborhood of ant_i location and possibly pick up an object (see algorithm in figure 4),
 - (c) Else (ant_i is already carrying an object O) look at the 8 cells around ant_i and possibly drop O (see the algorithm in figure 5),
4. Until Stopping criterion.

Tabla 7.1 Algoritmo AntClass

Como se puede ver este algoritmo es fácil de implementar pero tiene sus limitaciones debido a que trabaja sobre el tablero de dos dimensiones. El movimiento de las hormigas no es totalmente aleatorio, la hormiga anti selecciona una posición aleatoria de las ocho posible, la hormiga anti se mueve a la siguiente posición con una probabilidad P-dirección.

El criterio de parada de este algoritmo es simplemente el número de iteraciones.

7.2.4 Algoritmo Ant-Based Clustering con Neural Gas

Los algoritmos convencionales Ant-Based Clustering y las redes neural gas se combinan para producir un algoritmo de clasificación no supervisado que explota las fortalezas de ambas técnicas. El algoritmo Ant-Based Clustering detecta clases existentes en un conjunto de datos de entrenamiento, y al mismo tiempo entrena con varias redes neural gas. En una segunda etapa, estas redes se utilizan para clasificar vectores previamente no vistos como entradas en las cuales detecta las clases para el algoritmo Ant-Based. El algoritmo

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

propuesto elimina la necesidad de cambiar el número de agentes y las dimensiones del ambiente al ocuparse de las grandes bases de datos. Montes de Oca

Este fenómeno ha inspirado la creación de algoritmos de closterización para el análisis de datos exploratorio donde insectos son simulados por agentes reactivos simples que actúan en un tablero de dos dimensiones (Lumer y Faieta,1994). En este trabajo se propone una hibridación de un algoritmo Ant-Based Clustering con el growing neural gas networks.

Hibridación del Modelo

El modelo híbrido propuesto consiste en un sistema de agentes, donde cada uno contiene un growing neural gas networks. La diferencia principal del modelo híbrido es: los agentes aprenden la topología de su ambiente y como la exploran. La growing neural gas networks de un agente se está adaptando siempre para reflejar la distribución espacial de datos sobre el tablero, el agente utiliza la red para dirigir su búsqueda para localizaciones de inclinaciones favorables. Siempre que un agente tome un objeto de los datos, lo clasifica usando su propio neural gas network. Para predisponer su camino aleatorio para la localización de la neurona del ganador. La neurona del ganador se debe situar cerca de los objetos más similares a tomar porque la red refleja la distribución espacial de objetos en el ambiente. El objetivo principal de esto es crear clusters de más de alta calidad.

Después de que se cumpla un cierto criterio de parada, generalmente un número predefinido de iteraciones, cada agente tiene una neural gas network. Sin embargo, individualmente las redes pueden no cubrir el ambiente entero, una red colectiva es y se

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

utiliza en una segunda fase para clasificar más lejos los vectores de la entrada que no fueron utilizados durante la fase del entrenamiento. El vecindario es un área cuadrada de las localizaciones de 3×3 en el tablero.

Si más de una unidad está en la vecindad del objeto de datos, la unidad más cercana se selecciona para ser parte de la red colectiva. Por lo tanto, los clusters descubiertos por el algoritmo ant-based clustering se utilizan como las clases en las cuales los nuevos patrones de la entrada van a ser clasificados.

El algoritmo no supervisado resultante completo de la clasificación es:

1. Fase de la inicialización. (General).

- (a) Dispersar aleatoriamente los objetos de datos en un tablero cuadrado toroidal.
- (b) Cree y coloque aleatoriamente los agentes en el tablero.
 - i. Inicialice una growing neural gas network embebida en cada agente.

2. fase de Entrenamiento-closterización. (Por agente).

- (a) Movimiento aleatoriamente.
- (b) Si hay cualquier objeto de datos dentro del área de percepción entonces se entrena la red embebida con ellos.
- (c) Continúe hasta que se cumpla un criterio de parada, con el cambio siguiente:

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

- i. Si un objeto acaba de cogerse, clasifíquelo usando la red.
- ii. Fije el título hacia la localización de la neurona del ganador.

3. Fase de la clasificación. (General).

- (a) Extraer clusters del tablero. Los clusters se convierten en clases para la clasificación.
- (b) De todas las redes individuales, determine qué neuronas serán parte de la red colectiva.
- (c) Clasifique los vectores de entrada para la prueba con la red colectiva.

En dicho trabajo se hace un análisis de la importancia de la comunicación entre los agentes, la comunicación entre agentes en sistemas de inteligencia colectiva (swarm intelligence) y más generalmente en sistemas multiagentes, es crucial coordinar las actividades de los agentes para alcanzar una meta particular a nivel colectivo. De la perspectiva de un agente, el problema consiste en el establecimiento de las políticas de la comunicación que determinan lo que, cuando, y cómo comunicase con otros agentes. Las políticas de la comunicación dependerán generalmente de la naturaleza del problema que se esté solucionado. Esto significa que la solubilidad de problemas por los sistemas inteligentes colectivos depende entre otras cosas, de las políticas de la comunicación de los agentes, y el ajuste un conjunto incorrecto de políticas en los agentes puede dar lugar en encontrar soluciones pobres o aún a la insolubilidad de problemas (Montes de Oca).

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

Como caso de estudio, está el ver los efectos de dejar a los agentes utilizar diversas políticas de la comunicación en los algoritmos ant-based clustering. Para ver los efectos de usar diversas políticas de la comunicación sobre de estos algoritmos.

7.2.5 Algoritmo ANT-Miner

Otro algoritmo para la minería de datos es el llamado Ant-Miner (Ant Colony-based Data Miner). La meta del Ant-Miner es extraer reglas de clasificación de los datos. El algoritmo es inspirado por la investigación sobre el comportamiento de las colonias de la hormiga y de los conceptos y de los principios de minería de datos. El funcionamiento del Ant-Miner fue comparado con CN2, un algoritmo bien conocido de la minería de datos para la clasificación, en seis conjuntos de datos del dominio público. Los resultados proporcionan evidencia de: (a) El algoritmo Ant-Miner es competitivo con CN2 con respecto a exactitud; y (b) las listas de la regla descubiertas por el Ant-Miner son considerablemente más simples que las descubiertas por CN2 (Rafael S. Parpinelli).

El Ant-Miner tiene la tarea de llevar a cabo la clasificación que es un proceso de la minería de datos. En esta tarea la meta es asignar cada caja (objeto, expediente, o caso) a una clase, fuera de un sistema de clases predefinidas, basado en los valores de algunas cualidades (llamadas las cualidades del calculador) para el caso. En el contexto de la tarea de la clasificación de la minería de datos, el conocimiento descubierto se expresa a menudo bajo la forma de reglas SI ENTONCES, como sigue:

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

SI ENTONCES < class>. El antecedente de la regla (SI parte) contiene un sistema de condiciones, conectado generalmente por un operador lógico de la conjunción (Y). Aquí se refiere a cada condición de la regla como término, de modo que el antecedente de la regla sea una conjunción lógica de términos en la forma: SI term1 Y term2 Y... cada término es un triple, por ejemplo. <atributo, *operador*, *valor*>, tal como <Genero = femenino>.

El consecuente de la regla (ENTONCES parte) especifica la clase prevista para los casos cuyas cualidades del calculador satisfacen todos los términos especificados en el antecedente de la regla. De un punto de vista de la minería de datos, esta clase de representación de conocimiento tiene la ventaja de ser intuitivo comprensible para el usuario, mientras el número de reglas descubiertas y el número de términos en antecedentes de la regla no sean grandes. Lo mejor de nuestro conocimiento, el uso del algoritmo ACO para descubrir reglas de la clasificación, en el contexto de la minería de datos, sigue siendo un área de investigación poco explorada. Realmente, el único algoritmo de la hormiga desarrollado en esa época para la minería de datos es un algoritmo para closterización [Monmarché N. 1999], que es una tárea de la minería de datos muy diferente de la tárea de la clasificación tratada por el Ant-Miner.

ANT-MINER: Es un algoritmo de optimización propuesto de colonia de hormigas para el descubrimiento de las reglas de la clasificación.

A. Una descripción general del Ant-Miner

En un algoritmo de ACO cada hormiga incrementalmente construye/modifica una solución para el problema.

Una descripción de alto nivel del Ant-Miner se muestra en algoritmo el cual sigue un acercamiento al cubrimiento secuencial para descubrir una lista de reglas de la clasificación que cubren todos, o casi todos, los casos de entrenamiento. Al principio, la lista de reglas descubiertas está vacía y el sistema del entrenamiento consiste en todas las cajas de

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

entrenamiento. Cada iteración del algoritmo, corresponde a un número de ejecuciones, donde se descubre una regla de la clasificación. Esta regla se agrega a la lista de reglas descubiertas, y los casos de entrenamiento que son cubiertos correctamente por esta regla (es decir, casos que satisfacen la regla antecedente y que hacen la clase predecir por el consiguiente de la regla) se quitan del sistema del entrenamiento. Este proceso se realiza iterativamente mientras que el número de cajas de entrenamiento destapadas es mayor que un umbral definido por el usuario, llamado `Max_uncovered_cases`.

Cada iteración consiste en tres pasos, abarcando la construcción de la regla, la poda de la regla, y la actualización de la feromona, detallada como sigue. Primero, Antt comienza con una regla vacía, es decir, una regla sin término en su antecedente, y agrega un término a la vez a su regla parcial actual. La regla parcial actual construida por una hormiga corresponde a la trayectoria parcial actual seguida por esa hormiga. Semejantemente, la opción de un término que se agregará a la regla parcial actual corresponde a la opción de la dirección en la cual la trayectoria actual será extendida. La opción del término que se agregará a la regla parcial actual y de la cantidad de η depende de una función heurística problema-dependiente asociada a cada término, como será discutido detalladamente en las τ feromona subdivisiones siguientes. Antt guarda el agregar uno-término-en-untime a su regla parcial actual hasta que uno de los dos criterios de parada siguientes se cumpla:

Cualquier término que se agrega a la regla puede hacer el cubrimiento de la regla, un número de casos más pequeños que un umbral definido por el usuario, llamado `Min_cases_per_rule` (número mínimo de casos cubiertos por regla).

Todos los atributos han sido utilizados ya por la hormiga, de modo que no haya atributos que se puedan agregar al antecedente de la regla.

En uno de los artículos es comparado el funcionamiento del Ant-Miner con el funcionamiento del algoritmo conocido como C4.5 en seis conjuntos. Los resultados

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

proporcionan evidencia de que los sistemas de la regla descubiertos por Ant-Miner son más simples que los sistemas de la regla descubiertos por C4.5.

El sistema también agrega una regla por default a la última posición pasada de la lista de reglas. La regla de default tiene un antecedente vacío y tiene un consiguiente el predecir la mayoría de las clases el conjunto de casos de entrenamiento los casos de que no son cubiertos por ninguna regla. Esta regla por default se aplica automáticamente si ninguna de las reglas anteriores en la lista cubre nuevo caso de clasificación. Una vez que la lista de reglas es completa en el sistema está finalmente listo para clasificar un nuevo caso de prueba, no visto durante el entrenamiento. Para hacer esto los intentos del sistema para aplicar las reglas descubiertas, en orden. La primera regla que cubre el nuevo caso es aplicada - es decir el caso se asigna la clase prevista por el consiguiente de esa regla.

Algunos de los elementos claves del algoritmo son: Construcción de las reglas, función heurística, poda de la regla, actualización de la feromona y parámetros de sistema.

7.2.6 Algoritmo CLICK

Existe otro algoritmo de closterización con colonias de hormigas el cual se centra en datos categóricos. Las técnicas categóricas que existen tienen defectos significantes en términos de funcionamiento, los closter que detectan, y su capacidad de localizar closters en sub-espacios. Este trabajo introduce un algoritmo nuevo llamado CLICK (Clustering Categorical Data using K-partite Maximal Cliques), que los closterizan en los grupos de datos categóricos basados en un método de la búsqueda para las pandillas máximas del k-partite. El algoritmo CLICK puede detectar closters del sub-espacio, y supera acercamientos anteriores por un factor de dos a tres. Este escala mejor que uno de los métodos existentes para los grandes grupos de datos dimensionales. Estos resultados se demuestran en un estudio comprensivo del funcionamiento en el sintético y conjuntos de datos verdaderos (Peters and Zaki 2004).

CAPÍTULO VII- OPTIMIZACIÓN CON COLONIA DE HORMIGAS APLICADO A MINERÍA DE DATOS

7.3 Conclusiones

Como se puede ver, actualmente existen diversos algoritmos de colonias de hormigas que realizan tareas de la minería de datos, en especial para la clasificación y closterización, como es el caso del ANT-LGP, ANT-BASED Clustering, Ant-Miner, AntClass, entre otros, por otra parte podemos ver que algunos de los trabajos analizados son algoritmos híbridos con otras técnicas como son: Neural Gas. Sin embargo muchos de estos están limitados debido a que trabajan sobre una matriz de dos dimensiones, esto limita el vecindario de un nodo a sus ocho vecinos.

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

8.1 Introducción

Dentro de este capítulo se describe el trabajo realizado hasta el momento, por lo que se muestran los distintos algoritmos de colonias de hormigas que han sido implementados hasta el momento para analizar el comportamiento de dichos algoritmos de optimización, y de esta manera ver el comportamiento de este tipo de algoritmos en los diferentes benchmarks que están reportados en la literatura como parte del estado del arte del problema del clique máximo, para esto se hace una descripción de cada uno de los procesos utilizados en cada algoritmo.

8.2 Algoritmos Implementados

Los algoritmos con los que se han estado trabajando son los siguientes:

Se implemento inicialmente un algoritmo de colonia de hormigas para resolver el problema del agente viajero TSP, el cual fue elaborado en base al algoritmo Ant System propuesto por Dorigo en 1991, con la finalidad de tener el primer acercamiento en la implementación de los algoritmos de colonias de hormigas, posteriormente se implementaron varios modelos de colonias de hormigas para resolver el problema del clique máximo, estos algoritmos fueron probados con instancias disponibles en internet, a continuación se muestra un esquema del modelo base de este tipo de algoritmos, ver figura. 8.1.

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

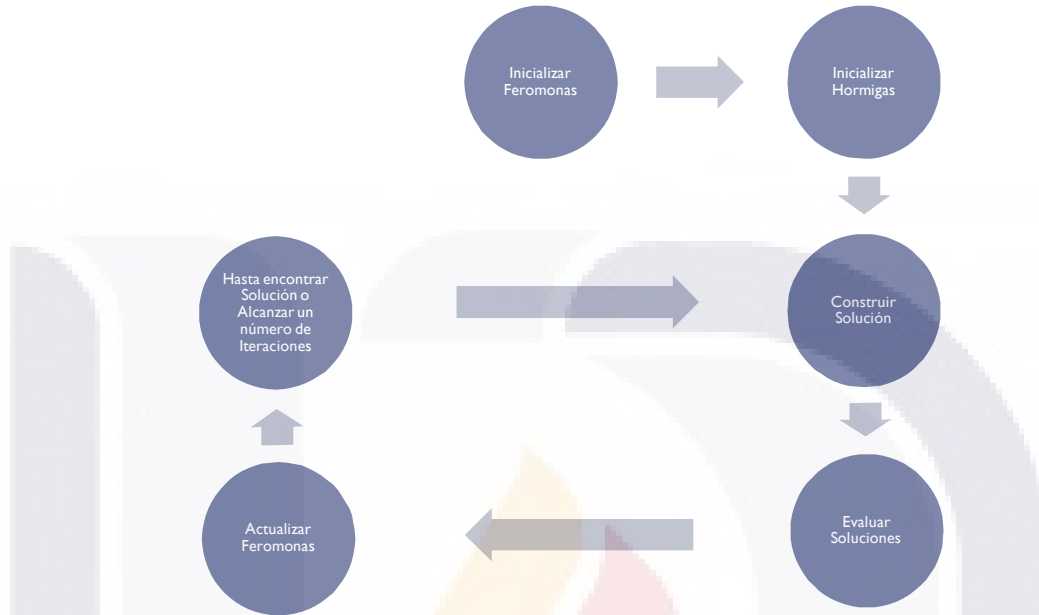


Figura. 8.1. Modelo general de un algoritmo de colonia de hormigas.

El primer algoritmo de colonias de hormigas para resolver el problema del clique máximo fue propuesto por Fenet y Solnon, aunque ellos mismos han implementado cuatro algoritmos más que son Vertex-AC, Vertex-AC+LS, Edge-AC y Edge-AC+LS. Las siglas AC de estos algoritmos significan Ant Clique, los algoritmos Vertex-AC, Vertex-AC+LS tiene la palabra Vertex debido a que la feromona es depositada en sus vértices y el segundo cuenta con las siglas LS porque al algoritmo Vertex-AC se le agregó una búsqueda local (Local Search), los algoritmos Edge-AC y Edge-AC+LS tiene la palabra Edge debido a que la feromona es depositada en los arcos.

Debido a los resultados reportados de estos cuatro algoritmos nosotros elegimos trabajar con el algoritmo de Edge-AC+LS, ya que es el que obtiene mejores resultado.

A continuación se muestra una implementación del Algoritmo Clique-Hormigas base (ACH) para resolver el problema del clique máximo. Este algoritmo está basado en el

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

algoritmo Edge-AC+LS propuesto recientemente por Fenet y Solnon. El algoritmo ACH implementa una nueva forma de colocar las hormigas en los vértices así como un optimizador local que permite crecer el tamaño del clique obtenido. Los resultados experimentales en este primer acercamiento nos dan buenos resultados los cuales son mostrados en el capítulo siguiente, ya que se realizaron modificaciones y los resultados de este algoritmo se presentaron en el CИСCI 2006 (Ponce et al., 2006).

Inicialización de la feromona: Las hormigas se comunican a través de la feromona depositada en los arcos del grafo. La concentración de feromona en el arco (v_i, v_j) es denotada por $\tau(v_i, v_j)$, una propuesta de Fenet y Solnon sobre cómo inicializar el rastro de feromona es de utilizar un τ_{min} o τ_{max} el cual está basado en el algoritmo MAX-MIN de Stutzle T. y Hoos H.H. con el propósito de tener una mayor exploración del espacio de búsqueda.

Construcción de los cliques con las hormigas: Se selecciona aleatoriamente un vértice inicial, y iterativamente se escoge vértices para agregar al clique. Dentro de un conjunto de candidatos que contiene todos los vértices que están conectados con todos los vértices del clique.

8.2.1 Algoritmo del Ant Clique Hormiga (ACH)

Este algoritmo es el primero que se implemento e intenta se una replica del algoritmo Edge-AC+LS

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

Algoritmo del Ant-Clique

Inicializar los rastros de feromonas

Repetir

Para k en $1..nbAnts$ **Hacer:**

 construya el clique C_k

 Actualizar los rastros de feromonas w.r.t. $\{C_1, \dots, C_{nbAnts}\}$

Hasta Alcanzar el Número de Ciclos o Encontrar la solución óptima

Tabla 8.1 Algoritmo Ant Clique base

Este algoritmo propuesto generó como resultado que la aproximación promedio al óptimo de 29 “benchmark” de la DIMACS para el problema del clique máximo es de 93.65%.

Se realizó un diseño de experimento obteniendo mejores resultados, los cuales se publicaron en el segundo congreso de investigación en el posgrado (Ponce et al., 2006).

8.2.2 Algoritmo de Colonia de Hormigas con Selección por Criterio de Metròpoli aplicado al Problema del Clique Maximo (MAC).

Obtener el grado de los vertices candidatos

Ordenarlos por el grado de mayor a menor

Si el numero de vertices del grado mayor es 1 **Entonces**

 Seleccionarlo como siguiente nodo

Sino

 Escoger uno de los vertices con mayor grado con una probabilidad $p(v_i)$

Tabla 8.2 Construccion de un Clique tomando en cuenta el grado del vertice

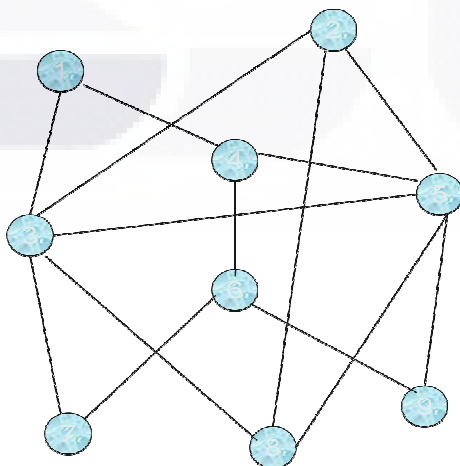
CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

La obtención del grado de los vértices candidatos se debe de hacer en base a la lista de candidatos y no en relación al grafo original ya que esto nos aumenta la probabilidad de encontrar mejores soluciones debido a que se selecciona como siguiente vértice uno que tiene todavía muchas relaciones con los vértices candidatos restantes.

Con este algoritmo propuesto fue ejecutado en 20 “benchmark”s de la DIMACS para el problema del clique máximo de los cuales en el 50% de los “benchmark”s en los que se ejecutó el algoritmo obtiene el óptimo, y en la solución más lejana queda a un 19% de la solución óptima. La ventaja de este algoritmo con el reportado en el CISCI’2006 (Ponce et al., 2006) es la velocidad.

8.2.3 Algoritmo de Colonia de hormigas para el problema del clique máximo

A continuación se muestra el proceso del algoritmo en base a la siguiente instancia.



8.2 Instancia del Grafo de prueba

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

La simbología utilizada en el algoritmo es: C es la solución la cual contiene el clique donde fue colocada inicialmente la hormiga k, Candidatos es la lista de los nodos que pueden ser seleccionados, V_i el vértice i.

El depósito de feromonas puede hacerse de dos maneras, ya que puede hacerse en el vértice o en la arista por lo cual la representación varia siendo un vector en el primer caso y una matriz en la segunda.

| | |
|---|------|
| | 1 |
| 1 | 0.01 |
| 2 | 0.01 |
| 3 | 0.01 |
| 4 | 0.01 |
| 5 | 0.01 |
| 6 | 0.01 |
| 7 | 0.01 |
| 8 | 0.01 |
| 9 | 0.01 |

Tabla 8.3 Vector de Feromonas

En esta representación se pone cero en donde no hay relación y en las otra una concentración muy pequeña.

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|------|------|------|------|------|------|------|------|------|
| 1 | 0 | 0 | 0.01 | 0.01 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0.01 | 0 | 0.01 | 0 | 0 | 0.01 | 0 |
| 3 | 0.01 | 0.01 | 0 | 0 | 0.01 | 0 | 0.01 | 0.01 | 0 |
| 4 | 0.01 | 0 | 0 | 0 | 0.01 | 0.01 | 0 | 0 | 0 |
| 5 | 0 | 0.01 | 0.01 | 0.01 | 0 | 0 | 0 | 0.01 | 0.01 |
| 6 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0.01 | 0 | 0.01 |
| 7 | 0 | 0 | 0.01 | 0 | 0 | 0.01 | 0 | 0 | 0 |
| 8 | 0 | 0.01 | 0.01 | 0 | 0.01 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0.01 | 0.01 | 0 | 0 | 0 |

Tabla 8.4 Matriz de Feromonas

Se colocan las hormigas aleatoriamente en una posición inicial, las cuales son colocadas en una matrix que representa la lista tabú.

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | | | | | | | | |
| 2 | 7 | | | | | | | | |
| 3 | 8 | | | | | | | | |
| 4 | 2 | | | | | | | | |

Tabla 8.5 Inicialización de la lista Tabú

La lista tabú es de dimensiones de $n \times m$ donde n es el número de hormigas con que está trabajando el algoritmo y m es el número de nodos con que cuenta el grafo.

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

Para cada hormiga se construye una solución tomando en cuenta el algoritmo mostrado a continuación.

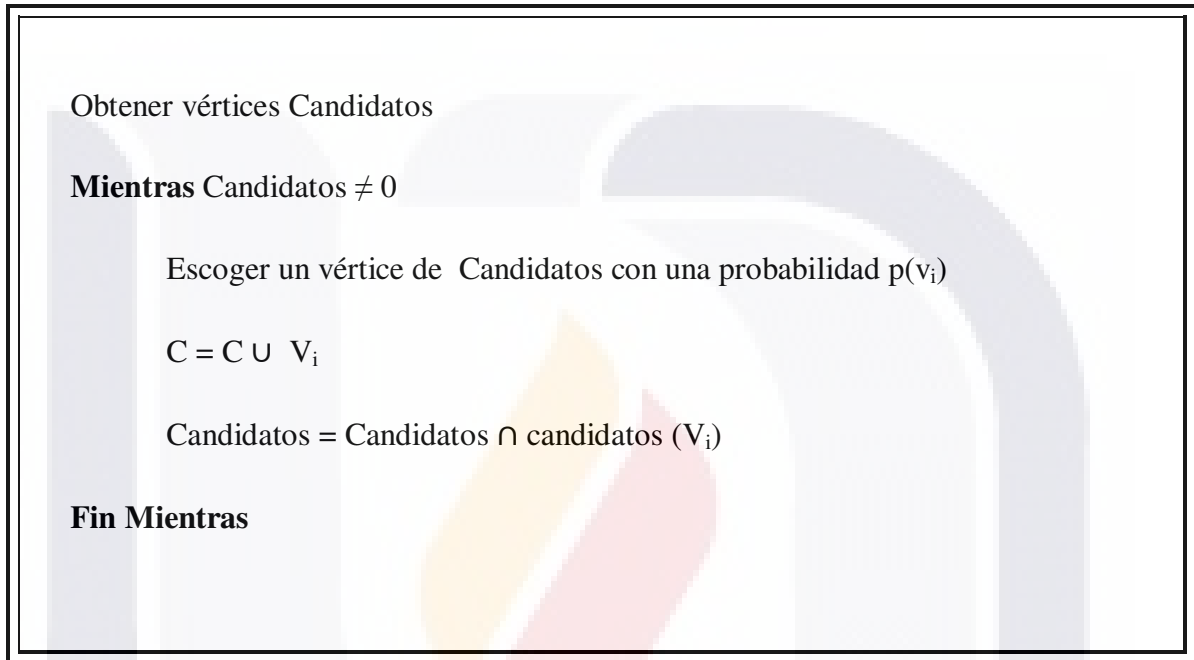


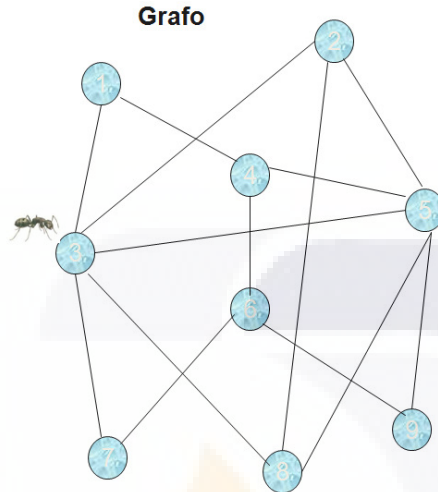
Tabla 8.6 Algoritmo para construir un Clique

$C = \emptyset$ // C representa la solución la cual corresponde a cada fila de la lista tabú

Candidatos // Son los nodos que pueden ser seleccionados

Se toma la hormiga 1 de la lista tabú para conocer la posición inicial y a partir de esta se obtiene la lista de los vértices candidatos. En el ejemplo la primera hormiga en la lista tabú está en el nodo 3.

