



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

OPTIMIZACIÓN DE RUTAS MEDIANTE COLONIA
DE HORMIGAS APLICADA AL PROBLEMA DE
RECOLECCIÓN DE RESIDUOS SÓLIDOS EN CIUDAD
UNIVERSITARIA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

MATEMÁTICA

PRESENTA:

ELIZABETH ALMA MANCERA GALVÁN

DIRECTOR DE TESIS:

DRA. BEATRIZ AURORA GARRO LICÓN



Ciudad Universitaria, Cd. Mx., 2016



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

OPTIMIZACIÓN DE RUTAS MEDIANTE COLONIA
DE HORMIGAS APLICADA AL PROBLEMA DE
RECOLECCIÓN DE RESIDUOS SÓLIDOS EN CIUDAD
UNIVERSITARIA

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

MATEMÁTICA

PRESENTA:

ELIZABETH ALMA MANCERA GALVÁN

DIRECTOR DE TESIS:

DRA. BEATRIZ AURORA GARRO LICÓN



Ciudad Universitaria, Cd. Mx., 2016

*Dedicado a
mi familia y a Panchito*

Agradecimientos

Quisiera expresar mi más sincero agradecimiento y aprecio a la Dra. Beatriz Garro Licón por su guía, infinita paciencia y ayuda a lo largo del desarrollo de este proyecto. También me gustaría agradecer a la Dra. Katya Rodríguez Vázquez por todo el apoyo que recibí para concluir esta tesis. He aprendido mucho al trabajar con ustedes.

Gracias a los doctores Verónica Arriola, José de Jesús Galaviz y Gustavo de la Cruz. Me dieron muchas sugerencias constructivas y comentarios, no sólo sobre la estructura, sino también sobre el contenido de la tesis y gracias a su experiencia profesional y conocimiento, este trabajo pudo ser mejorado.

Me gustaría dar gracias a mis padres y hermanos. Gracias por permitir que mi educación llegara hasta este punto. El término de esta tesis significa mucho para ellos, sin su apoyo este trabajo no habría sido posible.

Quisiera agradecer a Javi porque este es un logro, que en muchos sentidos no habría sido posible sin tu compañía. Gracias por tu paciencia, que hizo frente a mi obstinada ansiedad durante varias horas que dediqué a este trabajo. Sobre todo muchas gracias por tu apoyo y amor.

Gracias a mis amigos y compañeros de universidad: Héctor, Jorge, Michelle, Pamela y Pedro. Tengo los mejores recuerdos y anécdotas de esa etapa universitaria debido a ustedes.

Por último expreso mi enorme agradecimiento a esta universidad, que me abrió sus puertas del conocimiento.

Índice general

Agradecimientos	v
Lista de Figuras	v
Lista de Tablas	vii
Glosario	ix
Siglas	xi
1 Introducción	1
1.1 Motivación	1
1.2 Problema a resolver	2
1.3 Objetivo General	2
1.3.1 Objetivos Particulares	3
1.4 Restricciones	3
1.5 Aportaciones	3
1.6 Estructura de la Tesis	4
2 El Problema de Ruteo de Vehículos	5
2.1 Introducción	5
2.2 Definición del Problema de Ruteo	5
2.2.1 Elementos y Función Objetivo de un problema de ruteo	6
2.2.1.1 Tipo de Red	6
2.2.1.2 Los Clientes	6
2.2.1.3 Los Vehículos	7
2.2.1.4 Función Objetivo	7
2.3 Variantes del Problema de Ruteo	7
2.3.1 Traveling Salesman Problem (TSP)	8
2.3.2 Problema del cartero chino (CPP)	8
2.3.3 Otras variantes en el problema de ruteo	9
2.4 Resumen	9
3 Métodos de Solución	11
3.1 Introducción	11
3.2 Métodos Exactos	11
3.2.1 Programación Dinámica	11
3.2.2 Ramificación-Acotamiento	12
3.2.3 Ramificación-Corte	12
3.2.4 Ramificación-Evaluación	12
3.3 Métodos aproximados	13
3.3.1 Metaheurísticas	13

3.3.2	Algoritmos de Ahorros	14
3.3.3	Método <i>k-opt</i>	14
3.3.4	Métodos No Bioinspirados	15
3.3.4.1	Búsqueda Tabú	15
3.3.4.2	Recocido Simulado	16
3.3.5	Métodos Bio-inspirados	16
3.3.5.1	Algoritmos Evolutivos	16
3.4	Optimización por Colonia de Hormigas	17
3.4.1	De las colonias de hormigas a las hormigas artificiales	18
3.4.2	Representación del Problema	19
3.4.3	Hormigas Artificiales	20
3.4.4	Descripción de la Metaheurística ACO	20
3.4.5	Modelos	23
3.4.5.1	Ant System	23
3.4.5.2	Elitist Ant System	24
3.4.5.3	Max-Min Ant System	25
3.4.5.4	Ant Colony System	26
3.5	Resumen	27
4	Estado del arte	29
4.1	Métodos exactos	29
4.2	Métodos aproximados	30
4.2.1	Algoritmos ACO	30
4.2.2	Algoritmos Híbridos	33
4.3	Métodos de Solución Aplicados al Problema de Recolección de Residuos	34
4.4	Resumen	36
5	Caso de Estudio	37
5.1	Sistema de Manejo de Residuos Sólidos en CU	37
5.1.1	Generación	37
5.1.2	Almacenamiento de Residuos	38
5.1.3	Recolección	38
5.1.4	Transferencia de Residuos	39
5.1.5	Selección y Disposición Final	39
5.2	Sistema de Recolección en CU	39
6	Metodología	43
6.1	Descripción del problema	43
6.2	Representación matemática	44
6.2.1	Diseño de grafo: caso ATSP	44
6.2.2	Formulación para el rediseño de rutas	45
6.2.3	Diseño de grafo: caso ACVRP	46
6.2.4	Formulación para un nuevo diseño de rutas	47
6.3	Método de Solución	49
6.3.1	Caso ATSP	49
6.3.1.1	Algoritmo 3-opt	49
6.3.2	Caso ACVRP	51
6.4	Resultados	53
7	Experimentación y Análisis de Resultados	55
7.1	Resultados para ATSP	55
7.1.1	Efecto de los parámetros en la ejecución de los algoritmos	55

7.1.1.1	Pruebas a los parámetros del algoritmo AS	56
7.1.1.2	Pruebas a los parámetros del algoritmo EAS	57
7.1.1.3	Pruebas a los parámetros del algoritmo MMAS	58
7.1.1.4	Pruebas a los parámetros del algoritmo ACS	59
7.1.2	Rediseño de Rutas de Recolección en Ciudad Universitaria	62
7.2	Resultados para ACVRP	63
7.2.1	Ajuste de Parámetros	65
7.2.2	Diseño de Rutas de Recolección en Ciudad Universitaria	67
8	Conclusiones	77
8.1	Trabajos Futuros	79
	Anexos	81
	Anexo A Mapas de las Rutas de RSU operantes en CU	83
	Anexo B Análisis del efecto de los parámetros en los algoritmos ACO al ser aplicados al problema de recolección de residuos sólidos visto como un TSP	89
B.1	Algoritmo AS	89
B.2	Algoritmo EAS	94
B.3	Algoritmo MMAS	102
B.4	Algoritmo ACS	110
	Anexo C Resultados estadísticos de los algoritmos ACO	121
	Bibliografía	139

Lista de Figuras

Figura 2.1	Solución de un VRP sencillo	5
Figura 2.2	Algunas variantes del problema de ruteo	8
Figura 3.1	Algoritmo de ahorros	14
Figura 3.2	Experimento realizado por Deneubourg.	18
Figura 5.1	Área cubierta por cada una de las cinco rutas en Ciudad Universitaria y puntos de recolección asociados a cada ruta	40
Figura 6.1	Diagrama de Metodología	43
Figura 6.2	Reubicación de aristas en el algoritmo β -opt	50
Figura 6.3	Relocalización de nodo	52
Figura 6.4	Intercambio de nodo con $k = 1$	52
Figura 6.5	Operador de Cruce	53
Figura 6.6	Diagrama de flujo de ACO	54
Figura 7.1	Área cubierta por cada una de las cuatro rutas en la Zona Escolar, diseñadas con el algoritmo MMAS	70
Figura 7.2	Área cubierta por cada una de las cuatro rutas en la Zona Escolar diseñadas con el algoritmo ACS	71
Figura 7.3	Área cubierta por cada una de las dos rutas diseñadas por la DGOyC	76
Figura 7.4	Área cubierta por cada una de las dos rutas diseñadas con el algoritmo ACS y el algoritmo MMAS	76
Figura A.1	Ruta del Circuito Deportivo diseñada por la DGOyC	83
Figura A.2	Rediseño de ruta para el Circuito Deportivo con ACO	83
Figura A.3	Ruta del Circuito Exterior diseñada por la DGOyC	84
Figura A.4	Rediseño de ruta para el Circuito Exterior con ACO	84
Figura A.5	Ruta del Circuito de la Investigación Científica diseñada por la DGOyC	85
Figura A.6	Rediseño de ruta para el Circuito de la Investigación Científica con ACO	85
Figura A.7	Ruta del Circuito Escolar diseñada por la DGOyC	86
Figura A.8	Rediseño de ruta para el Circuito Escolar con ACO	86
Figura A.9	Ruta del Circuito CCU diseñada por la DGOyC	87
Figura A.10	Rediseño de ruta para el Circuito CCU con ACO	87
Figura B.1	Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito Deportivo	91
Figura B.2	Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito Exterior	91
Figura B.3	Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito de la Investigación Científica	92
Figura B.4	Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito Escolar	93

Figura B.5	Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito CCU	94
Figura B.6	Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito Deportivo	97
Figura B.7	Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito Exterior	98
Figura B.8	Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito de la Investigación Científica	99
Figura B.9	Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito Escolar	100
Figura B.10	Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito CCU	101
Figura B.11	Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito Deportivo	105
Figura B.12	Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito Exterior	106
Figura B.13	Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito de la Investigación Científica	107
Figura B.14	Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito Escolar	108
Figura B.15	Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito CCU	109
Figura B.16	Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito Deportivo	113
Figura B.17	Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito Exterior	114
Figura B.18	Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito de la Investigación Científica	115
Figura B.19	Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito Escolar	116
Figura B.20	Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito CCU	117
Figura B.21	Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Circuito Deportivo). Las barras indican la desviación estándar . .	118
Figura B.22	Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Circuito Exterior). Las barras indican la desviación estándar . . .	118
Figura B.23	Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Circuito de la Investigación Científica). Las barras indican la desviación estándar	119
Figura B.24	Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Circuito Escolar). Las barras indican la desviación estándar . . .	119
Figura B.25	Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Circuito CCU). Las barras indican la desviación estándar	120

Lista de Tablas

Tabla 2.1	Algunas variantes del VRP	10
Tabla 3.1	Similitudes y diferencias entre hormigas reales y artificiales.	20
Tabla 3.2	Algoritmos más populares de ACO	23
Tabla 5.1	Características de los circuitos de recolección	41
Tabla 6.1	Notación utilizada en el rediseño de rutas (ATSP)	45
Tabla 6.2	Notación utilizada en el rediseño de rutas (ACVRP)	48
Tabla 7.1	Valores constantes de los parámetros	56
Tabla 7.2	Valores variables de los parámetros del algoritmo AS	56
Tabla 7.3	Mejores resultados por cada parámetro del algoritmo AS aplicado a la Zona Escolar	57
Tabla 7.4	Valores variables de los parámetros del algoritmo EAS	57
Tabla 7.5	Mejores resultados por cada parámetro del algoritmo EAS aplicado a la Zona Escolar	58
Tabla 7.6	Valores variables de los parámetros del algoritmo MMAS	58
Tabla 7.7	Mejores resultados por cada parámetro del algoritmo MMAS aplicado a la Zona Escolar	59
Tabla 7.8	Valores variables de los parámetros del algoritmo ACS	60
Tabla 7.9	Mejores resultados por cada parámetro del algoritmo ACS aplicado a la Zona Escolar	61
Tabla 7.10	Secuencia de visita en los circuitos de CU obtenida con los algoritmos ACO	63
Tabla 7.11	Comparación entre rutas actuales y rediseño de rutas en Ciudad Universitaria	63
Tabla 7.12	Características de las instancias	64
Tabla 7.13	Valores constantes de los parámetros	64
Tabla 7.14	Valores variables de los parámetros de los algoritmos MMAS y ACS	65
Tabla 7.15	Configuraciones de parámetros para el algoritmo MMAS	66
Tabla 7.16	Configuraciones de parámetros para el algoritmo ACS	67
Tabla 7.17	Resultados obtenidos con MMAS	68
Tabla 7.18	Resultados obtenidos con ACS	68
Tabla 7.19	Características de las rutas obtenidas con los algoritmos MMAS y ACS	69
Tabla 7.20	Secuencia de visita de los puntos de recolección para la Zona Escolar (algoritmos ACS y MMAS)	71
Tabla 7.21	Características del conjunto de rutas de la DGOyC y del rediseño obtenido con ACO	73
Tabla 7.22	Características de las rutas de recolección de la Zona Cultural	73
Tabla 7.23	Resultados obtenidos con MMAS	74
Tabla 7.24	Resultados obtenidos con ACS	74
Tabla 7.25	Características de las rutas obtenidas con los algoritmos MMAS y ACS para la Zona Cultural	75
Tabla 7.26	Secuencia de visita de los puntos de recolección para la Zona Cultural (algoritmos ACS y MMAS)	76
Tabla C.1	Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito Deportivo	121

Tabla C.2	Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito Exterior	121
Tabla C.3	Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito de la Investigación Científica	122
Tabla C.4	Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito Escolar	122
Tabla C.5	Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito CCU	123
Tabla C.6	Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito Deportivo	123
Tabla C.7	Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito Exterior	124
Tabla C.8	Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito de la Investigación Científica	124
Tabla C.9	Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito Escolar	125
Tabla C.10	Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito CCU	125
Tabla C.11	Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito Deportivo	126
Tabla C.12	Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito Exterior	126
Tabla C.13	Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito de la Investigación Científica	127
Tabla C.14	Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito Escolar	127
Tabla C.15	Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito CCU	128
Tabla C.16	Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito Deportivo	129
Tabla C.17	Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito Exterior	130
Tabla C.18	Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito de la Investigación Científica	131
Tabla C.19	Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito Escolar	132
Tabla C.20	Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito CCU	133
Tabla C.21	Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado a la instancia P_CA_050	134
Tabla C.22	Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado a la instancia P_CA_050	135
Tabla C.23	Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado a la instancia P_CA_100	136
Tabla C.24	Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado a la instancia P_CA_100	137

Glosario

- Algoritmos bioinspirados** Algoritmos inspirados en el comportamiento que algunas especies muestran.
- Almacenamiento** Acción de retener temporalmente los residuos una vez generados y antes de ser recogidos o procesados por el servicio de recolección.
- ANOVA** ANalysis Of VAriance; Prueba estadística utilizada para comparar la media de varios grupos, con el fin de averiguar si diferentes características afectan la media.
- Asimétrico** Gráfica en donde el costo por cruzar un arco varía según la dirección.
- Compostaje** Proceso de manejo de residuos sólidos, por medio del cual se degradan bioquímicamente los residuos sólidos orgánicos, bajo condiciones controladas, hasta el punto en que el producto final puede ser manejado, embodegado y aplicado al suelo, sin que afecte negativamente el medio ambiente.
- Contenedor** Recipiente utilizado para el almacenamiento temporal de los residuos.
- Depósito** Sitio del cual parte los vehículos al iniciar su ruta, incluso en este lugar también son guardados cuando no se encuentran en operación.
- Disposición final** Acción de ubicar permanentemente los residuos sólidos. Proceso final de la manipulación y de la eliminación de los desechos sólidos.
- Estación de transferencia** Hace referencia a los sitios autorizados para el depósito temporal de residuos sólidos antes de ser trasladados a la disposición final.
- Estigmergia** Comunicación indirecta a través de la modificación del entorno.
- Feromona** Sustancia química liberada por algunos seres vivos que permite la comunicación entre ellos.
- Grafo** Conjunto finito no vacío de vértices, los cuales pueden estar conectados mediante aristas o arcos.
- Hormiga artificial** Agente computacional que construye una solución a un problema, y que deposita una cantidad de feromona por las conexiones que atraviesa.
- Metaheurística** Conjunto de conceptos algorítmicos que pueden ser utilizados para definir métodos heurísticos aplicables a un amplio conjunto de problemas diferentes.
- Métodos aproximados** Algoritmos que utilizan heurísticas para reducir el tiempo de cómputo, pero producen soluciones que no son necesariamente óptimas.
- Métodos exactos** Algoritmos que siempre producen una solución óptima.
- Recolección** Acción de retirar los residuos de la fuente que los generó y trasladarlos hacia una estación de transferencia, sitios de disposición final o instalaciones para tratamiento.
- Relleno sanitario** Instalación de ingeniería para la disposición de los residuos sólidos urbanos, diseñada y operada para minimizar los impactos a la salud pública y al ambiente.
- Residuo inorgánico** Son materiales que no se descomponen de forma natural o tardan largo tiempo en degradarse, como el plástico, el vidrio, el papel y los metales.
- Residuo orgánico** Son los residuos proveniente de podas en parques y jardines; residuos de actividades agropecuarias como estiércol y residuos de beneficios; y residuos domésticos como restos de comida y jardín.
- Residuos** Materiales que han cumplido su función y que tienen valor a través del rehúso o reciclaje.
- Residuos sólidos** Residuo que es susceptible de reutilización y reciclaje, tales como vidrio, papel, cartón, plásticos, laminados de materiales reciclables, aluminios y otros materiales no peligrosos, no considerados como de manejo especial.
- Transferencia** Es el traspaso de los residuos desde los vehículos recolectores a otros vehículos con más capacidad y para el transporte a distancia, se realiza en las Estaciones de Transferencia.

Siglas

2L-CVRP	Two-dimensional Loading Vehicle Routing Problem.
ACO	Ant Colony Optimization.
ACS	Ant Colony System.
ACVRP	Asymmetric Capacitated Vehicle Routing Problem.
ARP	Arc Routing Problem.
AS	Ant System.
ASRank	Rank-Based Ant System.
ATSP	Asymmetric Traveling Salesman Problem.
BWAS	Best-Worst Ant System.
CARP	Capacitated Arc Routing Problem.
CCU	Centro Cultural Universitario.
CPP	Chinese Postman Problem.
CU	Ciudad Universitaria.
CVRP	Capacitated Vehicle Routing Problem.
DGOyC	Dirección General Obras y Conservación.
DVRP	Dynamic Vehicle Routing Problem.
EA	Evolutionary Algorithm.
EAS	Elitist Ant System.
GA	Genetic Algorithm.
GRASP	Greedy Randomized Adaptive Search.
HCAS	Hyper Cube Framework Ant System.
IF-VRP	Intermediate Facilities Vehicle Routing Problem.
LNS	Large Neighborhood Search.
MDVRP	Multiple Depot Vehicle Routing Problem.
MMAS	Max-Min Ant System.
NI	Nearest Insertion.
NN	Nearest Neighbor.
NP	Non-Deterministic Polynomial-Time.
PL	Programación Lineal.
PVRP	Periodic Vehicle Routing Problem.
RNN	Randomized Nearest Neighbor.
RSU	Residuos Sólidos Urbanos.
SA	Simulating Annealing.
SDVRP	Split Delivery Vehicle Routing Problem.
SIG	Sistemas de Información Geográfica.
SVRP	Stochastic Vehicle Routing Problem.
TS	Tabu Search.
TSP	Traveling Salesman Problem.
VRP	Vehicle Routing Problem.
VRPB	Vehicle Routing Problem with Backhauls.
VRPSC	Vehicle Routing Problem with Stochastic Clients.
VRPSD	Vehicle Routing Problem with Stochastic Demands.
VRPSDP	Vehicle Routing Problem with Simultaneous Delivery and Pickup.
VRPSSTT	Vehicle Routing Problem with Stochastic Service and Travel Times.
VRPTW	Vehicle Routing Problem with Time Windows.