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN



8.3 Colocación de la hormiga en la posición inicial.

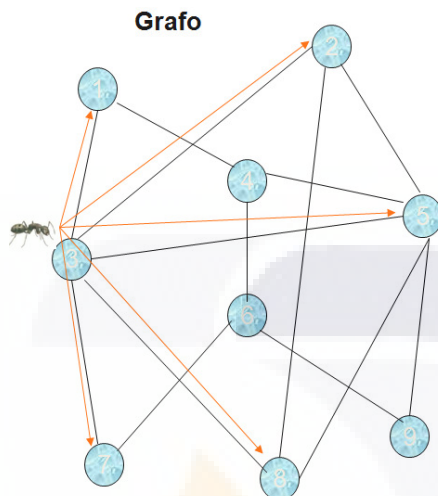
$C = \{3\}$

Se obtienen los candidatos en caso de no haber se pasaría a la siguiente hormiga, esto se debe hacer hasta terminar con todas las hormigas, una vez que se tienen todas las soluciones construidas por las hormigas, se evalúan y se seleccionan la mejor que es la que contiene el mayor número de nodos.

Se selecciona el nodo de mayor grado como siguiente en caso de ser único,

Sino se selecciona en base a la fórmula de probabilidad.

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN











8.4 Nodos candidatos de 3 en el grafo de prueba

Candidatos {1,2,5,7,8}

Mientras Candidatos $\neq 0$

- Escoger un vértice de Candidatos en base al criterio mostado en Tabla. 8.2

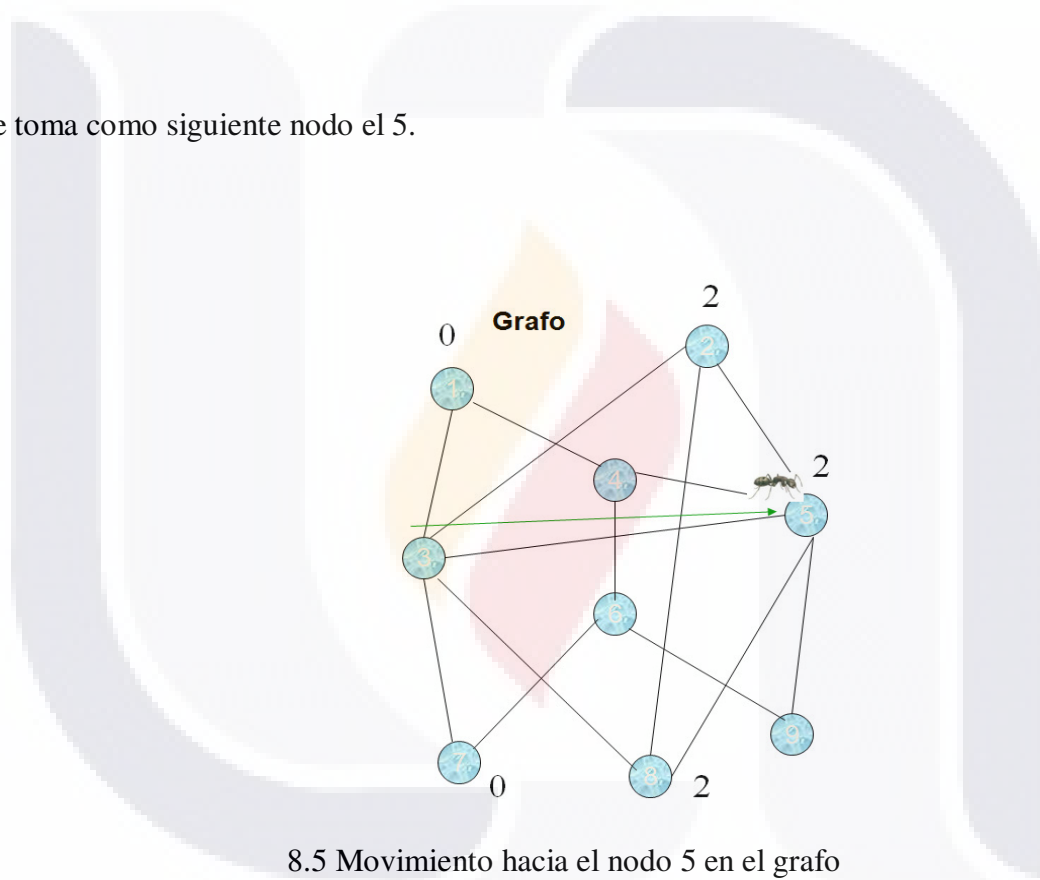
Procedimiento para seleccionar el vértice

-  Obtener el grado de los vértices candidatos
-  {0, 2, 2, 0, 2 }
-  Ordenarlos por el grado de mayor a menor
-  {2, 5, 8, 1, 7}
-  **Si** el número de vértices del grado mayor es 1
- Entonces**
-  Seleccionarlo como siguiente nodo
-  **Sino**
-  Escoger uno de los vértices con mayor grado con una probabilidad $p(v_i)$,

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

Como hay tres nodos con grado 2, se selecciona el siguiente nodo a través de la fórmula de probabilidad, por lo cual se calcula 3 probabilidades que son $P(2)$, $P(5)$ y $P(8)$. Para este caso como es la primera iteración todos tienen la misma concentración de feromonas, por lo cual la probabilidad de seleccionarlos es la misma $1/3$.

Se toma como siguiente nodo el 5.



- $V_i = 5$
- $C = C \cup \{5\}$ // $C = \{3, 5\}$

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 5 | | | | | | | |
| 2 | 7 | | | | | | | | |
| 3 | 8 | | | | | | | | |
| 4 | 2 | | | | | | | | |






Tabla 8.7 Introducción del elemento 5 a la lista Tabú

- Candidatos = Candidatos {1,2,5,7,8} \cap cadidatos (v_i) {2, 3, 4, 8}
- Candidatos = {2, 8}




Candidatos $\neq 0$

- Escoger un vértice de Candidatos en base al criterio mostado en Tabla. 8.2

Procedimiento para seleccionar el vértice

-  Obtener el grado de los vértices candidatos
-  {1, 1}
-  Ordenarlos por el grado de mayor a menor
-  {2, 8}
-  **Si** el número de vértices del grado mayor es 1

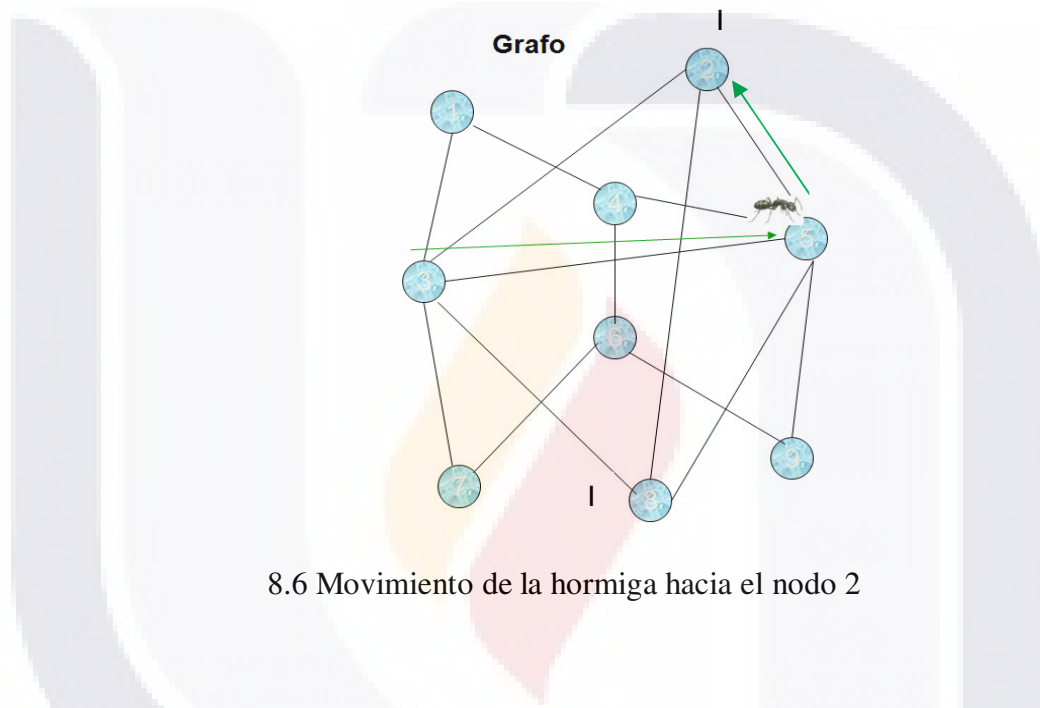
Entonces

-  Seleccionarlo como siguiente nodo
-  **Sino**
-  Escoger uno de los vértices con mayor grado con una probabilidad $p(v_i)$,

Como hay dos nodos con grado 2, se selecciona el siguiente nodo a través de la fórmula de probabilidad, por lo cual se calcula 3 probabilidades que son $P(2)$ y $P(8)$. Para este caso

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

como es la primera iteración todos tienen la misma concentración de feromonas, por lo cual la probabilidad de seleccionarlos es la misma 1/2.



8.6 Movimiento de la hormiga hacia el nodo 2

- $V_i = 2$
- $C = C \cup \{2\} // C = \{3, 5, 2\}$

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 5 | 2 | | | | | | |
| 2 | 7 | | | | | | | | |
| 3 | 8 | | | | | | | | |
| 4 | 2 | | | | | | | | |

Tabla 8.8 Introducción del elemento 2 a la lista Tabú




CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

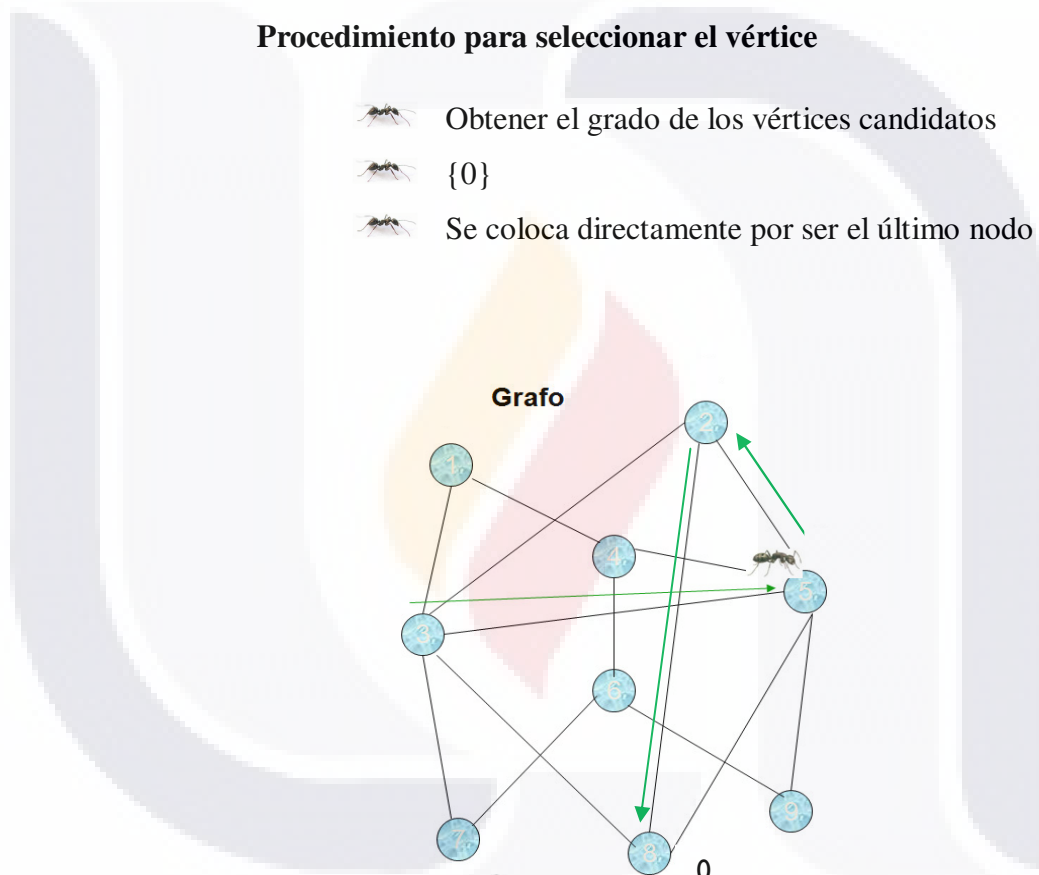
- Candidatos = Candidatos {2, 8} \cap candidatos (v_i) {3, 5, 8}
- Candidatos = {8}

Candidatos $\neq 0$

- Escoger un vértice de Candidatos en base al criterio mostado en Tabla. 8.2

Procedimiento para seleccionar el vértice

-  Obtener el grado de los vértices candidatos
-  {0}
-  Se coloca directamente por ser el último nodo



8.7 Movimiento de la hormiga al nodo 8

- $V_i = 8$
- $C = C \cup \{2\} // C = \{3, 5, 2, 8\}$

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 5 | 2 | 8 | | | | | |
| 2 | 7 | | | | | | | | |
| 3 | 8 | | | | | | | | |
| 4 | 2 | | | | | | | | |

Tabla 8.9 Introducción del elemento 8 a la lista Tabú

- Candidatos = Candidatos {8} \cap candidatos (vi) {3, 5}
- Candidatos = { }

Candidatos = 0

Regresa el clique C={3, 5, 2, 8}

Actualizar Feromonas

La actualización de la feromona se hace en base a la fórmula, esta se aplica para todos los nodos que fueron seleccionados en cada solución.

El incremento puede tomarse como una constante o puede estar en función de la solución (C/C_{best}) la utilidad de la solución que se está evaluando entre la utilidad de la mejor solución encontrada hasta el momento.

Y esto se modifica en el vector de las feromonas

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

8.2.4 Algoritmo de Colonia de Hormigas con Expansión aplicado al Problema del Clique Máximo (EXP-AC)

En el SIMS 2006 (Ponce et al, 2006), se propuso un algoritmo para resolver el problema del clique máximo en el cual la diferencia se encuentra en la parte de como se escoge el siguiente vértice dentro del proceso de construcción del clique la cual se muestra a continuación.

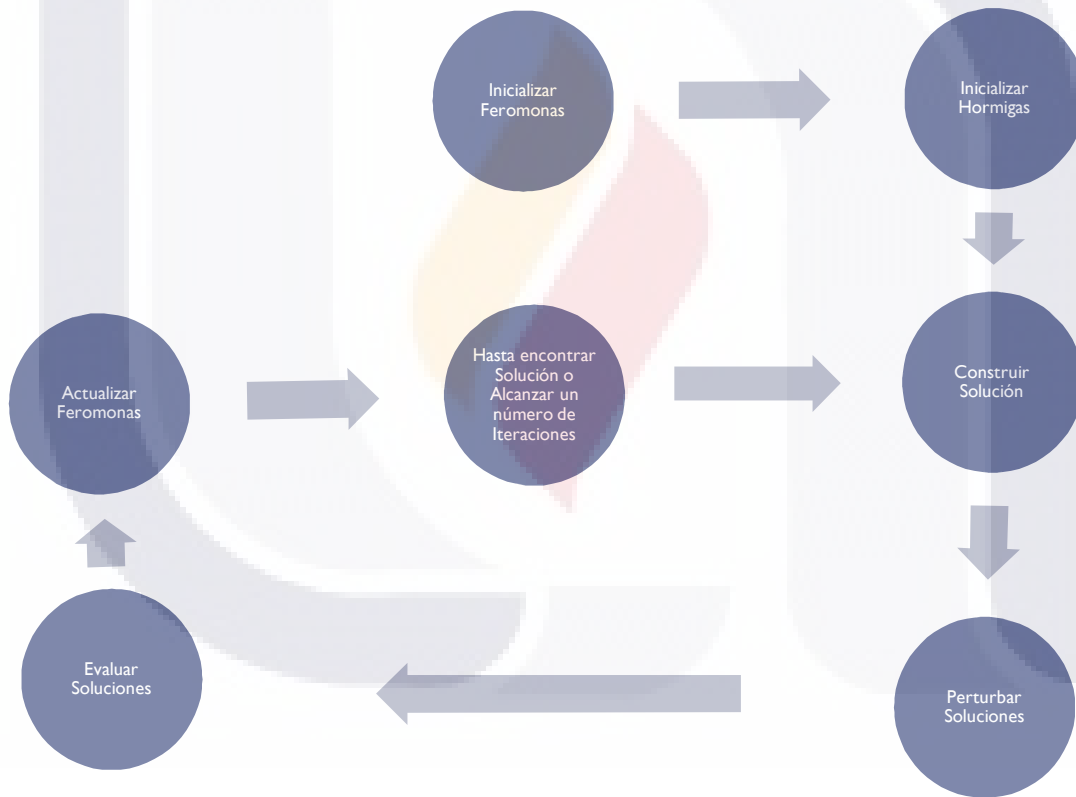


Figura 8.8. Modelo del un algoritmo de colonia de hormigas con expansión del clique (La expansión consiste en la perturbación de las soluciones encontradas)

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

| |
|---|
| <p>Inicializar los rastros de feromonas</p> <p>Inicializar las hormigas</p> <p>Repetir</p> <p style="padding-left: 40px;">Para k en $1..nbAnts$ Hacer:</p> <p style="padding-left: 80px;">construir solución (clique) C_k</p> <p style="padding-left: 40px;">Actualizar los rastros de feromonas w.r.t. $\{C_1, \dots, C_{nbAnts}\}$ tomando en cuenta τ_{max}</p> <p style="padding-left: 40px;"><i>Expansión del clique</i></p> <p>Hasta Alcanzar el Número de Ciclos o Encontrar la solución óptima</p> |
|---|

Tabla 8.10 Algoritmo Ant Clique con Expansión

Inicialización de la feromona: Se hace a través de depositar la feromona en los arcos del grafo. La concentración de feromona en el arco (v_i, v_j) es denotada por $\tau(v_i, v_j)$.

Inicialización de las hormigas: Nosotros trabajamos con dos maneras de inicializar a las hormigas en los vértices del grafo en el cual se va a trabajar para la búsqueda de los cliques una es la forma que normalmente es utilizada, es decir de una manera aleatoria y la segunda

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

es una propuesta que se realizó, la cual consiste en colocar un porcentaje de las hormigas en vértices que tienen mayor grado y el resto de las hormigas de manera aleatoria.

Construcción de los cliques con las hormigas: Se toma el vértice inicial de la hormiga que va a construir la solución, el cual fue seleccionado en base al paso anterior por cualquiera de las dos formas, iterativamente se obtienen el conjunto de vértices con que se relaciona y posteriormente se obtiene los vértices candidatos del nodo donde se encuentra la hormiga actualmente, para esto son eliminados los nodos que ya están contenidos en la solución parcial (clique) y se escoge el siguiente vértice (nodo) para agregar al clique en base a la fórmula de probabilidad utilizada. Dentro del conjunto de candidatos que contiene todos los vértices que están conectados con todos los vértices del clique.

Expansión del clique: consiste en perturbar las soluciones (cliques maximales) obtenidas hasta el momento en la iteración, es decir se retira uno de los vértices contenidos en la solución y se busca dentro de los vértices que no forman parte de la solución existe algún vértice que se relacione con todos los que se están contenidos en el conjunto solución, si existe algún elemento este es agregado al clique. Se repite ésta tarea con todos los vértices del clique obtenido.




En este trabajo se reporto que la aproximación promedio al óptimo del algoritmo es 97.58% el cual puede ser considerado como bueno, sin embargo la aproximación del algoritmo reportado por Solnon y Fenet es de 99.36%, este resultado fue obtenido al utilizar el algoritmo en 36 de los “benchmark”s de la DIMACS.

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

8.3 Complejidad de los algoritmos implementados

En este apartado de la sección, se muestra la manera como se calculó la complejidad del algoritmo parte por parte del algoritmo base, así como la complejidad total de los diversos algoritmos implementados.

Para el algoritmo base se tiene 3 funciones utilizadas para la lectura de los benchmark, estas funciones son:

| | |
|---|------------|
|  | get_edge; |
|  | get_params |
|  | readgraph |

Complejidad de la función get_edge

```
{
    int byte, bit;
    char mask;
    int k;
    if (i<j) { k = i; i = j; j = k;}
    bit = 7-(j & 0x00000007);
    byte = j >> 3;
    mask = masks[bit];
    return( (Bitmap[i][byte] & mask)==mask );}
```

En base al código anterior, no existe ningún ciclo por lo que se ejecutan solo la instrucciones que pueden se representadas con cualquier variable en este caso usaremos n. Por lo que la función de complejidad puede verse como **O(n)**.

Complejidad de la función get_params

```
while (!stop && (c = *pp++) != '\0'){
```

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

```
switch (c)
{
    case 'c':
        while ((c = *pp++) != '\n' && c != '\0');
        break;
    case 'p':
        sscanf(pp, "%s %d %d\n", tmp, &Nr_vert, &Nr_edges);
        stop = 1;
        break;
    default:
        break;
}
```

En el código anterior podemos ver que existen dos ciclos while, este tipo de estructura tienen una comportamiento logarítmico, con el paso del tiempo se acerca uno a la solución, sin embargo cuando conocemos cuantas veces va realizarse antes de llegar a la solución y que es finito este limite, puede ser tomado como un ciclo for por lo que el número de iteraciones para leer los parámetros es: $(n)(n)$ donde n y n son iguales. Por lo que la función de complejidad puede verse como $O(nn)$ u $O(n^2)$.








Complejidad de la función readgraph

```
for (i = 0; i < N; i++)
    for (j = 0; j < N; j++)
        if (get_edge(i, j))
            bitmap[(j+1)/CHARBITS][i+1] |= (1 << ((j+1) % CHARBITS));
```

En el código anterior podemos ver que existen dos ciclos for por lo que el número de iteraciones para leer el grafo de la instancia de prueba es: $(n)(n)$ donde n y n son iguales. Por lo que la función de complejidad puede verse como $O(nn)$ u $O(n^2)$.

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

Por otra parte las funciones utilizadas en el algoritmo base son:

| | |
|---|----------------------|
|  | inicializaf |
|  | inicializacandidatos |
|  | probabilidadi; |
|  | escogerv |
|  | actualizarf |
|  | inicializah |
|  | construyeclique |

A continuación se mostrara el código de las funciones bases del algoritmo para determinar la complejidad de los algoritmos.

Complejidad de la función inicializaf

```
for (x=1;x<=n;x++)  
{ for (y=1;y<=n;y++)  
  { if (grafo[x][y]==1)  
    grafo[x][y]=c; } }}
```

n = número de nodos que componen el grafo original

En base al código anterior, el número de iteraciones para inicializar las feromonas es: $(n)(n)$ donde **n** y **n** son iguales. Por lo que la función de complejidad puede verse como **O(nn)** u **O(n²)**.

Complejidad de la función inicializacandidatos

```
for (z=1;z<=n;z++)
```

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

```
{ candidatos[z]=0;
  candidatosj[z]=0; }
```

n = número de nodos que componen el grafo original

En base al código anterior, el número de iteraciones para inicializar las feromonas es: **(n)**. Por lo que la función de complejidad puede verse como **O(n)**.

Complejidad de la función probabilidadi

```
for (x=1;x<=y;x++)
  { probabilidad[x]=pow(grafo [clique[nh][vf]][candidatos[x]],alfa)/ totalf; }
for (x=1;x<=y;x++) //ciclo para sacar las probabilidades continuas de 0-1
  { p=p+probabilidad[x];
    probabilidad[x]=p; }
```

y = es el número de nodos candidatos que pueden ser seleccionados (permitidos)

En base al código anterior, el número de iteraciones para obtener la probabilidad de seleccionar un nodo es: **(y)(y)** donde **y** es un número entre 1-n por lo que se puede tomar como **n**. Por lo que la función de complejidad puede verse como **O(yy)** u **O(y²)** o **O(n²)**.

Complejidad de la funciones escogerv

```
probabilidadi();
vertice= random(99)+1;
while (aleatorio >probabilidad[x])
  x++;
```

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

```
if (aleatorio==1) {x--;}
```

```
vertice=candidatos[x];
```

x = varía de 1 a n

En base al código anterior, la función probabilidadi tiene complejidad de $O(n^2)$ y el número de iteraciones para el ciclo while es: **(X)** donde **X** es un número entre 1-n por lo que se puede tomar como **n**. Por lo que la función de complejidad puede verse como $O(n^2) + O(n) = O(n^2)$.

Complejidad de la función actualizarf

```
for (int y=1;y<t;y++)
for(int x=y+1;x<=t;x++)
{ grafo[clique[nh][y]][clique[nh][x]]=(grafo[clique[nh][y]][clique[nh][x]]*ro)+
tt/ttamclique;
  grafo[clique[nh][x]][clique[nh][y]]=(grafo[clique[nh][x]][clique[nh][y]]*ro)+
tt/ttamclique;
  if ((grafo[clique[nh][y]][clique[nh][x]])>6)
  { grafo[clique[nh][y]][clique[nh][x]]=6;
    grafo[clique[nh][x]][clique[nh][y]]=6;}
}
```

En base al código anterior, la función contiene dos ciclos anidados para realizaciones la actualización de la feromona. Por lo que la función de complejidad puede verse como $O(nn)$ u $O(n^2)$.

Complejidad de la función inicializah

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

```
for(x=1;x<=nhormigas;x++)
    clique[x][1]=random(numvertices)+1; //Se selecciona vf y se asigna a C
```

En base al código anterior, la función contiene un ciclo para colocar las hormigas en una posición inicial. Por lo que la función de complejidad puede verse como **O(n)**.

Complejidad de la función construyeclique

```
{
// sacar lista de candidatos
for (int x=1;x<=numvertices;x++)
    if (grafo[clique[nh][1]][x]!=0)
        { candidatos[y]=x; y++; }
//construir el clique
while (y!=0)
{ clique[nh][nvertices]=escogerv(nh,nvertices-1,y);
nvertices++;
y=1;
while (candidatos[y]!=0)
    y++;
y--; }
//ciclo para guardar mejor solución
for (y=1;y<nvertices;y++)
    cliquemax[y]=clique[nh][y];
// actualización de la feromona
actualizarf(nh, nvertices-1);
}
```

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

En base al código anterior, el ciclo para obtener candidatos tiene una complejidad $O(n)$ la construcción del clique tiene complejidad de $O(n^2)$, guardar el la mejor solución es de complejidad $O(n)$ y la actualización de la feromona tiene complejidad de $O(n^2)$. Por lo que la función de complejidad puede verse como $O(n) + O(n^2) + O(n) + O(n^2) = 2 O(n^2) + 2 O(n)$ esto se reduce finalmente a una complejidad $O(n^2)$.

Las funciones creadas para implementar las estrategias propuestas son:



grados



extenderclique

Complejidad de la función grados

```
for (x=1;x<=numvertices;x++)
{
    for (y=1;y<=numvertices;y++)
    {
        if (grafo[x][y]>0)
            grado[x]=grado[x]+1;}
    grado_ordenado[x][0]=x;}

for (x=1;x<numvertices;x++)
{
    for (y=x+1;y<=numvertices;y++)
    {
        if (grado_ordenado[x][1]<grado_ordenado[y][1])
        {temp=grado_ordenado[x][1];
        grado_ordenado[x][1]=grado_ordenado[y][1];
        grado_ordenado[y][1]=temp;
        temp=grado_ordenado[x][0];
        grado_ordenado[x][0]=grado_ordenado[y][0];
```

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

```
        grado_ordenado[y][0]=temp;}
    }
}
```

En base al código anterior, el ciclo para obtener el grado tiene una complejidad $O(n^2)$, y la ordenación de por grado tiene complejidad de $O(n^2)$. Por lo que la función de complejidad puede verse como $O(n^2) + O(n^2) = 2 O(n^2)$ esto se reduce finalmente a una complejidad $O(n^2)$.

Complejidad de la función extenderclique

```
for (z=c;z>0;z--)
{
    t=tamca;
    na=clique[ncliques][z];
    clique[ncliques][z]=0;
    for (x=z;x<=tamca;x++)
        clique[ncliques][x]=clique[ncliques][x+1];
    tamca--;
    for (x=1; x<N; x++)
    {
        b=0;
        for (y=1; y<=tamca; y++)
            if (grafo[x][clique[ncliques][y]]>0)
                b++;
        if ((b==(tamca))&&(x != na))
            { tamca++;          clique[ncliques][tamca]=x;}
    }
    if (tamca<t)
        { clique[ncliques][t]=na; tamca++;}
    if ((tamca)>(tamclique))
        { tamclique= tamca;
```

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

```

    for (y=1;y<=tamca;y++)
        cliquemax[y]=clique[ncliques][y];}
    if ((tamca)>(tamcliquet))
        {tamcliquet= tamca;
        for (y=1;y<tamca;y++)
            cliquemaxt[y]=clique[ncliques][y];}
    }

```

En el código anterior, el ciclo para obtener el grado tiene una complejidad $O(n) * (O(n) + O(n^2) + O(n)) = O(n) * (O(n^2) + 2O(n))$ esto se reduce finalmente a $O(n) * O(n^2)$ por lo que queda una complejidad $O(n^3)$.

8.3.1 Complejidad del Algoritmo ACH

Complejidad del Algoritmo

| | |
|---|----------|
| readgraph(); | $O(n^2)$ |
| inicializaf(); | $O(n^2)$ |
| grados(); | $O(n^2)$ |
| inicializah(); | $O(n)$ |
| for(int niteraciones=1;niteraciones<=nc;niteraciones++) | $O(n)$ |
| { for(ncliques=1;ncliques<=nhormigas;ncliques++) | $O(n)$ |
| { construyeclique();} | $O(n^2)$ |
| actualizaf();} | $O(n)$ |

La complejidad total del primer algoritmo queda como $O(n^2) + O(n^2) + O(n^2) + O(n) + O(n) * O(n) * O(n^2) + O(n)$ esto es igual a $3O(n^2) + 2O(n) + O(n^4)$ por lo que la complejidad del algoritmo es $O(n^4)$

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

8.3.2 Complejidad del Algoritmo

Complejidad del Algoritmo

| | |
|---|----------|
| readgraph(); | $O(n^2)$ |
| inicializaf(); | $O(n^2)$ |
| grados(); | $O(n^2)$ |
| inicializah(); | $O(n^2)$ |
| for(int niteraciones=1;niteraciones<=nc;niteraciones++) | $O(n)$ |
| { for(ncliques=1;ncliques<=nhormigas;ncliques++) | $O(n)$ |
| { construyeclique();} | $O(n^2)$ |
| actualizaf(); } | $O(n)$ |

La complejidad total del segundo algoritmo queda como $O(n^2) + O(n^2) + O(n^2) + O(n^2) + O(n) * O(n) * (O(n^2) + O(n))$ esto es igual a $4O(n^2) + O(n^2) * (O(n^2) + O(n)) = 4O(n^2) + O(n^4) + O(n^3) = O(n^4) + O(n^3) + 4O(n^2)$ por lo que la complejidad del algoritmo es $O(n^4)$

8.3.3 Complejidad del Algoritmo

Complejidad del Algoritmo

| | |
|---|----------|
| readgraph(); | $O(n^2)$ |
| inicializaf(); | $O(n^2)$ |
| grados(); | $O(n^2)$ |
| inicializah(); | $O(n^2)$ |
| for(int niteraciones=1;niteraciones<=nc;niteraciones++) | $O(n)$ |
| { for(ncliques=1;ncliques<=nhormigas;ncliques++) | $O(n)$ |
| { construyeclique();} | $O(n^2)$ |
| actualizaf(); | $O(n)$ |
| extenderclique();} | $O(n^3)$ |

CAPÍTULO VIII.- DESARROLLO Y VALIDACIÓN

La complejidad total del segundo algoritmo queda como $O(n^2) + O(n^2) + O(n^2) + O(n^2) + O(n) * O(n) * (O(n^2) + O(n) + O(n^3))$ esto es igual a $4O(n^2) + O(n^2) * (O(n^2) + O(n) + O(n^3)) = 4O(n^2) + O(n^4) + O(n^3) + O(n^5) = O(n^5) + O(n^4) + O(n^3) + 4O(n^2)$ por lo que la complejidad del algoritmo es $O(n^5)$

8.4 Conclusiones

Como se puede ver se ha trabajado con la implementación de varios algoritmos los cuales fueron descritos en este capítulo, para esto fueron explicados cada uno de los procesos tratando de que fueran lo más claro posible e incluso de describió paso a paso como se contruye un clique, se tienen publicaciones de tres, de los cuales dos son para resolver el problema del clique máximo, la implementación de estos algoritmos han permitido conocer más la implementación y el poder ver donde podemos implementar otras técnicas, como los optimizadores locales lo cuales se implementan en el momento en que se construye la solución ó la implementación de estrategias como es la de tomar la mejor solución y tratarla de mejorar, como fue el tratar de expandir el clique.

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

9.1 Introducción

En este capítulo se muestran los resultados que han sido obtenidos por los distintos algoritmos que fueron descritos en el capítulo anterior, así como el análisis e interpretación de estos, se mencionan también los benchmarks que fueron utilizados para cada uno de los problemas tratado y donde pueden ser encontrados.

9.2 Resultados Obtenidos

Los algoritmos ACO dependen de los parámetros α que es el factor de peso (importancia) de la feromona, y ρ es el porcentaje de evaporación de la feromona. Se puede incrementar la diversificación si decrecemos el valor de α , ya que las hormigas tienen menos sensibilidad al rastro de feromonas, y si se incrementa ρ , la evaporación de la feromona es más lenta. Cuando se incrementa la habilidad de exploración de las hormigas, estas pueden encontrar mejores soluciones pero esto implica más tiempo. Tomando en cuenta estos factores se realizó el diseño de experimentos para cada uno de los algoritmos reportados en el capítulo anterior para ver con cuales valores de los parámetros se obtenían mejores resultados, este tipo de experimento es importante realizarlo en cada algoritmo ya que través de este se define los valores óptimos de los parámetros, dichos valores son los que hacen que el algoritmo obtenga los mejores resultados, por lo tanto son los que se utilizan con otros algoritmos para poder hacer comparaciones con nuestro algoritmo.

Para realizar los experimentos se utilizaron los benchmarks que se encuentran disponibles en la DIMACS, ya que estos son los que fueron utilizados en la mayoría de los artículos que fueron analizados.

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

En la Tabla 9.1, se puede ver un ejemplo de uno de los experimentos realizados al algoritmo del clique máximo, en el cual los campos ó parámetros que se están tomando en cuenta son los siguientes:

Hormigas.- Corresponde al número de hormigas con que se va a trabajar que sería el equivalente a decir cuál es la población de la colonia.

τ_0 .- Es la concentración inicial de la feromona, α la importancia que le damos a la feromona.

ρ .- Es el factor de evaporación de la feromona.

τ_{max} .- Es el nivel de concentración máxima que puede tomar la feromona.

Ch.- La forma de colocar a las hormigas en su vértice inicial.

Nc.- El número de ciclos que se correrá el algoritmo, y se refiere al número de veces que cada uno de los agentes intentará crear una solución.

Cabe mencionar que en la primera iteración no hay rastro de feromona alguno ó es una concentración muy pequeña para todas, y que mientras aumenta el número de iteraciones, los agentes utilizarán los rastros generados en las iteraciones anteriores.

| Hormigas | T_0 | A | P | τ_{max} | Ch | Nc |
|----------|-------|-----|------|--------------|----|-----|
| 30 | 0.01 | 1 | 0.99 | 6 | 1 | 100 |
| 50 | | 0.5 | 0.98 | | 2 | |
| 100 | | | | | | |

Tabla 9.1 Diseño de Experimento

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

Para llevar a cabo el diseño de experimentos de los algoritmos desarrollados para resolver el problema del clique máximo se tomaron como instancia del problema un los “benchmark” de la DIMACS que fue la instancia C500.9, la cual fue utilizada por Fenet y Solnon, esto para realizar el diseño de experimento mostrado anteriormente en la tabla 9.1 los resultados son mostrados en la tabla 9.2.

| T_0 | A | P | Nh | Ch | τ_{max} | Nc | Tam. Clique |
|-------|-----|------|-----|----|--------------|-----|-------------|
| 0.01 | 1 | 0.98 | 30 | 1 | 6 | 100 | 50 |
| 0.01 | 1 | 0.98 | 30 | 2 | 6 | 100 | 50 |
| 0.01 | 1 | 0.99 | 30 | 1 | 6 | 100 | 51 |
| 0.01 | 1 | 0.99 | 30 | 2 | 6 | 100 | 52 |
| 0.01 | 0.5 | 0.98 | 30 | 1 | 6 | 100 | 50 |
| 0.01 | 0.5 | 0.98 | 30 | 2 | 6 | 100 | 50 |
| 0.01 | 0.5 | 0.99 | 30 | 1 | 6 | 100 | 51 |
| 0.01 | 0.5 | 0.99 | 30 | 2 | 6 | 100 | 51 |
| 0.01 | 1 | 0.98 | 50 | 1 | 6 | 100 | 52 |
| 0.01 | 1 | 0.98 | 50 | 2 | 6 | 100 | 51 |
| 0.01 | 1 | 0.99 | 50 | 1 | 6 | 100 | 50 |
| 0.01 | 1 | 0.99 | 50 | 2 | 6 | 100 | 53 |
| 0.01 | 0.5 | 0.98 | 50 | 1 | 6 | 100 | 50 |
| 0.01 | 0.5 | 0.98 | 50 | 2 | 6 | 100 | 51 |
| 0.01 | 0.5 | 0.99 | 50 | 1 | 6 | 100 | 51 |
| 0.01 | 0.5 | 0.99 | 50 | 2 | 6 | 100 | 51 |
| 0.01 | 1 | 0.98 | 100 | 1 | 6 | 100 | 51 |
| 0.01 | 1 | 0.98 | 100 | 2 | 6 | 100 | 51 |
| 0.01 | 1 | 0.99 | 100 | 1 | 6 | 100 | 52 |
| 0.01 | 1 | 0.99 | 100 | 2 | 6 | 100 | 52 |
| 0.01 | 0.5 | 0.98 | 100 | 1 | 6 | 100 | 51 |
| 0.01 | 0.5 | 0.98 | 100 | 2 | 6 | 100 | 51 |
| 0.01 | 0.5 | 0.99 | 100 | 1 | 6 | 100 | 51 |
| 0.01 | 0.5 | 0.99 | 100 | 2 | 6 | 100 | 51 |

Tabla 9.2. Resultados del Diseño de Experimento realizado con el benchmark C500.9

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

En base a los resultados obtenidos del diseño de experimento en la instancia C500.9, tomamos los parámetros siguientes $\tau_0=0.01$, $\alpha=1$, $\rho=0.99$, hormigas=100, $Ch=2$, $\tau_{max}=6$ y $Nc=100$ esto a través de un análisis empírico, dichos valores fueron utilizados para correr el algoritmo con los demás “benchmark”, debido a que fueron los valores con los que se obtuvieron los mejores resultados a simple vista, debido a que los mejores resultados obtenidos fueron:

0.01, 1, 0.99, 30, 2, 6, 100 con un clique de tamaño 52

0.01, 1, 0.99, 50, 2, 6, 100 con un clique de tamaño 53

0.01, 1, 0.99, 100, 2, 6, 100 con un clique de tamaño 52

El primer algoritmo fue ejecutado en 29 de los 36 “benchmark” de la DIMACS, y los resultados obtenidos se muestran en la Tabla 9.3.

| Grafo | Tamaño del Clique benchmark | Tamaño del Clique encontrado por ACH | Aproximación al óptimo |
|------------|-----------------------------|--------------------------------------|------------------------|
| Brock200_2 | 12 | 12 | 100% |
| Brock200_4 | 17 | 17 | 100% |
| Brock400_2 | 29 | 24 | 83% |
| Brock400_4 | 33 | 24 | 73% |
| Brock800_2 | 24 | 20 | 83% |
| Brock800_4 | 26 | 19 | 73% |
| C2000.5 | 16 | 15 | 94% |
| C4000.5 | 18 | 16 | 89% |

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

| | | | |
|----------------|----|----|------|
| C125.9 | 34 | 34 | 100% |
| C250.9 | 44 | 43 | 98% |
| C500.9 | 57 | 52 | 91% |
| DSJC500.5 | 13 | 13 | 100% |
| DSJC1000.5 | 15 | 14 | 93% |
| Gen200_p0.9_44 | 44 | 44 | 100% |
| Gen200_p0.9_55 | 55 | 55 | 100% |
| Gen400_p0.9_55 | 55 | 50 | 91% |
| Gen400_p0.9_65 | 65 | 55 | 85% |
| Gen400_p0.9_75 | 75 | 60 | 80% |
| Hamming8_4 | 16 | 16 | 100% |
| Hamming10_4 | 40 | 38 | 95% |
| Keller4 | 11 | 11 | 100% |
| Keller5 | 27 | 26 | 96% |
| P_hat300_1 | 8 | 8 | 100% |
| P_hat300_2 | 25 | 25 | 100% |
| P_hat300_3 | 36 | 36 | 100% |
| P_hat700_1 | 11 | 11 | 100% |
| P_hat700_2 | 44 | 44 | 100% |
| P_hat700_3 | 62 | 62 | 100% |
| P_hat1500_1 | 12 | 11 | 92% |

Tabla 9.3 Resultados Obtenidos en los 29 “benchmark”

Como se puede ver en los resultados mostrados algoritmo propuesto el cual introduce una nueva forma de colocar las hormigas en los vértices que es tomando en cuenta el grado de estos lo cual puede incrementar la probabilidad de encontrar cliques más grandes, el ACH

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

genera como resultado que la aproximación promedio al óptimo de 29 “benchmark” de la DIMACS para el problema del clique máximo es de 93.65%.

Mientras que el segundo algoritmo fue en 20 de los 36 “benchmarks” de la DIMACS, y los resultados obtenidos se muestran en la Tabla 9.3, donde en la columna “grafo” esta el nombre que tiene el grafo en la DIMACS, la columna “Tamaño del clique “benchmark”” es el tamaño que tiene el clique máximo (Solución Óptima) y Tamaño del Clique encontrado por ACH desarrollado.

| Grafo | Tamaño del Clique “benchmark” | Tamaño del Clique encontrado por ACH |
|------------|-------------------------------|--------------------------------------|
| C125.9 | 34 | 34* |
| C250.9 | 44 | 44* |
| C500.9 | 57 | 53 |
| C1000.9 | 68 | 63 |
| C2000.9 | 78 | 72 |
| DSJC500.5 | 13 | 13* |
| DSJC1000.5 | 15 | 13 |
| Brock200_2 | 12 | 12* |
| Brock200_4 | 17 | 17* |
| Brock400_2 | 29 | 24 |
| Brock400_4 | 33 | 28 |
| Brock800_2 | 24 | 21 |
| Brock800_4 | 26 | 21 |
| C2000.5 | 16 | 16* |

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

| | | |
|-------------|----|-----|
| C4000.5 | 18 | 16 |
| Hamming8_4 | 16 | 16* |
| Keller4 | 11 | 11* |
| P_hat300_1 | 8 | 8* |
| P_hat700_1 | 11 | 10 |
| P_hat1500_1 | 12 | 12* |

Tabla 9.3 Resultados del algoritmo en los 20 benchmarks

En la tabla anterior los resultados que cuentan con un “*” son en los que se alcanzó la solución óptima de los “benchmarks”.

Se desarrollo un segundo diseño de experimento el cual se muestra en la tabla 9.5

| Hormigas | Número de Iteraciones | Colocacion de las Hormigas |
|----------|-----------------------|----------------------------|
| 30 | 100 | 100% |
| 50 | 500 | 75% |
| | | 50% |

Tabla 9.5 Diseño de experimento para análisis de los parámetros

Se realizó en base al diseño de experimento mostrado en la tabla anterior un análisis de la varianza tomando como variable de respuesta la diferencia entre el total de nodos del clique y el total de nodos encontrado por el algoritmo.

Los factores número de hormigas y número de ciclos fueron estadísticamente significativos, la colocación de las hormigas no fue estadísticamente significativo (Vea la Tabla 9.6).

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

| Source | Type III Sum of Squares | df | Mean Square | F | Sig. |
|-----------------|-------------------------|-----|-------------|----------|------|
| Corrected Model | 2.867 | 4 | .717 | 8.290 | .000 |
| Intercept | 212.817 | 1 | 212.817 | 2461.620 | .000 |
| NHORMIG | .817 | 1 | .817 | 9.446 | .002 |
| NCICLOS | 2.017 | 1 | 2.017 | 23.326 | .000 |
| COLOCACI | 3.333E-02 | 2 | 1.667E-02 | .193 | .825 |
| Error | 20.317 | 235 | 8.645E-02 | | |
| Total | 236.000 | 240 | | | |
| Corrected Total | 23.183 | 239 | | | |

a R Squared = .124 (Adjusted R Squared = .109)

Tabla 9.6.-Resultados del análisis de la varianza

En base a los resultados obtenidos del diseño de experimentos que se muestran en la tabla 13, se obtiene que la mejor combinación de parámetro es la 2,2,3 lo cual significa poner 50 hormigas, correr el algoritmo 500 ciclos y colocar las hormigas 50% en los vértices de mayor grado y 50% aleatoriamente. Una vez obtenidos estos resultados pasamos a correr los 36 benchmark de la DIMACS con esta combinación de parámetros 20 veces cada uno.

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

Nota: Las pantallas de los sistemas pueden ser vistas en el anexo 5.

| Grafo | Tamaño del Clique bechmark | Tamaño del Clique encontrado por ACH | ACH con opt. K-OPT | ABOMC | Aproximación al óptimo |
|-------------|----------------------------|--------------------------------------|--------------------|-------|------------------------|
| Brock200_1 | 21 | - | 21 | 21 | |
| Brock200_2 | 12 | 12 | 12 | 11 | 100% |
| Brock200_3 | | - | 15 | | |
| Brock200_4 | 17 | 17 | 17 | 17 | 100% |
| Brock400_1 | 27 | - | 24 | 24 | |
| Brock400_2 | 29 | 24 | 24 | 25 | 83% |
| Brock400_3 | | - | 24 | | |
| Brock400_4 | 33 | 24 | 24 | | 73% |
| Brock800_1 | | - | 20 | 20 | |
| Brock800_2 | 24 | 20 | 20 | 21 | 83% |
| Brock800_3 | | - | 18 | | |
| Brock800_4 | 26 | 19 | 19 | | 73% |
| c-fat200-1 | | - | 12 | | |
| c-fat200-2 | | - | 24 | | |
| c-fat200-5 | | - | 58 | | |
| c-fat500-1 | 14 | - | 14 | 14 | |
| c-fat500-5 | | - | 64 | 64 | |
| c-fat500-10 | | - | 126 | 126 | |
| C125.9 | 34 | 34 | 34 | | 100% |
| C250.9 | 44 | 43 | 44 | 44 | 98% |

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

| | | | | | |
|----------------|-----|----|-----|-----|------|
| C500.9 | 57 | 52 | 55 | 55 | 91% |
| C1000.9 | 68 | - | 64 | 63 | |
| C2000.5 | 16 | 15 | 15 | 16 | 94% |
| C2000.9 | 77 | - | | 71 | |
| C4000.5 | 18 | 16 | | | 89% |
| DSJC500.5 | 13 | 13 | 13 | 13 | 100% |
| DSJC1000.5 | 15 | 14 | | 15 | 93% |
| Gen200_p0.9_44 | 44 | 44 | 43 | 40 | 100% |
| Gen200_p0.9_55 | 55 | 55 | 55 | 55 | 100% |
| Gen400_p0.9_55 | 55 | 50 | 51 | 51 | 91% |
| Gen400_p0.9_65 | 65 | 55 | 65 | 65 | 85% |
| Gen400_p0.9_75 | 75 | 60 | 75 | 75 | 80% |
| Hamming6_2 | 32 | - | 32 | 32 | |
| Hamming6_4 | 4 | - | 4 | 4 | |
| Hamming8_2 | 128 | - | 128 | 128 | |
| Hamming8_4 | 16 | 16 | 16 | 16 | 100% |
| Hamming10_2 | 512 | - | | 512 | |
| Hamming10_4 | 40 | 38 | 36 | 40 | 95% |
| Johnson8-2-4 | 4 | - | 4 | 4 | |
| Johnson8-4-4 | 14 | - | 14 | 14 | |
| Johnson16-2-4 | 8 | - | 8 | 8 | |
| Johnson32-2-4 | 16 | - | 16 | 16 | |
| Keller4 | 11 | 11 | 11 | 11 | 100% |
| Keller5 | 27 | 26 | | | 96% |

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

| | | | | | |
|--------------|------|----|-----|-----|------|
| Keller6 | 59 | - | | 48 | |
| Mann_a9 | 16 | - | 16 | 16 | |
| Mann_a27 | 126 | - | 123 | 126 | |
| Mann_a45 | 345 | - | | | |
| Mann_a81 | 1100 | - | | | |
| P_hat300_1 | 8 | 8 | 8 | | 100% |
| P_hat300_2 | 25 | 25 | 25 | | 100% |
| P_hat300_3 | 36 | 36 | 36 | | 100% |
| P_hat500_1 | | - | 9 | | |
| P_hat500_2 | 36 | - | 36 | 36 | |
| P_hat500_3 | 50 | - | 50 | 50 | |
| P_hat700_1 | 11 | 11 | 11 | | 100% |
| P_hat700_2 | 44 | 44 | 44 | | 100% |
| P_hat700_3 | 62 | 62 | 62 | | 100% |
| P_hat1000_1 | | - | 10 | | |
| P_hat1000_2 | 46 | - | 46 | 46 | |
| P_hat1000_3 | 68 | - | 66 | 65 | |
| P_hat1500_1 | 12 | 11 | 11 | | 92% |
| P_hat1500_2 | 65 | - | | 65 | |
| P_hat1500_3 | 94 | - | | 93 | |
| San400_0.7_1 | 40 | - | 31 | 40 | |
| San400_0.7_2 | 30 | - | 18 | 30 | |
| San400_0.7_3 | 22 | - | 17 | 17 | |
| San400_0.9_1 | 100 | - | 100 | 100 | |

CAPÍTULO IX.- REPORTE Y DISCUSIÓN DE RESULTADOS

| | | | | | |
|-------------|----|---|----|----|--|
| San1000 | 15 | - | 10 | 15 | |
| Sanr400_0.5 | 13 | - | 13 | 13 | |
| San400_0.7 | 21 | - | 21 | 21 | |

Tabla 9.7 Comparación con el algoritmo ABOMC

9.3 Conclusiones

Es posible ver que los resultados obtenidos son buenos y es posible mejorar un algoritmo al incorporar estrategias de exploración y/o explotación adecuadas, las estrategias implementadas fueron un operador de expansión, una estrategia de selección basada en el grado de relación que tiene con los nodos candidatos y la forma de colocar las hormigas la cual también está relacionada con el grado de los nodos ó vértices, los resultados de estos algoritmos se han publicado en distintos eventos.

CAPÍTULO X.- CONCLUSIONES

10.1 Conclusiones

En este trabajo, hemos presentado diversos algoritmos de colonias de hormigas para resolver los problemas del clique máximo. Es importante aclarar que este es un problema difícil de resolver con técnicas tradicionales cuando se tratan instancias grandes de los problemas y las instancias utilizadas en los bechmars van desde 125 nodos del más pequeño y hasta 4000 nodos.

Se realizo un diseño de experimento con el cual a partir de un análisis empirico se determino que los mejores valores del algoritmo eran los parámetros siguientes $\tau_0=0.01$, $\alpha=1$, $\rho=0.99$, hormigas=100, Ch=2, $\tau_{\max}=6$ y Nc=100, este se realizo con la instancia c500.9 de la DIMACS.

Con estos resultados se corrieron 29 de los benchmarks teniendo un acercamiento al optimo de 93.6%, Posteriormente fue implementado el algoritmo MAC el cual reporto una mejora al obtener un acercamiento al optimo del 96.8% con lo que demostró que la propuesta del uso de información heurística como es el grado de los vértices al inicio por si solos y posteriormente el grado de relación que tiene los vértices candidatos.

El enfoque propuesto de cada algoritmo implementado fue descrito a detalle y presentamos varios resultados en los capítulos 8 y 9. Con el desarrollo de estos algoritmos se observó que son fáciles de implementar y que no tienen mucho costo computacional. Se realizó un análisis de la complejidad de cada uno de los algoritmos implementados, dando como resultado las siguientes complejidades fueron $O(n^4)$ para el primer algoritmo implementado, $O(n^4)$ para le algoritmo MAC que utiliza búsquedas locales y $O(n^5)$ que es el algoritmo que utiliza el operador de expansión. Los resultados fueron comparados con algunos de los resultados reportados en la literatura, donde se demostró que algunos de los

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO X.- CONCLUSIONES

algoritmos desarrollados obtuvieron resultados parecidos ó mejores a los resultados que reporta la literatura.

Como se pudo ver, se presentó un estudio sobre los algoritmos de colonias de hormigas, que son una metaheurística basada en el comportamiento de hormigas naturales, en especial de las especies que son consideradas como forrajeras. Se desarrollaron varios modelos para resolver el problema del clique máximo. La optimización con colonias de hormigas es un tema de investigación relativamente nuevo y de gran interés para muchos investigadores en el área de la Inteligencia Artificial, aunque ha tenido un gran crecimiento en los últimos años, todavía se espera que surjan una gran cantidad de resultados favorables al trabajar con estos algoritmos, ya que actualmente los ACO han demostrado ser útiles para resolver distintos problemas al encontrar la secuencia óptima en el proceso para llegar a la solución óptima, algunos de los problemas que se han manejado son: el problema del agente viajero, asignación de tareas, closterización, mochila, clique máximo, etc., actualmente existe aún una gran cantidad de líneas de investigación las cuales no ha sido exploradas, en las cuales se pueden hacer aportaciones y crear conocimiento nuevo como es el caso del desarrollo de modelos adaptativos y los sistemas híbridos, para tratar de obtener mejores resultados, así mismo hay un interés por el desarrollo de algoritmos de colonias de hormigas aplicado a problemas de minería de datos, por lo que se pudo ver que si es posible diseñar e implementar un modelo de colonias de hormigas que sea más eficiente a los ya existentes para resolver el problemas de closterización.

Como se pudo ver en el capítulo 3 se realizó una comparación de algunos de los algoritmos de colonias de hormigas aplicado al problema del clique máximo identificando las características de cada uno de estos, tratando de identificar sus ventajas y limitaciones.

En parte final de esta tesis, propusimos una estrategia para poder encontrar los mejores parámetros necesarios para la aplicación y lograr así, un porcentaje de convergencia más alto, usando un menor tiempo de procesamiento computacional.