Capítulo 1

Introducción

La optimización combinatoria es un campo de las matemáticas discretas, que se relaciona con combinatoria, teoría de algoritmos, investigación de operaciones y ciencias de la computación. En los problemas relacionados a este campo, se busca la solución óptima en un espacio a lo más numerable de posibles soluciones.

Dentro de los problemas de optimización combinatoria se encuentran el problema de agente viajero TSP (*Traveling Salesman Problem*, por sus siglas en inglés) y el problema de ruteo de vehículos, VRP (*Vehicle Routing Problem*, por sus siglas en inglés). Este último consiste en encontrar una serie de rutas que visiten a un conjunto de clientes haciendo uso de una flota de vehículos, de tal manera que el costo sea mínimo y se cumplan ciertas restricciones asociadas al problema. Algunas restricciones son, por ejemplo, capacidad limitada del transporte, número de vehículos a usar, tiempos máximos de espera para descargar la mercancía, entre otras.

El VRP es considerado como un problema NP-completo, ya que el tiempo necesario para resolverlo aumenta exponencialmente con respecto al tamaño del problema. Debido a su complejidad, muchas son las técnicas que se han implementado para su resolución. Tal es el caso de técnicas de ramificación y acotamiento, ramificación y corte, programación dinámica y programación lineal. Sin embargo, resultan ineficientes para problemas a gran escala, por lo que en las últimas décadas se han implementado diversos métodos aproximados, que si bien, no siempre encuentran óptimos globales, las soluciones que dan son buenas aproximaciones y el tiempo de solución es razonable.

Uno de los métodos aproximados que puede resolver el problema de VRP, es la metaheurística de optimización por colonia de hormigas, ACO (*Ant Colony Optimization* por sus siglas en inglés), propuesta por Marco Dorigo en 1992. ACO es un algoritmo que se basa en la metáfora de recolección de alimento que llevan a cabo las hormigas para su supervivencia. Estos insectos, en la búsqueda de alguna fuente de alimentación, encuentran la ruta más corta entre el nido y dicha fuente.

El algoritmo ACO ha ayudado a resolver problemas de optimización enfocados en: ruteo de vehículos (Rizzoli, Oliverio, Montemanni & Gambardella, 2004), asignación cuadrática (Luca M. Gambardella, Taillard & Dorigo, 1999), problemas del grafo coloreado (Salari & Eshghi, 2008), problemas del agente viajero (Dorigo & Luca M Gambardella, 1997), problemas de gestión en telecomunicaciones (Di Caro & Dorigo, 1998), entre otros. Particularmente, en este trabajo se implementará ACO en un problema de VRP, el cual consiste en diseñar el conjunto de rutas óptimo para distintos vehículos en la recolección de residuos sólidos en Ciudad Universitaria (CU).

A continuación, se detalla la motivación de este trabajo y la razón del porqué es tan importante resolver el problema de recolección de residuos sólidos en la zona escolar más extensa de la República Mexicana, perteneciente a la Universidad Nacional Autónoma de México (UNAM).

1.1 Motivación

El problema de VRP se ha resuelto utilizando metaheurísticas y algoritmos exactos. Las aplicaciones reales de dichas técnicas se diversifican en repartición de alimentos, rutas de autobuses escolares, paquetería y mensajería, transportación de productos e incluso, recolección de residuos. Este último tipo de aplicaciones se analizarán en este trabajo pues es el problema que se busca resolver. Entre aplicaciones de dicho problema, de recolección de residuos, se encuentra el caso en el municipio de Sant Boi de Llobregat en Barcelona, donde se

tuvo que modificar su sistema de recolección de residuos en el año 2000, a causa de la aplicación del Plan Nacional para Residuos Urbanos en España (Bautista, Fernández & Pereira, 2008). El problema se resolvió utilizando un algoritmo de colonia de hormigas.

Otro ejemplo es Onitsha, una ciudad localizada en Nigeria, cuyo sistema de manejo de residuos era inadecuado e ineficiente, los costos eran muy altos debido a la mala planeación y la cobertura no era suficiente para la demanda en esa ciudad (Ogwueleka, T. C., 2009). Por estas razones, se desarrolló un método heurístico, que al ser implementado condujo a una reducción en los costos de recolección.

En los últimos años la importancia de llevar a cabo una correcta recolección de residuos ha sido tema fundamental en México. El crecimiento de la población, los diferentes esquemas sociales en los que se divide nuestro país y el impacto ecológico de una inconsciencia humana, han propiciado que en México se requieran sistemas de recolección de basura más eficientes.

Mundialmente existen proyectos donde se han aplicado sistemas de VRP para la recolección de residuos. Hasta el día de hoy, en México, las técnicas de optimización no han sido explotadas para la recolección de residuos, siendo éste un problema grave, ya que el tránsito excesivo de vehículos, la falta de organización, vehículos recolectores en mal estado y zonas de difícil acceso hacen el problema más complejo de lo que ya es.

La motivación de esta investigación consiste en ayudar a resolver el problema de optimización en la recolección de residuos sólidos en Ciudad Universitaria; una comunidad estudiantil, que está comprometida con el desarrollo económico, social y ambiental de nuestro país.

Ciudad Universitaria es el segundo terreno de gran extensión en el Distrito Federal, con aproximadamente 730 hectáreas y una población mayor a las 150,000 personas, lo que se traduce en un total de 15 toneladas¹ de basura en promedio generadas cada día. La Dirección General de Obras y Conservación (DGOyC), es la dependencia encargada de planear la recolección de residuos sólidos en Ciudad Universitaria, dividiéndola en cinco zonas y cada una de éstas tiene asignada una ruta de recolección. Sin embargo, hasta la fecha, no se ha aplicado algún método de optimización al sistema. Se espera que esta propuesta ayude a la disminución de costos, la resolución de una problemática tan fuerte como lo es el manejo de residuos y sobre todo, ser ejemplo para otras universidades, instituciones y/o empresas de que la ciencia puede ser aplicada en beneficio de la sociedad en nuestro país.

1.2 Problema a resolver

Resolver, mediante algoritmos bioinspirados, el problema de optimización de la ruta de recolección de residuos sólidos en Ciudad Universitaria de la Universidad Nacional Autónoma de México.

1.3 Objetivo General

Dado un problema de recolección de residuos sólidos, visto como un problema de ruteo de vehículos, en Ciudad Universitaria; implementar el algoritmo metaheurístico basado en Optimización por Colonia de Hormigas (ACO) para la optimización de las rutas.

¹Información proporcionada por la Dirección General de Obras y Conservación de la UNAM.

1.3.1 Objetivos Particulares

- Plantear el problema de recolección de residuos en CU, como cinco subproblemas del agente viajero asimétrico.
- Implementar cuatro variantes de ACO, dado los cinco subproblemas del agente viajero asimétrico.
- Analizar el efecto de los parámetros involucrados en los algoritmos ACO, al ser aplicado a cada subproblema del agente viajero asimétrico.
- Obtener resultados al aplicar el algoritmo de ACO a los subproblemas del agente viajero asimétrico.
- Plantear el problema de recolección de residuos en CU, como un VRP asimétrico y con capacidad.
- Implementar los dos mejores algoritmos (previo análisis) basados en ACO, dado un problema de VRP asimétrico y con capacidad.
- Implementar heurísticas para mejorar el desempeño de los algoritmos ACO.
- Obtener resultados al aplicar los algoritmos ACO al VRP asimétrico y con capacidad.
- Comparar los resultados obtenidos con el sistema actual de recolección de residuos sólidos utilizado por la Dirección General de Obras y Conservación de la UNAM.

1.4 Restricciones

Las restricciones del problema a resolver son las siguientes:

- El problema de ruteo de vehículos sólo será aplicado a Ciudad Universitaria.
- Sólo se utilizarán los algoritmos ACO: *Ant Colony System*, *Max-Min Ant System*, *Elitist Ant System* y *Ant System*.
- El problema se verá como un ACVRP y como subproblemas TSP asimétricos.
- Las rutas que se optimizarán son sólo aquellas en las que se recolectan residuos sólidos, por la mañana.

1.5 Aportaciones

Las aportaciones de este trabajo son las siguientes:

- El uso de técnicas de optimización basadas en ACO, para resolver el problema de recolección de residuos sólidos en Ciudad Universitaria.
- Este trabajo es hasta ahora el único en resolver el problema de recolección de residuos sólidos en Ciudad Universitaria, visto como un problema de ruteo y resuelto con técnicas bioinspiradas.
- Análisis de parámetros de los algoritmos ACO y estudio del desempeño de dichos algoritmos, al ser aplicados al problema de recolección de residuos sólidos en Ciudad Universitaria.
- Se implementaron técnicas de mejora a los algoritmos MMAS Y ACS para explotar el desempeño de los mismos y analizar el desempeño en el problema a resolver.
- Disminución de costos en tiempo y la planificación eficiente de la ruta para los vehículos recolectores de residuos sólidos.
- Mejoras, en cuanto a disminución de distancias, en las rutas de recolección que actualmente operan en

Ciudad Universitaria.

- Obtención de nuevas rutas, que pueden ser implementadas en el sistema de recolección de residuos sólidos de Ciudad Universitaria.

1.6 Estructura de la Tesis

La tesis se divide en siete capítulos, siendo el primero esta Introducción. En el Capítulo 2, se definirán los conceptos básicos del problema de ruteo de vehículos y diversas modificaciones del mismo. Mientras que en el Capítulo 3 se introducen algunos métodos de solución, así como aspectos teóricos de la metaheurística de optimización por colonia de hormigas y se describen algunas de sus variantes.

En el Capítulo 4 se presenta el estado del arte en relación el problema de ruteo de vehículos y su resolución, utilizando diferentes técnicas clásicas de optimización. Así también, aquellas investigaciones que tratan el problema de recolección de residuos como problema de ruteo de vehículos, y lo resuelven tanto con técnicas bioinspiradas como con métodos clásicos.

A continuación, en el Capítulo 5 se presenta a detalle el caso de estudio, es decir el sistema de recolección de residuos sólidos urbanos en Ciudad Universitaria.

En el Capítulo 6 se desarrolla la metodología propuesta y las restricciones del problema desarrollado en este trabajo.

En el Capítulo 7 se presentan los resultados experimentales, comparando los algoritmos propuestos para resolver el problema de recolección de residuos sólidos en Ciudad Universitaria.

Por último, en el Capítulo 8, se presentan las conclusiones del trabajo y las líneas de futura investigación, que pueden derivarse de esta tesis.

Capítulo 2

El Problema de Ruteo de Vehículos

2.1 Introducción

Este capítulo está dedicado a presentar una descripción a detalle del Problema de Ruteo de Vehículos (VRP, por sus siglas en inglés), empezando por la definición de los elementos que lo componen y la función objetivo del problema. Después se presentan diversas variantes y la explicación de cómo el VRP se encuentra relacionado al problema del agente viajero (TSP, por sus siglas en inglés).

2.2 Definición del Problema de Ruteo

Actualmente, la aplicación de modelos matemáticos a problemas reales es de gran importancia, debido al impacto positivo en costos, tiempo y calidad del servicio. Una de estas aplicaciones es el problema de ruteo de vehículos (VRP, por sus siglas en inglés) que, de manera general, consiste en establecer un conjunto de rutas de costo mínimo que, uno o varios vehículos deben recorrer con el objetivo de proporcionar un servicio a diversos clientes (Toth & Vigo, 2001). Dentro de las situaciones reales a las que ha sido aplicado exitosamente, se encuentran por ejemplo: el sector de transporte, distribución y logística.

Con el propósito de ejemplificar el problema de ruteo, consideremos un fábrica que desea distribuir sus productos a 10 tiendas. Cada una de estas tiendas ha hecho un pedido de cierto número de productos. La fábrica, para atender los pedidos, cuenta con tres vehículos. En este caso, el problema consiste en asignarle a cada vehículo una ruta, de tal manera que todas las tiendas sean visitadas. En la Figura 2.1 se presenta una solución a este problema propuesto.

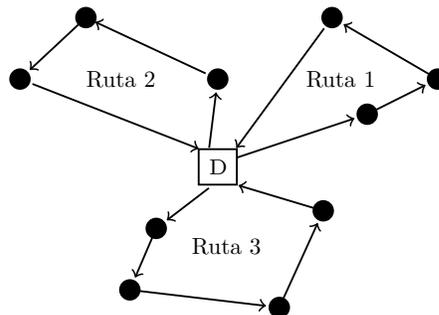


Figura 2.1: Solución de un VRP sencillo

A pesar de que la idea detrás del VRP es bastante sencilla, el problema pertenece a la clase de los NP-difícil (Lenstra & Kan, 1981), pues el tiempo necesario para resolverlo crece exponencialmente conforme la dimensión del problema aumenta.

Originalmente, el VRP propuesto por Dantzig y Ramser (1959), consistía en visitar un conjunto de clientes una sola vez, con una flota de vehículos que iniciaban y terminaban su ruta en un depósito central. El objetivo era minimizar la distancia total recorrida. Sin embargo, este planteamiento ha tenido que ser extendido, ya que los problemas reales tienen características y restricciones más complejas.

Algunas de las restricciones y características de los problemas de ruteo han dado pie a la creación de

diferentes variantes del VRP. Las restricciones hacen referencia a ciertas condiciones que se deben cumplir como: múltiples visitas a los clientes, horarios restrictivos de las visitas, vehículos previamente asignados a visitar ciertos clientes, volumen máximo de carga para los vehículos, entre otras; mientras que las características se encuentran relacionadas a la descripción del problema; por ejemplo: múltiples depósitos, no conocer desde el inicio la demanda de los clientes, disponer de diferentes tipos de vehículos, etcétera.

Antes de describir algunas variantes del problema de ruteo, primero se mencionan, en la siguiente sección, algunos elementos que las condicionan.

2.2.1 Elementos y Función Objetivo de un problema de ruteo

Un problema de ruteo cuenta con tres elementos principales: los clientes, el tipo de red y los vehículos (Rizzoli *et al.*, 2004). En esta sección se hace una descripción de dichos elementos y además, los posibles objetivos que pueden establecerse en un problema de ruteo.

2.2.1.1 Tipo de Red

La red es el elemento que describe las posibles rutas a seguir para llegar a un destino. Generalmente se representa mediante un grafo dirigido o no dirigido, dependiendo del planteamiento del problema.

Además, es importante mencionar que se puede hacer una clasificación general de dos clases de problemas de ruteo: uno de nodos y otro de arcos. El ruteo de arcos hace referencia a problemas donde los clientes son representados por el conjunto de arcos del grafo. Por otro lado, el ruteo por nodos se refiere a problemas donde los clientes son representados por el conjunto de vértices del grafo. Asimismo, en el ruteo por nodos, puede existir un depósito (espacio del cual los vehículos parten), que es representado por un nodo y a las aristas o arcos se les puede asociar un costo. Dicho costo hace referencia, por ejemplo, a la distancia entre los clientes o al tiempo requerido para dirigirse de un cliente a otro.

2.2.1.2 Los Clientes

La característica esencial de los clientes es tener asociada una demanda, la cual puede entenderse como una cantidad de bienes que tiene que ser cubierta, ya sea recibéndola o entregándola a cierto vehículo de carga.

Adicionalmente, los clientes pueden establecer horarios restringidos durante los cuales deben ser atendidos. Dichos horarios reciben el nombre ventanas de tiempo. Estas ventanas de tiempo pueden ocasionar costos adicionales en caso de no ser respetadas. Por ejemplo, consideremos un vehículo que debe recoger mercancía de un almacén que abre a las 8 a.m (límite inferior de la ventana) y cierra a las 2 p.m (límite superior de la ventana). Si el vehículo llega antes de las 8 a.m, deberá esperar hasta que abran; en caso de llegar después de las 2 p.m no alcanzará a recoger a tiempo los productos.

En el caso donde el vehículo llega antes de las 8 a.m, el costo adicional se genera al considerar que el vehículo podría atender a otro cliente a esa hora e ir al almacén más tarde, y de esta manera evitar el tiempo de espera.

Adicional a las ventanas de tiempo, que restringen el período de tiempo durante el día, también se puede tener una restricción en el conjunto de días durante los cuales los clientes deben ser atendidos.

Por último, los clientes puede tener asociado un tiempo de servicio, el cual se entiende como el estimado de cuánto se tarda el vehículo en atender al cliente.

2.2.1.3 Los Vehículos

La característica inmediata asociada a los vehículos es la capacidad de los mismos, no obstante, algunas formulaciones consideran capacidad ilimitada. La capacidad, en el sentido más sencillo, es entendida como la cantidad de bienes que pueden ser trasladados por los vehículos, sin embargo, esta idea puede extenderse a una representación bidimensional, por ejemplo: considerar la superficie del vehículo como un espacio en donde se debe acomodar el cargamento de manera óptima. Otra característica que se puede asociar al vehículo es un costo fijo por utilizarlo.

En unos problemas, el vehículo podría salir del depósito a recolectar la demanda de los clientes y regresar a dicho depósito, en otros, podría ser inexistente el depósito y el vehículo solamente debería transportar los productos de un cliente a otro.

Por otra parte, también existen restricciones de accesibilidad, es decir, que un vehículo no pueda transitar por ciertas calles, debido al peso o a la dimensión. Lo anterior supone, por ejemplo, utilizar diferentes tipos de vehículos, entre lo cuales alguno tenga características particulares que le permitan el acceso restringido. Ahora bien, cuando todos los vehículos tienen las mismas características se dice que la flota es homogénea, en caso contrario se le llama heterogénea.

Finalmente, cabe mencionar que no siempre el número de vehículos disponibles es suficiente para satisfacer las demandas, en este caso, algunos vehículos tendrán permitido realizar más de un recorrido y tendrán que adaptarse al entorno al seguir la mejor ruta que en ese momento se presente.

2.2.1.4 Función Objetivo

En relación a la función objetivo, pese a que la más natural es intentar minimizar la distancia, hay otras que también son utilizadas:

- Minimizar el número de vehículos;
- Minimizar costos de la operación (distancia total recorrida o tiempo de viaje) y el costo fijo asociado a los vehículos;
- Minimizar tiempo de espera;
- Maximizar el servicio al cliente;
- Maximizar eficiencia del vehículo, calculada como el porcentaje de capacidad de carga.

2.3 Variantes del Problema de Ruteo

Combinando los elementos mencionados anteriormente, se pueden establecer diferentes variantes del problema de ruteo, sea por nodos o por arcos. Y, si bien, los problemas de ruteo son generalmente entendidos como problemas de ruteo por nodos, y son los que han tenido un papel dominante dentro de la literatura, es importante hacer la distinción entre ambos.

De manera muy general, los problemas de ruteo por nodos son conocidos como VRP, cuya definición se presentó al inicio de este capítulo. Los problemas de ruteo por arcos se engloban en una categoría conocida como ARP (*Arc Routing Problem*). Cabe mencionar, que es común encontrar la transformación de problemas de ruteo por arcos en problemas de ruteo por nodos, para aplicar los métodos de solución desarrollados para éstos en los primeros.

En ambos casos, ruteo por nodos y ruteo por arcos, se han establecido variantes al combinar los elementos mencionados en la Sección 2.2.1 (Figura 2.2). Por ejemplo, si sólo se cuenta con un vehículo, el problema del VRP es reducido a uno del agente viajero (TSP); mientras que el ARP, se reduce a un problema del cartero chino (CPP). A continuación se ahonda un poco en este par de problemas, pues son la base de los problemas de ruteo al ser los más simples de entender y que a su vez, cuentan con menos restricciones.