TESIS TESIS TESIS TESIS TESIS

CAPÍTULO X.- CONCLUSIONES

10.2 Trabajo Futuro

Parte del trabajo futuro incluye lo siguiente:

1. Estudiar otras técnicas heurísticas para tomar ideas sobre mecanismos que nos ayuden a que el modelo sea más eficiente.
2. Incorporar nuevas estrategias como son el caso de la migración como se maneja en el trabajo de Pratt et al. del 2005 y/o sistemas con varias colonias como Sim 2003.
3. Demostrar por medios estadísticos cual de los dos algoritmos desarrollados de colonias de hormigas para el problema de la mochila es mejor.
4. Aplicar los algoritmos a problemas de optimización reales.
5. Buscar problemas prácticos en la sociedad, los cuales puedan ser resuelto a través de los algoritmos implementados este.
6. Trabajar en el estudio del comportamiento de los parámetros de los algoritmos para diferentes tipos de densidades de los grafos.
7. Hacer un diseño de experimento para determinar cuál es el porcentaje de hormigas que conviene colocar.

ANEXO

Anexo 1 Publicaciones

LIBROS PUBLICADOS

1. Julio Ponce and Adem Karahoca, **Data Mining and Discovery Knowledge in Real Life Application**, Aceptado Noviembre del 2008, Fecha de publicación: Enero de 2009, In-Tech, Austria <http://sciyo.com/search?q=julio+ponce>, <http://intechweb.org/search.php?q=julio+ponce>
2. Francisco Javier Álvarez Rodríguez, Jaime Muñoz Arteaga, Juan Pedro Cardona Salas, Mónica Brizuela Sandoval, Fatima Sayuri Quezada Aguilera, Julio Cesar Ponce Gallegos. **Interpretación del Modelo de Madurez de Capacidades (CMM) para pequeñas industrias de software**, Textos Universitarios, ISBN 978-970-728-020-5, 200 Ejemplares, Primera Edición, Septiembre 2008.

CAPÍTULOS EN LIBROS PUBLICADOS

3. Alberto Ochoa, Arturo Hernández, Laura Cruz, Julio Ponce, Fernando Montes, Liang Li & Lenka Janacek, **Artificial Societies and Social Simulation using Ant Colony, Particle Swam Optimization and Cultural Algorithms**, Enero 2010, In-Tech, Austria. <http://sciyo.com/search?q=julio+ponce>, <http://intechweb.org/search.php?q=julio+ponce>

ANEXO

4. Julio Ponce, Felipe Padilla, Alberto Ochoa, Alejandro Padilla, Eunice Ponce de León, Fátima Quezada, **Ant Colony Algorithm for Clustering through of Cliques**, Noviembre del 2009

5. Julio Ponce, Alberto Hernandez, Alberto Ochoa, Felipe Padilla, Alejandro Padilla, Francisco Alvarez y Eunice Ponce De León. **Data Mining in Web Applications**, Aceptado Noviembre del 2008, Fecha de publicación: Enero de 2009, Austria.
<http://sciyo.com/search?q=julio+ponce>, <http://intechweb.org/search.php?q=julio+ponce>

6. Alberto Ochoa, Sayuri Quezada, Julio Ponce, Francisco Ornelas, Dolores Torres, Christian Correa, Jaime de la Torre, Miguel Meza. **From Russia with Disdain: Simulating a Civil War**. Artificial Intelligence for Humans: Service Robots and Social Modeling, Grigori Sidorov (Ed.), ISBN: 978-607-00-0478-0, Primera Edición, 500 Ejemplares, Octubre del 2008, México D.F.

CAPÍTULOS EN LIBROS EN REVISION

7. Alberto Ochoa, Julio Ponce, Alberto Hernández, Eliza Coli, Anna Tordanelli4 & Felipe Padilla, **Multiagents System applied on a Cyberbullying Model for a Social Network**, Enviado Febrero del 2010, Canada.

ANEXO

PUBLICACIONES EN JOURNAL Y/O REVISTAS

8. Alberto Ochoa, Arturo Hernández, S. Jöns, Fernando Montes, Julio Ponce and Lenka Janacek, **Dyoram's Representation Using a Mosaic Image**, The International Journal of Virtual Reality, 2009, 8(3): 1-4 <http://www.ijvr.org/issues/issue3-2009/1.pdf>
9. Alberto Ochoa, Julio Ponce, Arturo Hernández & Liang Li . **Resolution of a Combinatorial Problem using Cultural Algorithms**. Journal of Computers, vol. 4, No. 8, August 2009, ISSN : 1796-203X. <http://www.academypublisher.com/jcp/vol04/no08/index.html>
10. Carlos Alberto Ochoa Ortiz Zezzatti, José Alberto Hernández Aguilar , Sául González Campos, Arnulfo Castro Vázquez and Julio Cesar Ponce Gallegos. **Evaluating the Authority in a Weblog Community**, Computación y Sistemas Vol. 11 No. 4, 2008, pp 370-380, ISSN 1405-5546, Article received on October 31, 2007, accepted on March 17 2008, México. <http://www.ejournal.unam.mx/cys/vol11-04/CYS011000405.pdf>
11. Julio C. Ponce, Carlos A. Ochoa, Miguel A. Meza, Alejandro Padilla y Felipe Padilla, **ACHPM: Algoritmo de Optimización con Colonia de Hormigas para el Problema de la Mochila**, Revista Iberoamericana de Sistemas, Cibernética e Informática Volumen 3 - Número 2, ISSN: 1690-8627, Florida, USA. <http://www.iiisci.org/Journal/RISCI/Contents.asp?var=&Previous=ISS0802>

ANEXO

12. Alberto Ochoa, Julio C. Ponce, Antonio Zamarron, Alberto Hernández, Alejandro Padilla, Francisco Álvarez, **A Game Board Implementing Data Mining and Cultural Algorithms**, Hífen, Uruguaiana, v. 31, n. 59, pag. 191, ISSN 0103-1155, Noviembre del 2007, Uruguaiana, Brasil.
<http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/issue/view/292>
13. Alberto Ochoa, Rosario Baltasar y Julio Ponce, **Translation of Summaries Obtained from Two Different Languages**, International Journal of Translation, Vol. 19, No. 1, Jan-Jun 2007, Pag. 163-173, ISSN 0940-9819, India
14. Ponce J., Ponce de León E., Padilla F. Padilla y A. Ochoa, **Algoritmo De Colonia De Hormigas Para El Problema Del Clique Máximo Con Un Optimizador Local K-Opt**, Hífen, Uruguaiana, v. 30, n. 58, pag. 191, ISSN 0103-1155, Noviembre del 2006, Uruguaiana, Brasil.
<http://revistaseletronicas.pucrs.br/ojs/index.php/hifen/issue/view/290>

PUBLICACIONES EN CONGRESOS INTERNACIONALES

15. Julio Ponce, Felipe Padilla, Alberto Ochoa, Alejandro Padilla, Eunice Ponce de León and Fatima Sayuri Quezada. **Ant Colony Algorithm for Clustering through of Cliques**. II Workshop of Hybrid Intelligent Systems en el MICAI 2009, 9-13 de Noviembre del 2009, Guanajuato, México

ANEXO

16. Alberto Ochoa, Julio Ponce, Alejandro Padilla, Fatima Sayuri Quezada y Miguel Meza. **Six degrees of separation in a graph to a Social Networking.** XXII Congreso Nacional y VIII Internacional de Informática y Computación ANIEI 2009, 21-23 de Octubre del 2009, Ensenada, Baja California
17. Miguel Meza, Francisco Álvarez, Alejandro Padilla, Jaime Muñoz y Julio Ponce. **Metodología de implementación de OA's en la materia de Lenguajes de Computación mediante B-learning.** XXII Congreso Nacional y VIII Internacional de Informática y Computación ANIEI 2009, 21-23 de Octubre del 2009, Ensenada, Baja California.
18. Julio Ponce, Felipe Padilla, Alberto Ochoa, Alejandro Padilla y Miguel Meza. **Algoritmo de Colonia de Hormigas para Clusterización.** XXII Congreso Nacional y VIII Internacional de Informática y Computación ANIEI 2009, 21-23 de Octubre del 2009, Ensenada, Baja California
19. Enrique Luna, Julio Ponce, Humberto Ambriz, Jorge Mondragón y Juan Antonio Nungaray. **Repositorio de Métricas construido a partir de la aplicación de herramientas de la Minería de Datos en Proyectos de Desarrollo de Software.** I Workshop de Ingeniería de Software dentro del XII Congreso Internacional Sudamericano de Ingeniería de Sistemas e Informática CISAICI, 6-10 de Octubre del 2009, Arica, Chile. <http://www.ijvr.org/issues/issue3-2009/1.pdf>

ANEXO

20. Ulises Ibarra Hernández, Francisco Javier Álvarez Rodríguez, Julio César Ponce Gallegos, Arturo Barajas Savedra and Juan Muñoz López. **Relaciones entre MoProSoft y Six Sigma: Un Camino Hacia el Desarrollo de un Modelo de Interpretación para Implantar la “Gestión de Procesos” en PyMEs.** Simposio de Ingeniería de Software SIS 2009, Realizado en el ENC’09, 21-22 de Septiembre de 2009, Mexico, DF
21. Alberto Ochoa, Shakley Bhiridzov, Julio Ponce, Fernando Montes, Cristiano Castelfranchi, Alberto Hernandez. **Six Degrees of Separation in a Graph to a Social Networking.** 6th Conference on Applications of Social Network Analysis (ASNA), 27-28 de Agosto del 2009, University of Zurich / ETH Zurich
22. Alberto Ochoa, Sayuri Quezada, Julio Ponce, Francisco Ornelas, Dolores Torres, Christian Correa, Jaime de la Torre, Miguel Meza. **From Russia with Disdain: Simulating a Civil War by Means of Predator/Prey Game and Cultural Algorithms.** Seventh Mexican International Conference on Artificial Intelligence. 27-31 de Octubre del 2008, Atizapan, México
23. Julio Ponce, Francisco Ornelas, Alberto Ochoa, Antonio Zamarrón, Jöns Sánchez, **Hybrid System for Determining a Ranking in Eurovision Song Contest.** 9th Mexican International Conference on Computer Science, ENC’08, II Workshop on Data Mining, 6-10 de Octubre del 2008, Mexicali, B.C., México

ANEXO

24. Del Abra Juárez I.J., Meza de Luna, M.A. y Ponce Gallegos, J.C. **Aplicación Inteligente que Contribuya a la Educación Superior.** XXI Congreso Nacional y VII Internacional de Informática y Computación ANIEI 2008, 1-3 de Octubre del 2008, Monterrey, Nuevo León, ISBN 978-970-15-1438-2.
25. Ramos Sánchez, M.A., Meza de Luna, M.A. y Ponce Gallegos, J.C. **Administrador de Páginas Web con Técnicas de Inteligencia Artificial.** XXI Congreso Nacional y VII Internacional de Informática y Computación ANIEI 2008, 1-3 de Octubre del 2008, Monterrey, Nuevo León, ISBN 978-970-15-1438-2.
26. Ochoa, Alberto Hernández, Arturo Sánchez, Jónhs Muñoz-Zavala, Angel Ponce, Julio. **Determining the Ranking of a New Participant in Eurovision Using Cultural Algorithms and Data Mining.** 18th International Conference on Electronics, Communications and Computers, 2008. CONIELECOMP 2008, Publication Date: 3-5 March 2008, On page(s): 47-52, Location: Puebla, Mexico, ISBN: 978-0-7695-3120-5, Digital Object Identifier: 10.1109/CONIELECOMP.2008.27, Posted online: 2008-03-12 10:18:24.0
<http://ieeexplore.ieee.org/Xplore/login.jsp?url=/iel5/4470492/4470493/04470509.pdf?arnumber=4470509>
27. Martínez F., C. Ochoa, Ponce J, Muñoz J. y García E.. **Assisted location of web advertisement zones using interactive genetic algorithms guided by event tracking.** Memorias del 3er Congreso Internacional de Computación Evolutiva COMCEV'07, Octubre 2007, ISBN 970-728-055-7.

ANEXO

28. Ochoa C., Ponce J., Baltasar R., Álvarez F, Quezada F., Torres M. y Pérez C. **Baharastar- Simulador de Algoritmos Culturales Para la Minería de Datos Sociales.** Memorias del 3er Congreso Internacional de Computación Evolutiva COMCEV'07, Octubre 2007, ISBN 970-728-055-7.
29. J.F. Meza de Luna, F.J. Luna Rosas, A. Padilla Díaz, F. Padilla Díaz y J. C. Ponce Gallegos. **Métodos Evolutivos para la Asignación de Horarios Docentes en el CBTis 168 de Aguascalientes México.** Memorias del 3er Congreso Internacional de Computación Evolutiva COMCEV'07, Octubre 2007, ISBN 970-728-055-7.
30. Miguel Meza, Francisco Álvarez, Alejandro Padilla, Jaime Muñoz, Felipe Padilla y Julio Ponce. **Sistemas Tutores Inteligentes en la Educación Superior, Caso Practico UAA.** Memorias del 3er Congreso Internacional de Computación Evolutiva COMCEV'07, Octubre 2007, ISBN 970-728-055-7
31. Julio Ponce, Alberto Ochoa, Wolfram Pietsch, and Zé Zolezzi-Hatsukimi. **Ankara: Identify Bipolar Síndrome in User of Orkut with Data Mining.** Memorias del ENC 2007. ISBN 9780769528991 P2899, IEEE ENC 2007 29 Books Box 1 of 3, Septiembre 2007, Morelia, Michoacán
32. Ponce J., Padilla F., Padilla A., Ochoa C. y Meza M. **ACHPM: Algoritmo De Optimización Con Colonia De Hormigas Para El Problema De La Mochila.** Memorias de la 6ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2007

ANEXO

33. Meza M., Álvarez F., Padilla A., Muñoz J. y Ponce J.. **Importancia De La Adaptabilidad Combinada Con Objetos De Aprendizaje.** Memorias de la 6ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2007
34. Padilla F., Padilla A. y Ponce J. **Sistemas Clasificadores de Aprendizaje con aprendizaje por refuerzo.** Memorias de la 6ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2007
35. Ponce J., Ponce de León E., Padilla F. Padilla y A. Ochoa. **Algoritmo De Colonia De Hormigas Para El Problema Del Clique Máximo Con Un Optimizador Local K-Opt.** 11º Simposio de Informática y 6º Mostra de Software Académico SIMS 2006, Hífen, Uruguaiana, v. 30, n. 58, pag. 191, ISSN 0103-1155, Noviembre del 2006, Uruguaiana, Brasil.
36. Ponce J., Ponce de León E., Padilla F. y Padilla A. **Implementación De Un Algoritmo De Colonia De Hormigas Para El Problema Del Clique Máximo.** Memorias de la 5ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2006, pag. 70-73, ISBN: 980-6560-88-4 Colección 980-6560-89 Volumen 1, Julio del 2006, Orlando Florida, USA.
37. Ponce J., Ponce de León E., Meza M.A. y Padilla A. **Implementación Del Método De Acciones De Grupo Para La Conjetura De Erdős Y Hável.** Memorias del 2º Congreso Mexicano de Computación Evolutiva COMCEV'05, pag. 145-149, ISBN: 970-728-024-7, Mayo del 2005, Aguascalientes, México

ANEXO

PUBLICACIONES A NIVEL ESTATAL Y NACIONAL

38. Julio Ponce, Alejandro Padilla, Felipe Padilla, Eunice Ponce de León, Alberto Ochoa. **Algoritmo de Colonias de Hormigas con Aprendizaje.** Memorias del Cuarto Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, Octubre del 2008, Aguascalientes, México
39. Julio Ponce, Felipe Padilla, Alejandro Padilla, Eunice Ponce de León. **Algoritmo de Colonia de Hormigas aplicado al problema de Bin Parking.** Memorias del Tercer Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, Octubre del 2007, Aguascalientes, México.
40. Miguel Angel Meza de Luna, Alejandro Padilla Díaz y Julio Cesar Ponce Gallegos. **Nivel Pedagógico con Asesoría Inteligente.** Memorias del 8° Seminario de Investigación, Universidad Autónoma de Aguascalientes y CONCYTEA, Noviembre del 2007, Aguascalientes, México, ISSN: 1870-4921.
41. Meza M., Padilla A. y Ponce J. **Sistema Hipermedia Adaptativos Para el Proceso de Enseñanza Aprendizaje de Lenguajes de Programación.** Memorias del 7° Seminario de Investigación, Universidad Autónoma de Aguascalientes y CONCYTEA, Noviembre del 2006, Aguascalientes, México, ISSN: 1870-4921.

ANEXO

42. Padilla F., Ponce J., Padilla A. y Ponce de León E. **Implementación de un Algoritmo De Colonia De Hormigas para el Problema Del Clique Máximo.** Memorias del 7° Seminario de Investigación, Universidad Autónoma de Aguascalientes y CONCYTEA, Noviembre del 2006, Aguascalientes, México, ISSN: 1870-4921.
43. Ponce J., Ponce de León E., Padilla F. y Padilla A. **Algoritmos De Colonia De Hormigas Aplicados Al Problema De La Mochila.** Memorias del Segundo Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, Noviembre del 2006, Aguascalientes, México.
44. García A., Padilla F., Padilla A., Ochoa A. y Ponce J. **Análisis Del Uso De Algoritmos Genéticos En La Optimización De Funciones Matemáticas.** Memorias del Segundo Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, Noviembre del 2006, Aguascalientes, México.
45. Ponce J., Ponce de León E., Padilla F. y Padilla A. **Algoritmo De Colonia De Hormigas Aplicado Al Problema Del Clique Máximo.** Memorias del Segundo Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, Noviembre del 2006, Aguascalientes, México.
46. Miguel Angel Meza de Luna, Alejandro Padilla Díaz y Julio Cesar Ponce Gallegos. **Sistema Para la Extracción de Estilos de Aprendizaje.** Memorias del XII Sposium de Investigación y Desarrollo Tecnológico. Pag. 68, ISSN 1405-7042.

ANEXO

47. Hernández Martínez E., Padilla Díaz A. Ochoa Ortiz Zezzatti y Ponce Gallegos J. **Ontologías con Optimización de Búsqueda Basada en Colonias De Hormigas Para la Solución de Problemas de Identificadores de Nombre.** Memorias del Octavo Verano de la ciencia Región centro, San Luís Potosí, México, Agosto del 2006.
48. García Domínguez A., Padilla Díaz A. Ochoa Ortiz Zezzatti y Ponce Gallegos J. **Análisis del uso de Algoritmos Genéticos en la Optimización.** Memorias del Octavo Verano de la ciencia Región centro, San Luís Potosí, México, Agosto del 2006.
49. Ponce J., Ponce de León E., Padilla F. y Padilla A. **Diseño De Un Modelo De Optimización Con Colonias De Hormigas.** Memorias del Primer Congreso Estatal La Investigación en el Posgrado, Universidad Autónoma de Aguascalientes, pag. 137, ISBN: 970-728-039-5, Noviembre del 2005, Aguascalientes, México.
http://www.uaa.mx/investigacion/memoria/ponencias/mesa_ieisc/doc/ponce_gallegos.doc
50. Ponce J., Padilla F., Padilla A. y Meza M. **Hormigas Atómicas.** Gaceta de la Universidad Autónoma de Aguascalientes, Diciembre de 2005, Aguascalientes, México.
51. Padilla A., Padilla F., Meza de Luna M. y Ponce J. **Club De Ciencias.** Gaceta de la Universidad Autónoma de Aguascalientes, Mayo de 2005, Aguascalientes, México

ANEXO

52. Padilla A., Padilla F., Meza de Luna M. y Ponce J. **Enseñanza De Las Matemáticas.** Gaceta de la Universidad Autónoma de Aguascalientes, Octubre de 2004, Aguascalientes, México.

53. Ponce de León E., Padilla F. y Ponce J. **Nueva Implementación Del Método De Acciones De Grupo Para La Conjetura De Hável.** Memorias del Décimo Sposium de Investigación y Desarrollo Tecnológico. Pag. 98, ISSN 1405-7042, Noviembre del 2003, Aguascalientes, México.

PUBLICACIONES EN CARTELES

54. Julio Ponce, Felipe Padilla, Alberto Ochoa, Alejandro Padilla y Eunice Ponce de León. **Closterización con Colonia de Hormigas.** 5º Congreso Estatal La Investigación en el Posgrado, Octubre de 2009, Realizado por la Universidad Autónoma de Aguascalientes, Aguascalientes, Ags.

55. Julio Ponce, Felipe Padilla, Eunice Ponce de León, Alejandro Padilla, Alberto Ochoa. **Algoritmo De Colonias De Hormigas Con Aprendizaje Diferencial Para Resolver El Problema Del Agente Viajero.** IV Seminario de Inteligencia Artificial: Un Panorama de Aplicaciones, Realizado en la Facultad de Ciencias de la Universidad Central de Venezuela, Diciembre de 2008, Caracas, Venezuela

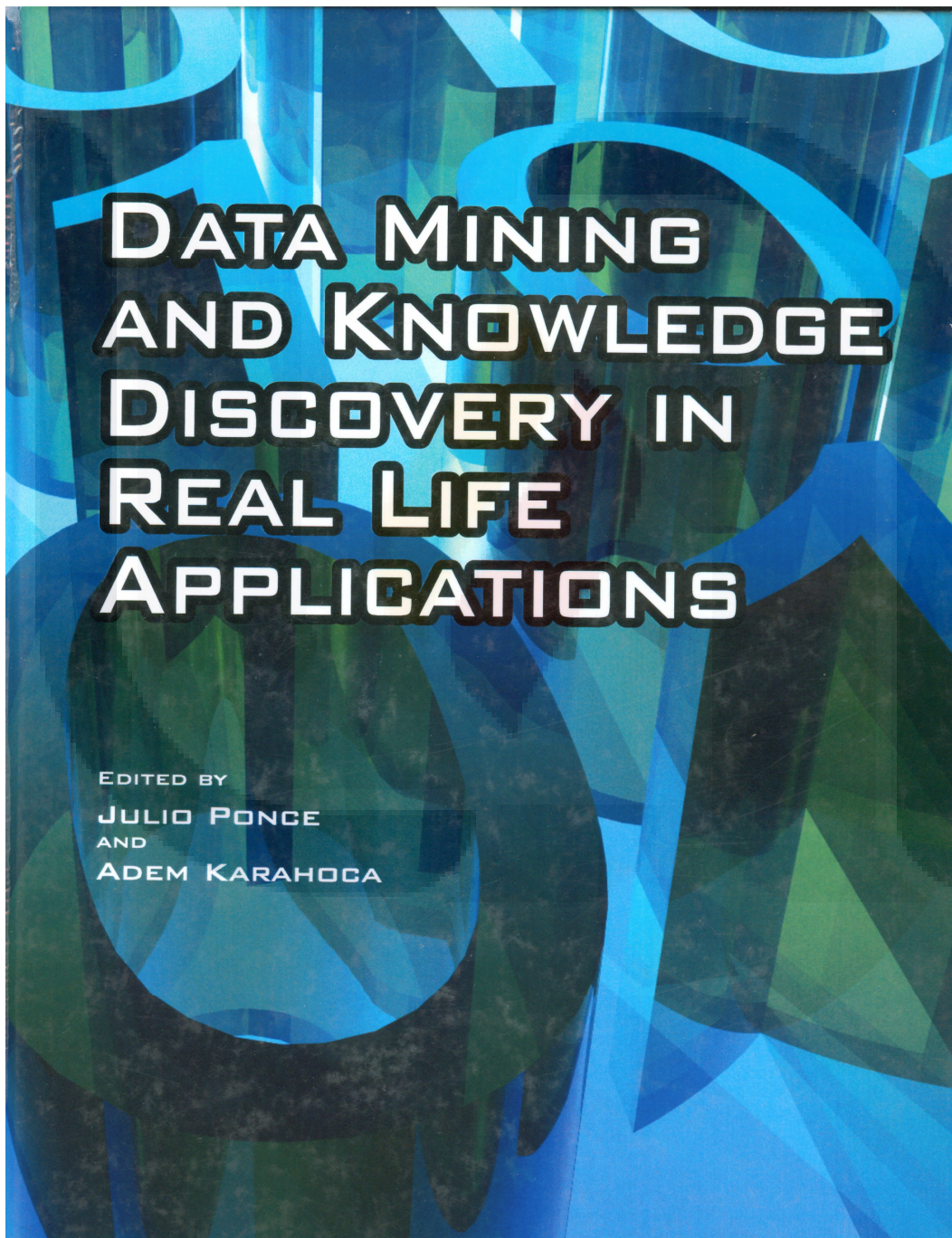
ANEXO

56. Alberto Ochoa, Antonio Zamarrón, Julio Ponce, Dzoara Baydavletov & Ernesto Mancilla. **Analyzing Competences using Fuzzy Logic**. 6° Congreso Internacional de Computó en Optimización y Software CICos 2008, Junio 2008, Cuernavaca, Morelos.
57. Alberto Ochoa, Antonio Zamarrón, Julio Ponce, Dzoara Baydavletov & Ernesto Mancilla. **Poverty in Mexico: An approach from the perspective of data mining**. 6° Congreso Internacional de Computó en Optimización y Software CICos 2008, Junio 2008, Cuernavaca, Morelos.



ANEXO

Anexo 2 Portada del Libro de Minería de Datos



ANEXO

Anexo 3 Artículo Relevante

Algoritmo de Colonia de Hormigas para el Problema del Clique Máximo con un Optimizador Local K-opt

Julio C. Ponce¹, Eunice E. Ponce de León¹, Alejandro Padilla¹, Felipe Padilla¹, Alberto Ochoa O. Zezzatti²

¹Departamento de Sistemas Electrónicos – Universidad Autónoma de Aguascalientes
Aguascalientes, México

²Facultad de Ingeniería Eléctrica – Universidad Autónoma de Zacatecas
Zacatecas, México

julk_cpg@yahoo.com.mx, {eponce, apadilla}@correo.uaa.mx,
fpadilla2000@yahoo.com, megamax8@hotmail.com

Resumen. En este artículo se muestra un Algoritmo de Colonias de Hormigas (ACH) para resolver el problema del clique máximo (PCM) el cual contiene un optimizador local K-opt en el momento de construir un clique. Esta implementación se hizo en la forma de seleccionar los nodos que forman parte del clique dentro del proceso de selección de un nodo durante la construcción de la solución, ya que se implementó el método de búsqueda local. Los resultados experimentales en este primer acercamiento nos dan buenos resultados.

Resumo. Este artigo de pesquisa descreve o algoritmo da Colônia da Formigas (ACF) pra resolver o problema da encontrar o subgraph completo mais grande dentro do graph (problema del clique máximo (PCM), em espanhol), este algoritmo tem uma ferramenta ótima local k-opt pra construir a subgraph. Esta execução ficou em a forma da escolher os nodos dentro do processo da seleção de um nodo, ficou executado o método da pesquisa local. Os resultados da o experimento em este primeiro aproximação mostraram bom resultados.

Keywords. Optimización con Colonia de Hormigas, Clique Máximo.

1. Introducción

ANEXO

El problema del clique máximo es un problema de optimización combinatoria que se clasifica dentro de los problemas NP- duros los cuales son difíciles de resolver. Debido a su complejidad las técnicas convencionales exactas (exhaustivas) tardan mucho tiempo para dar una solución, por lo tanto es necesario desarrollar algoritmos heurísticos que lo resuelvan alcanzando una solución cercana al óptimo en un tiempo razonable. Este problema tiene aplicaciones reales como son: teoría de códigos, diagnóstico de errores, visión computacional, análisis de agrupamiento, recuperación de información, aprendizaje automático, minería de datos, entre otras. Por esta razón es importante usar nuevas técnicas heurísticas y/o metaheurísticas para tratar de resolver este problema, las cuales obtengan mejores resultados en un tiempo polinomial.