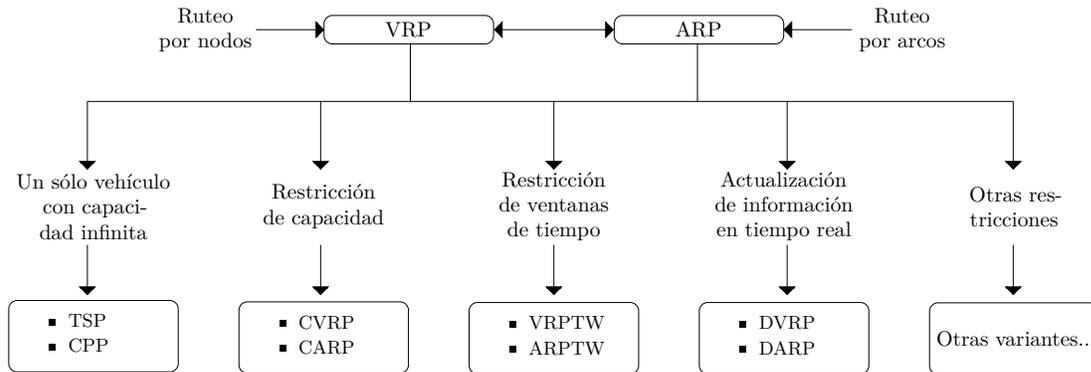


Figura 2.2: Algunas variantes del problema de ruteo

2.3.1 Problema del Agente Viajero (TSP)

El VRP se basa en el problema de optimización del agente viajero, dicho de otra manera, el VRP puede ser visto como un TSP si se considera un solo vehículo que es representado por el agente viajero. En general, el TSP consiste en un agente que debe visitar una lista de ciudades una sola vez, comenzando y terminando en la misma ciudad. El objetivo del problema es determinar el camino más corto que debe tomar el agente viajero.

El origen del problema del agente viajero se remonta a 1800s, cuando W. R Hamilton y T. Kirkman querían determinar *tours* entre los vértices de un icosaedro. Dicho *tour* debía empezar y terminar en el mismo punto, y cada vértice debía ser visitado una sola vez. Más tarde, a este tipo de *tour* se le llamó circuito Hamiltoniano.

A pesar de la aparición del problema en el siglo XIX, fue hasta 1930 cuando Karl Menger hizo una formulación matemática del TSP. En el planteamiento de Menger, se buscaba encontrar el camino más corto para un cartero y un viajero de negocios, refiriéndose a este problema como *Messenger Problem*. El nombre del agente viajero fue introducido por primera vez por Julia Robinson en 1949 (Chaitin, da Costa & Doria, 2012).

Si bien es cierto que el planteamiento del TSP es muy fácil de entender, intentar resolverlo no es tan sencillo. Por ejemplo, si consideramos que el agente debe visitar 10 ciudades, el número de combinaciones de posibles soluciones es de 3,628,800, pues el orden de este problema es $O(n!)$.

2.3.2 Problema del Cartero Chino (CPP)

El planteamiento del ARP empezó con Leonard Euler en 1735 (aunque no se le conocía con el nombre de ARP), al estudiar el problema de los puentes de Königsberg. El problema de los puentes de Königsberg, consiste en saber si una persona puede cruzar todos los puentes una sola vez, regresando al lugar del cual partió. Euler se basó en teoría de grafos para modelar el problema, el cual está descrito de la siguiente manera: encontrar un circuito que recorre todas las aristas una sola vez, partiendo y regresando al mismo vértice. Hoy en día, si este tipo de circuito existe, se le denomina como Circuito Euleriano, en honor a Leonard Euler.

Hasta 1962, se planteó la posibilidad de encontrar un circuito Euleriano de costo mínimo, conociéndose este planteamiento como el problema del cartero chino: CPP, *Chinese Postman Problem* (Kwan, M. K.). El CPP fue planteado de la siguiente manera: un cartero debe entregar la correspondencia en un vecindario, para lograrlo, él necesita recorrer todas las calles del vecindario y regresar a la oficina de correos. El objetivo es diseñar una ruta para el cartero, de tal manera que camine la distancia más corta.

Por otro lado, al considerar la complejidad de las versiones descritas en esta subsección, el VRP, TSP y también el ARP tienen una complejidad NP-difícil, sin embargo, el CPP puede serlo o no¹. Esto depende de la dirección de las aristas: si el grafo sólo contiene aristas sin dirección o con dirección, el problema se puede resolver en tiempo polinomial, por el contrario, cuando en la red que modela el problema se incluyen arcos dirigidos y no dirigidos el problema se convierte en NP-difícil.

2.3.3 Otras variantes en el problema de ruteo

En el problema original del VRP y del ARP, no se incluyen muchas características y restricciones que son necesarias para modelar problemas reales. Lo cual ha generado el surgimiento de variantes que incluyan las restricciones originales y otras adicionales. En la Tabla 2.1 se describen algunas de las variantes más conocidas del VRP. Cabe mencionar que la restricción de capacidad está prácticamente presente en todos los problemas de ruteo que modelan situaciones reales, pues resulta ilógico pensar en vehículos cuya capacidad sea infinita. En caso de omitirse tal restricción, se parte del supuesto que la capacidad del vehículo es suficiente para satisfacer la demanda de todo los clientes.²

Si bien, en la Tabla 2.1 sólo se muestran variantes del VRP, en el caso del ARP también se pueden considerar extensiones del problema original, al añadir características y restricciones como: múltiples depósitos y ventanas de tiempo. De hecho, así como en los problemas de ruteo por nodos (VRP), se involucra la capacidad de los vehículos, también en el ruteo por arcos puede añadirse tal característica. En 1981, Golden y Wong propusieron el problema de ruteo por arcos con capacidad (CARP, *Capacitated Arc Routing Problem*), el cual consiste en determinar un conjunto de rutas óptimas, de tal manera que se cumplan las siguientes características: cada arista debe ser visitada por un solo vehículo; las rutas inician y terminan en un depósito; la capacidad del vehículo no puede ser excedida.

En aplicaciones reales, los arcos pueden verse como las calles. Un ejemplo es, otra vez, la recolección de residuos, que en ciertos lugares se realiza por acera. Bajo esta situación, la capacidad asociada a las aceras (arcos) corresponde a la cantidad de residuos que se genera en dichas aceras, y que debe ser recolectada por un vehículo con capacidad limitada.

Por último, vale la pena mencionar, que muchas veces se crean nuevas variantes basadas en las necesidades del problema real que se quiera resolver.

2.4 Resumen

En este capítulo se presentó la definición del problema de ruteo por arcos y por nodos, la función objetivo que puede plantearse, así como los elementos que componen al problema. También se describieron diversas variantes, que pueden considerarse como extensiones del problema de ruteo original.

¹En Lenstra, J. K. y A. H. G. R. Kan, Complexity of vehicle routing and scheduling problems, 1981. se habla sobre la complejidad de estos problemas.

²Algunos autores, al hablar del planteamiento más básico del VRP incluyen la restricción de capacidad, sin embargo en este trabajo, se utilizará VRP cuando no haya restricción de capacidad y CVRP en caso contrario.

Tabla 2.1: Algunas variantes del VRP

Nombre de la variante	Objetivo	Restricciones
Capacitated VRP (CVRP)	Encontrar un conjunto de rutas que minimice la distancia total, de tal manera que la capacidad del vehículo no sea rebasada.	Capacidad de los vehículos.
VRP with Time Windows (VRPTW) Pullen y Webb (1967)	Encontrar un conjunto de rutas que: 1)minimice la distancia total recorrida 2)minimice el tiempo de espera. ²	Los clientes estrictamente deben ser visitados dentro de intervalos de tiempo.
Multiple Depot VRP (MDVRP) Kulkarni y Bhave (1985)	Minimizar la distancia total recorrida, donde el vehículo debe regresar al mismo depósito del cual partió.	Varios depósitos de los que parten los vehículos.
VRP with Simultaneous Delivery and Pickup (VRPSDP) Min, Hokey (1989)	Minimizar la distancia total recorrida, donde los vehículos primero entregan los productos a los clientes y después recogen los productos de los mismos.	Los clientes envían y reciben productos entre ellos.
VRP with Backhauls (VRPB) Pullen y Webb (1967)	Minimizar la distancia total recorrida, donde los vehículos atienden a dos tipos de clientes. En una ruta primero se atienden los clientes de ida.	Los clientes se clasifican en dos grupos: clientes de ida (los vehículos les entregan producto) y clientes de vuelta (ellos entregan producto al vehículo).
Split Delivery VRP (SDVRP) Dror y Trudeau (1989)	Minimizar la distancia total recorrida, permitiendo que un cliente sea atendido por varios vehículos siempre y cuando el costo total del recorrido se reduzca.	La demanda del cliente puede ser atendida por más de un vehículo.
Stochastic VRP (SVRP) Tillman, F. A. (1969) Tres variantes: VRP with Stochastic Demands (VRPSD), VRP with Stochastic Clients (VRPSC), VRP with Stochastic Service and Travel Times (VRPSSTT)	Minimizar la distancia total recorrida, considerando valores aleatorios en la demanda, clientes o tiempos de servicio y traslado.	Incertidumbre en el problema, ocasionada por: demandas modeladas como una variable aleatoria, clientes presentes con cierta probabilidad y tiempos de servicio y viaje modelados como variables aleatorias.
Periodic VRP (PVRP) Russell e Igo (1979)	Determinar una asignación óptima de qué días deben ser visitados los clientes y establecer un conjunto de rutas que minimice la distancia total recorrida	Planeación de rutas sobre un horizonte de tiempo. El vehículo no tiene la restricción de regresar al depósito el mismo día. El cliente debe ser visitado al menos una vez dentro del período de tiempo.
Dynamic VRP (DVRP) Wilson y Colvin (1977)	Minimizar la distancia total recorrida, considerando una replaneación de rutas una vez que llegan nuevos pedidos. ³	Desconocimiento de demandas desde el principio. Los clientes pueden hacer pedidos una vez que el proceso de distribución ha comenzado.
Intermediate Facilities VRP (IF-VRP)	Minimizar la distancia total recorrida, donde los vehículos tienen que ir a una <i>instalación intermedia</i> antes de regresar al depósito.	El vehículo antes de regresar al depósito, debe ir forzosamente a otro vértice (no un cliente), conocido como <i>instalación intermedia</i> .

²Para considerar el caso donde el vehículo llega después del límite superior de la ventana, pueden establecerse como no factibles las soluciones en las que el vehículo no llega a tiempo o, de manera menos restrictiva, se puede penalizar la función objetivo si se presenta tal situación.

³Es importante notar que hay una diferencia con el SVRP, pues si bien en ambos se desconoce cierta información, uno es dinámico y otro estocástico como sus nombres lo indican. El término dinámico hace referencia a la *evolución* de la información, es decir, el cambio que puede sufrir dicha información durante la ejecución de las rutas. Mientras que la palabra estocástico se asocia a la *calidad*, es decir al grado de incertidumbre de los datos. Más información sobre *evolución* y *calidad* en Psaraftis, H. N. (1980)

Capítulo 3

Métodos de Solución

3.1 Introducción

La complejidad del VRP ha motivado el desarrollo de diferentes técnicas para lograr su resolución. Estas técnicas o métodos, pueden ser agrupados de manera muy general en dos categorías: métodos exactos y métodos aproximados o heurísticos. Los métodos exactos se diferencian de los otros, principalmente porque son capaces de dar soluciones óptimas en un período finito de tiempo polinomial. Para lograr esta tarea, los métodos exactos examinan exhaustivamente todas las soluciones, lo cual involucra un costo computacional alto, principalmente en problemas donde el espacio de búsqueda es muy grande y de complejo modelado. Dado que, para la mayoría de los problemas reales se desconoce su solución óptima y el espacio de búsqueda puede presentar una dimensionalidad muy alta, es prácticamente imposible que los algoritmos exactos sean aplicados para su solución. Lo anterior ha llevado a la creación e implementación de los llamados algoritmos de aproximación que determinan una solución (la mejor) muy cercana a la óptima en un tiempo razonable.

En este capítulo se describen algunos de los métodos exactos y aproximados, que han sido utilizados para resolver el VRP. Principalmente, se hace énfasis en los algoritmos ACO, método que es utilizado para resolver el problema de recolección de residuos que motiva esta tesis.

3.2 Métodos Exactos

La manera más simple de resolver el VRP es enumerando el conjunto de posibles soluciones y escogiendo la mejor de éstas, pero tal estrategia resulta poco eficiente cuando la dimensión del problema es muy grande. Por otra parte, agregar restricciones al problema lo hace más complejo, por lo que es necesario utilizar métodos de solución que enumeren de forma más inteligente.

Ejemplos de este tipo de métodos exactos son: programación dinámica y algoritmos de ramificación (ramificación-acotamiento, ramificación-corte, y ramificación-evaluación). Se dice que estos métodos hacen una enumeración inteligente de todas las soluciones, porque, aunque buscan en el espacio completo de soluciones, no lo hacen de forma exhaustiva. Estos métodos hacen referencia a la frase *Divide y vencerás*, es decir, si un problema es difícil de resolver, éste puede ser particionado en subproblemas más sencillos, de tal manera que al final se pueda seleccionar la mejor solución generada por cada uno. A continuación se describen algunos de estos métodos.

3.2.1 Programación Dinámica

La programación dinámica es un método enfocado en solucionar problemas que pueden ser separados en etapas o subproblemas independientes. De manera general, este método se basa en una idea muy simple: para encontrarle solución a un problema dado, primero se resuelven subproblemas más simples, y a continuación se combinan las soluciones de los subproblemas para encontrar el óptimo del problema original (Taha, 2004).

La programación dinámica soluciona el problema al ser dividido en *etapas*, las cuales, a su vez, contienen

un número finito de *estados*.¹ En cada etapa se debe tomar una decisión, la cual asocia un estado actual con el inicio de la siguiente etapa. Se puede decir que la programación dinámica es un proceso de toma de decisiones, las cuales influyen en las decisiones futuras.

Lo anterior significa que la decisión tomada en una etapa depende sólo del estado en el que se encuentra y no del proceso para llegar a ese estado. Esto quiere decir que toda la información, que se requiere para tomar decisiones a partir del estado actual, se encuentra en él mismo. Adicionalmente, una vez que se conoce el óptimo global, las soluciones parciales asociadas a diferentes etapas del problema también son óptimas.

3.2.2 Ramificación-Acotamiento

La idea general del método de ramificación-acotamiento es explorar un árbol de soluciones, eliminando ramas que no conducen a la solución óptima. Este método, al igual que la programación dinámica, divide el problema inicial, de tal manera que el espacio de búsqueda es reducido cuando una solución no es prometedora. En el árbol, el nodo raíz es el problema original a resolver y los otros nodos representan los subproblemas generados mediante la estrategia de ramificación (Morrison, Jacobson, Sauppe & Sewell, 2016; Talbi, E. G., 2009).

La estrategia de ramificación busca particionar el espacio de soluciones, la cual se realiza al seleccionar un nodo entre los no explorados. Por otra parte, la estrategia de acotamiento se refiere a la poda de soluciones no potenciales, con el fin de evitar la búsqueda exhaustiva.

La poda consiste en calcular cotas en cada una de las posibles soluciones o nodos que pueden ser encontrados en caso de ramificar el nodo. Dichas cotas, son comparadas con la mejor solución conocida hasta ese momento. En caso de que la cota no mejore a tal solución, no será necesario seguir explorando ese nodo, debido a que en esa parte del árbol no puede existir el óptimo global, y por tanto puede ser eliminado. Las cotas inferiores pueden generarse mediante la relajación del problema, es decir, quitar una o varias restricciones para hacer al problema más sencillo. El proceso de dividir y podar continúa, descartando los subproblemas que se puede comprobar no contienen la solución óptima global o encontrando soluciones locales a los subproblemas, de tal manera que vayan sustituyendo a la mejor solución conocida hasta ese momento.

3.2.3 Ramificación-Corte

Los algoritmos de ramificación-corte son resultado de la combinación de los métodos de ramificación y acotamiento con los métodos de planos de corte. En general, este método consiste en agregar planos de corte (restricciones) al algoritmo de ramificación-acotamiento, siempre y cuando no se haya logrado podar un nodo, después de haber resuelto la relajación de un subproblema (Morrison *et al.*, 2016).

Los cortes representan restricciones violadas, los cuales pueden o no agregarse a la formulación del problema. Cuando no se puede podar el nodo, se busca alguna restricción violada por el óptimo de la relajación; si existe, la restricción se añade, si no, se continúa con la ramificación. Este tipo de métodos son útiles cuando el número de restricciones es muy grande.

3.2.4 Ramificación-Evaluación

Este método integra la generación de columnas (variables) en cada nodo del algoritmo de ramificación y acotamiento. La generación de columnas es una técnica que empieza con una parte del problema, al considerar

¹Información que se necesita en la etapa para tomar una decisión y que relaciona la etapa actual con la siguiente.

un número pequeño de variables. A esta parte del problema, se le conoce como problema maestro, el cual debe ser resuelto para después identificar qué variables deben agregarse al modelo (evaluación). Para saber qué columnas entrarán al problema se hace uso de un problema auxiliar llamado *pricing problem*. De esta manera, se va agrandando el problema hasta finalmente, resolverlo por completo. En caso de que alguna columna no pueda entrar al problema se hace la ramificación del mismo (Morrison *et al.*, 2016)

A diferencia del método de ramificación-corte, la ramificación-evaluación es un método útil para resolver problemas con un número grande de variables (Vance, P. H., 1998).

3.3 Métodos aproximados

Como ya se mencionó, los métodos exactos permiten resolver problemas de optimización combinatoria al encontrar la solución óptima. Sin embargo, cuando los espacios de búsqueda son grandes y no es posible encontrar la solución en tiempo polinomial, se recurre a otro tipo de técnicas de optimización. Los métodos aproximados o heurísticas han abordado exitosamente este tipo de problemas, debido a que no necesitan formalizar rigurosamente la cercanía a la solución óptima, no requieren conocimiento específico sobre el problema y su desempeño es visible de manera experimental.

La idea relacionada al concepto *heurística* es aplicar la experiencia y conocimiento para resolver un problema. Este término proviene del griego *heuriskein*, cuyo significado es “descubrir”. Una definición más formal y enfocada en problemas de optimización es dada por Hromkovic, J. (2010):

“Una heurística es una técnica robusta para el diseño de algoritmos aleatorios para los problemas de optimización (...), para los que uno no es capaz de garantizar a la vez la eficiencia y la calidad de las soluciones factibles calculadas, ni siquiera acotar con probabilidad constante $p > 0$.”

En esta definición, los métodos de búsqueda local pueden ser vistos como algoritmos heurísticos si se quita el término *aleatorio*. Por otro lado, *robusto* hace referencia a la posibilidad de ser aplicado a una gran variedad de problemas de optimización (Hromkovic, J., 2010).

Dentro de las heurísticas se encuentra un grupo de algoritmos denominado metaheurísticas, los cuales son algoritmos más generales, que con mínimas variaciones pueden resolver una gama amplia de problemas. En la siguiente sección se describen estas técnicas.

3.3.1 Metaheurísticas

La palabra metaheurística, además del término *heuriskein*, se compone del sufijo *meta* que significa “ir más allá”, se puede decir que las metaheurísticas buscan resolver problemas que otras heurísticas no han podido solucionar con éxito (Osman & Laporte, 1996). El término metaheurística fue introducido por vez primera por Glover, F. en 1986, y desde entonces varias definiciones se han propuesto. A continuación se cita una de estas definiciones:

“Una metaheurística se define formalmente como un proceso de generación iterativo que guía a una heurística subordinada al combinar inteligentemente diferentes conceptos para explorar y explotar el espacio de búsqueda, se usan estrategias de aprendizaje para estructurar la información con el objetivo de encontrar eficientemente soluciones cercanas al óptimo.” (Osman & Laporte, 1996)

Se puede decir que las metaheurísticas son algoritmos aproximados independientes al problema, es decir, que no requieren de conocimiento previo del problema a resolver, por lo que las metaheurísticas son aplicables en diferentes ámbitos. Estas técnicas pueden ser utilizadas como esqueleto para el desarrollo de una heurística, que al hacer uso de información del problema buscan encontrar soluciones del alta calidad. En general, las metaheurísticas desconocen si han encontrado la mejor solución, más aún, no son capaces de decir qué tan cercana es la solución que proporcionan al óptimo del problema, sin embargo, son métodos sencillos que permiten resolver problemas complejos en un tiempo razonable.