Se han utilizados diferentes heurísticas para tratar de resolver este problema, como son: Búsqueda Local [Battiti and Protasi 2001, Katayama, Hamamoto, Narihisa 2004], Algoritmos Genéticos [Marchiori 2002], Búsqueda Tabú [Soriano and Gendreau 1996] y Algoritmos de Optimización de Colonia de Hormigas (OCH) [Solnon and Fenet 2006], este último muestra unos de los resultados mas recientes. Los algoritmos de optimización de colonias de hormigas es una metaheurística bioinspirada basada en el comportamiento de las hormigas naturales, en como establecen el camino más adecuado entre el hormiguero y una fuente de alimento [Dorigo, Maniezzo and Colomi 1996].

2. Descripción del Problema del Clique Máximo

Dado un grafo no dirigido cualquiera $G=(V,E)$, en el cual $V=\{1,2,\dots,n\}$ es el conjunto de los vértices del grafo y E es el conjunto de aristas. Un clique es un conjunto C de vértices donde todo par de vértices de C esta conectado con una arista en G , es decir C es un subgrafo completo. Un clique es parcial si este forma parte de otro clique, de otra forma es maximal. La meta del PCM es el encontrar un clique máximo, es decir el clique maximal con mayor cardinalidad. Se puede observar gráficamente un ejemplo de un clique en figura (1).

ANEXO

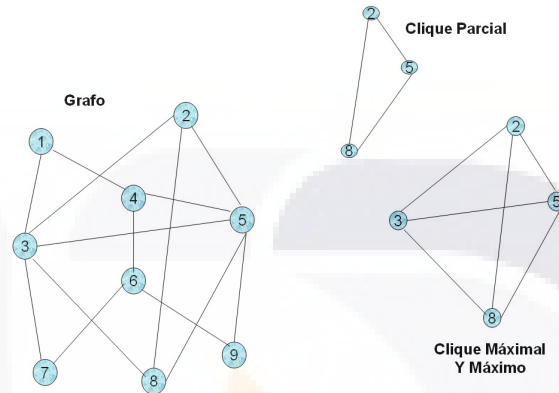


Figura 1. Ejemplo de un clique máximo

3. Algoritmo Ant Clique (AC)

Los algoritmos OCH han sido utilizados para resolver diferentes problemas de optimización combinatoria que están clasificados dentro de los NP-Duros [Dorigo, Di Caro and Gambardella 1999]. La idea principal del OCH es modelar los problemas para buscar el camino de costo mínimo en un grafo. Las hormigas recorren el grafo en busca de buenos caminos (soluciones). Cada hormiga es un agente que tiene un comportamiento simple de modo que no siempre encuentra caminos de calidad cuando esta sola. Las hormigas encuentran mejores caminos como resultado de la cooperación global entre la colonia. Esta cooperación se realiza de una manera indirecta al depositar la feromona, una sustancia que es depositada por una hormiga en su recorrido.

El algoritmo de colonia de hormigas general para el problema del clique máximo propuesto por Fenet y Solnon se muestra en la figura 2

Inicializar los rastros de feromonas

Repetir

ANEXO

Para k en $1..nbAnts$ **Hacer:** construya el clique C_k

Actualizar los rastros de feromonas w.r.t. $\{C_1, \dots, C_{nbAnts}\}$

Hasta Alcanzar el Número de Ciclos o Encontrar la solución optima

Figura 2. Seudocódigo del algoritmo del Ant-Clique

Inicialización de la feromona: Las hormigas se comunican a través de la feromona la cual es depositada en las aristas del grafo. La concentración de feromona en la arista (v_i, v_j) es denotada por $\tau(v_i, v_j)$, el rastro de feromona inicial es denotada por c , aunque se puede utilizar los límites τ_{\min} ó τ_{\max} los cuales están basados en el algoritmo MAX-MIN [Stutzle and Hoos 2000] con el propósito de tener una mayor exploración del espacio de búsqueda.

Construcción de los cliques con las hormigas: Se selecciona aleatoriamente un vértice inicial, y iterativamente se escoge vértices para agregar al clique. Dentro de un conjunto de candidatos que contiene todos los vértices que están conectados con todos los vértices del clique, ver figura 3.

Escoger aleatoriamente el primer vértice $v_f \in V$

$C \leftarrow \{v_f\}$

Candidatos $\leftarrow \{v_i / (v_f, v_i) \in E\}$

Mientras Candidato $\neq 0$ **Hacer**

Escoger un vértice $v_i \in$ Candidatos con una probabilidad $p(v_i)$, ver Ec. (2)

ANEXO

$$C \leftarrow C \cup \{v_i\}$$

$$\text{Candidatos} \leftarrow \text{Candidatos} \cap \{v_j / (v_i, v_j) \in E\}$$

Fin Mientras

Regresa C

Figura 3. Algoritmo para construir un clique

La actualización del rastro de feromona utiliza la Ec. (1).

$$\tau_{ij}(t+n) = \rho \tau_{ij}(t) + \Delta \tau_{ij} \quad (1)$$

$$p(v_i) = \frac{[\tau_{c(v_i)}]^\alpha}{\sum_{v_j \text{ candidatos}} [\tau_{c(v_j)}]^\alpha} \quad (2)$$

4. Algoritmo Propuesto

El algoritmo propuesto está basado en el algoritmo mostrado anteriormente la diferencia se encuentra en la parte de como se escoge el siguiente vértice dentro del proceso de construcción del clique la cual se muestra en la figura 4.

Obtener el grado de los vértices candidatos

Ordenarlos por el grado de mayor a menor

Si el número de vértices del grado mayor es 1 **Entonces**

 Seleccionarlo como siguiente nodo

ANEXO

Sino

Escoger uno de los vértices con mayor grado con una probabilidad $p(v_i)$, ver Ec. 2

Figura 4. Seudocódigo para seleccionar el vértice candidato

La obtención del grado de los vértices candidatos se debe de hacer en base a la lista de candidatos y no en relación al grafo original ya que esto nos aumenta la probabilidad de encontrar mejores soluciones debido a que se selecciona como siguiente vértice uno que tiene todavía muchas relaciones con los vértices candidatos restantes.

4.1. Diseño de Experimentos

Los algoritmos OCH dependen de los parámetros α que es el factor de peso (importancia) de la feromona, y ρ es el porcentaje de evaporación de la feromona. Si decrecemos el valor de α , las hormigas tienen menos sensibilidad al rastro de feromonas, y si se incrementa ρ , la evaporación de la feromona es más lenta. Cuando se incrementa la habilidad de exploración de las hormigas, estas pueden encontrar mejores soluciones pero esto implica más tiempo. Tomando en cuenta estos parámetros se ejecutó el algoritmo con los siguientes valores: número de hormigas=100, concentración inicial de la feromona $\tau_0 = 0.01$, importancia de la feromona $\alpha = 1$, factor de evaporación de la feromona $\rho = 0.99$, concentración máxima que puede tomar la feromona τ_{\max} , número de ciclos que se ejecuta el algoritmo $N_c = 100$, estos valores fueron escogidos en base a los resultados obtenidos al implementar un primer algoritmo a principios del 2006 en el cual se colocaron las hormigas dentro del grafo en los vértices con mayor grado [Ponce, Ponce de León, Padilla, Padilla 2006].

Para llevar a cabo el diseño de experimentos tomamos los benchmark de la DIMACS [DIMACS] que son los utilizados actualmente a nivel internacional para este problema.

ANEXO

5. Resultados

Se decidió resolver el problema con la ejecución del software con los parámetros antes mencionados en el algoritmo, en 20 de los 36 benchmarks de la DIMACS , y los resultados obtenidos se muestran en la Tabla 1, donde en la columna “grafo” esta el nombre que tiene el grafo en la DIMACS, La columna “Tamaño del clique benchmark” es el tamaño que tiene el clique máximo(Solución Optima) y Tamaño del Clique encontrado por ACH desarrollado.

Tabla 1. Resultados del algoritmo en los benchmarks

| Grafo | Tamaño del Clique benchmark | Tamaño del Clique encontrado por ACH |
|-------------|-----------------------------|--------------------------------------|
| C125.9 | 34 | 34* |
| C250.9 | 44 | 44* |
| C500.9 | 57 | 53 |
| C1000.9 | 68 | 63 |
| C2000.9 | 78 | 72 |
| DSJC500.5 | 13 | 13* |
| DSJC1000.5 | 15 | 13 |
| Brock200_2 | 12 | 12* |
| Brock200_4 | 17 | 17* |
| Brock400_2 | 29 | 24 |
| Brock400_4 | 33 | 28 |
| Brock800_2 | 24 | 21 |
| Brock800_4 | 26 | 21 |
| C2000.5 | 16 | 16* |
| C4000.5 | 18 | 16 |
| Hamming8_4 | 16 | 16* |
| Keller4 | 11 | 11* |
| P_hat300_1 | 8 | 8* |
| P_hat700_1 | 11 | 10 |
| P_hat1500_1 | 12 | 12* |

En la tabla anterior los resultados que cuentan con un “*” son en los que se alcanzo la solución optima de los benchmarks.

ANEXO

6. Conclusiones

En este artículo presentamos un algoritmo basado en colonia de hormigas el cual introduce un optimizador local k-opt al escoger un nuevo vértice, el cual toma en cuenta el grado de estos lo cual puede incrementar la probabilidad de encontrar cliques más grandes.

El algoritmo propuesto fue ejecutado en 20 benchmarks de la DIMACS para el problema del clique máximo de los cuales en el 50% de los benchmarks en los que se ejecutó el algoritmo obtiene el óptimo, y en la solución más lejana queda a un 19% de la solución óptima. La ventaja de este algoritmo con el reportado en el CISCI'2006 es la velocidad.

Trabajo futuro: Es importante hacer un estudio del comportamiento de los parámetros y del algoritmo, así como hacer un diseño de experimento más amplio para determinar cuales son los mejores valores para los parámetros y comparar los dos algoritmos implementados.

7. Referencias

- Battiti R. and Protasi M.(2001) Reactive local search for the maximum clique problem. *Algorithmica*, 29(4): 610–637.
- DIMACS Center for Discrete Mathematics and Theoretical Computer Science <http://dimacs.rutgers.edu/pub/challenge/graph/benchmarks/>
- Dorigo, M.; Maniezzo, V. and Colorni, A. (1996) Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man And Cybernetics –Part B: Cybernetics*, 26:1, 29-41.
- Dorigo M., Di Caro G., and Gambardella L.M.(1999) Ant algorithms for discrete optimization. *Artificial Life*, 5(2): 137–172.
- Fenet S. and Solnon C. (2003) Searching for Maximum Cliques with Ant Colony Optimization *EvoWorkshops 2003*, LNCS 2611, 236–245.
- Katayama, Hamamoto, Narihisa (2004) Solving the Maximum Clique Problem by K-opt Local Search. *ACM Symposium on Applied Computing*, 1021-1025.
- Marchiori E.(2002) Genetic, iterated and multistart local search for the maximum clique problem. In *Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim*, volume 2279, 112–121.

ANEXO

- Ponce J., Ponce de León E., Padilla F. Padilla A. (2006) Implementación de un Algoritmo de Colonia de Hormigas para el Problema del Clique Máximo. En CISCI'2006 volumen 1: 70-73.
- Solnon C. and Fenet S. (2006) A study of ACO capabilities for solving the Maximum Clique Problem Journal of Heuristic, Volume 12 - Number 3. 155-180.
- Soriano P. and Gendreau M. (1996) `Diversification strategies in tabu search algorithms for the maximum clique problem', Ann. Oper. Res., Vol. 63: 189-207.
- Stutzle T. and Hoos H.H. (2000) MAX-MIN Ant System. Journal of Future Generation Computer Systems, 16: 889-914.

ANEXO

Anexo 4 Trabajo Realizado

1. Participación como evaluador No-Ciego en el CISCI 2007
2. Participación Activa En La Primera Reunión Regional De Cuerpos Académicos en Inteligencia Artificial (Aguascalientes, Febrero del 2007)
3. Se Apoyo En El Desarrollo De El Panel: Retos Y Desafíos De La Inteligencia Artificial En Los Próximos 20 Años en el ENC06 (San Luis Potosí, Septiembre del 2006)
4. Asistencia Congreso Iberoamericano En Sistemas, Cibernética E Informática CISCI (Orlando Florida, Julio 2006)
5. Asistencia al Seminario Ofrecido Por La Línea De Investigación De “Metaheurísticas” (Agosto Diciembre del 2006)
6. Asistencia al Seminario Ofrecido Por La Academia De Inteligencia Artificial (Agosto Diciembre del 2006)
7. Asistencia al Seminario De “Seguridad Mediante Matemáticas Discretas” impartido por el Dr. Miguel Vargas Martín (1° al 23 de Agosto del 2006)

ANEXO

8. Miembro del comité Organizador Del Tercer Congreso Mexicano De Computación Evolutiva COMCEV'07 A Realizarse En Octubre Del 2007 En La Ciudad De Aguascalientes (desde Julio del 2006)

9. Colaborador en el proyecto PIIInf 06-2 F.T. 7-2 Diseño y evaluación de un sistema tutor inteligente para el autoaprendizaje en el nivel de licenciatura.

Enero del 2006 a la Fecha

Participantes del Proyecto

- MC. Miguel Angel Meza de Luna
- Mtro. Alejandro Padilla Díaz
- Dr. Felipe Padilla Díaz
- Ing. Julio César Ponce Gallegos

10. PIIInf 05-3 F.T. 6-2 Metaheurísticas y métodos de Monte Carlo para el problema de selección de subconjuntos de características.

Enero del 2006 a la Fecha

Participantes del Proyecto

- Dra. Eunice Ponce de León Senti
- Dr. Felipe Padilla Díaz
- MC. Elva Díaz Díaz
- Mtra. Aurora Torres Soto
- Mtra. Ma. Dolores Torres Soto
- Ing. Julio César Ponce Gallegos

ANEXO

Anexo 5 Ponencias

1. Ant Colony Algorithm for Clustering through of Cliques, II Workshop of Hybrid Intelligent Systems, 9 de Noviembre del 2009, Guanajuato, México
2. Six degrees of separation in a graph to a Social Networking, XXII Congreso Nacional y VIII Internacional de Informática y Computación ANIEI 2009, 22 de Octubre del 2009, Ensenada, Baja California
3. Metodología de implementación de OA's en la materia de Lenguajes de Computación mediante B-learning, XXII Congreso Nacional y VIII Internacional de Informática y Computación ANIEI 2009, 22 de Octubre del 2009, Ensenada, Baja California
4. Algoritmo de Colonia de Hormigas para Clusterización, XXII Congreso Nacional y VIII Internacional de Informática y Computación ANIEI 2009, 22 de Octubre del 2009, Ensenada, Baja California
5. Conferencia: Algoritmos basados en Inteligencia Colectiva para la solución de problemas de optimización, Congreso de Ciencias Exactas de la Universidad Autónoma de Aguascalientes, 6 de Octubre del 2009, Aguascalientes, Ags
6. Colonias de hormigas aplicadas a Data Mining, Seminario/Taller de Data Minig, 28 de Febrero del 2009, Aguascalientes, Ags

ANEXO

7. Desarrollo de una versión animada de la ruta de la Glucólisis para facilitar la Enseñanza-Aprendizaje en las Aulas de Estudio Tercera exposición de Mini-Proyectos de Investigación, Noviembre del 2008, Aguascalientes, Ags.
8. Minería de Datos Aplicada a Redes Sociales, 30° Aniversario de la creación del Instituto Tecnológico del Llano, 31 de Octubre del 2008, El Llano, Ags., México
9. From Russia with Disdain: Simulating a Civil War by Means of Predator/Prey Game and Cultural Algorithms, Seventh Mexican International Conference on Artificial Intelligence 27 de Octubre del 2008, Atizapan, México
10. Algoritmo de Colonias de Hormigas con Aprendizaje Cuarto Congreso Estatal “La Investigación en el Posgrado”, 17 de Octubre del 2008, Aguascalientes, México
11. Hybrid System for Determining a Ranking in Eurovision Song Contest, 9th Mexican International Conference on Computer Science, ENC’08, 6 de Octubre del 2008, Mexicali, B.C., México
12. Administrador de Páginas Web con Técnicas de Inteligencia Artificial XXI Congreso Nacional y VII Internacional de Informática y Computación ANIEI 2008, 1° de Octubre del 2008, Monterrey, Nuevo León

ANEXO

13. Optimización con Colonia de Hormigas, Semana Tecnológica 2008 del Instituto Tecnológico Superior de Irapuato 25 de Septiembre del 2008, Irapuato, Guanajuato
14. Determining Ranking of a new Participant in Eurovisión and Data Mining Using Cultural Algorithms, XVIII International Conference on Electronics Communications and Computers, 4 de Marzo del 2008, Cholula, Puebla
15. Implementación de la IA en el Juego del SUDOKU, Segunda exposición de Mini-Proyectos de Investigación, Noviembre del 2007, Aguascalientes, Ags.
16. La Inteligencia Artificial Aplicada en la Robotica Lego, Segunda exposición de Mini-Proyectos de Investigación, Noviembre del 2007, Aguascalientes, Ags.
17. Algoritmo de Colonia de Hormigas aplicado al problema de Bin Parking, Tercer Congreso Estatal La Investigación en el Posgrado realizado el 26 de Octubre del 2007, Aguascalientes, Ags.
18. Conferencia “Optimización de Procesos Productivos con Colonias de Hormigas”, XXVI Aniversario del Plantel CETis 80 el 22 de Octubre del 2007, Aguascalientes, Ags.

ANEXO

19. Assisted location of web advertisement zones using interactive genetic algorithms guided by event tracking, Congreso Internacional de Computación Evolutiva COMCEV'07 4 de Octubre del 2007 Aguascalientes, Ags.
20. Ahankara: Identifying Bipolar Síndrome in Users of Orkut with Data Mining, Workshop de Minería de Datos, Dentro del 8° Encuentro Internacional de Computación 2007 Septiembre de 2007, Morelia, Michoacán
21. Conferencia “Objetos de Aprendizaje en la Teoría de Sistemas”, Como parte de las actividades de la Universidad la Concordia el 12 de Septiembre de 2007, Aguascalientes, Ags.
22. Diseño de un Modelo Adaptativo para Resolver el Problema del Viajante (TSP) Basado en Optimización de Colonias de Hormigas (ACO), Seminario Organizado por la Academia de Inteligencia Artificial realizado del 1al 3 de Agosto de 2007, Aguascalientes, Ags.
23. Conferencia “Optimización de Colonia de Hormigas Aplicado al Problema del Clique Máximo” Symposium Internacional de Optimización y Ecuaciones Diferenciales Abril del 2007 Aguascalientes, México
24. Conferencia Magistral “Optimización de Procesos Mediante Colonia de Hormigas” Primer Simposio en Ciencias Computacionales Marzo del 2007 Tepezala, Aguascalientes

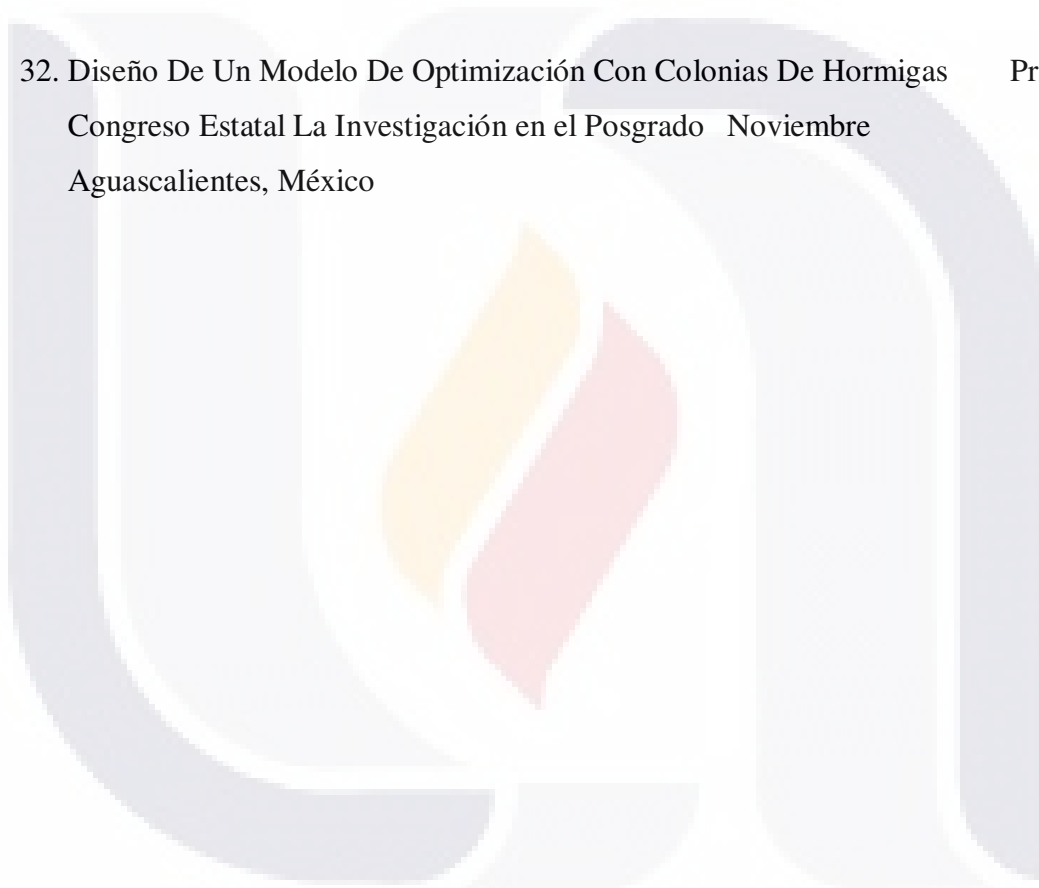
ANEXO

25. Algorithm Of Ant Colony Applied To Subset Problems In Real Time Coloquio del Posgrado en Ciencias Exactas, Sistemas y de la Información Enero del 2007 Aguascalientes, México
26. Aplicación De La Técnica De Colonia De Hormigas En Un Dominio De Aplicación Para Optimización En el Primer Foro de Ciencias Computacionales, Noviembre del 2006 Zacatecas, México
27. Algoritmos De Colonia De Hormigas Aplicados Al Problema De La Mochila Segundo Congreso Estatal La Investigación en el Posgrado Noviembre del 2006 Aguascalientes, México
28. Análisis Del Uso De Algoritmos Genéticos En La Optimización De Funciones Matemáticas Segundo Congreso Estatal La Investigación en el Posgrado Noviembre del 2006 Aguascalientes, México
29. Algoritmo De Colonia De Hormigas Aplicado Al Problema Del Clique Máximo Segundo Congreso Estatal La Investigación en el Posgrado Noviembre del 2006 Aguascalientes, México
30. Implementación De Un Algoritmo De Colonia De Hormigas Para El Problema Del Clique Máximo 5ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CИСCI 2006 Julio del 2006 Orlando Florida, USA.

ANEXO

31. Búsqueda Dispersa Para El Problema De Selección De Subconjuntos De Características 5ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2006 Julio del 2006 Orlando Florida, USA.

32. Diseño De Un Modelo De Optimización Con Colonias De Hormigas Primer Congreso Estatal La Investigación en el Posgrado Noviembre 2005 Aguascalientes, México

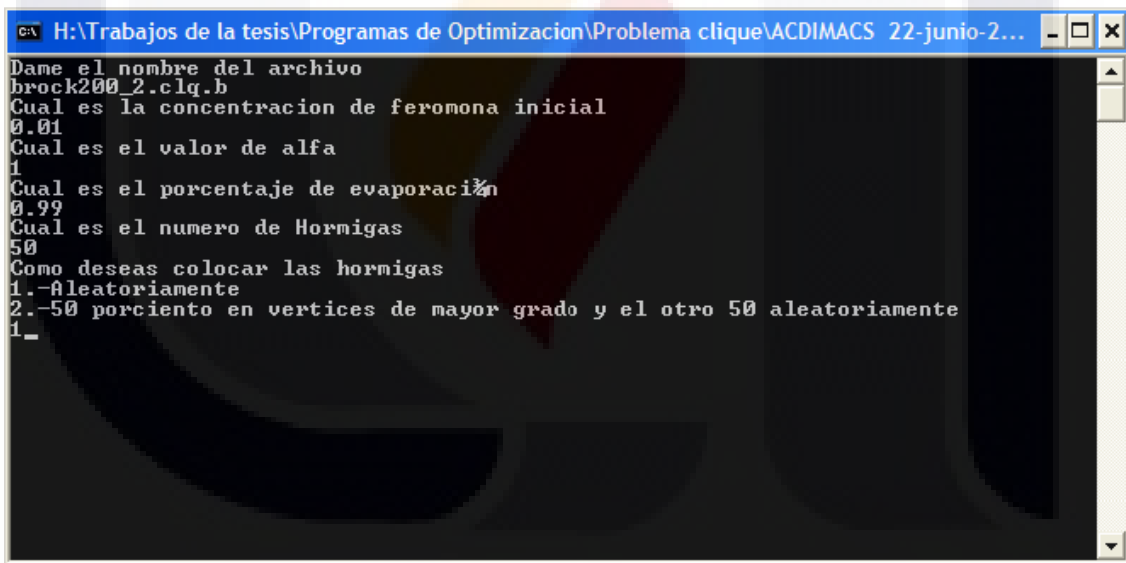


ANEXO

Anexo 6 Pantallas de los Sistemas

Sistema reportado en el CISCI 2006 en Orlando para resolver el problema del clique máximo, el cual fue elaborado en Borland C v. 5.01.

Al ejecutar el programa se piden los parámetros con los cuales se va a ejecutar el sistema, como se puede ver en la Figura 1, cabe mencionar que en este caso no se utiliza el parámetro Beta debido a que no se tienen distancias.



```

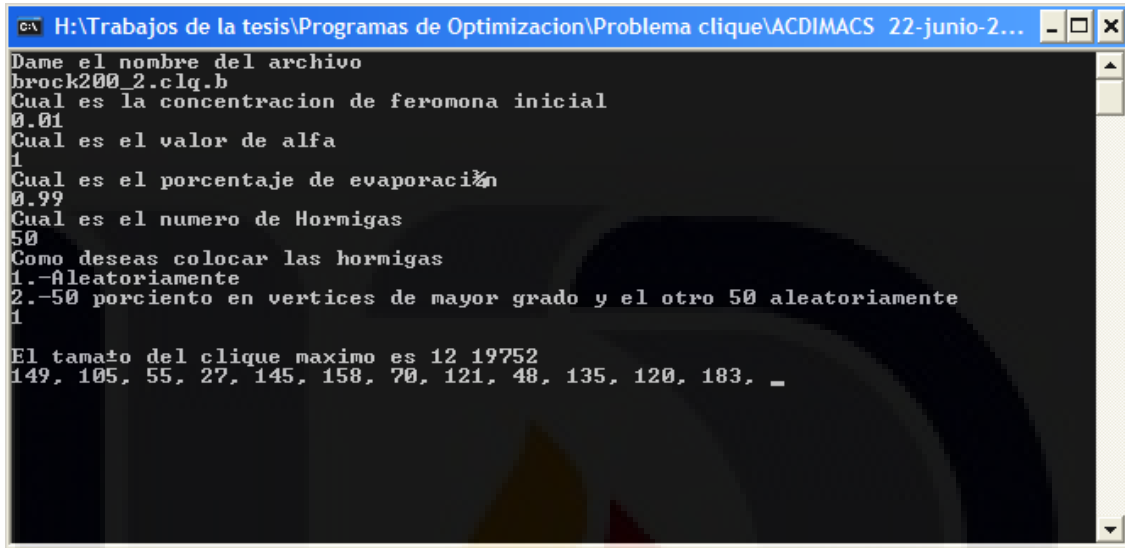
C:\> H:\Trabajos de la tesis\Programas de Optimizacion\Problema clique\ACDIMACS 22-junio-2...
Dame el nombre del archivo
brock200_2.clq.b
Cual es la concentracion de feromona inicial
0.01
Cual es el valor de alfa
1
Cual es el porcentaje de evaporacion
0.99
Cual es el numero de Hormigas
50
Como deseas colocar las hormigas
1.-Aleatoriamente
2.-50 porciento en vertices de mayor grado y el otro 50 aleatoriamente
1_

```

Figura1. Pantalla donde se muestra los datos que solicita este sistema

Una vez introducidos los parámetros el sistema hace la búsqueda y muestra el mejor resultado obtenido, ver Figura 2.