Ejemplos de metaheurísticas incluyen: algoritmos bioinspirados, búsqueda tabú, recocido simulado, algoritmos de búsqueda local, entre otros. A continuación se explican algunas de estas metaheurísticas, además se incluyen: la heurística algoritmo de ahorros y el algoritmo k -opt, estos dos últimos debido a que son de los más utilizados en problemas de ruteo. Por último se da una explicación detallada de ACO (*Ant Colony Optimization*), ya que esta metaheurística es la técnica utilizada para resolver el problema propuesto en esta tesis.

3.3.2 Algoritmos de Ahorros

El algoritmo de ahorros es un algoritmo constructivo, propuesto por Clarke y Wright (1964), para resolver el problema de ruteo de vehículos. La idea detrás del algoritmo es que si dos rutas diferentes pueden ser combinadas en una sola (ver Figura 3.1), entonces se podrá obtener un ahorro generado por tal combinación.

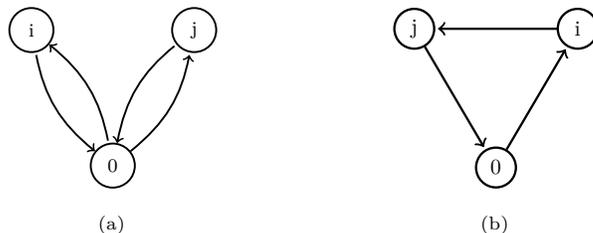


Figura 3.1: Algoritmo de ahorros

El ahorro se calcula de la siguiente manera: $s_{ij} = c_{i0} + c_{j0} - c_{ij}$. Lo anterior se traduce en que el ahorro s al ir del cliente i al j , es igual a la suma de los costos que hay entre el cliente i y el depósito (0) y entre el cliente j y el depósito, menos el costo generado por ir del cliente i a j .

El ahorro es calculado para todas las posibles combinaciones de clientes y se escoge la ruta que genera mayor ahorro.

3.3.3 Método k -opt

El algoritmo k -opt es un método de búsqueda local, enfocado en la resolución del TSP (*Traveling Salesman Problem*). Dado que los algoritmos k -opt son muy utilizados para mejorar las soluciones obtenidas con los algoritmos de hormigas (que se describirán mas adelante), se considera necesario explicar su funcionamiento en esta sección. Para profundizar en el algoritmo k -opt, primero se introducen de manera muy general los métodos de búsqueda local.

Los métodos de búsqueda local son quizá, la metaheurística más simple. En estos algoritmos se comienza con una solución, que va siendo reemplazada por otra que mejore la función objetivo. En cada paso, a la solución actual se le aplica un “movimiento” que permite explorar el vecindario de dicha solución, entendiendo

como vecindario al conjunto de soluciones alcanzables desde una solución inicial mediante el “movimiento”. Tal “movimiento” es una transformación de la solución, por ejemplo, cambiar el valor de una variable, o de un conjunto de variables. Este proceso se repite hasta que se cumpla algún criterio de paro, por ejemplo que la solución no mejore bajo ninguna transformación o se cumpla un número de iteraciones.

Estos algoritmos tienen la ventaja de usar poca memoria, pues es solo a partir de la solución actual que se encuentra una nueva. No obstante, una desventaja es que, no hay un método para determinar el tamaño del vecindario. Por un lado, si la cardinalidad del vecindario es pequeña, las soluciones no serán de buena calidad, por otro si el tamaño del vecindario es grande, también lo será el costo computacional.

La técnica *k-opt* inicia con un ciclo Hamiltoniano, al cual se le aplica una transformación que consiste en quitar k aristas del ciclo, resultando así en k caminos ajenos. Estos caminos son conectados entre sí, al añadir k nuevas aristas, produciéndose un nuevo ciclo. Para este ciclo se calcula la función objetivo y si mejora a la solución anterior, entonces esta última es reemplazada. En caso de que no haya una mejora, se hace otro intercambio de k aristas.

El funcionamiento de estos algoritmos no es totalmente satisfactorio, sin embargo, son simples, rápidos y forman parte de la base de metaheurísticas más complejas.

3.3.4 Métodos No Bioinspirados

En esta sección se describe la búsqueda tabú y el recocido simulado, metaheurísticas que no están incluidas en los métodos bioinspirados.

3.3.4.1 Búsqueda Tabú

Esta metaheurística fue propuesta por Glover, F. en 1986. La idea de esta técnica es dirigir una heurística de búsqueda local, en combinación con un mecanismo de memoria que toma en cuenta información sobre las soluciones pasadas.

El funcionamiento del algoritmo es el siguiente: empezar desde una solución inicial, definir su vecindario y moverse a la mejor solución contenida en tal vecindario. La característica importante es que el vecindario está restringido, pues sólo contiene soluciones que no se encuentran en la *lista tabú*. Esta lista almacena información de soluciones visitadas anteriormente y que no pueden ser visitadas nuevamente durante determinado número de iteraciones. Al evitar que se caiga en tales soluciones, se permite la exploración de nuevas regiones.

Es importante mencionar que cuando una *solución tabú* ofrece mejores resultados, se puede eliminar su característica de tabú, mediante un *criterio de aspiración*. Este criterio permite que soluciones clasificadas como tabú se vuelvan admisibles si cumplen cierta condición. Por ejemplo: si se toma como criterio el número de apariciones de cada solución en la lista; entonces, la que tenga una incidencia mayor será removida de la lista, dejando ser tabú.

Como característica interesante del algoritmo, se observa que la nueva solución no es necesariamente mejor que la solución existente. Adicionalmente, en el algoritmo de búsqueda tabú más básico, no hay noción de elecciones estocásticas, pues las soluciones vecinas son generadas a través de movimientos como el de inserción o el de intercambio.

3.3.4.2 Recocido Simulado

La técnica de recocido simulado (SA, por sus siglas en inglés *Simulating Annealing*) es aplicada a la optimización por primera vez por Kirkpatrick, Gelatt y Vecchi (1983). Esta metaheurística tiene su analogía en el proceso físico de recocido con sólidos, que tiene por objetivo hacerlos más resistentes o cristalinos al buscar su estado de mínima energía. A nivel físico, este proceso consta de dos fases. En la primera, el material es calentado a temperaturas muy altas, hasta que la energía del sistema es máxima. En la segunda fase, la temperatura desciende lentamente hasta formar una estructura cristalina, correspondiente a un estado de energía mínima. Sin embargo, dicha estructura cristalina puede quedar con imperfecciones si la temperatura con la que se enfría el sólido desciende bruscamente (Talbi, E. G., 2009).

El algoritmo inicia con una solución aleatoria (en el proceso físico la solución corresponde a un estado del sistema), y se fija una temperatura inicial. En cada iteración se compara el costo (corresponde a la energía del estado) de la solución actual con la de una nueva solución: si el costo de la nueva solución es menor, entonces ésta reemplaza a la solución actual. En caso contrario, la nueva solución se acepta con cierta probabilidad, la cual se calcula a partir de la distribución de Boltzmann.

De manera parecida al recocido de metales, en el algoritmo de optimización, inicialmente la temperatura es alta y, mientras el algoritmo corre hay un proceso lento de enfriamiento. Al comienzo, cuando la temperatura es alta es más frecuente que las soluciones empeoren, lo que permite salir de óptimos locales. Con el paso de las iteraciones, la temperatura desciende y, a su vez la probabilidad de aceptar peores soluciones disminuye, de este modo el algoritmo se concentra en encontrar soluciones en ciertos espacios de búsqueda (Sörensen, K., 2013). El algoritmo se detiene cuando se alcanza una temperatura final, previamente fijada.

3.3.5 Métodos Bio-inspirados

Como su nombre lo indica, los algoritmos bio-inspirados se basan en sistemas naturales para resolver problemas complejos de optimización. Por ejemplo, emulan la evolución de las especies (algoritmos evolutivos) o comportamientos sociales en insectos (inteligencia de enjambres). Estos algoritmos tienen la característica de ser métodos poblacionales, es decir, que para encontrar la mejor solución parten de un conjunto de soluciones (población), que serán mejoradas en un tiempo determinado, hasta converger a dicha solución dentro de un espacio de búsqueda específico. A continuación, se introducen, de manera muy general, los algoritmos evolutivos, haciendo énfasis en los algoritmos genéticos; además, se profundiza en los algoritmos de optimización basados en colonia de hormigas, que forman parte de inteligencia de enjambres.

3.3.5.1 Algoritmos Evolutivos

Los algoritmos evolutivos (EA, por sus siglas en inglés, *Evolutionary Algorithm*) están inspirados en la teoría evolutiva de Charles Darwin. Esta teoría combina conceptos de supervivencia, selección, reproducción y variación. Dentro de este grupo se encuentran los algoritmos genéticos (GA, por sus siglas en inglés, *Genetic Algorithm*), introducidos por Holland, J. H. en 1975.

En los GA los individuos de la población son evaluados mediante una función de aptitud, la cual permite seleccionar a los mejor adaptados. La selección es estocástica y se relaciona con la función de aptitud del individuo de la siguiente manera: entre mejor sea la evaluación de la función de aptitud de un individuo, mayor probabilidad de ser elegido (Yang, Lin, Lin, Gao & Ye, 2013). De esta manera, dicha función es la clave para determinar qué solución resuelve mejor el problema, pues de cada generación se seleccionan los individuos mejor adaptados. En consecuencia, la población al reproducirse tendrá descendientes con características de los padres mejor adaptados.

Por otra parte, el algoritmo utiliza dos operadores de variación, que tienen como objetivo la mejora de las soluciones. Estos operadores son: el operador de cruza y el operador de muta. El primero consiste en elegir dos individuos (soluciones) los cuales se *reproducen* y generan dos hijos con características heredadas de ambas soluciones. Al utilizar información de soluciones existentes se permite la explotación del espacio de búsqueda. El operador de muta consiste en hacer una modificación al individuo, al cambiar una o varias características del mismo, lo que permite la exploración de otras soluciones. El proceso concluye hasta que se cumpla un criterio de paro.

3.4 Optimización por Colonia de Hormigas

Los algoritmos basados en colonia de hormigas se encuentran dentro de las metaheurísticas fundamentadas en la inteligencia de enjambres (*Swarm Intelligence*). Este tipo de metaheurísticas simulan el comportamiento social de enjambres de insectos o grupos de animales que resuelven en conjunto alguna necesidad, por ejemplo: la búsqueda de alimento, el diseño de sus nidos, la regulación de la temperatura, la búsqueda del camino más corto de un nido hasta la fuente de alimentación, etc. Estas necesidades han sido observadas y llevadas al área de las ciencias computacionales, donde se ha podido emular su comportamiento para la solución de problemas reales de optimización para el hombre.

En la naturaleza, los insectos pertenecientes a un enjambre no son capaces de realizar tareas complicadas de forma individual. Sin embargo, a partir de interacciones directas o indirectas y de un trabajo en conjunto, logran llevarlas a cabo de manera exitosa, mostrando un comportamiento inteligente y complejo. Ejemplos de estos insectos sociales son: termitas, abejas y hormigas.

Dichos insectos sociales hacen uso de la estigmergia, término introducido por el biólogo Pierre-Paul Grasse en 1959, y que hace referencia a un tipo de comunicación. La estigmergia es definida como la estimulación que reciben los trabajadores (en un sentido biológico) por los logros alcanzados. Este tipo de comunicación tiene dos características principales: es indirecta, es decir, los insectos intercambian información a través del entorno, y dos, es local, lo cual significa que sólo los insectos que visiten el lugar asociado a la estigmergia tendrán acceso a ella (Dorigo, Birattari & Stützle, 2006). Por ejemplo, las hormigas intercambian información mediante feromonas.

A continuación, se profundiza en los algoritmos de hormigas, metaheurística que se basa en el comportamiento de estos insectos y que será aplicada para resolver un problema de ruteo en la recolección de residuos sólidos.

En la Sección 3.4.1 se describe el comportamiento de las hormigas que inspiró los algoritmos ACO. Después, en las Secciones 3.4.2 y 3.4.3, se describe la representación de un problema de optimización al cual se aplican los algoritmos ACO. También se describen las características de las hormigas artificiales y se resaltan sus diferencias y similitudes con respecto a las hormigas reales. Se sigue con la Sección 3.4, en la cual se describe el algoritmo ACO de manera general y también tres procedimientos básicos de los algoritmos de hormigas: *construcción de soluciones*, *actualización de feromona* y *acciones del demonio*. Finalmente, se incluyen algunos modelos de hormigas que surgieron por primera vez con el Sistema de Hormigas (AS por sus siglas en inglés), los cuales son: Sistema de Hormigas Elitista (EAS, por sus siglas en inglés), Sistema de Hormigas Max-Min (MMAS, por sus siglas en inglés) y Sistema de Colonia de Hormigas (ACS, por sus siglas en inglés) La mayor parte de lo expuesto sobre optimización por colonia de hormigas se basa en los siguientes trabajos: Dorigo *et al.* (2006, 1999, 1991), Dorigo y Stützle (2004), Stützle y Hoos (2000).

3.4.1 De las colonias de hormigas a las hormigas artificiales

Las hormigas son insectos sociales que viven en una colonia y realizan sus actividades con el fin de servir a la misma, es decir, buscan el desarrollo y bien común, más que el individual. En la colonia, la vida está organizada de tal manera que cada individuo tiene un trabajo propio que cumplir, por ejemplo están las hormigas encargadas de defender el nido, otras que deben cuidar los huevos y las que se encargan de encontrar la comida. Esta última actividad *auto-organizada*,² es particularmente la que inspira los algoritmos de optimización por colonia de hormigas (ACO, por sus siglas en inglés), metaheurística propuesta por Marco Dorigo en 1992, en su tesis de doctorado.

El comportamiento de las hormigas para buscar comida es muy singular, pues se ha observado que estos insectos son capaces de elegir el camino más corto entre varios caminos que conectan el nido de la colonia y la fuente de alimento. Esto es muy interesante, porque las hormigas son casi ciegas, y por tanto, el sentido de la vista puede ser excluido como medio de ayuda para la búsqueda del camino más corto (Dorigo, Maniezzo & Colorni, 1996).

Un experimento, realizado por Deneubourg, Aron, Goss y Pasteels (1990), permitió mostrar cómo cierta especie de hormiga seleccionaba el camino para encontrar la comida. En su trabajo, dichos autores demostraron que las hormigas, cuando están en busca de comida, exudan una sustancia química llamada feromona. Además, demostraron que las hormigas son capaces de hallar el camino más corto entre el nido y la fuente de alimento, debido a que siguen el rastro de feromona que depositan a su paso.

El experimento mencionado, consistía en separar el nido de la fuente de alimento mediante un puente con dos ramas de diferente longitud. En seguida, se dejaba escoger a las hormigas qué camino seguir. Los autores se dieron cuenta de que las hormigas, después de cierto tiempo, muestran una tendencia a recorrer la rama más corta (Ver Figura 3.2).

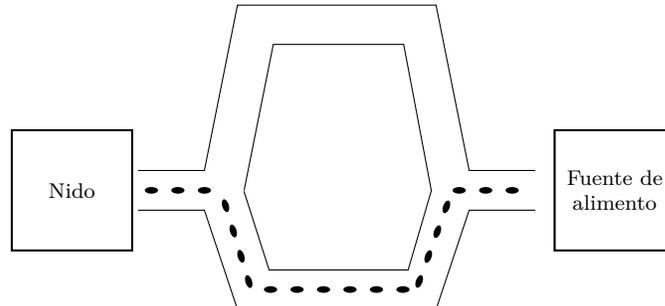


Figura 3.2: Experimento realizado por Deneubourg.

Si partimos del supuesto que las hormigas caminan a velocidad constante del nido a la fuente de alimento y viceversa; es claro que las hormigas, que han tomado el camino más corto, son las primeras en llegar a la fuente de alimento y, por tanto, también en regresar al nido. Dado que las hormigas depositan feromona a su paso, dicha sustancia se acumulará más rápido en el camino más corto, lo cual implica una mayor cantidad de feromona en tal camino. Por otra parte, la feromona se evapora en un intervalo corto de tiempo, por lo que, mientras más tiempo pasa, es menos detectable por las hormigas. De esta manera, cuando una hormiga recorre el camino más corto, el tiempo de evaporación es menor comparado con el tiempo de evaporación del camino largo, esto debido a que la feromona es reforzada por cada hormiga que recorre el camino. Y si bien, al principio de la búsqueda la elección del camino es aleatoria, al transcurrir el tiempo comenzarán a inclinarse por la selección del camino más corto, simplemente porque perciben una mayor concentración de

²Comportamiento complejo que surge debido a la interacción de agentes simples.

feromona en este camino.

El comportamiento antes descrito, ha inspirado el funcionamiento de los algoritmos de hormigas. En dichos algoritmos, el comportamiento de las hormigas reales es simulado mediante el uso de *hormigas artificiales*. En los algoritmos de hormigas, dichas *hormigas artificiales aprenden* sobre el espacio de búsqueda durante las iteraciones y utilizan esta experiencia adquirida para construir, cada vez, mejores soluciones. Este proceso de construcción puede entenderse como una toma secuencial de decisiones, que se rige a través de una regla de transición estocástica. De manera general, se puede decir que la optimización por colonia de hormigas consiste en una construcción estocástica iterativa de soluciones para el problema de optimización que se busca resolver.

Las metaheurísticas basadas en ACO, han sido aplicadas a diversos problemas de optimización, sin embargo, la aplicación más directa ha sido en problemas de ruteo, debido al mapeo natural de similitudes que existe entre ellas (Di Caro, Ducatelle & Gambardella, 2008).

3.4.2 Representación del Problema

A continuación se describe la representación del problema de optimización que sirve de base para los algoritmos de hormigas (metaheurísticas ACO) (Dorigo & Stützle, 2004).

Un problema de optimización es una 3-tupla $P = (S, \Omega, f)$, donde: S es el espacio de búsqueda definido sobre un conjunto X de variables de decisión, es decir, las variables para las que queremos encontrar el mejor valor, para hallar el valor óptimo de la función objetivo. Ω es el conjunto de restricciones asociadas al problema. f es la función objetivo, tal que $f : S \rightarrow \mathbb{R}$.

El espacio de búsqueda queda definido de la siguiente manera: dado un conjunto de variables $X = \{x_1, x_2, \dots, x_n\}$, con dominios D_i , respectivamente, una solución $s \in S$ es una asignación en la que a una variable de decisión x_i se le asigna un valor de su dominio $v_i \in D_i$. Además, una solución s es factible si cada variable satisface todas las restricciones Ω .

Dado lo anterior, el problema a optimizar por los algoritmos ACO es caracterizado por lo siguiente:

- Las variables de decisión son llamadas *componentes solución* y se denotan por c_i . El conjunto de componentes se representa por C .
- Las hormigas se mueven en un *grafo de construcción* $G_C = (V, E)$. Generalmente, los nodos son las componentes, y a las aristas se les llama conexiones. De hecho, en este trabajo se asumirá dicha representación.
- Se definen los *estados* del problema como soluciones parciales y se representan como una sucesión de componentes $\chi = (c_1, c_2, \dots, c_n)$. El conjunto de todas las posibles soluciones se denota por \mathcal{X} .
- Las componentes y las conexiones pueden tener un par de valores asociados: por un lado, un parámetro de *rastro de feromona*; por otro lado, la *información heurística*. Para este trabajo, se asumirá que son las conexiones a las que se les asocia la feromona, denotada por τ_{ij} y la información heurística, representada por η_{ij} .
- El conjunto de soluciones candidatas S es un subconjunto de \mathcal{X} .
- El conjunto de estados factibles $\tilde{\mathcal{X}}$ está definido por el conjunto finito de restricciones, con $\tilde{\mathcal{X}} \subseteq \mathcal{X}$.
- El conjunto de soluciones óptimas es denotado por S^* , con $S^* \subseteq \tilde{\mathcal{X}}$ y $S^* \subseteq S$.

A continuación, en la Sección 3.4.3 se explica el comportamiento y características de las hormigas artificiales en los algoritmos ACO.

3.4.3 Hormigas Artificiales

Como ya se ha mencionado, la metaheurística ACO sigue de manera muy estrecha el comportamiento de las hormigas reales, cuando éstas se encuentran buscando comida. De esta manera, en los algoritmos ACO se utilizan *hormigas artificiales*, las cuales reproducen el comportamiento de las hormigas reales..

De acuerdo a Dorigo y Stützle (2004), las hormigas artificiales tienen las siguientes características:

- Las hormigas buscan soluciones factibles de costo mínimo.
- Las hormigas artificiales, denotadas por k , tienen una memoria M^k que les permite guardar el camino recorrido.
- Cada hormiga tiene asignado un estado inicial del problema χ_i^k , y sólo se puede mover a otro estado que sea factible.
- Cada hormiga obedece una regla de decisión estocástica, la cual está en función de τ_{ij}, η_{ij} y las restricciones propias del problema a solucionar.
- Cuando una componente es añadida al estado actual, el rastro de feromona es actualizado.
- Cuando una hormiga ha construido una solución, es capaz de regresar por el mismo camino y actualizar la feromona asociada al camino total recorrido.

Los algoritmos ACO imitan el comportamiento de las hormigas, sin embargo, no son réplicas exactas. Esto se debe a que los algoritmos no emulan ciertas capacidades de las hormigas, por ejemplo: marcas visuales, otros tipos de feromonas y sistemas químico-biológicos complejos característicos de cualquier ser vivo. Por otro lado, los algoritmos ACO, que aquí se presentan, se basan sólo en la simulación de la metáfora de la búsqueda de alimento de las hormigas tomando en cuenta una feromona y otras características ajenas a las hormigas reales, con el fin de mejorar el desempeño de los algoritmos. En la Tabla 3.1 se describen similitudes y diferencias entre las hormigas reales y artificiales (Dorigo *et al.*, 1999).

Tabla 3.1: Similitudes y diferencias entre hormigas reales y artificiales.

Similitudes	Diferencias
Las hormigas reales y artificiales, encuentran los caminos más cortos de manera iterativa.	Las hormigas artificiales viven en un mundo discreto. Los movimientos de un estado a otro son secuenciales.
Las hormigas reales y artificiales, trabajan de manera conjunta para encontrar buenas soluciones a una tarea específica.	Las hormigas artificiales tienen una memoria interna que les permite almacenar las componentes que han sido visitadas.
Tanto las hormigas reales como las artificiales se comunican de manera indirecta mediante la feromona que depositan a su paso.	Las hormigas artificiales depositan una cantidad de feromona en función de la calidad de la solución encontrada.
Tanto las hormigas artificiales, como las reales, deciden probabilísticamente, qué camino seguir.	La probabilidad de que una hormiga artificial seleccione una componente combina feromona e información heurística.
El rastro de feromona decrece mediante el proceso de evaporación.	Las hormigas artificiales pueden depositar la feromona una vez que han encontrado una solución al problema.

3.4.4 Descripción de la Metaheurística ACO

Los algoritmos ACO pueden describirse como métodos iterativos, en los que en cada iteración las hormigas deciden a dónde moverse, es decir, a qué nodo avanzar, con la tarea de construir una solución. Para realizar esta tarea de construcción, las hormigas artificiales de la colonia se mueven concurrentemente y de manera asíncrona (es decir, dos o más hormigas no se mueven al mismo tiempo) en una grafo conexo $G_C = (V, A)$,

llamado grafo de construcción, como ya fue mencionado. El conjunto de componentes es denotado por C y coincide con el conjunto de nodos representado por V . Estos nodos se encuentran conectados mediante el conjunto de arcos A (Dorigo & Stützle, 2004).

Una parte esencial de los algoritmos ACO, es la combinación de información heurística y el rastro de feromona. La información heurística, también llamada visibilidad, mide qué tan deseable es una componente, con base en la información que se tiene de forma *a priori* del problema. Cabe mencionar, que las hormigas no modifican esta información. En cuanto al rastro de feromona, intuitivamente se puede decir que mide lo deseable que es una componente con base en lo que han aprendido las hormigas. En este caso, el rastro de feromona es modificable por las hormigas. El parámetro τ_{ij} refleja el rastro de feromona, mientras que η_{ij} representa la información heurística, ambos parámetros son utilizados en el algoritmo para calcular la probabilidad que permitirá a las hormigas decidir cómo moverse a través del grafo de construcción.

En general, la metaheurística ACO está compuesta por tres procedimientos (que se explican más adelante): construcción de soluciones, actualización de feromona y acciones del demonio. Estos procedimientos pueden ser implementados puede ser implementados en paralelo o secuencialmente, así mismo puede haber interacción entre ellos, según el problema que se quiera resolver (Algoritmo 1).

Algoritmo 1 Metaheurística ACO

Procedimiento Metaheurística ACO
Mientras No se cumpla criterio de paro **Hacer**
Actividades Programadas
 Construcción de Soluciones
 Acciones del demonio { % Opcional}
 Actualización de feromona
Fin Actividades Programadas
Fin Mientras
Fin Procedimiento

Construcción de Soluciones. Como el nombre lo indica, este procedimiento es utilizado para construir una solución completa al ir agregando secuencialmente componentes del conjunto C a soluciones parciales (χ). Al inicializar el algoritmo, se determina un rastro de feromona inicial τ_0 , que será depositado en las conexiones. Este rastro de feromona será modificado por las hormigas en cada iteración.

Inicialmente se tiene una solución parcial vacía, a la cual se le añade una componente en cada iteración. Esta componente se selecciona probabilísticamente dentro de un grupo de vecinos factibles, mediante la Ecuación 3.1:

$$P(c_{h+1} = j | \chi_h) = \begin{cases} \frac{F_{ij}(\tau_{ij})}{\sum_{(i,l) \in N_i^k} F_{il}(\tau_{il})} & \text{si } j \in N_i^k \\ 0 & \text{en otro caso} \end{cases} \quad (3.1)$$

donde j está en el conjunto de vecinos de i (N_i^k , si y sólo si la sucesión $\chi_{h+1} = (c_1, c_2, \dots, c_h, j)$, no viola las restricciones Ω . Para determinar el vecindario (N_i^k) factible, la hormiga guarda en su memoria M^k , qué componentes ha visitado y en qué orden; de esta manera, no repetirá componentes ni violará restricciones. $F_{ij}(\tau_{ij})$ es una función monótona y decreciente que depende generalmente de τ_{ij} y η_{ij} ; pero que varía, según el problema que se quiera resolver, así como por la variante de ACO que se utilice por lo que se describirá a detalle más adelante en cada variante ACO. Este proceso de construcción, puede verse como un camino en el grafo de construcción $G_C = (V, E)$.

Si durante el proceso de construcción realizado por cada hormiga, la solución parcial a construir no pertenece a las soluciones candidatas ($\chi_h \notin S$) y no hay otra componente que se pueda agregar a la solución ($N_i^k = \emptyset$), se descarta χ_h y la hormiga deberá construir otra solución.

El procedimiento completo está expresado en el Algoritmo 2.

Algoritmo 2 Procedimiento Construcción de Soluciones

```

1: Procedimiento Construcción de Soluciones
2:   Seleccionar un nodo de inicio  $c_1$ 
3:    $h \leftarrow 1$ 
4:    $\chi_h \leftarrow (c_1)$ 
5:   Mientras  $\chi_h \notin S$  and  $N_i^k \neq \emptyset$  Hacer
6:      $j \leftarrow$  Seleccionar nodo siguiente( $\chi_h$ )
7:      $\chi_h \leftarrow \chi_h \otimes j$ 
8:   Fin Mientras
9:   Si  $\chi_h \in S$  Entonces
10:    Devolver  $\chi_h$ 
11:  Si No
12:    Abortar
13:  Fin Si
14: Fin Procedimiento
  
```

Actualización de feromona Después de que se ha generado una solución, se modifica el rastro de feromona en cada componente. La actualización de feromona se lleva a cabo en dos pasos: evaporación y depósito de feromona. Ambos pasos tienen lugar en todas las conexiones del grafo, sin embargo, su función es diferente, incluso contraria. Lo anterior se debe a que al depositar feromona se busca potencializar las buenas soluciones, mientras que a través de la evaporación, se quiere evitar que el algoritmo converja prematuramente a soluciones subóptimas.

El procedimiento de actualización de feromona se describe en el Algoritmo 3. Donde L , es una solución construida y ρ es la tasa de evaporación con $\rho \in (0, 1]$. Este parámetro generalmente se fija desde el inicio y es utilizado para limitar la cantidad de feromona acumulada en las conexiones, y así, enfocarse en caminos más recientes. Además, busca que las peores soluciones sean *olvidadas*. Lo último se debe a que las conexiones poco recorridas o no elegidas no son buenas decisiones y, por tanto, al reducir la cantidad de feromona en los mismos, las hormigas los encontrarán menos atractivos en cada iteración.

Algoritmo 3 Procedimiento Actualización de Feromona

```

1: Procedimiento Actualización de Feromona
2: Para todo  $(i, j) \in L$  Hacer
3:    $\tau_{ij} \leftarrow (1 - \rho)\tau_{ij}$ 
4: Fin Para
5: Para todo  $s \in \hat{S}_m$  Hacer
6:   Para todo  $(i, j) \in s$  Hacer
7:      $\tau_{ij} \leftarrow \tau_{ij} + q_f(s|S_1, S_2, \dots, S_m)$ 
8:   Fin Para
9: Fin Para
10: Fin Procedimiento
  
```

S_m es el conjunto de soluciones generadas en la iteración m . $q_f : S_m \rightarrow \mathbb{R}$, es una función tal que si $f(s_1) \leq f(s_2)$ entonces $q_f(s_2) \leq q_f(s_1) \forall s_1 \neq s_2 \in \hat{S}_m$. Dicha función, llamada *función de calidad*, determina la cantidad de feromona a agregar en la conexión (i, j) , según la calidad de la solución encontrada. En otras palabras, mientras mejor sea la solución asociada a (i, j) , mayor será la feromona depositada, pues si el valor de la función objetivo de la solución s_1 es mejor que el valor de la solución s_2 , la calidad de la solución s_1 será al menos tan buena como la calidad de la solución s_2 .

El conjunto \hat{S}_m , determina qué soluciones se tomarán para depositar la feromona. El depósito tiene el objetivo de representar la *deseabilidad aprendida*, al escoger las componentes.

Es importante mencionar, que este procedimiento es muy general, y las variantes de la metaheurística ACO utilizan diferentes *funciones de calidad*, así como diferentes conjuntos \hat{S}_m . Para algunos algoritmos,

\hat{S}_m contiene todas las soluciones generadas en la iteración m , es decir, $\hat{S}_m = S_m$; para otros, \hat{S}_m también se compone por la mejor solución encontrada hasta el momento, denotada por s_{bs} , o por la mejor solución de la iteración (s_{ib}).

Acciones del demonio. Este procedimiento consiste en aplicar ciertas acciones que no son ejecutables directamente por las hormigas. Ejemplos de estas acciones son: depositar una cantidad extra de feromona sólo en conexiones asociadas a ciertas soluciones y la aplicación de algoritmos de búsqueda local que permitan mejorar las soluciones encontradas por las hormigas. Cabe señalar que este procedimiento es opcional.

3.4.5 Modelos

A partir de que la metaheurística ACO fue introducida por Marco Dorigo en 1991, se han desarrollado diversas propuestas que buscan mejorar el funcionamiento del algoritmo original. En esta sección se presentan los algoritmos AS, EAS, ACS y MMAS. En la Tabla 3.2 se muestra cronológicamente la aparición de estos y otros algoritmos, así como quiénes los propusieron.

Tabla 3.2: Algoritmos más populares de ACO

Año	Algoritmo	Autor
1991	<i>Ant System</i> (AS)	Marco Dorigo
1992	<i>Elitist AS</i> (EAS)	Marco Dorigo
1996	<i>Ant Colony System</i> (ACS)	Dorigo & Gambardella
1996	<i>Max-Min AS</i> (MMAS)	Stützle & Hoos
1997	<i>Rank-Based AS</i> (ASRank)	Bullnheimer, Hartl & Strauss
2000	<i>Best-Worst AS</i> (BWAS)	Cordon, de Viana, Herrera & Moreno
2001	<i>Hyper-Cube Framework AS</i> (HCAS)	Blum, Roli & Dorigo

3.4.5.1 Ant System

El primer algoritmo dentro de la metaheurística ACO fue el *Ant System*. Originalmente el AS tenía tres variantes *ant-density*, *ant quantity* y *ant cycle* (Dorigo *et al.*, 1991). En el caso del *ant-density* y del *ant-quality* la feromona es depositada mientras las hormigas construyen sus soluciones. Por otro lado, en la tercer variante, la actualización de feromona se realiza una vez que todas las hormigas han construido su solución. Debido a que la última variante arrojaba mejores resultados, ésta ha sido adoptada como el algoritmo AS y éste es el algoritmo que se describe en esta sección.

Construcción de la Solución. Siguiendo el procedimiento de construcción de soluciones descrito en la Sección 3.4.4, la Ecuación (3.1) se convierte en la Ecuación (3.2), en el algoritmo AS. Dicha Ecuación, es una regla de transición de estados, llamada *regla de proporción aleatoria*. Es así, que una hormiga k ubicada en la componente i , elige moverse a la componente j , con la siguiente probabilidad:

$$P_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{u \in N_i^k} (\tau_{iu})^\alpha (\eta_{iu})^\beta} & \text{si } j \in N_i^k \\ 0 & \text{en otro caso} \end{cases} \quad (3.2)$$

En esta variante se observa que la función $F_{ij}(\tau_{ij})$ es una combinación del rastro de feromona (τ_{ij}) y la visibilidad (η_{ij}). Más aún, la importancia relativa de ambos parámetros está ponderada por α y β . El papel que juegan ambos parámetros es el siguiente: si $\alpha = 0$ se tiene que la regla de transición queda expresada en términos de la información heurística η_{ij} , por lo que el algoritmo se reduce a uno del tipo glotón. Por

otro lado, si β se anula, la probabilidad de ir a la componente j , desde la i , depende exclusivamente de los rastros de feromona, en este caso, el algoritmo puede terminar en un rápido estancamiento, pues las hormigas tenderán a seguir siempre los caminos con mayor cantidad de feromona (Dorigo & Stützle, 2004). En resumen, si α tiene un valor alto significa que se le da mayor importancia al rastro de feromona. Por el contrario, si β es muy grande, las hormigas tienden a elegir con base en el conocimiento a priori del problema. De aquí la importancia de equilibrar ambos parámetros (Ver Solnon, C., 2013, Capítulo 9).

Si se trata del TSP, la información heurística suele calcularse como $1/d_{ij}$, donde d_{ij} representa la distancia entre las componentes en este caso, ciudades i y j . Es claro, que entre más cerca esté j de i , mayor será el valor de η_{ij} , lo que supone una mayor probabilidad de ir a la ciudad j , estando en i .

Actualización de feromona. Como ya se explicó en la Sección 3.4.4 la actualización de feromona involucra la evaporación y el depósito de la misma. Los valores de feromona son modificados con el objetivo de que, en futuras iteraciones, las hormigas construyan soluciones similares a las mejores soluciones previamente construidas. En el algoritmo AS, la cantidad de feromona a depositar, tomando en cuenta la evaporación en las conexiones, sigue la Ecuación (3.3):

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3.3)$$

$\Delta\tau_{ij}^k$ representa la función de calidad y el conjunto $\hat{S}_m = S_m$. En el caso del TSP así como de algunos problemas de ruteo se calcula como:

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{si la hormiga } k \text{ atravesó el arco } (i, j) \\ 0 & \text{en otro caso} \end{cases} \quad (3.4)$$

Donde Q es una constante y L_k es la longitud del tour construido por la hormiga k , es decir, la evaluación de la solución encontrada $f(s)$.

Cabe mencionar, que la actualización de la feromona es hecha por **todas** las hormigas y es esta característica la que distingue este algoritmo de hormigas de los demás.

Acciones del demonio. En el AS básico no se tienen acciones del demonio.

3.4.5.2 Elitist Ant System

El algoritmo *Elitist Ant System* (EAS) fue la primer mejora hecha al AS, sólo un año después de la propuesta del mismo. La regla de transición de estados es la misma que en AS, la cual está dada en la Ecuación 3.2. Sin embargo, en el algoritmo EAS se introduce la acción del demonio con la que se busca darle un peso adicional a la mejor solución conocida hasta el momento (desde ahora, llamada mejor solución global, s_{bs}). Específicamente, se agrega una cantidad de feromona extra en las conexiones que pertenecen a la mejor solución global (hormiga elitista).

La actualización de la feromona modificada es la siguiente:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k + e\Delta\tau_{ij}^{bs} \quad (3.5)$$

Los dos primeros términos de la Ecuación (3.5) son los mismos que el AS utiliza para la actualización de la feromona. El tercer término $e\Delta\tau_{ij}^{bs}$ representa el depósito extra de feromona, ya mencionado, donde e es un parámetro que pondera la influencia de la mejor solución global s_{bs} . En este algoritmo, la función de calidad toma como conjunto de referencia a todas las soluciones generadas en la iteración, más la mejor solución global $\hat{S}_m = S_m \cup \{s_{bs}\}$. Es importante mencionar que $\Delta\tau_{ij}^{bs}$ sólo ha de depositarse en arcos que pertenecen a la mejor solución encontrada.

El EAS tiene la ventaja de converger más rápido a la solución en comparación con el AS, aunque esto no es del todo positivo, pues provoca que la exploración de nuevas soluciones disminuya.

3.4.5.3 Max-Min Ant System

El algoritmo *Max-Min Ant System* (MMAS) toma como base el algoritmo AS, al utilizar el mismo procedimiento para construir soluciones. Sin embargo, las características que hacen propio al MMAS son:

- Una explotación de la mejor solución global o la encontrada en la iteración actual. Esta explotación puede llevar al estancamiento en el que todas las hormigas siguen la misma ruta.
- El rastro de feromona está acotado inferior y superiormente, lo cual ayuda a contrarrestar el estancamiento.
- Los niveles de feromona son reinicializados si, durante cierto número de iteraciones, la solución no ha mejorado. Esto sirve para dar mayor fuerza a la exploración temprana.

Actualización de feromona. Al igual que en el algoritmo AS, los niveles de feromona son actualizados una vez que todas las hormigas han construido una solución. La actualización de feromona explota la mejor solución de una sola hormiga, ya sea la mejor global o la mejor de la iteración. La actualización de feromona sigue la siguiente ecuación:

$$\tau_{ij} = [(1 - \rho)\tau_{ij} + \Delta\tau_{ij}^{best}]_{\tau_{min}}^{\tau_{max}} \quad (3.6)$$

donde τ_{min} y τ_{max} son los límites inferior y superior de feromona, respectivamente. $\Delta\tau_{ij}^{best}$ es la función de calidad que representa la aportación de feromona realizada por la mejor solución global o la mejor solución de la iteración actual. De esta manera, $\hat{S}_m = \{s_{bs}\}$ o $\hat{S}_m = \{s_{ib}\}$. Cuando se decide usar s_{bs} , la exploración se ve limitada pues las soluciones tienden a concentrarse cerca de ese valor. Por otro lado, si sólo se usa s_{ib} aumenta la probabilidad de explorar el espacio de búsqueda, ya que las soluciones pueden cambiar considerablemente de iteración en iteración. Por esta razón, también puede considerarse utilizar una combinación de ambas soluciones para encontrar mejores soluciones (Stützle & Hoos, 2000). $\Delta\tau_{ij}^{best}$ es calculada para el TSP como:

$$\Delta\tau_{ij}^{best} = \begin{cases} \frac{1}{L_{best}} & \text{si } (i, j) \text{ pertenece al mejor tour} \\ 0 & \text{en otro caso} \end{cases} \quad (3.7)$$

L_{best} es la longitud del mejor tour encontrado en la iteración actual o desde el inicio del algoritmo.