ANEXO



```
cmd H:\Trabajos de la tesis\Programas de Optimización\Problema clique\ACDIMACS 22-junio-2... - □ ×
Dame el nombre del archivo
brock200_2.clq.b
Cual es la concentracion de feromona inicial
0.01
Cual es el valor de alfa
1
Cual es el porcentaje de evaporación
0.99
Cual es el numero de Hormigas
50
Como deseas colocar las hormigas
1.-Aleatoriamente
2.-50 porciento en vertices de mayor grado y el otro 50 aleatoriamente
1
El tamaño del clique maximo es 12 19752
149, 105, 55, 27, 145, 158, 70, 121, 48, 135, 120, 183, _
```

Figura 2. Pantalla después de hacer el proceso de búsqueda

Sistema reportado en el SIM 2006 en Brasil para resolver el problema del clique máximo, fue elaborado en Borland C v. 5.01.

Al ejecutar el programa se piden los parámetros con los cuales se va a ejecutar el sistema, como se muestra ver en la Figura 3, cabe mencionar que en este caso no se utiliza el parámetro Beta debido a que no se tienen distancias, ni los parámetros de número de hormigas y la forma de colocar las hormigas debido a que es el sistema que realiza el diseño de experimento y estos valores fueron asignados internamente.

ANEXO

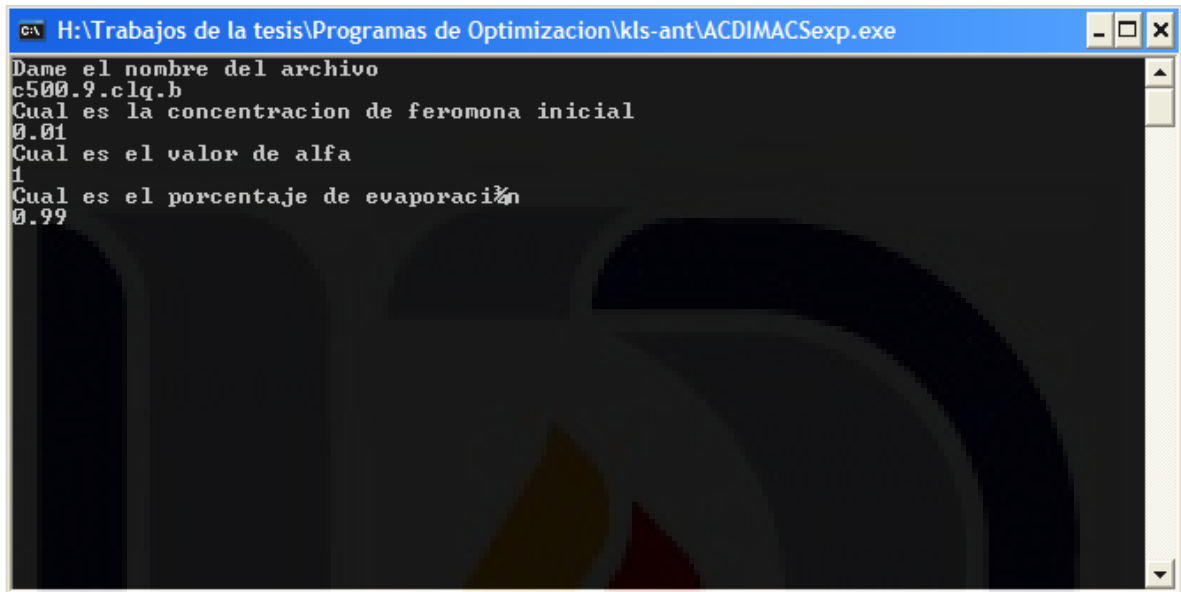


Figura3. Pantalla donde se muestra los datos que solicita este sistema

Una vez introducidos los parámetros el sistema hace la búsqueda y muestra el resultado obtenido a través de un archivo de texto llamado experimentos en el cual se muestra la siguiente información: nombre del benchmark ejecutado, la concentración inicial de la feromona, el valor de Alfa, el porcentaje de evaporación, el número de hormigas y como están colocadas como encabezado, después aparece cuatro números que son, el número de hormigas, número de ciclos, y un tercer número que indica la forma en que fueron colocadas las hormigas (1.- Aleatoreamente, 2.-75% aleatoreamente y 25% en nodos de mayor grado y por ultimo 3.-50% y 50% . En ver Figura 4.

ANEXO

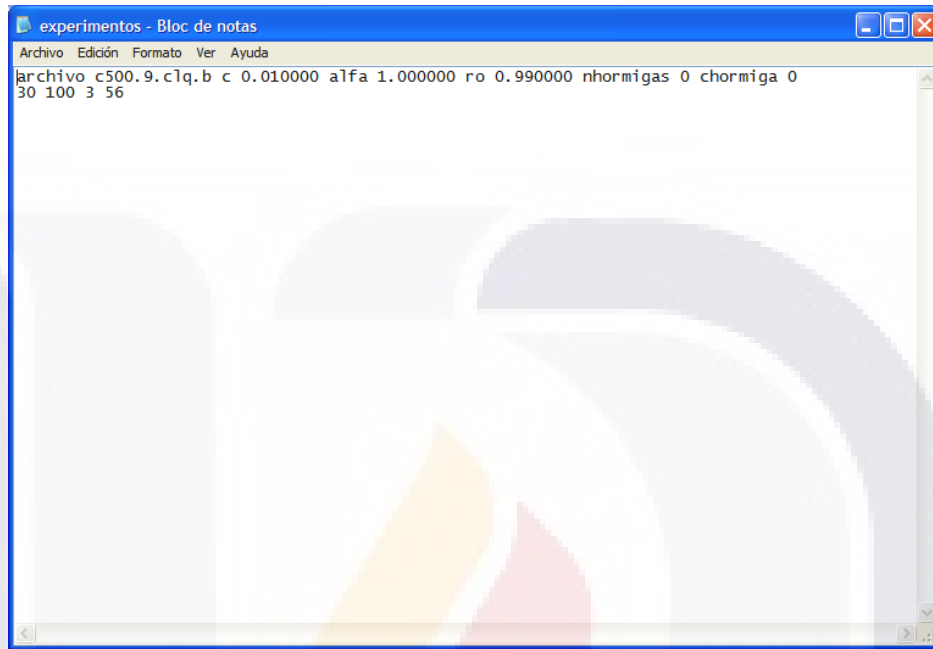


Figura 4. Archivo txt generado

Sistema reportado en el CИСCI 2007 en Orlando para resolver el problema de la mochila, fue elaborado en Borland C v. 5.01.

Al ejecutar este sistema, no se pide ningún parámetro, actualmente todo esta por medio de programación, lo que hace el sistema es abrir un archivo .dat dentro del cual se tienen los datos del problema a resolver que son: el número de Objetos, la capacidad de la mochila y en forma de lista se enumeran los objetos, se pone su peso y su utilidad del lado derecho ver Figura 5.

ANEXO

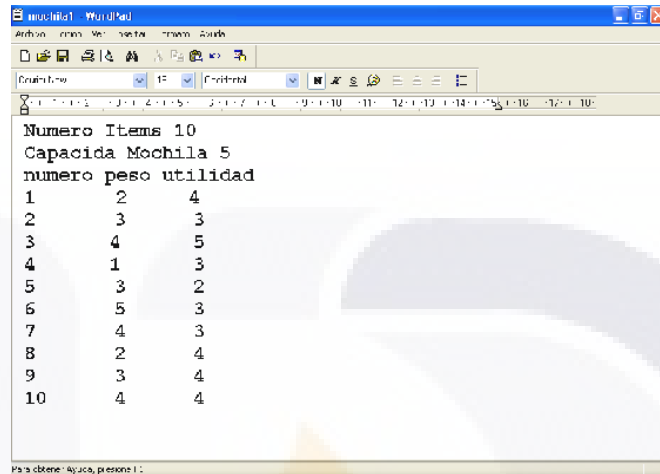


Figura 5. Forma del archivo .dat

Una vez leído este archivo el sistema muestra como salida la información del archivo leído y la solución en la cual se dice cuales objetos son los seleccionados, cual es el peso total de dichos objetos y la utilidad obtenida al seleccionarlos, ver la Figura 6.

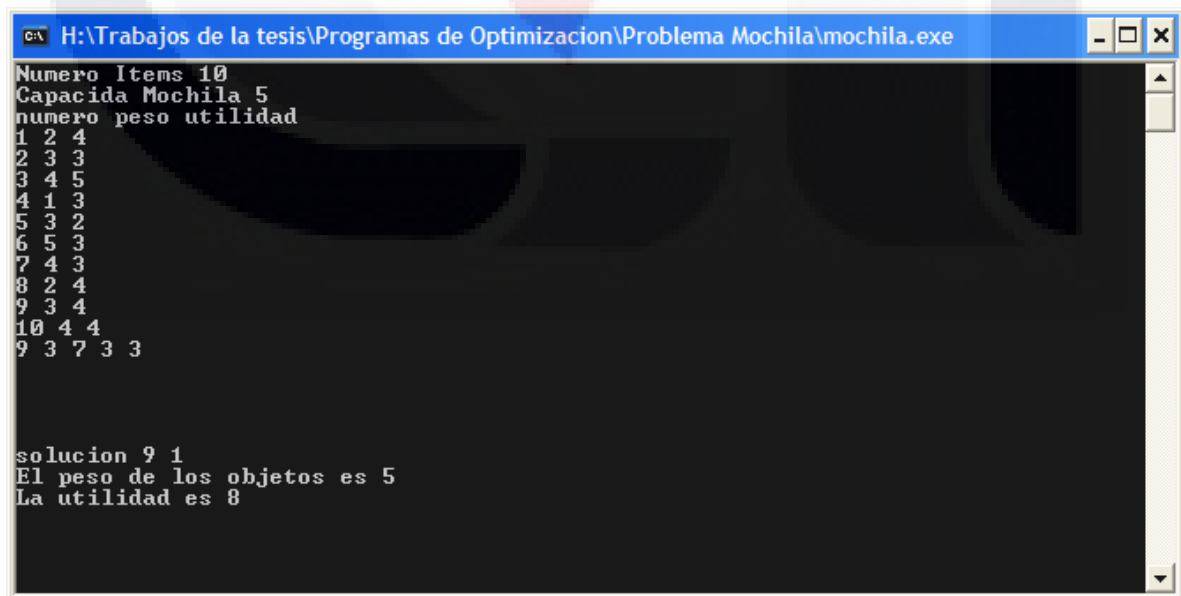


Figura 6 Pantalla donde se muestra la solución al problema de la mochila

ANEXO

Anexo 7 Miembro del Comité Revisor en Congresos

| Evento | Lugar |
|--|---------------------------|
| Hybrid Artificial Intelligence Systems 2010 | España |
| 2a Conferencia Iberoamericana en Generación, Comunicación y Gerencia del Conocimiento 2009 | Orlando, Florida |
| 9a Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2010 | Orlando, Florida |
| XXII Congreso Nacional y VIII Internacional de Informática y Computación | Ensenada, Baja California |
| ANIEI 2009 | |
| Mexican International Conference on Artificial Intelligence 2009 | CIMAT |
| Hybrid Artificial Intelligence Systems 2009 | España |
| 1a Conferencia Iberoamericana en Generación, Comunicación y Gerencia del Conocimiento 2009 | Orlando, Florida |

ANEXO

8a Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2009 Orlando, Florida

7ma Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2008 Orlando, Florida

Tercer Congreso Internacional de Computación Evolutiva COMCEV'07 Aguascalientes, México

6ta Conferencia Iberoamericana en Sistemas, Cibernética e Informática CISCI 2007 Orlando, Florida

ANEXO

Anexo 8 Bibliografía Artículos Relacionados con la Tabla de Problemas Tratados con ACO

Abraham A y Ramos Victor (2003). **Web Usage Mining Using Artificial Ant Colony Clustering and Linear Genetic Programming**

Alaya I., Solnon C. and Ghedira K.(2004) **ANT ALGORITHM FOR THE MULTIDIMENSIONAL KNAPSACK PROBLEM**, Dans Proceedings of international conference on Bioinspired Methods and their Applications (BIOMA 2004), Slovenia. pp. 63-72.

Bui, T. N., T. H. Nguyen, and J. R. Rizzo Jr. (2009). **Parallel Shared Memory Strategies for Ant-Based Optimization Algorithms**. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'09), ACM Press, Seattle, pp. 1-8.

Bui T.N. y Rizzo J.R.(2004) **Finding maximum cliques with distributed ants**. In K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E.K. Burke, P.J. Darwen, D. Dasgupta, D. Floreano, J.A. Foster, M. Harman, O. Holland, P.L. Lanzi, L. Spector, A. Tettamanzi, D. Thierens, and A.M. Tyrell, editors, Genetic and Evolutionary Computation - GECCO 2004, volume 3102 of Incs, pages 24{35. Springer-Verlag, 2004.

ANEXO

Bullnheimer B., Hartl R. y Strauss C.(1999a) **“An improved ant system algorithm for the vehicle routing problem”**. *Annals of Operations Research* (Dawind, Feichtinger and Hartl (eds.): Nonlinear Economic Dynamics and Control.

Bullnheimer, B., Hartl, R.F. y Strauss, C. (1999b). **Applying the Ant System to the vehicle routing problem**, in: S. Voß, S. Martello, I.H. Osman, C. Roucairol (Eds.), *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer Academic Publishers, Boston, MA.

Cordón O., de Viana I. F., Herrera F, & Morena LL. (2000) **A new ACO Model Integrating Evolutionary Computation Concepts: the Best-Worst Ant System**. *In Second International Workshop on Ant Algorithms*.

Dorigo, M. (1992) **Optimization, learning and natural algorithms**, Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, (in Italian).

Dorigo M. y Gambardella L., (1997a) **Ant colonies for the traveling salesman problem**, in *BioSystems*, 1997.

Dorigo M. y Gambardella L., (1997b) **“Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”**. *IEEE Trans. on Evolut. Comp.*, Vol 1, No. 1, pp. 53-66.

ANEXO

Dorigo, M., Maniezzo, V. y Colorni, A. (1991) **Ant System: An Autocatalytic Optimizing Process.** *Technical report 91-016 revised, Dipartimento de Electronia, Politecnico di Milano, Milan.*

Dorigo, M.; Maniezzo, V. y Colorni, A. (1996): **Ant System: Optimization by a Colony of Cooperating Agents.** *IEEE Transactions on Systems, Man And Cybernetics – Part B: Cybernetics, 26:1, 29-41.*

Dorigo, M. y Stützle, T.(2004) **Ant Colony Optimization.** *A Bradford Book, The MIT Press Cambridge, Massachusetts, London England.*

Elliman D. G. and Youssef S.M. (2004) **Reactive Prohibition-based Ant Colony Optimisation (RPACO): A New Parallel Architecture for Constrained Clique Sub-Graphs,** Computer Science Technical Report No. NOTTCS-TR-2004-7 in University of Nottingham.

Fenet, S. y Solnon, C.(2003) **Searching for Maximum Cliques with Ant Colony Optimization.** *EvoWorkshops 2003, LNCS 2611, 236–245.*

ANEXO

Fidanova S. (2002) **ACO Algorithm for MKP Using Different Heuristic Information**
5th Int Conference of Numerical Methods and Applications, Lecture Notes in
Computer Science No 2542, Springer, Germany , 2002, 434-440.

Fidanova S. (1999) **Evolutionary Algorithm for Multiple Knapsack Problem.** PPSNVII-
Workshop 2002.

Fidanova S. (2004) **Ant Colony Optimization for Multiple Knapsack Problem and
Model Bias**, NAA'04, *lecture Notes in Computer Sciences*, Springer, Germany, 282-
289

Gambardella, L.M. y Dorigo, M. (1995) **Ant-Q: A Reinforcement Learning approach to
the traveling salesman problem**

Gambardella, L.M. y Dorigo, M. (1996) **Solving symmetric and asymmetric TSPs by ant
colonies**, in: Proceedings of the IEEE Conference on Evolutionary Computation,
ICEC'96, IEEE Press, New York, pp. 622–627.

Handl J., J. Knowles and M. Dorigo (2003a). **Ant-based clustering: a comparative study
of its relative performance with respect to k-means, average link and ld-som.**
Technical Report TR/IRIDIA/2003-24. IRIDIA, Universite Libre de Bruxelles,
Belgium.

ANEXO

Handl J., J. Knowles and M. Dorigo (2003b). On **the performance of ant-based clustering**. In Design and Application of Hybrid Intelligent Systems, Vol. 104 of Frontiers in Artificial Intelligence and Applications (pp. 204-213). Amsterdam, The Netherlands: IOS Press.

Leguizamón, G. y Michalewicz, Z.(1999)**A New Version of Ant System for Subset Problems**, Congress on Evolutionary Computation pp1459-1464.

Monmarché N., M. Slimane y G. Venturini (1997) **AntClass: discovery of clusters in numeric data by an hybridization of an ant colony with the kmeans algorithm**. Technical Report No.213, E3I, Laboratoire d'Informatique, University of Tours

Monmarché N.(1999) **On data clustering with artificial ants**. In Freitas, A., editor, AAAI-99 & GECCO-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions, pages 23-26.

Montes de Oca M., Garrido L and Aguirre J. L. (2005a) **An hybridization of an antbased clustering algorithm with growing neural gas networks for classification tasks**

ANEXO

Montes de Oca M., Garrido L. and Aguirre J. L. (2005b) **Effects of Inter-agent Communication in Ant-Based Clustering Algorithms: A case Study on Communication Policies in Swarm Systems**

Montgomery, J., & Randall, M. (2002). **The Accumulated Experience Ant Colony for the Travelling Salesman Problem.** *International Journal of Computational Intelligence and Applications*, 189–198.

Parpinelli R. S., Lopes H. S., and Freitas A. A. (2001). **Data Mining with an Ant Colony Optimization Algorithm.**

Ponce J., Ponce de León E., Padilla F. y Padilla A. (2006a) **Implementación De Un Algoritmo De Colonia De Hormigas Para El Problema Del Clique Máximo.** *En CISCI'2006 volume 1 pages70-73, Orlando Florida.*

Ponce J., Ponce de León E., Padilla F. y Padilla A. (2006a) **Implementación De Un Algoritmo De Colonia De Hormigas Para El Problema Del Clique Máximo.** *En CISCI'2006 volume 1 pages70-73, Orlando Florida.*

Ponce J., Ponce de León E., Padilla F. y Padilla A. (2006b) **Algoritmos De Colonia De Hormigas Aplicados Al Problema De La Mochila.** *En “Segundo Congreso Estatal La Investigación en el Posgrado”, Aguascalientes*

ANEXO

Ponce J., Ponce de León E., Padilla F. y Padilla A. (2006c) **Algoritmo De Colonia De Hormigas Aplicado Al Problema Del Clique Máximo.** En “*Segundo Congreso Estatal La Investigación en el Posgrado*”, Aguascalientes.

Ponce J., Ponce de León E., Padilla F. Padilla y A. Zezzatti A.(2006) **Algoritmo de Colonia de Hormigas para el Problema del Clique Máximo con un Optimizador Local k-opt.** En *11° Simposio de Informática y 6° Mostra de Software Academico SIMS 2006, Uruguaiiana Brasil. Hífen, Uruguaiiana, v. 30, n. 58, 2006.*

Ramos V., Muge F., Pina P. (2002), **Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies,** Soft Computing Systems - Design, Management and Applications, 2nd Int. Conf. on Hybrid Intelligent Systems, IOS Press, pp. 500-509.

Ramos V. y Merelo J. J. (2002). **Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning,** in E. Alba, F. Herrera, J.J. Merelo et al. (Eds.), AEB'02 - 1st Int. Conf. On Metaheuristics, Evolutionary and Bio-Inspired Algorithms, pp. 284-293, Mérida, Spain.

Ramos V. y Almeida F. (2000). **Artificial Ant Colonies in Digital Image Habitats - A Mass Behaviour Effect Study on Pattern Recognition,** in Marco Dorigo, Martin

ANEXO

Middendorf and Thomas Stützle (Eds.), Proc. of ANTS'00 – 2nd Int. Workshop on Ant Algorithms, pp. 113-116, Brussels, Belgium.

Ren, Z.-G., Feng Z.-R., Ke L.-J. and Zhang Z.-J. (2010). **New ideas for applying ant colony optimization to the set covering problem.** Computers & Industrial Engineering, Elsevier.

Sim K. M. and Sun W.H. (2002) **“Multiple ant-colony optimization for network routing”**, in *Proc. 1st Int. Symp. Cyberworld*, Tokyo, Japan, November 2002, pp. 277–281.

Sim K. M. and Sun W.H. (2003) **“ Ant Colony Optimization for Routing and Load - Balancing: Survey and New Directions”**, IEEE Transactions on Systems, Man, and Cybernetics—PART A: Systems and Humans, Vol. 33, No. 5, September 2003.

Solnon C. (2000) **Solving Permutation Constraint Satisfaction Problems with Artificial Ants.** *Proceedings of the 14th European Conference on Artificial Intelligence (pp. 118-122).* Amsterdam, IOS Press.

Solnon C. (2002) **Ants can solve Constraint Satisfaction Problems.** IEEE transactions on evolutionary computation, vol. 6, no. 4, august.

ANEXO

Solnon C. (2008). **Combining two pheromone structures for solving the car sequencing problem with Ant Colony Optimization.** European Journal of Operational Research 191, 1043–1055

Solnon C. and Bridge D. (2006) **An Ant Colony Optimization Meta-Heuristic for Subset Selection Problems,** in *N.Nedjah and L.M.Mourelle (eds.), Systems Engineering using Swarm Particle Optimisation*, pp.7-29, Nova Science Publishers.

Solnon, C. y Fenet, S.(2004) **Investigating ACO capabilities for solving the Maximum Clique Problem.** Rapport de recherche RR-LIRIS-2004-021 , Soumis à Journal of Heuristics.

Solnon C. y Fenet S. (2005) **A study of ACO capabilities for solving the Maximum Clique Problem** Journal of Heuristic, Springer.

Stutzle, T. y Dorigo, M.(1999). **ACO Algorithms for the Traveling Salesman Problem,** in: P.Neittaanmaki, J. Periaux, K.Miettinen, M.M. Makela(Eds.), *Evolutionary Algorithms in Engineering and Computer Science*, Wiley, Chichester, UK., pp. 163-183.

Stützle,T. y Hoos, H.(1997) **The MAX-MIN Ant System and local search for the traveling salesman problem,** in: T. Bäck, Z. Michalewicz, X. Yao (Eds.),

ANEXO

Proceedings of IEEE-ICEC-EPS'97, IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference, IEEE Press, New York, pp. 309–314.

Peters M. and Zaki M. J. (2004), **CLICK: Clustering Categorical Data using K-partite Maximal Cliques**. TR 04-11, CS Dept., RPI.

Pratt S., Sumpter D., Mallon E. & Franks N.(2005). **An agent-based model of collective nest choice by the ant *Temnothorax albipennis***. ANIMAL BEHAVIOUR, 2005, 70, 1023–1036, ELSEVIER.