A continuación se explica la función de los límites de feromona incluidos en la Ecuación (3.6), así como la inicialización y reinicialización del rastro de feromona.

Límites de rastro de feromona. Como ya se mencionó, el algoritmo MMAS se caracteriza por acotar el rastro de feromona. Cada vez que el rastro de feromona se actualiza, se debe asegurar que el valor se encuentre en el intervalo $[\tau_{min}, \tau_{max}]$. El límite inferior permite que todas las conexiones tengan al menos una cantidad mínima (τ_{min}) de feromona. Lo anterior se traduce en asignar a las conexiones una pequeña probabilidad mayor a cero, de ser elegidas, de esta manera habrá mayor exploración en el espacio de búsqueda. Por otra parte, la diferencia de valores de feromona entre conexiones disminuye y con esto, la probabilidad de escoger componentes con distintas cantidades de feromona, también se reduce y por tanto, habrá mayor exploración en el espacio de búsqueda. Finalmente, el límite superior evita que la cantidad de feromona crezca sin medida, lo que previene la prematura convergencia a subóptimos.

Inicialización y reinicialización de rastros de feromona. En este algoritmo se quiere dar mayor exploración al inicio. Esto se consigue al inicializar la feromona con valores muy altos, para que después de la primer iteración coincidan con τ_{max} .

Por otra parte, el mecanismo de reinicialización permite que el algoritmo explore caminos con baja probabilidad de ser escogidos. Este mecanismo es activado cuando el algoritmo se acerca a estancamientos.

3.4.5.4 Ant Colony System

A pesar de que en sus inicios, el algoritmo AS mostró ser un método capaz de encontrar buenas soluciones a problemas de optimización, no era tan competente en comparación con algoritmos bien establecidos para ciertos problemas. De esta manera, se introdujeron modificaciones al AS y se crearon variantes que lo mejoraron, como las ya presentadas en secciones anteriores. Otra variante es el algoritmo *Ant Colony System* (ACS) propuesto por Dorigo y Gambardella (1997), cuya principal diferencia con el algoritmo AS es el uso de una regla de transición que permite establecer de manera directa el balance entre la explotación y la exploración.

Otro par de modificaciones introducidas al algoritmo ACS son: al final de cada iteración, la evaporación y depósito de feromona, sólo se llevan a cabo en las conexiones que pertenecen a la mejor solución global (actualización global); y, cada vez que una hormiga ha utilizado un arco, la cantidad de feromona de ese arco disminuye (actualización local).

A continuación, se explica el procedimiento del algoritmo ACS.

Construcción de soluciones. Para construir las soluciones, las hormigas utilizan la Ecuación (3.8), conocida como *regla proporcional pseudo-aleatoria*. Una hormiga k ubicada en la componente i se moverá a la j de acuerdo a:

$$j = \begin{cases} \operatorname{argmax}_{l \in N_i^k} \{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta\} & \text{si } q \leq q_0, \\ J & \text{en otro caso} \end{cases} \quad (3.8)$$

donde j es el siguiente nodo a visitar, q es una variable aleatoria uniforme sobre el intervalo $[0, 1]$ y q_0 es un parámetro que toma valores en el intervalo $[0, 1]$. Como antes, η_{ij} es la información heurística, definida como el inverso de la distancia de i a j . Y, α y β son los parámetros que determinan la influencia relativa de τ_{ij} y η_{ij} , respectivamente. Esta regla tiene una doble función, pues si $q \leq q_0$ se beneficia la explotación de tal manera que se acelera la convergencia al mejor tour, de hecho $S_m = \{\operatorname{argmax}_q q_f | s \in S^{best}\}$. Por otra parte, si $q \geq q_0$ se potencializa la exploración, pues J denota el camino a seguir de acuerdo a la Ecuación 3.2, esto permite evitar la convergencia temprana hacia óptimos locales.

En el algoritmo ACS, la actualización de feromona queda dividida en dos: un proceso local y otro global.

Actualización local de feromona. La regla de actualización utilizada en este algoritmo permite que todas las hormigas modifiquen los niveles de feromona después de atravesar cada conexión. Esta regla, conocida como regla de actualización local de feromona, se muestra a continuación:

$$\tau_{ij} = (1 - \psi)\tau_{ij} + \psi\tau_0 \quad (3.9)$$

donde ψ , es un coeficiente de decaimiento de feromona que toma valores en $(0, 1]$, y τ_0 es el valor inicial de rastro de feromona. La Ecuación (3.9) incluye la evaporación y un depósito determinado por τ_0 . Con esta regla, cada vez que una hormiga visita un conexión, ésta pierde feromona. De esta manera, mientras más hormigas pasen por una conexión, menos atractivas serán las conexiones, lo cual favorece la exploración de las conexiones no visitadas (Dorigo *et al.*, 1999) y previene estancamientos.

Actualización global de feromona. Para realizar la actualización global, se toma en cuenta una sola hormiga, aquella que generó la mejor solución global. La Ecuación (3.10) es utilizada para añadir feromona a las componentes, después de cada iteración.

$$\tau_{ij} = \begin{cases} (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{best} & \text{si } (i, j) \text{ está en el mejor tour,} \\ \tau_{ij} & \text{en otro caso} \end{cases} \quad (3.10)$$

donde $\Delta\tau_{ij}^{best}$ se calcula mediante la Ecuación 3.7.

Hay que recalcar que esta actualización sólo es aplicada a las conexiones que pertenecen a la mejor solución global. Además, este proceso pertenece a las acciones del demonio, ya que las hormigas no actualizan

directamente el rastro de feromona. Con esta regla se busca reforzar la cantidad de feromona en aquellos tours más cortos.

3.5 Resumen

En este capítulo se revisaron algunos métodos de solución que han sido aplicados al VRP y sus variantes. Por un lado, los métodos exactos como Programación Dinámica y técnicas de Ramificación son buenos para resolver problemas pequeños. Por otro lado, los métodos aproximados son más adecuados para problemas más grandes. Particularmente las metaheurísticas han mostrado dar soluciones de buena calidad a los problemas de optimización combinatoria.

Capítulo 4

Estado del arte

Hay una amplia gama de métodos para resolver el problema de ruteo de vehículos, que pueden ser clasificados de manera muy general en dos tipos: exactos y aproximados. Los métodos exactos buscan encontrar un óptimo global, mientras que los métodos aproximados, como su nombre lo indica, intentan encontrar una buena aproximación a la solución. Entre los métodos exactos se encuentran, por ejemplo, las técnicas de ramificación-acotamiento, y ramificación-corte. En el caso de los métodos aproximados, las técnicas más utilizadas son: metaheurísticas y algoritmos de búsqueda local.

En este capítulo se presenta una revisión de la literatura enfocada en la solución del VRP y del clásico problema del agente viajero. A continuación, en la primera sección de este capítulo, se presenta la revisión de algunos artículos en los que se utilizan métodos exactos para resolver el VRP y el TSP. En la Sección 4.2 se presentan varios trabajos enfocados en solucionar el VRP mediante métodos aproximados, específicamente, la sección se encuentra dividida según el tipo de técnica aplicada: en la Subsección 4.2.1 se utiliza ACO y en la Subsección 4.2.2 se utilizan algoritmos híbridos. Por último, hay un apartado exclusivo dedicado a la presentación de diversos trabajos que utilizan métodos exactos y ACO para solver el problema de ruteo aplicado a la recolección de residuos.

4.1 Métodos exactos

Como ya se mencionó, en el trabajo de Dantzig y Ramser (1959), se introdujo por primera vez el VRP. En éste se propuso una aproximación algorítmica para resolver el VRP en la distribución de gasolina entre varias estaciones de combustible.

A partir de dicha propuesta, se han desarrollado y mejorado distintas técnicas para resolver el VRP en diversas aplicaciones. A continuación, se presentan algunos trabajos en donde se propone el uso de los algoritmos exactos para la solución del VRP y algunas de sus variantes.

Christiansen y Lysgaard (2007) utilizan la técnica de ramificación y evaluación para resolver el CVRPSD. En este trabajo el problema maestro se obtiene a partir de la relajación entera de las variables; cambio en las restricciones del problema, entre otras. Después, ellos se basan en la técnica de descomposición de Dantzig-Wolfe¹ para generar los subproblemas que permitan identificar qué columnas o variables han de ser incluidas en el problema maestro. Una vez que se han generado estos subproblemas (problemas del camino más corto), son resueltos mediante programación dinámica. En cuanto a las características particulares de este problema de ruteo, las demandas no se conocen desde un inicio, lo que implica una desventaja debido a que se puede no satisfacer la demanda del cliente. Si el vehículo es incapaz de atender el cliente, entonces debe regresar al depósito a replanear la ruta. A este tipo de situación se le llama *falla*, la cual es incluida en la función objetivo como un costo adicional. Los resultados que se obtienen con el algoritmo propuesto son satisfactorios, sin embargo se ven sesgados ya que la demanda es modelada como una distribución Poisson.

Desrochers, Desrosiers y Solomon (1992) también utilizan el método de ramificación y evaluación para resolver un problema de ruteo con ventanas de tiempo. Y al igual que los autores del artículo presentado anteriormente, las columnas o variables a incluir en el problema maestro se determinan al resolver el subproblema mediante programación dinámica. A diferencia del trabajo anterior, en este trabajo el subproblema

¹El método de descomposición de Dantzig-Wolfe permite resolver problemas de programación lineal grandes, al transformarlos en subproblemas más pequeños y más fáciles de resolver.

es un problema del camino más corto con limitaciones de recursos. En este trabajo se menciona que aunque los resultados son buenos, el algoritmo no es competitivo para problemas a gran escala.

Baldacci y Mingozzi (2009) proponen un algoritmo que puede ser utilizado para resolver cinco variantes del VRP. Este algoritmo, que aplica técnicas de acotamiento a diferentes relajaciones del problema, provee mejores cotas inferiores en comparación con otros algoritmos exactos. Tales cotas se traducen en soluciones más cercanas al óptimo para problemas que, por su dimensión, son difíciles de resolver con métodos exactos. Cabe señalar, que aunque es un método exacto, los autores mencionan que son problemas intratables por su dimensión. En consecuencia, para algunos casos no agregan todas las restricciones, por lo que no pueden asegurar la optimalidad de las soluciones encontradas.

Por otra parte, Archetti, Bianchessi y Speranza (2014) utilizan dos algoritmos de ramificación y corte para resolver el problema de ruteo de vehículos con entrega dividida (SDVRP). Ambos algoritmos propuestos difieren en la formulación en la que se basan: uno utiliza una formulación de dos índices (las variables utilizadas para definir las rutas tienen dos subíndices), y el otro una formulación de flujo de una sola mercancía *Single Commodity*. Los autores concluyen que los mejores resultados se obtienen al utilizar un algoritmo basado en una formulación de dos índices, sin embargo no se alcanza la solución óptima para todos los problemas en los que hacen pruebas.

4.2 Métodos aproximados

En esta sección se presenta una revisión de los trabajos enfocados en resolver el VRP mediante métodos aproximados. Específicamente, se hace énfasis en aquéllos trabajos que usan ACO, pues es el algoritmo a utilizar para resolver el problema propuesto en esta tesis. Además, se incluyen métodos como recocido simulado (SA, por sus siglas en inglés), algoritmos genéticos (GA, por sus siglas en inglés) y búsqueda tabú (TS, por sus siglas en inglés) que, en combinación con ACO son utilizados para resolver el problema de ruteo.

4.2.1 Algoritmos ACO

Dorigo y Luca M Gambardella (1997) plantean resolver el TSP simétrico y asimétrico mediante el algoritmo sistema de colonia de hormigas (*Ant Colony System*, ACS) que integra una búsqueda local *3-opt*. Se propone, por primera vez, el uso de una lista de nodos candidatos para reducir el número de ciudades a las que se puede mover una hormiga dada su ubicación actual. De esta forma, se intenta mejorar la calidad de las soluciones y aumentar la rapidez del algoritmo. Experimentalmente se muestra que el algoritmo propuesto obtiene mejores resultados que un GA.

Varios trabajos, inspirados en la propuesta de Dorigo y Luca M Gambardella (1997), incluyen una lista de candidatos en el algoritmo ACS. Por ejemplo, para resolver el TSP, Hlaing y Khine (2011) consideran una lista de candidatos inamovible. Adicionalmente, proponen la implementación de una regla dinámica para actualizar la feromona. Esta estrategia consiste en usar la entropía para ajustar el valor del parámetro que determina la influencia de la información heurística. Con este par de modificaciones se mejora el funcionamiento del algoritmo, así como la rapidez de convergencia.

Asimismo, Bell y McMullen (2004) incorporan una lista de candidatos al algoritmo ACS para resolver el CVRP. Pero más interesante aún, presentan el uso de múltiples colonias de hormigas, de manera que el rastro de feromona dejado por una primer colonia será ignorado por las demás. Los autores, al analizar el impacto de las modificaciones, sugieren que su uso ayudaría a mejorar la capacidad de los algoritmos de hormigas para solucionar problemas a gran escala.

C. H. Chen y Ting, 2006 presentan el algoritmo IACS como una mejora al ACS, para resolver el CVRP. Las modificaciones que presenta este algoritmo son: (1) el uso de una búsqueda local *2-opt* aplicada a las S mejores hormigas, (2) la reinicialización de los niveles de feromona y (3) el reforzamiento de la feromona combinando la mejor solución de la iteración con la mejor global. Los resultados muestran que este algoritmo compite con SA, TS yGAs, sin embargo el tiempo de ejecución es mayor.

En el trabajo de Yu, Yang y Yao (2009) se presenta el algoritmo IACO, en el cual se implementa un operador de muta y una estrategia llamada *ant-weight*. Con estas modificaciones se pretende explorar distintas soluciones y actualizar la feromona en función de la calidad de las soluciones obtenidas por las hormigas. El objetivo es minimizar la distancia en un CVRP. Se muestra experimentalmente que el operador de muta produce mejores soluciones que la estrategia *ant-weight*. No obstante, al implementar ambas técnicas se obtienen mejores soluciones y se reduce el tiempo de ejecución que si se aplican por separado.

Wang, Y. P. en 2013 propone un algoritmo ACS que integra la búsqueda local Pareto, con el objetivo de mejorar las soluciones encontradas. Este algoritmo adaptativo, llamado AAACA, es usado para resolver los problemas VRP y CVRP con flota heterogénea. Además, el objetivo es minimizar el tiempo de recorrido más el tiempo de servicio.

Diferentes variantes del VRP se han desarrollado con la finalidad de hacer una mejor representación de problemas reales. Por ejemplo, Arias-Rojas, Jiménez y Montoya-Torres (2013) tratan un problema inspirado en el ruteo de un autobús escolar. En el artículo, se propone descomponer el VRP en subproblemas TSP, que son resueltos usando un algoritmo de hormigas. Para generar los subproblemas se agrupan las paradas en rutas y se les asigna un vehículo, de tal manera que no se viole la restricción de capacidad del autobús. Los resultados indican una reducción en la distancia recorrida por los autobuses, en comparación con el modelo empírico utilizado anteriormente.

Otros autores han presentado su trabajo en el estudio y solución del DVRP que, como ya se explicó, se caracteriza por la llegada de nuevas órdenes mientras los vehículos se encuentran en ruta, lo que implica una replaneación de la asignación inicial de las rutas. Por ejemplo Montemanni, Gambardella, Rizzoli y Donati (2005), proponen que los vehículos, una vez que han salido del depósito, no tengan que replanear su ruta. Esta propuesta se basa en plantear el DVRP como una sucesión de VRP estáticos. Tal sucesión se genera al dividir el día en rebanadas de tiempo, a las que se les asocia un VRP con flota heterogénea y que considera solamente órdenes conocidas al inicio de la ruta. Con dicha división, se evita que los vehículos que están fuera del depósito tengan que modificar su ruta inicial, y más aún, se evita un aumento en el costo. El problema es resuelto mediante un algoritmo ACS y el algoritmo es desarrollado de tal forma, que entre rebanadas de tiempo se conserve la feromona. Esto, con el fin de pasar información sobre buenas soluciones. El algoritmo es más eficiente al ser comparado con un GRASP (*Greedy Randomized Adaptive Search Procedures*).

En el trabajo de Fuellerer, Doerner, Hartl e Iori (2009) se estudia el problema de ruteo de vehículos de carga en dos dimensiones (2L-CVRP, *Two-dimensional Loading Vehicle Routing Problem*). En el 2L-CVRP algunos vehículos se caracterizan por tener una superficie en la cual se debe acomodar el cargamento. Este trabajo propone por primera vez considerar orientación variable de la carga. El algoritmo propuesto para resolver dicho problema se basa en el ACO, desarrollado por Reimann, Doerner y Hartl (2004) para un CVRP. Este problema tiene la peculiaridad de situar la carga en el vehículo de forma restrictiva, por ejemplo: la carga no puede colocarse una encima de otra o, al hacer la entrega a un cliente, la mercancía de los clientes sucesivos no pueden ser movida. Por tanto, se debe verificar la factibilidad de las rutas, es decir, dada una sucesión de clientes asociada a una ruta se debe comprobar que las mercancías puedan ser colocadas en la superficie del vehículo respetando las restricciones de carga. Para probar la factibilidad, se utilizan diversos métodos: el procedimiento de cotas inferiores de Martello y Vigo (1998), las heurísticas *Bottom-left fill* y *Toucher Perimeter Algorithm*, una búsqueda local que modifica las posiciones de dos ítems y, por último, si ninguno de los métodos descritos funciona, un algoritmo de ramificación y acotamiento. Además, al aplicar el ACO se pueden usar más vehículos de los permitidos, pero con una penalización, y en las rutas asociadas a esos vehículos no se actualizará la feromona.

Gajpal y Abad (2009) propusieron una metodología basada en ACO, para resolver dos variantes del VRP: el problema de ruteo de vehículos con entrega y colecta simultánea (VRPSDP) y el problema de ruteo de vehículos de ida y vuelta (VRPB). En su investigación utilizan dos técnicas de búsqueda local multiruta: intercambio e inserción de clientes y el intercambio de subrutas. Además, cuando una hormiga encuentra una solución no factible regresa al depósito y al construir la siguiente ruta, el primer nodo visitado es aquél al que no pudo ir en la ruta anterior. El algoritmo es probado en tres tipos de problemas: el primero es un caso real aplicado a la distribución de libros desde una librería central a sus sucursales, el segundo y tercero son problemas *benchmark*. Los resultados muestran que se obtiene la solución óptima exacta para el primer problema. Mejores soluciones se obtienen para el segundo y tercer caso, al comparar ACO con LNS (*Large Neighborhood Search*), además se encuentran nuevas soluciones en cada caso.

Luca M Gambardella, Taillard y Agazzi (1999) propusieron el algoritmo MACS-VRPTW que, hasta ahora, ha servido de base para resolver el VRPTW. La idea de este algoritmo es usar dos colonias independientes de hormigas, la primera (ACS-VEI) minimiza el número de vehículos, y la segunda (ACS-TIME), busca minimizar el tiempo de recorrido. Las dos colonias trabajan en paralelo, y cuando la ACS-VEI encuentra una solución con un vehículo menos, ambas se reinician y la ACS-TIME usa esa información para lograr su objetivo. La diferencia entre ambas colonias se refiere a la información heurística que utilizan: la colonia ACS-VEI involucra la urgencia de atender a un cliente debido a que la ventana de tiempo está cerca; por otro lado la colonia ACS-TIME solamente toma en cuenta las distancias entre los nodos.

Quizá, el caso más simple dentro de este conjunto de problemas de ruteo de vehículos, es el agente viajero temporal con ventanas de tiempo. En el trabajo de Favaretto, Moretti y Pellegrini (2004), se desarrolla un algoritmo basado en colonia de hormigas, con el objetivo de minimizar el tiempo de recorrido. Dado que en este se considera un sólo vehículo, no es necesario el uso de dos colonias de hormigas, sin embargo, la información heurística considerada es muy parecida a la de la colonia ACS-VEI.

Pellegrini, Favaretto y Moretti (2007) también plantean un problema multiobjetivo VRPTW con las siguientes características: múltiples visitas, flota de vehículos heterogénea, múltiples ventanas de tiempo, y máxima duración de las subrutas; características que permiten una mejor representación de situaciones reales. Para resolver el problema, se presentan dos algoritmos: el primero llamado *Multiple Ant Colony System* (M-ACS) el cual se basa en el algoritmo ACS y el segundo llamado *Multiple MAX-MIN Ant System* (M-MMAS) que tiene como base el algoritmo MMAS. Los algoritmos usan dos colonias de hormigas, al igual que Luca M Gambardella *et al.* (1999) pero, a diferencia de ellos, la implementación en este estudio es secuencial. Por otra parte, con el fin de considerar las múltiples ventanas, la regla de transición es modificada, ya que incorpora un nuevo parámetro que considera el número de ventanas de tiempo. Los algoritmos M-ACS y M-MMAS, al ser comparados con búsqueda tabú (TS) y vecino más cercano aleatorizado (RNN), obtienen mejores resultados. Estas mismas autoras, en otro trabajo, muestran experimentalmente que, al usar un número de hormigas mayor al número de nodos, se da un reforzamiento en el efecto de la actualización local de la feromona, lo que permite salir de óptimos locales (Favaretto, Moretti & Pellegrini, 2007).

Rizzoli *et al.* (2004) también han hecho uso de dos colonias de hormigas para resolver problemas de ruteo de vehículos. El trabajo de Rizzoli *et al.* es aplicado a dos casos reales: distribución de productos de una cadena de supermercados en Suiza y transporte de productos desde una fábrica hasta las tiendas de inventario en Italia. Los problemas son planteados como un VRPTW y una modificación del mismo que incluye recogida y entrega simultánea, ambos resueltos con una modificación al algoritmo MACS-VRPTW, al que llaman ANTRROUTE. Para el VRPTW, hay algunos supermercados a los que sólo ciertos vehículos pueden ingresar, por lo que al inicio de la ruta, cada hormiga escoge el tipo de vehículo; además se incorporó un costo de espera con el fin de que los vehículos no lleguen muy temprano a las tiendas. Por otro lado, al resolver el segundo problema, se dispone solamente de un vehículo, así que se remueve la colonia ACS-VEI del MACS-VRPTW. Las soluciones son comparadas con el método implementado en cada empresa y en ambos casos se reduce la distancia recorrida.

Otro par de trabajos que aplican el algoritmo de hormigas para resolver el VRPTW se han revisado. El

primero de Tan, Zhuo y Zhang (2006) en el que también usan dos colonias, sin embargo, para mejorar las soluciones usan la heurística de inserción secuencial de clientes. Los autores mencionan que una ruta generada por la colonia ACS-VEI podría excluir a algunos clientes, pero con la heurística mencionada se añadirían clientes a la ruta.

El segundo trabajo es propuesto por Katiyar, V. (2013), quienes implementan el algoritmo EN-ACS (*Enhanced Ant Colony System*) con el objetivo de minimizar la distancia solamente y no el número de vehículos usados. Además, se propone una nueva regla de actualización de feromona que considera sólo la mejor solución y la cantidad depositada es limitada por una cota superior e inferior.

Hasta ahora, la función objetivo ha estado relacionada a minimizar ya sea la distancia, el tiempo o costos generados por la flota de vehículos como un conjunto. Sin embargo, el objetivo también puede ser minimizar la máxima distancia recorrida por cada vehículo, lo que deriva en el uso de todos los vehículos con el fin de reducir la distancia recorrida de aquéllos con rutas más grandes. Un ejemplo de este problema de ruteo es el *min-max Multi-Depot Vehicle Routing Problem*, planteado por Venkata Narasimha, Kivelevitch, Sharma y Kumar (2013). Estos autores buscan resolver este problema mediante una extensión del algoritmo ACO propuesto por Bullnheimer, Hartl y Strauss (1999), al descomponer el problema en múltiples VRP. Además, para mejorar las soluciones usan una búsqueda local *2-opt* e integran una lista candidata que, como ya ha sido mencionado, supone una reducción en el tiempo de solución. Para demostrar la eficiencia del algoritmo, se comparan los resultados obtenidos por colonia de hormigas y por programación lineal (PL). Diferentes números de depósitos, ciudades y vehículos son usados en los experimentos, pero en promedio el algoritmo propuesto mejora las soluciones del PL en un 145.59%.

4.2.2 Algoritmos Híbridos

A partir de los algoritmos mencionados y otros que han sido estudiados, se han creado los algoritmos híbridos. Los algoritmos híbridos combinan métodos exactos, heurísticas y metaheurísticas, con el objetivo de mejorar las soluciones al aprovechar las mejores características de cada algoritmo. En esta sección se hará una breve revisión de algunos algoritmos híbridos que han sido aplicados al problema del agente viajero y al problema de ruteo de vehículos.

Dentro de las aplicaciones al TSP se encuentra el trabajo de Neissi y Mazloom (2009), ellos presentan una combinación de un método de búsqueda local *2-opt* con algoritmos genéticos. Los autores utilizan tres operadores de cruce y dos de muta, con el fin de determinar la mejor combinación de estos operadores, compuesta por un elemento de cada tipo que resuelva el problema. Además, observan que al incluir una búsqueda local *3-opt*, se alcanzan mejores soluciones que con una *2-opt*, sin embargo, también se consume más tiempo.

Gharehchopogh, Maleki y Farahmandian (2012) introducen, implementan y discuten una solución algorítmica híbrida para resolver el TSP, que se basa en algoritmos genéticos y optimización por colonia de hormigas. El objetivo de usar ambas metaheurísticas es evitar la rápida convergencia, que derive en estancamiento en óptimos locales. Este algoritmo se resuelve en primera instancia utilizando ACO y las soluciones que se generan sirven como población inicial para el algoritmo genético. Los autores muestran que la solución alcanzada por su algoritmo híbrido mejora la solución obtenida con ACO y GA.

Por otro lado X. Chen, Tan, Yang y Cheng (2012) exponen un algoritmo híbrido que, además de combinar colonia de hormigas y algoritmos genéticos, incluye recocido simulado. En el artículo se explica que, al combinar estos algoritmos, la convergencia a la solución será más rápida y se evitará el estancamiento. Los resultados obtenidos con el algoritmo híbrido se compararon con aquellos que se obtuvieron al aplicar por separado cada uno de los algoritmos básicos (SA, ACO y GA) y en todos los casos se reduce la distancia mínima recorrida.

En los problemas de ruteo existe la posibilidad de que una hormiga llegue a un callejón sin salida, es decir, que todos los nodos adyacentes ya los haya visitado. Yoshikawa y Otani (2010) desarrollan un sistema híbrido que combina búsqueda tabú con un algoritmo ACS para escapar de callejones sin salida en problemas de ruta más corta. El algoritmo propuesto es más eficiente al ser comparado con el de Dijkstra.

4.3 Métodos de Solución Aplicados al Problema de Recolección de Residuos

En los últimos años varios trabajos se han centrado en el área de manejo de residuos, desde el punto de vista ecológico hasta el punto de vista económico. Este manejo de residuos incluye diferentes etapas, como lo son: proceso, transporte, reciclaje y recolección, siendo esta última la de mayor costo (Beliën, De Boeck & Van Ackere, 2012). En esta sección, sólo se presentan algunos artículos enfocados en la recolección de residuos. Muchos de estos trabajos buscan optimizar el sistema de recolección existente en diversas zonas urbanas, cuyos objetivos principales son la reducción de transporte y de costos.

En la recolección de residuos, se pueden distinguir dos categorías: por contenedores y por aceras. Principalmente el VRP o TSP (si se trata de un sólo vehículo) están asociados al diseño de rutas de recolección por contenedores, en cambio si la recolección es por acera, es más adecuado tratarlo como un ARP (*Arc Routing Problem*) o un CPP (*Chinese Postman Problem*). En otras palabras, el problema puede ser modelado como un problema de ruteo de nodos o de arcos. Por ejemplo, Filipiak, Abdel-Malek, Hsieh y Meegoda (2009) plantean el problema de recolección como un problema de ruteo de arcos. Los autores utilizan el algoritmo del cartero chino para proponer rutas de recolección óptimas en una región de Nueva Jersey, Estados Unidos. Por otra parte Kulcar, T. (1996) representa el problema de recolección en Bruselas como un CARP con múltiples depósitos. El objetivo es minimizar los costos de transportación y, para resolverlo, utilizan el método de ramificación y acotamiento.

El objetivo de reducir los costos, puede verse como una reducción en la distancia, el tiempo o decidir que días se recogen los residuos, pues un menor número de días supone un menor costo. Tal es el caso de Mansini y Speranza (1998), quienes plantean un modelo de programación lineal con el objetivo de minimizar la máxima cantidad recolectada en un día y con esto reducir la frecuencia de recolección; con la restricción de que unos días se recoge basura orgánica y otros días inorgánica.

Otros autores también han usado programación lineal para abordar el problema de recolección de residuos. Por ejemplo, Bonomo, Durán, Larumbe y Marengo (2012) presentan un modelo para diseñar las rutas de recolección de contenedor en contenedor en una zona sur de Buenos Aires. Dicha zona es dividida en cuatro subzonas y cada una queda asociada a un problema del agente viajero. La singular función objetivo de este modelo reduce el *trabajo*² ejercido por cada uno de los vehículos, así como la distancia recorrida. Para minimizar el *trabajo*, se seleccionan las mejores soluciones obtenidas con PL y de éstas, se elige la de menor trabajo. Con el método desarrollado hay una reducción de hasta 40 % en las distancias recorridas, además se realizan diferentes experimentos para asignar más vehículos a cada subzona y con esto disminuir aún más, las distancias.

Otra técnica importante para la tarea de recolección de residuos es el uso de Sistemas de Información Geográfica (SIG) los cuales, mediante la representación espacial, han permitido la integración de parámetros como: mapas, posición de contenedores, tráfico en las calles e incluso la densidad del área estudiada. Esto conlleva a una representación más real de los problemas y, por tanto, ofrece soluciones muy eficientes. Tal es el caso de Marković, Janošević, Jovanović y Nikolić (2010), quienes usan un SIG con el fin de determinar

²El concepto de trabajo es tomado de la Física y es calculado como el producto de la distancia recorrida y la fuerza aplicada.

las distancias entre contenedores, incluyendo el depósito. Los autores plantean el problema como un CVRP con un depósito y un solo vehículo. Para resolver el problema se utiliza el algoritmo de ahorros de Clarke y Wright (1964) y es aplicado en la ciudad de Niš, Serbia.

Hemmelmayr, Doerner, Hartl y Rath (2013) desarrollan un método basado en búsqueda de entorno variable y programación dinámica para resolver tres problemas que consideran instalaciones intermedias (IF-VRP). Las instalaciones intermedias son locaciones en las que se puede o debe descargar antes de regresar al depósito. El primer y segundo problema son periódicos, es decir que no todos los días deben ir a las mismas instalaciones intermedias, la diferencia entre ambos es que en el primero los vehículos deben regresar vacíos al depósito y en el otro no es necesario que regresen vacíos. Además, las instalaciones intermedias tienen capacidad limitada. El tercer problema es un VRP con multidepósitos y con rutas interdepósitos. En este último, sólo se considera una recolección de desechos por día con la flexibilidad de regresar con una carga parcial al término del recorrido.

Mourao y Almeida (2000) también usan heurísticas para la optimización de recolección de desechos en Lisboa, Portugal. El problema es planteado como un CARP donde utilizan el método: asignar primero, rutear después. Este método tiene dos fases: en la primera, se crean grupos de clientes y, en la segunda, se crea una ruta para cada grupo de clientes.

Por otra parte, Ogwueleka, T. C. (2009) propone un método eficiente de recolección en Onitsha, una región de Nigeria, debido a la deficiencia del método empírico que se utilizaba en la zona para la recolección de residuos. El problema también es modelado como un CARP, y es resuelto mediante la heurística rutear primero, asignar después. Esta heurística tiene dos fases: la primera consiste en diseñar una ruta que pase por todos los nodos, en la segunda se particiona dicha ruta, de tal manera que cada vehículo sea asignado a una subruta. Los resultados obtenidos indican una disminución en la distancia recorrida, un ahorro monetario y, además, una reducción del número de vehículos usados.

Katkar, A. (2012) implementa la técnica de árbol de expansión mínima para resolver el problema de recolección de desechos en un área de la ciudad de Kolpapur, India. El planteamiento del problema incluye dos tipos de nodos: los que representan la posición de los contenedores y los que representan las áreas donde la recolección se hace de puerta en puerta (debido a la restricción del acceso del vehículo). La técnica es aplicada primero para minimizar la distancia en las áreas donde no puede transitar el vehículo y después para minimizar la distancia entre nodos. Los resultados que se obtienen muestran que, usando esta propuesta, se mejorará el sistema actual de recolección.

Nikolaos Karadimas ha realizado diversas investigaciones enfocadas en la recolección de residuos, en las que se hace uso del Sistema de Información Geográfica (SIG), que permite hacer un análisis de escenarios. Este autor señala lo importante que es un proceso de recolección eficiente, pues económicamente puede derivar en ahorros monetarios. Varios de estos trabajos han sido aplicados en un municipio de Atenas, Grecia y en ellos el problema es reducido a subproblemas TSP, los cuales son generados al particionar el escenario. Al hacer la partición, se quita la restricción de capacidad del vehículo, pues cada subárea es atendida por un solo vehículo cuyo volumen es suficiente para realizar la recolección. Para resolver cada TSP se ha propuesto el uso del algoritmo Ant Cycle (Karadimas, Kouzas, Anagnostopoulos & Loumos, 2005) y el ACS (Karadimas, Papatzelou & Loumos, 2007). Al comparar ambos algoritmos con el sistema empírico utilizado en el municipio se reduce la distancia recorrida.

En otro trabajo Karadimas, Doukas, Kolokathi y Defteraïou (2008) también usan un algoritmo ACS para optimizar la recolección de residuos en el municipio de Atenas y, adicionalmente, usan un *software* llamado *ArcGIS Network Analyst*. Los resultados obtenidos con el ACS son mejores en comparación con aquellos obtenidos por el *software*. Sin embargo, ambos métodos son más eficientes que el implementado en Atenas, mencionado en el párrafo anterior. Una desventaja que presenta esta técnica gira en torno al tiempo computacional generado al aplicar ACS, dado que éste es muy extenso. Por ello, los autores sugieren el uso de una heurística que mejore la solución inicial, buscando una reducción del tiempo.

Hornig y Fuentealba (2009) aplican un algoritmo ACO a la recolección de residuos por contenedores en un municipio de Chile. El objetivo del problema es minimizar la distancia entre secuencias parciales.³ La distancia total recorrida es calculada como la suma de las distancias de las secuencias parciales más la unión de las secuencias. Con el algoritmo propuesto hay una disminución en la distancia recorrida, con respecto del método utilizado en el municipio.

Debido a la complejidad de la representación real del problema de recolección de residuos, se han desarrollado modelos que integren restricciones que representen mejor el problema. Por ejemplo, se han propuesto métodos basados en el algoritmo AS que consideran restricciones para evitar vueltas prohibidas y penalizaciones derivadas del tiempo perdido en semáforos (Bautista & Pereira, 2004) y (Bautista *et al.*, 2008). En el primer trabajo, el problema es modelado como un CARP, sin embargo es transformado en un CVRP, con el fin de tomar en cuenta las vueltas no permitidas. Ellos proponen dos algoritmos basados en el AS, que difieren en el método constructivo que usan, uno se basa en el vecino más cercano, NN (*Nearest Neighbor*) y el otro en inserción más cercana, NI (*Nearest Insertion*). Los modelos propuestos son aplicados a un municipio de Barcelona: Sant Boi de Llobregat y al ser comparados con el antiguo método aplicado en esa región, se encontró una disminución favorable en la distancia recorrida.

Es importante mencionar que en este tipo de problemas, no todos los datos se conocen con certeza, de hecho la cantidad de residuos recolectada no siempre es la misma. Por esta razón es importante considerar elementos estocásticos en la formulación de los problemas de recolección de residuos. Tal es el caso de Ismail y Loh (2009) quienes plantean un VRPSD, que es resuelto mediante el algoritmo ACS. El algoritmo propuesto no es probado con datos reales, sin embargo lo prueban con un problema hipotético que considera 50 clientes.

Por último, es importante mencionar el trabajo de Enríquez (2015), en el que se plantea optimizar las rutas de recolección en Ciudad Universitaria, UNAM. En dicho trabajo se utiliza el software TransCAD, cuya técnica para encontrar un conjunto de rutas de recolección es desconocido, sin embargo, logran disminuir la distancia recorrida con el modelo empírico.

4.4 Resumen

En este capítulo se presentó parte de la bibliografía consultada que se relaciona con el problema a resolver (recolección de residuos), con el problema de ruteo de vehículos y con optimización por colonia de hormigas.

En cuanto al problema de ruteo, son diversas las técnicas usadas para resolverlo. No obstante, para problemas a gran escala las metaheurísticas han dado resultados satisfactorios. Tal es el caso de los algoritmos ACO, que como se ha podido observar a través de esta revisión, han sido modificados con el objetivo de reducir la brecha entre las soluciones obtenidas y las óptimas. Por otro lado, se puede inferir a partir de los trabajos revisados, que los algoritmos ACO encuentran mejores soluciones al problema del VRP en comparación con algoritmos exactos, heurísticas, algoritmos genéticos, búsqueda tabú y recocido simulado. Sin embargo, para atacar las desventajas de los algoritmos de hormigas, combinarlos con otras técnicas como las mencionadas anteriormente, es casi necesario.

En materia de recolección de residuos, en la mayoría de los artículos el algoritmo que se propone es aplicado a zonas urbanas, pues es importante observar su efecto en situaciones reales, donde la reducción del costo visto como distancia, tiempo, etc. beneficia la economía y sobre todo, el bienestar social. Adicionalmente, notamos que para este problema varios autores aplican los SIG, con el fin de que los planteamientos se apeguen más a la realidad, ya que con esta información es posible un mejor diseño de rutas.

³Ver Lazo, P. A., *Aplicación de un Procedimiento de Ruteo para la Recolección de Contenedores de Residuos Domiciliarios en San Pedro de la Paz*, 2005.

Capítulo 5

Caso de Estudio

En este Capítulo se describe el manejo actual de residuos sólidos urbanos (RSU) en Ciudad Universitaria, el cual comprende las siguientes etapas: generación, almacenamiento, recolección, transferencia y disposición final; siendo la tercera etapa en la que se enfoca este trabajo. Después, en la Sección 5.2, se ahonda en el sistema de recolección de RSU que actualmente se encuentra implementado en Ciudad Universitaria (CU) y que es el caso de estudio de esta investigación. Asimismo, se presenta el diseño de los circuitos que hoy en día se siguen para realizar la recolección.

5.1 Sistema de Manejo de Residuos Sólidos en CU

En el Distrito Federal, de acuerdo al Inventario de Residuos sólidos del Distrito Federal del 2013, se estima que diariamente se generan 12,816 toneladas de residuos. Inherentes a la generación de tal cantidad de residuos, están los problemas de sanidad, como transmisión de enfermedades infecciosas y proliferación de fauna nociva; así como contaminación de suelos, aire y agua. De hecho, en el año 2011 el relleno sanitario Poniente, lugar a donde eran llevados todos los residuos generados en el Distrito Federal, fue cerrado debido a que su capacidad fue rebasada. Por tanto, ha sido necesario tomar medidas, como la publicación en el 2013 de la Ley General para la Prevención y Gestión Integral de los Residuos, en la cual uno de los objetivos es la separación en residuos orgánicos y residuos inorgánicos.