ANEXO

Anexo 9 Análisis Estadístico

| | | Notes |
|-------------------------------|---------------------------------------|--|
| Output Created | | 10 Mar 06 23:29:30 |
| Comments | | |
| Input | Data | C:\Análisis Multivariante\Julio\brock\Brock.sav |
| | Filter | <none> |
| | Weight | <none> |
| | Split File | <none> |
| | N of Rows in Working Data File | 240 |
| Missing Value Handling | Definition of Missing | User-defined missing values are treated as missing. |
| | Cases Used | Statistics are based on all cases with valid data for all variables in the model. |
| Syntax | | UNIANOVA clique BY param1 param2 param3 /METHOD = SSTYPE(3) /INTERCEPT = INCLUDE /EMMEANS = TABLES(param1*param2*param3) /PRINT = DESCRIPTIVE HOMOGENEITY /CRITERIA = ALPHA(.05) |

ANEXO

| | | |
|------------------|---------------------|----------------------------------|
| | | /DESIGN = param1 param2 param3 . |
| Resources | Elapsed Time | 0:00:00.11 |

| Levene's Test of Equality of Error Variances(a) | | | |
|---|------------|------------|-------------|
| Dependent Variable: Dif con clique | | | |
| F | df1 | df2 | Sig. |
| 9.498 | 11 | 228 | .000 |
| Tests the null hypothesis that the error variance of the dependent variable is equal across groups. | | | |
| a Design: Intercept+PARAM1+PARAM2+PARAM3 | | | |

| Tests of Between-Subjects Effects | | | | | |
|--|--------------------------------|-----------|--------------------|-----------|-------------|
| Dependent Variable: Dif con clique | | | | | |
| Source | Type III Sum of Squares | df | Mean Square | F | Sig. |
| Corrected Model | 14.375(a) | 4 | 3.594 | 17.623 | .000 |
| Intercept | 10626.704 | 1 | 10626.704 | 52112.522 | .000 |
| PARAM1 | .704 | 1 | .704 | 3.453 | .064 |
| PARAM2 | 13.538 | 1 | 13.538 | 66.387 | .000 |

ANEXO

| | | | | | |
|------------------------|-----------|-----|-----------|------|------|
| PARAM3 | .133 | 2 | 6.667E-02 | .327 | .721 |
| Error | 47.921 | 235 | .204 | | |
| Total | 10689.000 | 240 | | | |
| Corrected Total | 62.296 | 239 | | | |

a R Squared = .231 (Adjusted R Squared = .218)

Estimated Marginal Means

| nhorm * nciclos * colocacion hormigas | | | | | | | | |
|---------------------------------------|---------|---------------------|-------|------------|-------------------------|-------|-------------|-------------|
| Dependent Variable: Dif con clique | | | | | | | | |
| | | | Mean | Std. Error | 95% Confidence Interval | | Lower Bound | Upper Bound |
| Nhorm | nciclos | colocacion hormigas | | | | | | |
| 30 | 100 | 100 % | 6.929 | .065 | 6.801 | 7.058 | | |
| | | 75% | 6.979 | .065 | 6.851 | 7.108 | | |
| | | 50% | 6.929 | .065 | 6.801 | 7.058 | | |
| | 500 | 100 % | 6.454 | .065 | 6.326 | 6.583 | | |
| | | 75% | 6.504 | .065 | 6.376 | 6.633 | | |

ANEXO

| | | | | | | |
|-----------|------------|--------------|-------|------|-------|-------|
| | | 50% | 6.454 | .065 | 6.326 | 6.583 |
| 50 | 100 | 100 % | 6.821 | .065 | 6.692 | 6.949 |
| | | 75% | 6.871 | .065 | 6.742 | 6.999 |
| | | 50% | 6.821 | .065 | 6.692 | 6.949 |
| | 500 | 100 % | 6.346 | .065 | 6.217 | 6.474 |
| | | 75% | 6.396 | .065 | 6.267 | 6.524 |
| | | 50% | 6.346 | .065 | 6.217 | 6.474 |

Univariate Analysis of Variance

| | | |
|-----------------------|--------------------|---|
| Notes | | |
| Output Created | 10 Mar 06 23:32:39 | |
| Comments | | |
| Input | Data | C:\Análisis Multivariante\Julio\brock\Brock.sav |
| | Filter | <none> |
| | Weight | <none> |
| | Split File | <none> |

ANEXO

| | | |
|-------------------------------|---|---|
| | N of Rows in Working Data File | 240 |
| Missing Value Handling | Definition of Missing | User-defined missing values are treated as missing. |
| | Cases Used | Statistics are based on all cases with valid data for all variables in the model. |
| Syntax | UNIANOVA clique BY param2 /METHOD = SSTYPE(3) /INTERCEPT = INCLUDE /PRINT = DESCRIPTIVE HOMOGENEITY /CRITERIA = ALPHA(.05) /DESIGN = param2 . | |
| Resources | Elapsed Time | 0:00:00.03 |

| Between-Subjects Factors | | | |
|---------------------------------|----------|--------------------|----------|
| | | Value Label | N |
| nciclos | 1 | 100 | 120 |
| | 2 | 500 | 120 |

ANEXO

| Descriptive Statistics | | | |
|------------------------------------|--------|----------------|-----|
| Dependent Variable: Dif con clique | | | |
| nciclos | Mean | Std. Deviation | N |
| 100 | 6.8917 | .3620 | 120 |
| 500 | 6.4167 | .5279 | 120 |
| Total | 6.6542 | .5105 | 240 |

| Levene's Test of Equality of Error Variances(a) | | | |
|---|-----|-----|------|
| Dependent Variable: Dif con clique | | | |
| F | df1 | df2 | Sig. |
| 94.083 | 1 | 238 | .000 |
| Tests the null hypothesis that the error variance of the dependent variable is equal across groups. | | | |
| a Design: Intercept+PARAM2 | | | |

| Tests of Between-Subjects Effects | | | | | |
|--|-------------------------|----|-------------|--------|------|
| Dependent Variable: Dif con clique | | | | | |
| Source | Type III Sum of Squares | df | Mean Square | F | Sig. |
| Corrected Model | 13.537(a) | 1 | 13.537 | 66.079 | .000 |

ANEXO

| | | | | | |
|--|-----------|-----|-----------|-----------|------|
| Intercept | 10626.704 | 1 | 10626.704 | 51871.248 | .000 |
| PARAM2 | 13.538 | 1 | 13.538 | 66.079 | .000 |
| Error | 48.758 | 238 | .205 | | |
| Total | 10689.000 | 240 | | | |
| Corrected Total | 62.296 | 239 | | | |
| a R Squared = .217 (Adjusted R Squared = .214) | | | | | |

| Notes | | |
|-----------------------|---------------------------------------|--------|
| Output Created | 10 Mar 06 23:39:46 | |
| Comments | | |
| Input | Filter | <none> |
| | Weight | <none> |
| | Split File | <none> |
| | N of Rows in Working Data File | 240 |

ANEXO

| | | |
|-------------------------------|--|---|
| Missing Value Handling | Definition of Missing | User-defined missing values are treated as missing. |
| | Cases Used | Statistics are based on all cases with valid data for all variables in the model. |
| Syntax | UNIANOVA dif BY nhormiga nciclos colocaci /METHOD = SSTYPE(3) /INTERCEPT = INCLUDE /PRINT = DESCRIPTIVE HOMOGENEITY /CRITERIA = ALPHA(.05) /DESIGN = nhormiga nciclos colocaci . | |
| Resources | Elapsed Time | 0:00:00.06 |

| Descriptive Statistics | | | | | |
|-------------------------------|---------|--------------|--------|----------------|----|
| Dependent Variable: DIF | | | | | |
| NHORMIGA | NCICLOS | COLOCACI | Mean | Std. Deviation | N |
| 1 | 1 | 1 | 7.2000 | .6959 | 20 |
| | | 2 | 7.4500 | .8256 | 20 |
| | | 3 | 7.5000 | .6070 | 20 |
| | | Total | 7.3833 | .7152 | 60 |
| | 2 | 1 | 6.3500 | .8127 | 20 |

ANEXO

| | | | | | | |
|---|--------------|--------------|--------|--------|-------|----|
| | | 2 | 6.5000 | .6070 | 20 | |
| | | 3 | 6.4000 | .9403 | 20 | |
| | | Total | 6.4167 | .7874 | 60 | |
| | Total | 1 | 6.7750 | .8619 | 40 | |
| | | 2 | 6.9750 | .8619 | 40 | |
| | | 3 | 6.9500 | .9594 | 40 | |
| | | Total | 6.9000 | .8925 | 120 | |
| 2 | 1 | 1 | 7.1000 | .7182 | 20 | |
| | | 2 | 7.0500 | .8256 | 20 | |
| | | 3 | 7.1000 | .8522 | 20 | |
| | | Total | 7.0833 | .7874 | 60 | |
| | 2 | 1 | 6.1500 | .5871 | 20 | |
| | | 2 | 6.3000 | .5712 | 20 | |
| | | 3 | 6.1500 | .8127 | 20 | |
| | | Total | 6.2000 | .6587 | 60 | |
| | | Total | 1 | 6.6250 | .8066 | 40 |

ANEXO



| | | | | | | |
|--------------|--------------|--------------|--------------|--------|-------|-----|
| | | 2 | 6.6750 | .7970 | 40 | |
| | | 3 | 6.6250 | .9524 | 40 | |
| | | Total | 6.6417 | .8481 | 120 | |
| Total | 1 | 1 | 7.1500 | .6998 | 40 | |
| | | 2 | 7.2500 | .8397 | 40 | |
| | | 3 | 7.3000 | .7579 | 40 | |
| | | Total | 7.2333 | .7640 | 120 | |
| | 2 | 1 | 6.2500 | .7071 | 40 | |
| | | 2 | 6.4000 | .5905 | 40 | |
| | | 3 | 6.2750 | .8767 | 40 | |
| | | Total | 6.3083 | .7310 | 120 | |
| | Total | Total | 1 | 6.7000 | .8329 | 80 |
| | | | 2 | 6.8250 | .8385 | 80 |
| | | | 3 | 6.7875 | .9638 | 80 |
| | | | Total | 6.7708 | .8784 | 240 |

ANEXO

| Levene's Test of Equality of Error Variances(a) | | | |
|---|------------|------------|-------------|
| Dependent Variable: DIF | | | |
| F | df1 | df2 | Sig. |
| 1.006 | 11 | 228 | .441 |
| Tests the null hypothesis that the error variance of the dependent variable is equal across groups. | | | |
| a Design: Intercept+NHORMIGA+NCICLOS+COLOCACI | | | |

| Tests of Between-Subjects Effects | | | | | |
|--|--------------------------------|-----------|--------------------|-----------|-------------|
| Dependent Variable: DIF | | | | | |
| Source | Type III Sum of Squares | df | Mean Square | F | Sig. |
| Corrected Model | 56.000(a) | 4 | 14.000 | 25.624 | .000 |
| Intercept | 11002.604 | 1 | 11002.604 | 20137.818 | .000 |
| NHORMIGA | 4.004 | 1 | 4.004 | 7.329 | .007 |
| NCICLOS | 51.337 | 1 | 51.337 | 93.962 | .000 |
| COLOCACI | .658 | 2 | .329 | .602 | .548 |
| Error | 128.396 | 235 | .546 | | |
| Total | 11187.000 | 240 | | | |

ANEXO

| | | | | | |
|--|---------|-----|--|--|--|
| Corrected Total | 184.396 | 239 | | | |
| a R Squared = .304 (Adjusted R Squared = .292) | | | | | |

| Notes | | |
|-------------------------------|--|---|
| Output Created | 10 Mar 06 23:46:37 | |
| Comments | | |
| Input | Filter | <none> |
| | Weight | <none> |
| | Split File | <none> |
| | N of Rows in Working Data File | 240 |
| Missing Value Handling | Definition of Missing | User-defined missing values are treated as missing. |
| | Cases Used | Statistics are based on all cases with valid data for all variables in the model. |
| Syntax | UNIANOVA difcliq BY nhormig nciolos colocaci /METHOD = SSTYPE(3) | |

ANEXO

| | | |
|------------------|---------------------|--|
| | | /INTERCEPT = INCLUDE /PRINT = DESCRIPTIVE HOMOGENEITY /CRITERIA = ALPHA(.05) /DESIGN = nhormig nciclos colocaci . |
| Resources | Elapsed Time | 0:00:00.06 |

Descriptive Statistics
Dependent Variable: DIFCLIQ

| NHORMIG | NCICLOS | COLOCACI | Mean | Std. Deviation | N | |
|--------------|--------------|--------------|--------|----------------|-------|----|
| 1 | 1 | 1 | 1.0000 | .0000 | 20 | |
| | | 2 | 1.1500 | .4894 | 20 | |
| | | 3 | 1.0500 | .2236 | 20 | |
| | | Total | 1.0667 | .3117 | 60 | |
| | 2 | 2 | 1 | 1.0000 | .0000 | 20 |
| | | | 2 | .9000 | .3078 | 20 |
| | | | 3 | .9000 | .3078 | 20 |
| Total | | | .9333 | .2515 | 60 | |
| | Total | 1 | 1.0000 | .0000 | 40 | |

ANEXO

| | | | | | |
|---|--------------|--------------|--------|--------|-------|
| | | 2 | 1.0250 | .4229 | 40 |
| | | 3 | .9750 | .2762 | 40 |
| | | Total | 1.0000 | .2899 | 120 |
| 2 | 1 | 1 | 1.0000 | .0000 | 20 |
| | | 2 | 1.0000 | .0000 | 20 |
| | | 3 | 1.0000 | .0000 | 20 |
| | | Total | 1.0000 | .0000 | 60 |
| | 2 | 1 | .8000 | .4104 | 20 |
| | | 2 | .7500 | .4443 | 20 |
| | | 3 | .7500 | .4443 | 20 |
| | | Total | .7667 | .4265 | 60 |
| | Total | 1 | .9000 | .3038 | 40 |
| | | 2 | .8750 | .3349 | 40 |
| | | 3 | .8750 | .3349 | 40 |
| | | Total | .8833 | .3224 | 120 |
| | Total | 1 | 1 | 1.0000 | .0000 |

ANEXO

| | | | | | |
|--|--------------|--------------|--------|-------|-----|
| | | 2 | 1.0750 | .3499 | 40 |
| | | 3 | 1.0250 | .1581 | 40 |
| | | Total | 1.0333 | .2220 | 120 |
| | 2 | 1 | .9000 | .3038 | 40 |
| | | 2 | .8250 | .3848 | 40 |
| | | 3 | .8250 | .3848 | 40 |
| | | Total | .8500 | .3586 | 120 |
| | Total | 1 | .9500 | .2193 | 80 |
| | | 2 | .9500 | .3865 | 80 |
| | | 3 | .9250 | .3091 | 80 |
| | | Total | .9417 | .3115 | 240 |

| Levene's Test of Equality of Error Variances(a) | | | | |
|--|------------|------------|-------------|--|
| Dependent Variable: DIFCLIQ | | | | |
| F | df1 | df2 | Sig. | |
| 8.438 | 11 | 228 | .000 | |

ANEXO

Tests the null hypothesis that the error variance of the dependent variable is equal across groups.

a Design: Intercept+NHORMIG+NCICLOS+COLOCACI

| Tests of Between-Subjects Effects | | | | | |
|-----------------------------------|-------------------------|-----|-------------|----------|------|
| Dependent Variable: DIFCLIQ | | | | | |
| Source | Type III Sum of Squares | df | Mean Square | F | Sig. |
| Corrected Model | 2.867(a) | 4 | .717 | 8.290 | .000 |
| Intercept | 212.817 | 1 | 212.817 | 2461.620 | .000 |
| NHORMIG | .817 | 1 | .817 | 9.446 | .002 |
| NCICLOS | 2.017 | 1 | 2.017 | 23.326 | .000 |
| COLOCACI | 3.333E-02 | 2 | 1.667E-02 | .193 | .825 |
| Error | 20.317 | 235 | 8.645E-02 | | |
| Total | 236.000 | 240 | | | |
| Corrected Total | 23.183 | 239 | | | |

a R Squared = .124 (Adjusted R Squared = .109)

GLOSARIO

Glosario.

Algoritmos Bio-inspirados.-Son algoritmos inspirados en distintos comportamientos encontrados en la naturaleza, diferentes a la evolución de las especies.

Algoritmos de Colonias de Hormigas.- Son Algoritmos que emplean estrategias inspiradas en el comportamiento de las colonias de hormigas naturales, como es descubrir fuentes de alimentación, al establecer el camino más corto entre el hormiguero y las fuentes de alimento a través del uso de la feromona.

Algoritmos Exactos.- Garantizan encontrar el óptimo global de cualquier problema, pero tienen el grave inconveniente de que en ciertos problemas su tiempo de ejecución y/o los requisitos de memoria crecen de forma exponencial con el tamaño del problema.

Benchmark.- Es una instancia de un problema que se considera un reto o desafío de un problema en específico.

Computación Evolutiva.- Área que se encarga del desarrollo de algoritmos llamados “algoritmos evolutivos”, que emulan procesos evolutivos basados en la “supervivencia”, que es el principio del más apto en selección natural.

Feromona.- Es una sustancia química que utilizan las hormigas para comunicarse entre sí.

GLOSARIO

Inteligencia Artificial.- Área de ciencias de la computación que se encarga del estudio y simulación de la inteligencia.

Meta-heurística.-Conjunto de conceptos que pueden ser usados para definir métodos heurísticos que pueden ser aplicados en un conjunto amplio de problemas.

Minería de Datos.- Es un proceso de descubrimiento del conocimiento oculto en grandes y complejos conjuntos de datos (Bases de Datos).

BIBLIOGRAFÍA

Aarts E. and Lenstra J.K.(1997). **Local Search in Combinatorial Optimization**. J. Wiley & Sons, Chichester, UK.

Abraham A. (2003), ***i-Miner: A Web Usage Mining Framework Using Hierarchical Intelligent Systems***, The IEEE International Conference on Fuzzy Systems, FUZZ-IEEE'03, pp. 1129-1134.

Abraham A y Ramos Victor (2003). **Web Usage Mining Using Artificial Ant Colony Clustering and Linear Genetic Programming**

Abu-Khzaml F. N. , Baldwin N.E., Langston M. A. y Samatova N. F. (2005) **On the Relative Efficiency of Maximal Clique Enumeration Algorithms, with Application to High-Throughput Computational Biology**. SIAM J. Comput. Vol. 2: 1-6.

Akkoyunlu E.A. (1973) **The enumeration of maximal cliques of large graphs**. SIAM J. Comput. Vol. 2: 1-6.

Alaya I., Solnon C. and Ghedira K.(2004) **ANT ALGORITHM FOR THE MULTIDIMENSIONAL KNAPSACK PROBLEM**, Dans Proceedings of international conference on Bioinspired Methods and their Applications (BIOMA 2004), Slovenia. pp. 63-72.

BIBLIOGRAFÍA

Aljanaby A., Ku-Mahamud K. R. y Norwawi N. M.(2010). **Interacted Multiple Ant Colonies Optimization Approach to Enhance the Performance of Ant Colony Optimization Algorithms**, Computer and Information Science, vol. 3, No. 1, ISSN 1913-8989 (Print) ISSN 1913-8997 (Online).

Ambler A.P., Barrow H.G., Brown C.M., Burstall R.M. and Popplestone R.J.(1973) **A versatile computer-controlled assembly system**, In Proc. 3rd Int. J. Conf. Artif. Intell. 298-307.

Amin A.T. and Hakimi S.L. (1972) **Upper bounds on the order of a clique of a graph**. SIAM J.Appl. Math. Vol. 22: 569-573.

Auguston J.G. and Minker J. (1970) **An analisis of some graph theoretical cluster technique**. J. ACM Vol. 17: 571-588.

Babel L. (1991) **Finding maximum cliques in arbitrary and in special graphs**. Computing. Vol. 46: 321-341.

Babel L. (1994) **A fast algorithm for the maximum weight clique problem**. Computing. Vol. 52: 31-38.

Babel L and Tinhofer G. (1990) **A branch and bound algorithm for the maximum clique problem**. ZOR-Methods and Models of Operations Research. Vol. 34: 207-217.

BIBLIOGRAFÍA

Bäck T., Fogel D. B., y Michalewicz Z.(1997). **Handbook of Evolutionary Computation.** Institute of Physics Publishing and Oxford University Press.

Balas E. and Niehaus W. (1996). **Finding large cliques in arbitrary graphs by bipartite matching.**

Balas E. and Samuelsson H. (1977) **A Node Covering Algorithm.** Naval Research Logistics Quartely. Vol. 24: 213-233.

Balas E. and Toth P.(1985). **Branch and bound methods.** In: E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys(Eds), The traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, J. Wiley & Sons, Chichester, UK: 361-403.

Balas E. and Yu C.S. (1986). **Finding a maximum clique in an arbitrary graph.** SIAM J. Comput. Vol. 14: 1054-1068.

Balas E. and Xue J. (1991,1992) **Minimum weighted coloring of triangulated graph, with application to maximum weight vertex packing and clique finding in arbitrary graphs.** SIAM J. Comput. Vol. 2: 209-221(1991).

BIBLIOGRAFÍA

Barán B., Chaparro E. y Cáceres N.(1998) **A-Teams en la Optimización del Caudal Turbinado de una Represa Hidroeléctrica.** IBERAMIA-98, Lisboa-Portugal.

Bednarek A.R. and Taulbee O.E. (1966). **On maximal chains.** *Roum. Math. Pres et Appl.*,Vol. 11: 23-25.

Bernstein, D. S. (1999). **Reusing old policies to accelerate learning on new MDPs** (Technical Report UM-CS-1999-026). Dept. of Computer Science, Univ. of Massachusetts, Amherst, MA.

Bertoni A., Campadelli P. and Grossi G.(1997) **A discrete neural algorithm for the maximum clique problem: Analysis and circuit implementation.** Presented at WAE'97: Int. Workshop on Algorithm Engineering, Venice, Italy.

Bloedorn, E.; Christiansen, D.; Hill, W.; Skorupka, C.; Talbot, L. & Tivel J. (2001). **Mining for Network Intrusion Detection: How to Get Started,** *MITRE Technical Report*, August 2001

Blum, C y Roli, A. (2003): **Metaheuristics in Combinatorial Optimization: Overview and Conceptual Coamparison.** *ACM Cumputing Surveys*, Vol. 35, No. 3 September 2003, pp. 268-308.

BIBLIOGRAFÍA

Bomze I.M., Pelillo M. and Giacomini R.(1997) **Evolutionary approach to the maximum clique problem: Empirical evidence on a large scale**, in **Developments in Global Optimization**. I.M. Bomze, T. Csendes, R. Horst, and P.M. Pardalos(eds.), Kluwer: 95-108, Dordrecht.

Bomze I.M., Pelillo M. and Stix V. (1998). **Approximating the Maximum Weight Clique: An Evolutionary Game Theory Approach**. Manuscript in preparation.

Bonabeau, E., Theraulaz G., Deneuborg J.L., Arona S. y Camazine S. (1997). **Self-Organization in social insects**. *Tree* 12(5):188-193.

Bron C. and Kerbosch J.(1973) **Algorithm 457: Finding all cliques of an undirected graph**. *Commun.ACM*. Vol. 16: 575-577

Budinich M. and Matula (1976). **Bounds on the maximum clique of a graph**. *Disc. Appl. Math.* http://www.ts.infn.it/~mbh/MC_Bounds.ps.Z

Bui T.N. and Eppley P.H. (1995). **A hybrid genetic algorithm for the maximum clique problem**. *Proc. 6th Int. Conf. Genetic Algorithms*: 478-484.

Bui, T. N., T. H. Nguyen, and J. R. Rizzo Jr. (2009). **Parallel Shared Memory Strategies for Ant-Based Optimization Algorithms**. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'09)*, ACM Press, Seattle, pp. 1-8.

BIBLIOGRAFÍA

- Bui T.N. y Rizzo J.R.(2004) **Finding maximum cliques with distributed ants**. In K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E.K. Burke, P.J. Darwen, D. Dasgupta, D. Floreano, J.A. Foster, M. Harman, O. Holland, P.L. Lanzi, L. Spector, A. Tettamanzi, D. Thierens, and A.M. Tyrell, editors, Genetic and Evolutionary Computation - GECCO 2004, volume 3102 of *lnes*, pages 24{35. Springer-Verlag, 2004.
- Bullnheimer B., Hartl R. y Strauss C. (1999a) **“An improved ant system algorithm for the vehicle routing problem”**. *Annals of Operations Research* (Dawind, Feichtinger and Hartl (eds.): Nonlinear Economic Dynamics and Control.
- Bullnheimer, B., Hartl, R.F. y Strauss, C. (1999b). **Applying the Ant System to the vehicle routing problem**, in: S. Voß, S. Martello, I.H. Osman, C. Roucairol (Eds.), *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Kluwer Academic Publishers, Boston, MA.
- Cai, D.; Shao, Z.; He, X.; Yan, X. & Han J. (2005). **Mining Hidden Community in Heterogeneous Social Networks**, *Proceedings of LinkKDD'05*, Chicago, USA, August 2005, ACM.
- Carraghan R. and Pardalos P.M. (1990) **An exact algorithm for the maximum clique problem**. *Oper Res. Lett.*, Vol. 9: 375-382

BIBLIOGRAFÍA

Cerdá X., Barroso, A. Angulo E., Amor F. y Boulay R. (2009) **Nidos de verano y nidos de invierno en una hormiga nomada *Aphaenogaster senilis* Mayr, 1853.** Boletín de la Asociación Ibérica de Mirmecología, ISSN: 1989-7928, Iberomemymex. No 1

Carte B. and Park K. (1993). **How good are genetic algorithms at finding large cliques: An experimental study.** Technical Report BU-CS-93-015, Computer Science Dept., Boston University.

Cavique L., Rego C., Themido I. (2001) **A Scatter Search Algorithm for the Maximum Clique Problem.** in book “Essays and Surveys in Metaheuristics”, Kluwer Academic Publishers, 227-244.

Chen, M. y Han, J., (1996). ***Data mining: An overview from database perspective.*** IEEE Transactions on Knowledge and Data Eng.

Cherrett, J. (1976). **A Review of the status of Leaf – Cutting Ants and their Control.** Ann Appl. Biol. 84:124 –128.

Chvátal V. (1977) **Determining the stability number of a graph.** SIAM J. Comput.. Vol. 6: 643-662.

Coello C. A. (1998) **An Updated Survey of GA-Based Multiobjective Optimization Techniques.** Technical Report Lania-RD-98-08, Laboratorio Nacional de Informática Avanzada (LANIA), Xalapa, Veracruz, México.

BIBLIOGRAFÍA

- Coello C. A. (1999) **Constraint handling through a multiobjective optimization technique.** En A. S. Wu, editor, Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program, págs. 117–118, Orlando, Florida.
- Coello C. A. y Mariano C. E.(2002) Evolutionary Algorithms and Multiple Objective Optimization. En M. Ehrgott y X. Gandibleux, editors, Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys, págs. 277–331. Kluwer Academic Publishers, Boston, 2002.
- Cordón O., de Viana I. F., Herrera F, & Morena LL. (2000) **A new ACO Model Integrating Evolutionary Computation Concepts: the Best-Worst Ant System.** *In Second International Workshop on Ant Algorithms.*
- Corradi K. and Szabo S. (1990) **A combinatorial approach for Keller's conjecture.** Periodica Mathematica Hungarica. Vol. 21: 95-100.
- Deb K. (1999) **Evolutionary algorithms for multi-criterion optimization in engineering design.** En K. Miettinen, M. M. Mäkelä, P. Neittaanmäki, y J. Periaux, editors, Evolutionary Algorithms in Engineering and Computer Science, págs. 135–161, Chichester, UK. JohnWiley & Sons, Ltd.
- Della Croce F. and Tadei R.(1994) **A multi-KP modeling for the maximum clique problem.** Europ. J. Oper. Res. Vol. 73: 555-561.

BIBLIOGRAFÍA

Desler J.F. and Hakimi S.L. (1970) **On finding a maximum internally stable set of a graph.** Proc. of Fourth Annual Princeton Conference on Information Sciences and Systems. Vol. 4: 459-462. Princeton, NJ.

Digney, B. (1996). **Emergent hierarchical structures: Learning reactive/hierarchical relationships in reinforcement environments.** *From animals to animats 4: SAB 96.* MIT Press/Bradford Books.

Digney, B. (1998). **Learning hierarchical control structure for multiple tasks and changing environments.** *From animals to animats 5: SAB 98.*

DIMACS Center for Discrete Mathematics and Theoretical Computer Science
[http://dimacs.rutgers.edu/pub/challenge/graph/"benchmark"s/](http://dimacs.rutgers.edu/pub/challenge/graph/)

Dokas, P.; Ertöz, L.; Kumar, V.; Lazarevic, A.; Srivastava, J. & Tan, P.-N. (2002). **Data mining for network intrusion detection,** *Proceedings of NSF Workshop on Next Generation Data Mining,* pp. 21-30, November 2002

Dorigo, M. (1992) **Optimization, learning and natural algorithms,** Ph.D. Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, (in Italian).

BIBLIOGRAFÍA

Dorigo, M., Bonabeau, E. y Theraulaz, G.(2000) **Ant algorithms and stigmergy**, Future Generation Computer Systems 16, pp. 851-871

Dorigo M. y Gambardella L., (1997a) **Ant colonies for the traveling salesman problem**, in BioSystems, 1997.

Dorigo M. y Gambardella L., (1997b) **“Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”**. *IEEE Trans. on Evolut. Comp.*, Vol 1, No. 1, pp. 53-66.

Dorigo, M., Maniezzo, V. y Colorni, A. (1991) **Ant System: An Autocatalytic Optimizing Process**. *Technical report 91-016 revised, Dipartimento de Electronia, Politecnico di Milano, Milan.*

Dorigo, M.; Maniezzo, V. y Colorni, A. (1996): **Ant System: Optimization by a Colony of Cooperating Agents**. *IEEE Transactions on Systems, Man And Cybernetics – Part B: Cybernetics*, 26:1, 29-41.

Dorigo, M. y Stützle, T.(2004) **Ant Colony Optimization**. *A Bradford Book, The MIT Press Cambridge, Massachusetts, London England.*

Ebenegger C., Hammer P.L. and de Werra D. (1984) **Pseudo-boolean functions and stability af graphs**. *Ann. Discr Math.* Vol. 19: 83-98.

BIBLIOGRAFÍA

Elliman D. G. and Youssef S.M. (2004) **Reactive Prohibition-based Ant Colony Optimisation (RPACO): A New Parallel Architecture for Constrained Clique Sub-Graphs**, Computer Science Technical Report No. NOTTCS-TR-2004-7 in University of Nottingham.

Fenet, S. y Solnon, C.(2003) **Searching for Maximum Cliques with Ant Colony Optimization**. EvoWorkshops 2003, LNCS 2611, 236–245.

Fernández F. (2003). **Introducción a las hormigas de la región Neotropical**. Instituto de Investigación de Recursos Biológicos Alexander von Humboldt, Bogotá, Colombia.

Fidanova S. (1999) **Evolutionary Algorithm for Multiple Knapsack Problem**. PPSNVII-Workshop 2002.

Fidanova S. (2002) **ACO Algorithm for MKP Using Different Heuristic Information** 5th Int Conference of Numerical Methods and Applications, Lecture Notes in Computer Science No 2542, Springer, Germany , 2002, 434-440.

Fidanova S. (2004) **Ant Colony Optimization for Multiple Knapsack Problem and Model Bias**, *NAA'04, lecture Notes in Computer Sciences, Springer, Germany*, 282-289

BIBLIOGRAFÍA

Fonseca C. M. y Fleming P. J.(1993) **Genetic Algorithms for Multiobjective Optimization: Fórmulaion, Discussion and Generalization**. En S. Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms, págs. 416–423, San Mateo, California. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

Foster J.A. and Soule T. (1995) **Using genetic algorithms to find maximum cliques**, *Technical Report LAL 95-12, Dept. of Computer Science, U. Idaho.*

Funabiki N., Takefuji Y. and Lee K.C. (1992) **A neural network model for finding a near-maximum clique**. J. PParallel Distrib. Comput. Vol. 14: 340-344-445

Franks, N., Pratt, S., Mallon, E., Britton, N. & Sumpter, D. (2002). **Information flow, opinion polling and collective intelligence in house-hunting social insects**. Philosophical Transactions of the Royal Society of London, Series B, 357, 1567–1583.

Friden C., Hertz A. and de Werra M. (1990) **TABARIS: An exact algorithm based on tabú search for finding a maximum independent set in a graph**. Comput. Oper. Res., Vol. 17: 437-445.

Gambardella, L.M. y Dorigo, M. (1995) **Ant-Q: A Reinforcement Learning approach to the traveling salesman problem**. Twelfth International Conference on Machine Learning, A. Prieditis and S. Russell (Eds.), Morgan Kaufmann, pp. 252-260.

BIBLIOGRAFÍA

Gambardella, L.M. y Dorigo, M. (1996) **Solving symmetric and asymmetric TSPs by ant colonies**, in: Proceedings of the IEEE Conference on Evolutionary Computation, ICEC'96, IEEE Press, New York, pp. 622–627.

Garey M. and Johnson D. (1979) **Computers and Intractability- A guide to the Theory of NP-Completeness**. Freeman, San Francisco.

Garey , M. R. y Johnson, D. S. (1975): **Computers and Intractability**. *Library of Congress Cataloging in Publication Data*.

Garofalakis, M.; Rastogi, R.; Seshadri, S. & Shim, K. (1999). **Data Mining and the Web: Past, Present and Future**, *Proceedings of 2nd ACM International Workshop on Web Information and Data Management (WIDM)*, pp. 43-47, Missouri, USA, November 1999, ACM, Kansas City

Gendreau M., Picard J.C. and Zubieta L. (1988): **An efficient implicit enumeration algorithm for the maximum clique problem**. A. Kurzhanski et al. (eds) Lecture Notes in Economics and Mathematical Systems, Vol. 304: 70-91

Gendreau A., Salvail L. and Soriano P. (1993). **Solving the maximum clique problem using a tabú search approach**. Ann. Oper.Res., Vol. 41: 385-403.

BIBLIOGRAFÍA

Gerhads L. and Lindenberg W..(1979): **Clique detection for nondirected graphs: Two new algorithms.** *Computing*, Vol. 21: 295-322

Glover, F. (1998): **A template for scatter search and path relinking.** En J.-K. Hao y E. Lutton (eds) *Artificial Evolution*, Vol. 1363 de *Lecture Notes in Computer Science*, 13-54, Springer- Verlag.

Glover F. (1990): **Tabú searchPart II.** *ORSA J. Comput*, Vol. 2: 4-32

Glover F. (1989): **Tabú searchPart I.** *ORSA J. Comput*, Vol. 1: 190-260

Glover, F. y Kochenberger, G. (eds.) (2003): **Handbook of Metaheuristics**, *Kluwer Academic Publishers*.

Goldberg, D. E.(1989). **Genetic Algorithms in Search, Optimization and Machine Learning.** Addison- Wesley Publishing Company, Reading, Massachusetts.

Goldberg D.E. and Rivenburgh R.D.(1996) **Constructing cliques using restricted backtracking.** In (D.Johnson and M. Trick, eds.), *DIMACS Series in Discrete Mathematics and Theoretical Computer Science: Cliques, Coloring, and Satisfiability*, 75–88.

BIBLIOGRAFÍA

Grossman T. (1996): **Applying the INN model to de max clique problem.** In (D.Johnson and M. Trick, eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science: Cliques, Coloring, and Satisfiability, 75–88.

Hamilton, W.D. (1964). **The genetical evolution of social behavior.** Journal of Theoretical Biology 7:1-52.

Han, J., Kamber, M. (2001). **Data mining: Concepts and techniques.** Morgan Kauffmann Publishers.

Handl J., J. Knowles and M. Dorigo (2003a). **Ant-based clustering: a comparative study of its relative performance with respect to k-means, average link and ld-som.** Technical Report TR/IRIDIA/2003-24. IRIDIA, Universite Libre de Bruxelles, Belgium.

Handl J., J. Knowles and M. Dorigo (2003b). **On the performance of ant-based clustering.** In Design and Application of Hybrid Intelligent Systems, Vol. 104 of Frontiers in Artificial Intelligence and Applications (pp. 204-213). Amsterdam, The Netherlands: IOS Press.

Harary F. and Ross I.C.(1957). **A procedure for clique detection using the group matrix.** Sociometry, Vol. 20: 205-215

BIBLIOGRAFÍA

Hernández, J.; Ochoa, A.; Muñoz, J. & Burlak, G. (2006). **Detecting cheats in online student assessments using Data Mining**, *Proceedings of The 2006 International Conference on Data Mining (DMIN'2006)*, pp. 204-210, Las Vegas, USA, June 2006, Nevada City

Holland J.H. (1975): **Adaptation in Natural and Artificial Systems**. University of Michigan Press, Ann Arbor, MI.

Hopfield J.J and Tank D.W. (1985). **Neural" computation of decisions in optimization problems**. *Biol. Cybern.*, Vol. 52: 141-152

Horn R.A. and Johnson C.R. (1985). **Matrix Analysis**. Cambridge University Press. Cambridge, UK.

Houck D.J.(1974) **On the vertex packing problem**. Ph.D. dissertation, The Johns Hopkins University, Baltimore.

Houck D.J. and Vemuganti R.R. (1977) **An algorithm for the vertex packing problem**. *Oper. Res.*, Vol. 25: 773-787 .

Jagota A., Sanchis L. and Ganesan R. (1996): **Approximately solving maximum clique using neural networks and related heuristics**. In Second DIMACS

BIBLIOGRAFÍA

Implementation Challenge, D.S. Johnson and M.A. Trick (eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, v. 26, p. 169—203.

Jerrum M. (1992): **Large cliques elude the Metropolis process.** Random Structures and Algorithms., Vol. 3: 347-359.

Johnson H.C.(1976): **Cliques of a graph: Variations on the Bron-Kerboch algorithm.** Int. J. Comput. Inform. Sci., Vol. 5: 209-238.

Johnson L.F.(1975): **Determining cliques of a graph.** Proc.5th Manitoba Conf. on Numer. Math., 429-437.

Johnson D.S, Yannakakis M. and Papadimitriou C.H. (1988): **On generating all maximal independent sets.** Inform. Proc. Lett., Vol. 27: 119-123

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). **“Reinforcement Learning: A Survey”**, *Journal of Artificial Intelligence Research*, Volume 4.

Karp R.M. (1972): **Reducibility among combinatorial problems. In complexity of Computer Computations.** R.E. Miller and J.W. Thatcher(eds), Plenum Press, New York: 85-103.

Katayama K., Hamamoto A. y Narihisa H.(2004) **An Effective Local Search for the Maximum Clique Problem.** *Information Processing Letters* **95** (5): 503–511

BIBLIOGRAFÍA

Kennedy, Y.S. J. y Eberhart, R.(2001): **Swarm Intelligence**, *Morgan Kaufmann*.

Kerningham B. W. and LIN S. (1970). **An Efficient Heuristic Procedure for Partitioning Graphs**. Bell Syst. Tech. J. 49 (1970), 291-307.

Kikusts P. (1986): **Another algorithm determining the independence number of a graph**. Elektron. Inf. Verarb. Kybern. EIK 22::157-166.

Kirkpatrick S., Gelatt C.D. and Vecchi M.P. (1983): **Optimization by simulated annealing**. Science, Vol. 220: 671-680.

Kopf R. and Ruhe G. (1987): **A computational study of the weighted independent set problem for general graphs**. Found. Control Engin., Vol. 12: 167-180.

Laguna, M.; Glover, F. y Martí, R. (2003) **Scatter search and path relinking: Advances and applications**. Cap I en F. Glover y G. Kochenberger (eds) *Handbook on MetaHeuristics*.

Laguna, M. y Martí, R.(2002): **Scatter Search Methodology and Implementations in C**. *Kluwer Academia Publisher*.

BIBLIOGRAFÍA

Larrañaga, P. Etxberria, R. Lozano, J.A. y Muhlenbein, H. (2003): **Estimation of Distribution Algorithms Applied to Combinatorial Optimization Problems**, *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, No. 19, páginas 149-168.

Larrañaga, P. y Lozano, J.A. (2002): **Estimation of Distribution algorithms: A new tool for Evolutionary Computation**. *Kluwer Academic Publishers*.

Leguizamón, G. y Michalewicz, Z.(1999) **A New Version of Ant System for Subset Problems**, Congress on Evolutionary Computation pp1459-1464.

Leifman L.J.(1976). **On construction of all maximal complete subgraphs (cliques) of a graph**. **Technical Report**. Dept. of Mathematics. University of Haifa, Israel.

Lin F. and Lee K. (1992). **A parallel computation network for the maximum clique problem**, In Proc. 1st Int. Conf. Fuzzy Theory Tech., Baton Rouge, Louisiana.

Loukakis E. .(1983). **A new backtracking algorithm for generating the family of maximal independent sets of a graph**. *Comp. Math. Appl.*, Vol. 9: 583-589.

Loukakis E. and Tsouros C. (1983): **An algorithm for the maximum internally stable set in a weighted graph**. *Int. J. Comp. Math.*, Vol. 13: 117-129.

BIBLIOGRAFÍA

Loukakis E. and Tsouros C. (1982). **Determining the number of internal stability of a graph.** *Int. J. Comp. Math.*, Vol. 11: 207-220.

Lumer E. D. and Faieta B.(1994). **Diversity and adaptation in populations of clustering ants.** In *Proceedings of the 3th International Conference on Simulation of Adaptive Behavior: From Animals to Animals 3*, pages 501–508. MIT Press.

MacKay, W. P. (1991). **The role of ants and termites in desert communities.** pp. 113-150. In: G.A. Polis (ed). *The ecology of desert communities*. The University of Arizona Press, Tucson.

Mallon, E., Pratt, S. & Franks, N. (2001). **Individual and collective decision-making during nest site selection by the ant *Leptothorax albipennis*.** *Behavioral Ecology and Sociobiology*, 50, 352–359.

Mannino C. and Sassano A.(1996) **Edge projection and the maximum cardinality stable set problem.** *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 26, 205-219

Marchiori E. (1998) **A simple heuristic based genetic algorithm for the maximum clique problem.** *Proc. ACM Symp. Appl. Comput.* 366-373.

BIBLIOGRAFÍA

Marcus P.M.(1964). **Derivation of maximal compatibles using boolean algebra.** IBM J. Res. Develop., Vol. 8: 537-538.

Martí, R. y Laguna, M. (2003): **Scatter Search: Diseño básico y estrategias avanzadas.** *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, No. 19.

Matula D.W.(1976). **The largest clique size in a random graph.** Technical Report CS 760, Department of Computer Science, Southern Methodist University.

McGovern, A., and Barto, A. G. (2001). **Automatic Discovery of Subgoals in Reinforcement learning using Diverse Density.** Proceedings of the 18th International Conference on Machine Learning, pages 361-368.

Meeusen W. and Cuyvers L. (1975): **Clique detection in directed graphs: An new Algorithm.** J. Comput Appl. Math., Vol. 1: 185-193.

Mezura, E. (2004). **Alternative Techniques to Handle Constraints in Evolutionary Optimization.** Tesis de Doctorado. IPN, México.

Mezura, E. (2008). **Cómputo Bio-Inspirado, una mirada a lo natural.** Diario La Jornada de Oriente.

BIBLIOGRAFÍA

Miettien K.(2001)**Some methods for nonlinear multi-objctive optimization.** En E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, y D. Corne, editors, First International Conference on Evolutionary Multi-Criterion Optimization. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

Miramontes O. (2000) **Orden y Caos en la Organización Social de las Hormigas,** Ciencias, No. 059, julio

Monmarché N., M. Slimane y G. Venturini (1997) **AntClass: discovery of clusters in numeric data by an hybridization of an ant colony with the kmeans algorithm.** Technical Report No.213, E3I, Laboratoire d'Informatique, University of Tours

Monmarché N.(1999) **On data clustering with artificial ants.** In Freitas, A., editor, AAAI-99 & GECCO-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions, pages 23-26.

Montes de Oca M., Garrido L and Aguirre J. L. (2005a) **An hybridization of an antbased clustering algorithm with growing neural gas networks for classification tasks.** SAC, pp. 9-13.

Montes de Oca M., Garrido L. and Aguirre J. L. (2005b) **Effects of Inter-agent Communication in Ant-Based Clustering Algorithms: A case Study on Communication Policies in Swarm Systems.** MICAI, pp. 254-263

BIBLIOGRAFÍA

Montgomery, J., & Randall, M. (2002). **The Accumulated Experience Ant Colony for the Travelling Salesman Problem.** *International Journal of Computational Intelligence and Applications*, 189–198.

Murthy A.S., Parthasarathy G. and Sastry V.U.K. (1994) **Clique finding-A genetic approach.** Proc. 1st IEEE Conf. Evolutionary Comput.: 18-21.

Nemhauser G.L. and Sigismondi G. (1992): **A strong cutting plane/branch and bound algorithm for node packing.** *J. Opl. Res. Soc.* Vol. 43: 443-457.

Nemhauser G.L. and Trotter L.E.(1975): **Vertex pecking: Structural properties and algorithms.** *Math. Programming.* Vol. 8: 232-248.

Nemhauser G.L. and Wolsey L.A.(1988): **Integer and Combinatorial Optimization.** *J.Wiley & Sons, New York.*

Osteen R. E. (1974): **Clique detection algorithms based on line addition and line removal.** *SIAM J. Appl.Math.,* Vol. 26: 126-135.

Osteen R. E. and Tou J.T. (1973): **A clique-detection algorithm based on neighborhoods in graphs.** *Int. J. Comput. Inform. Sci.,* Vol. 2: 257-268.

BIBLIOGRAFÍA

Ouyang, Q., Kaplan P.D., Liu S. y Libchaber A. (1997) **DNA Solution of the Maximal Clique Problem.** *Journal Science*, pp. 446–449.

Pardalos P.M. and Desai N. (1991): **An algorithm for finding a maximum weighted independent set in an arbitrary graph.** *Int. J. Comput. Math.*, Vol. 38: 163-175.

Pardalos P.M. and Phillips A.T. (1990): **A global optimization approach for solving the maximum clique problem .** *Int. J. Comput. Math.*, Vol. 33: 209-216.

Pardalos P.M. and Rodgers G.P. (1992): **A branch and bound algorithm for the maximum clique problem.** *Comput. Oper. Res.*, Vol. 19: 363-375

Parpinelli R. S., Lopes H. S., and Freitas A. A. (2001). **Data Mining with an Ant Colony Optimization Algorithm.** *IEEE Transaction on Evolutionary Computation*, vol.6, no.4, pp.321-332.

Paull M.C. and Unger S.H. (1959): **Minimizing the number of states in incompletely specified sequential switching funtions.** *IRE Trans. Electr. Comput.*, Vol. EC-8: 356-367.

Peters M. and Zaki M. J. (2004), **CLICK: Clustering Categorical Data using K-partite Maximal Cliques.** TR 04-11, CS Dept., RPI.

BIBLIOGRAFÍA

Pelillo M.(1995) **Relaxation labeling networks for the maximum clique problem.** *J. Artif. Neural Networks.*, Vol. 2: 313-328.

Pelillo M. and Torsello A. (2006) **Payoff-Monotonic Game Dynamics and the Maximum Clique Problem.** *Neural Computation*, Volume 18, Issue 5, p.1215-12

Pettinger J. E. and Everson R. M. (2003) **Controlling Genetic Algorithms with Reinforcement Learning.** University of Exeter

Pineda H. (2008). **Teoría, Algoritmos y Aplicaciones en Computación.** Benemérita Universidad Autónoma de Puebla. ISBN: 978-607-7541-89-9

Polis, G. A. (1991). **Desert communities: an overview of patterns and processes.** Pp. 1-26. *In: G.A. Polis (ed). The ecology of desert communities.* The University of Arizona Press, Tucson.

Ponce J., Hernandez A., Ochoa, A., Padilla F., Padilla A., Alvarez F. y Ponce Eunice. (2009). **Data Mining and Knowledge Discovery in Real Life Applications, Book, Intechweb,** ISBN 978-3-902613-53-0, pp. 265-290

Ponce, J.; Ochoa, A.; Pietsch, W. & Zolezzi-Hatsukimi, Z. (2007). **Ahankara: Identify Bipolar Síndrome in User of Orkut with Data Mining, Proceedings of ENC 2007,** Michoacan, Mexico, September 2007, IEEE, Morelia City

BIBLIOGRAFÍA

Ponce J., Ponce de León E., Padilla F. y Padilla A. (2006a) **Implementación De Un Algoritmo De Colonia De Hormigas Para El Problema Del Clique Máximo.** *En CISCI'2006 volume 1 pages70-73, Orlando Florida.*

Ponce J., Ponce de León E., Padilla F. y Padilla A. (2006b) **Algoritmos De Colonia De Hormigas Aplicados Al Problema De La Mochila.** *En “Segundo Congreso Estatal La Investigación en el Posgrado”, Aguascalientes*

Ponce J., Ponce de León E., Padilla F. y Padilla A. (2006c) **Algoritmo De Colonia De Hormigas Aplicado Al Problema Del Clique Máximo.** *En “Segundo Congreso Estatal La Investigación en el Posgrado”, Aguascalientes.*

Ponce J., Ponce de León E., Padilla F. Padilla y A. Zezatti A.(2006) **Algoritmo de Colonia de Hormigas para el Problema del Clique Máximo con un Optimizador Local k-opt.** *En 11° Simposio de Informática y 6° Mostra de Software Academico SIMS 2006, Uruguaiiana Brasil. Hífen, Uruguaiiana, v. 30, n. 58, 2006.*

Pratt S., Sumpter D., Mallon E. & Franks N.(2005). **An agent-based model of collective nest choice by the ant *Temnothorax albipennis*.** *ANIMAL BEHAVIOUR*, 2005, 70, 1023–1036, ELSEVIER.

Preparata F.P., Metze G. and Chien R.T. (1967). **On the connection assignment problem of diagnosable systems,** *IEEE Trans. Electr. Comput.*, Vol. 16: 848-854.

BIBLIOGRAFÍA

Ramos V., Muge F., Pina P. (2002), **Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies**, Soft Computing Systems - Design, Management and Applications, 2nd Int. Conf. on Hybrid Intelligent Systems, IOS Press, pp. 500-509.

Ramos V. y Merelo J. J. (2002). **Self-Organized Stigmergic Document Maps: Environment as a Mechanism for Context Learning**, in E. Alba, F. Herrera, J.J. Merelo et al. (Eds.), AEB'02 - 1st Int. Conf. On Metaheuristics, Evolutionary and Bio-Inspired Algorithms, pp. 284-293, Mérida, Spain.

Ramos V. y Almeida F. (2000). **Artificial Ant Colonies in Digital Image Habitats - A Mass Behaviour Effect Study on Pattern Recognition**, in Marco Dorigo, Martin Middendorf and Thomas Stützle (Eds.), Proc. of ANTS'00 – 2nd Int. Workshop on Ant Algorithms, pp. 113-116, Brussels, Belgium.

Rayward-Smith, V. J. (Editor), Osman, I. H. (Editor), Reeves, C. R. (Editor), Smith, G. D. (Editor) (1996) : **“Modern Heuristic Search Methods”**. *John Wiley & Sons*.

Ren, Z.-G., Feng Z.-R., Ke L.-J. and Zhang Z.-J. (2010). **New ideas for applying ant colony optimization to the set covering problem**. Computers & Industrial Engineering, Elsevier.

BIBLIOGRAFÍA

Ribeiro, C.C. y Hansen, P. (eds) (2001): **Essay and Surveys in Metaheuristics**. Kluwer Academic Publishers

Robinson J.B. (1949), “**On the Hamiltonian game (a traveling-salesman problem)**”. RAND Research Memorandum RM-303.

Robson J..M. (1986). **Algorithms for maximum independent sets**. *J. Algorithms*, Vol. 7: 425-440.

Sait, S.M. y Youssef, H. (1999) : **Iterative Computer Algorithms with Applications in Engineering : Solving Combinatorial Optimization Problems**, IEEE Computer Society, California.

Schwartz A. (1993) **A reinforcement learning method for maximizing undiscounted rewards**. Proceedings of the Tenth International Conference on Machine Learning, pags. 298- 305. Morgan Kauffman.

Sen, S. (1998). **Special issue on evolution and learning in multi-agent systems**. *International Journal of Human-Computer Studies* 48.

Sim K. M. and Sun W.H. (2002) “**Multiple ant-colony optimization for network routing**”, in *Proc. 1st Int. Symp. Cyberworld*, Tokyo, Japan, November 2002, pp. 277–281.

BIBLIOGRAFÍA

Sim K. M. and Sun W.H. (2003) "**Ant Colony Optimization for Routing and Load - Balancing: Survey and New Directions**", IEEE Transactions on Systems, Man, and Cybernetics—PART A: Systems and Humans, Vol. 33, No. 5, September 2003.

Solnon C. (2000) **Solving Permutation Constraint Satisfaction Problems with Artificial Ants**. *Proceedings of the 14th European Conference on Artificial Intelligence* (pp. 118-122). Amsterdam, IOS Press.

Solnon C. (2002) **Ants can solve Constraint Satisfaction Problems**. IEEE transactions on evolutionary computation, vol. 6, no. 4, august.

Solnon C. (2008). **Combining two pheromone structures for solving the car sequencing problem with Ant Colony Optimization**. European Journal of Operational Research 191, 1043–1055

Solnon C. and Bridge D. (2006) **An Ant Colony Optimization Meta-Heuristic for Subset Selection Problems**, in N.Nedjah and L.M.Mourelle (eds.), Systems Engineering using Swarm Particle Optimisation, pp.7-29, Nova Science Publishers.

Solnon, C. y Fenet, S.(2004) **Investigating ACO capabilities for solving the Maximum Clique Problem**. Rapport de recherche RR-LIRIS-2004-021 , Soumis à Journal of Heuristics.

BIBLIOGRAFÍA

Solnon C. y Fenet S. (2005) **A study of ACO capabilities for solving the Maximum Clique Problem** Journal of Heuristic, Springer.

Soriano P., and Gendreau M. (1996) **Diversification strategies in tabú search algorithms for the maximum clique problem**, *Ann. Oper. Res.* Vol. 63: 189-207

Stutzle, T. y Dorigo, M.(1999). **ACO Algorithms for the Traveling Salesman Problem**, in: P.Neittaanmaki, J. Periaux, K.Miettinen, M.M. Makela(Eds.), *Evolutionary Algorithms in Engineering and Computer Science*, Wiley, Chichester, UK,, pp. 163-183.

Stützle,T. y Hoos, H.(1997) **The MAX-MIN Ant System and local search for the traveling salesman problem**, in: T. Bäck, Z. Michalewicz, X. Yao (Eds.), *Proceedings of IEEE-ICEC-EPS'97*, IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference, IEEE Press, New York, pp. 309–314.

Sutton and Barto (1998), **Reinforcement Learning**, A Bradford Book, The MIT Press, Cambridge, Massachussets, London, England

Tarjan R.E.(1972). **Finding a maximum clique**. *Technical Report 72-123 Computer Sci. Dept., Cornell University, Ithaca, NY.*

BIBLIOGRAFÍA

Tarjan R.E and Trojanowski A.E. (1977) **Finding a maximum independent set.** *SIAM J. Comput.*, Vol. 6: 537-546

Thrun, S. B., & Schwartz, A. (1995). **Finding structure in reinforcement learning.** *NIPS* 7 (pp. 385-392). San Mateo, CA: Morgan Kaufmann.

Tomita E., Kohata Y. and Takahashi H.(1988): **A simple algorithm for finding a maximum clique.** *Technical Report UEC-TR-C5.*

Tsukiyama S., Ide M., Aviyoshi H. and Shirakawa I.(1977): **A new algorithm for generating all the maximum independent sets** *SIAM J.Comput.*, Vol. 6: 505-517

Tsutsui, S. (2008) **Convergence Analysis of the Cunning Ant System with Entropy Measure.** *Proceedings of the 2008 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'08)*, pp. 783-789, CSREA press , Monte Carlo Resort, Las Vegas, Nevada, USA (July 14-17, 2008), 2008.7. ISBN: 1-60132-082-5.

van Veldhuizen D. A. (1999) **Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations.** PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio.

BIBLIOGRAFÍA

Varan, S. (2006). **Crime Pattern Detection Using Data Mining**, Oracle Corporation

Wilf H.S. (1986): **Spectral bounds for the clique and independence numbers of graphs.**
J. Combin. Theory B. Vol. 40: 3113-117.

Xue J. (1991): **Fast Algorithms for Vertex Packing and Related Problems.** *Ph.D. Thesis, GSIA, Carnegie Mellon University.*

Xue J. (1994): **Edge-maximal triangulated subgraphs and heuristics for the maximum clique problem.** *Networks.* Vol. 24: 109-120

Wahlstrom K., & Roddick J. (2000). **On the Impact of Knowledge Discovery and Data Mining,** *Proceedings of Australian Institute of Computer Ethics Conference (AiCE2000), Canberra, Australia, April 2000, Sydney City*

Weber, N. (1972). **Gardening Ants: The Attines.** *Memories of the American Philosophical Society.* 92: 1 – 46.

Weiss, G. (1997). **Distributed Artificial Intelligence meets Machine Learning.** *Proceedings of The European Conference on Artificial Intelligence ECAI-96 Workshop.* Berlin: Springer.

BIBLIOGRAFÍA

Wiley J. and Sons (1995). **Intelligent Hybrid Systems**. Suran Goonatilake and Sukhdev Khebbal, editors, London.

Wilson, E.O. (1971) *The Insect Societies*. Cambridge, MA: Belknap Press of Harvard University Press.

Wilson, E.O. (2000). *Ants. Standard methods for measuring and monitoring biodiversity*. Foreword. Pp. XV-XVI. In: D. Agosti, J.D. Majer, L.E. Alonso and T.R. Schultz (eds). Smithsonian Institution Press. Washington.

Zitzler E., Deb K., Thiele L., Coello C. A., y Cornde D., editors **Equipo Probabilístico de Algoritmos Evolutivos Multiobjetivo en Computación Paralela**. Proceedings of the First International conference on EMOO, 2001, Berlin, Germany, Marzo 2001. Springer-Verlag.

Zhang Q., Sun J. y Tsang E.(2005) **An Evolutionary Algorithm With Guided Mutation for the Maximum Clique Problem**, *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 9, NO. 2, APRIL 2005