Ciudad Universitaria, se localiza en la Delegación Coyoacán y con sus 730 hectáreas de superficie, representa 13.4% del área total de la delegación. Esta institución, aún siendo uno de los centros educativos más importantes del país, no es ajena a la problemática que supone el manejo de RSU, pues es la misma institución la encargada de coordinar el manejo de RSU generados en tal espacio. Y es a partir de la publicación de la Ley en el 2013 que en la UNAM se creó un nuevo programa para mejorar el manejo de RSU y así aprovechar al máximo los residuos que de alguna forma podrían ser reutilizables.

De las etapas mencionadas involucradas en el manejo de residuos sólidos la disposición final no es atendida por la UNAM, siendo el Sistema de Manejo de Residuos del Distrito Federal quien se encarga de la disposición final. A continuación, se describirá el manejo de residuos sólidos en Ciudad Universitaria de acuerdo a las etapas mencionadas. Dicha descripción se hace con el fin de contextualizar la recolección de residuos sólidos, que es la etapa que motiva este trabajo. La información que se presenta se obtuvo de: entrevistas con personal de la Dirección General de Obras y Conservación (DGOyC), personal que opera los vehículos recolectores y se encarga directamente de la recolección, manuales del sistema de RSU publicados por el Programa Universitario de Medio Ambiente (ECOPUMA) y del trabajo de Enríquez (2015).

5.1.1 Generación

Ciudad Universitaria, conocida como CU, tiene una población superior a las 150,000 personas y se estima que diariamente se generan 15 toneladas de basura. De manera que la generación de residuos promedio por persona es de 0.1 kg diarios.¹

¹Información proporcionada por la Dirección General de Obras y Conservación.

5.1.2 Almacenamiento de Residuos

Dentro de la mayoría de las facultades, institutos y demás dependencias de CU, la comunidad universitaria deposita los residuos en campanas cuya capacidad es de $0.1 m^3$, diferenciadas por residuos orgánicos y residuos inorgánicos. Sin embargo, este sistema busca ser reemplazado por un sistema de separación de RSU en cinco grupos: orgánicos, botellas de plástico, latas/vidrio/tetrapak, papel/cartón y otros.² Este nuevo sistema, al momento de las entrevistas con el personal de la DGOyC, solamente era realizado en la Facultad de Ciencias y en Veterinaria.

Ahora bien, no importa si la separación se hace en dos o cinco grupos, en ambos casos, el personal de limpieza de cada edificio es responsable de llevar los residuos a las artesas, lugares construidos y destinados para el almacenamiento temporal de los residuos y que están ubicados generalmente en los estacionamientos. En las artesas se ubican unos contenedores más grandes, los cuales cuentan con una capacidad de $0.8 m^3$ y son de color verde y gris: los primeros dedicados al almacenamiento de residuos orgánicos y los segundos para el almacenamiento de residuos inorgánicos y no reutilizables.

Cabe aclarar, que el número de contenedores difiere según el punto de recolección. Lo anterior se debe a que la cantidad de residuos generados en cada punto está en función del número de personas que se concentren en cada dependencia.

5.1.3 Recolección

De acuerdo a la Secretaría de Desarrollo Social (SEDESOL), en México los costos de recolección oscilan entre 70 y 85 % del costo total del manejo de residuos urbanos.³ Lo cual, por una parte, indica que la recolección es la etapa más cara dentro del manejo de RSU, pero por otra, también representa una ventana de oportunidad para la reducción de costos, al buscar un sistema de recolección funcional y más eficiente.

Los métodos de recolección varían dependiendo de las necesidades y del entorno, se tienen por ejemplo: *método de parada fija*, *método de acera*, *intradomiciliario* y *método de contenedores*. El *método de parada fija* consiste en que un camión se estaciona en cierto punto en un hora predeterminada, y así, los usuarios se dirigen al vehículo a depositar los residuos. El *método de acera* consiste en que los trabajadores recogen los recipientes con residuos sólidos, los cuales fueron previamente depositados por los usuarios en la acera. El *método intradomiciliario* es parecido al anterior, la diferencia se encuentra en que los operadores del vehículo ingresan a las viviendas a coleccionar los residuos. El último método, llamado por *contenedores*, es similar al primero, pues el vehículo debe estacionarse en ciertos puntos; sin embargo, bajo este sistema sólo se recogen los residuos previamente depositados en el contenedor. Este último método es el utilizado en Ciudad Universitaria.

En CU, la Dirección General de Obras y Conservación es la dependencia encargada de coordinar la recolección de residuos sólidos.⁴ En total, se recolectan alrededor de 15 toneladas de residuos cada día y dicha recolección es llevada a cabo de manera manual y mecánica. La colecta manual sólo se realiza en el casco central, en el resto de la universidad la colecta es mecánica. Para llevar a cabo la recolección mecánica se cuenta con 8 vehículos recolectores y un camión de volteo, cuya capacidad es de $15.3 m^3$ y $6.0 m^3$, respectivamente. Sin embargo, solamente se utilizan cinco vehículos recolectores, los cuales están asignados a cinco rutas diseñadas de manera empírica; el vehículo de volteo no es utilizado para la recolección de RSU.

²Actualmente ECOPUMA, se encuentra en contacto con las dependencias en CU, para poder aplicar exitosamente dicha separación de residuos.

³Consulta 29 de noviembre 2015 <http://www.arpet.org/docs/Mexico-Manual-de-gestion-integral-de-residuos-solidos.pdf>

⁴La recolección de residuos orgánicos es tratada aparte y llevada al centro de compostaje; los residuos peligrosos, son recolectados por una empresa externa

Cada vehículo recolector realiza generalmente un viaje a la Estación de transferencia de Coyoacán por la mañana y otro por la tarde. El horario para realizar la recolección es desde las 6:00 a.m hasta las 5:00 p.m de lunes a viernes y los sábados solamente por la mañana.

5.1.4 Transferencia de Residuos

Como ya se mencionó, los residuos generados en CU son llevados a la Estación de transferencia de Coyoacán, la cual tiene un horario de servicio de las 6:00 a las 22:00.⁵ Una vez que los camiones han descargado en la estación, regresan a CU. En caso de no haber recolectado los residuos sólidos en su totalidad, se retoman las rutas y se recolecta lo faltante, sin embargo, ya no son llevados nuevamente a la Estación de transferencia, pues las cantidades son pequeñas.

5.1.5 Selección y Disposición Final

Esta etapa está totalmente a cargo de la Dirección General de Servicios Urbanos de la Secretaría de Obras y Servicios del Distrito Federal. La selección de residuos consiste en recuperar material que pueda ser reutilizado (vidrio, cartón, papel, plástico, entre otros), el material que no es recuperado es transportado a sitios de Disposición final.⁶ Estos sitios, también conocidos como rellenos sanitarios, sirven como depósito de los residuos que ya no pueden ser útiles de ninguna manera, por lo que es importante que a estos lugares sólo lleguen residuos que ya no sean aprovechables.⁷

5.2 Sistema de Recolección en CU

Como ya se mencionó, la recolección de RSU en CU es cubierta por cinco rutas, las cuales son: Circuito CCU, Circuito Escolar, Circuito de la Investigación Científica, Circuito Exterior y Circuito Deportivo. Cada ruta es iniciada en los Talleres Centrales de Conservación o en los Talleres de Mantenimiento del Centro Cultural Universitario, lugares que sirven como depósito para los vehículos recolectores.

Ahora bien, es importante mencionar que, por políticas previamente establecidas, CU ha sido dividida en dos zonas: la primera llamada Zona Cultural, la cual tiene asignados los puntos de recolección del Circuito CCU y la segunda zona (que llamaremos Zona Escolar) los puntos de los circuitos restantes. En el caso de los vehículos que atienden la Zona Cultural, sólo pueden partir y regresar al Taller de Mantenimiento del Centro Cultural y para los vehículos que atienden la Zona Escolar, sólo pueden salir y regresar a los Talleres Centrales de Conservación. En la Figura 5.1 se observan las rutas ya establecidas, así como el conjunto de puntos de recolección asociados a cada ruta y los depósitos donde se inicia cada recorrido.

Adicionalmente, en el Anexo A, se muestra un mapa por cada ruta con el conjunto de puntos de recolección atendido por cada circuito, así como el orden en que se visitan los mismos. Este orden es ascendente, empezando por el Depósito (o Taller) donde se resguardan los vehículos, después pasan al punto uno y así consecutivamente hasta recorrer todos los puntos. Cuando han visitado todos los puntos de recolección, se dirigen a la Estación de transferencia a depositar la carga. Por otra parte, en la Tabla 5.1 se presentan las características generales (distancia total recorrida, número de puntos de recolección y volumen total) de cada una de las cinco rutas.

⁵Consulta 29 de noviembre 2015 http://www.consejeria.df.gob.mx/portal_old/uploads/gacetitas/4c8afe2c2a738.pdf

⁶Consulta 29 de noviembre 2015 <http://www.obras.df.gob.mx/recoleccion-transferencia-seleccion-y-disposicion-final/>

⁷Consulta 29 de noviembre 2015 <http://www.sedema.df.gob.mx/sedema/images/archivos/temas-ambientales/programas-generales/residuos-solidos/inventario-residuos-solidos-2013/capitulo-2.pdf>

La información relacionada con las rutas de recolección (mapas y orden de recolección) fue obtenida gracias a la DGOyC. Sin embargo, el volumen de residuos generado en cada punto de recolección no fue proporcionado por tal dependencia y se obtuvo del trabajo de Enríquez, 2015. En dicho trabajo, también se tienen mapas relacionados a las rutas de recolección, no obstante, hay diferencias con los que aquí se muestran. Estas diferencias probablemente se deban a la fuente que proporcionó los datos, sin embargo como ya ha sido mencionado, en este trabajo se utilizan los mapas que fueron facilitados por la DGOyC, dependencia encargada de la recolección de residuos sólidos en CU.

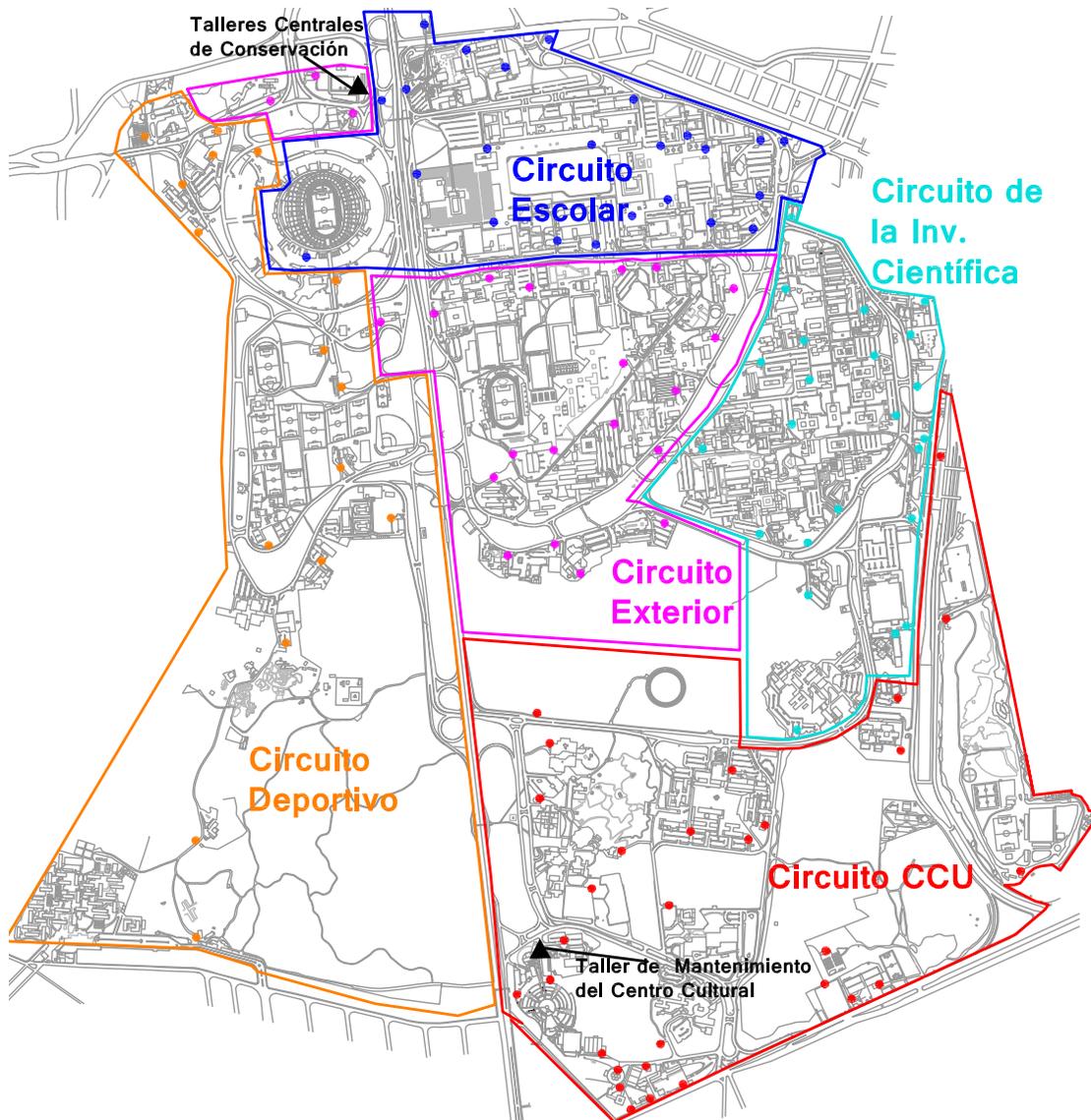


Figura 5.1: Área cubierta por cada una de las cinco rutas en Ciudad Universitaria y puntos de recolección asociados a cada ruta

Tabla 5.1: Características de los circuitos de recolección

Circuito	Número de puntos de recolección	Longitud de la Ruta	Volumen
C. Deportivo	16	43, 407 mts	11.3 m ³
C. Exterior	22 ⁸	38, 698 mts	16.3 m ³
C. de la Investigación Científica	23	32, 081 mts	14.3 m ³
C. Escolar	24	41, 808 mts	18.4 m ³
CCU	30 ⁹	34 134 mts	
Total	115	190,128 mts	30.6 m ³

Como ya se mencionó, el diseño de rutas de la recolección de residuos sólidos en Ciudad Universitaria se realiza de manera empírica, por lo que representa un problema de interés que, en esta tesis, se busca resolver al ser planteado como un problema de ruteo. Pero, antes de continuar con el planteamiento del problema, vale la pena mencionar algunos problemas que se identificaron en las entrevistas con el personal de la DGOyC y con operadores de los vehículos recolectores, así como en pleno recorrido de algunas rutas:

- Problemas de acceso para los vehículos recolectores a los centros de acopio (puntos de recolección).
- El número de contenedores en los centros de acopio es insuficiente en algunos casos, por lo que al llegar al punto de recolección los trabajadores deben juntarla primero antes de depositarla en el camión.
- Los operadores de los camiones no siempre respetan el orden de la ruta que nos fue señalada en las entrevistas con la DGOyC.
- La mayoría de las artesas están al aire libre, lo cual las hace proclives a animales en busca de alimentos o a los llamados pepenadores que recuperan materiales que puedan reutilizar o vender. En ambos casos, los residuos son dispersados alrededor de los contenedores.¹⁰

⁸En los mapas proporcionados por la DGOyC se muestran 25 puntos de recolección, sin embargo, uno corresponde a los Talleres de Conservación (Depósito) por lo que no está incluido tal cual como punto de recolección. Otro aparece fuera de los límites de CU, además en entrevistas con personal de la DGOyC no se nos pudo aclarar la ubicación correcta ni acceso a tal punto, por lo que se decidió no considerarlos al realizar la experimentación. Uno más, se encuentra también en el Circuito Deportivo y se tomó únicamente para ese circuito.

⁹En los mapas proporcionados por la DGOyC se muestran 31 puntos de recolección, sin embargo, en entrevistas con personal de la DGOyC no se nos pudo aclarar la ubicación correcta ni acceso a tal punto, por lo que se decidió no considerarlo al realizar la experimentación.

¹⁰En el nuevo Sistema de manejo de residuos se propone crear artesas cerradas bajo llave, a las cuales sólo tendrá acceso personal autorizado.

Capítulo 6

Metodología

En este Capítulo se describe a detalle la metodología a seguir en esta investigación, la cual se muestra en la Figura 6.1. De esta manera, en la Sección 6.1, se presenta el problema de recolección de residuos sólidos que se quiere resolver. En la Sección 6.2, el problema real es representado y formulado de forma matemática. En la Sección 6.3, se presenta el método de solución propuesto para resolver el problema, el cual es Optimización por Colonia de Hormigas.

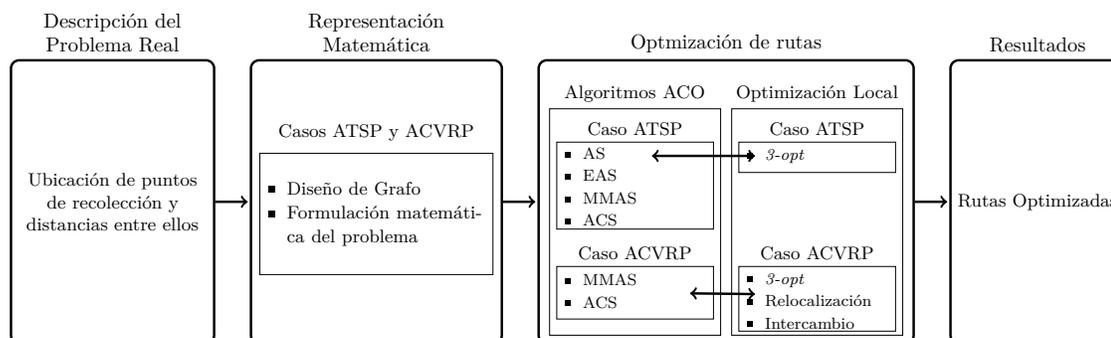


Figura 6.1: Diagrama de Metodología

6.1 Descripción del problema

En muchas ciudades, incluyendo la nuestra, el diseño de rutas de recolección ha sido generado por la experiencia de los encargados de la gestión de residuos o incluso por los propios operadores de los vehículos. Lo cual, no necesariamente es la mejor decisión, pues no aseguran seguir rutas óptimas. Un diseño de rutas inadecuado lleva a una cobertura insuficiente, problemas ambientales y de salubridad, así como costos elevados del servicio.

Por lo anterior, en este trabajo se busca encontrar un diseño de rutas de recolección lo más cercano al óptimo en Ciudad Universitaria. El objetivo es minimizar la distancia total recorrida, lo cual podría derivarse en una disminución de gastos de combustible, deterioro de los vehículos y ahorro de tiempo.

El problema de recolección de RSU se plantea en este trabajo como un problema de ruteo, el cual, como se vio en el Capítulo 2, puede ser visto como ruteo por nodos o ruteo por arcos. En este caso, se propone un modelo de problema de ruteo por nodos, debido a que el tipo de recolección (por contenedores) se ajusta más a ese tipo de problema.

Adicionalmente, el problema es resuelto de dos formas: en primer lugar se hace un rediseño de las rutas ya establecidas. Es decir, para cada uno de los cinco circuitos presentados en el capítulo anterior, se establece en qué orden deberían visitarse los puntos de recolección, para reducir la distancia total del recorrido. En este caso, se asume que la cantidad total generada en todos los puntos de recolección asociados a cada ruta, no rebasa la capacidad del vehículo, por lo que el problema se trabaja como un TSP asimétrico (ATSP).

Por otra parte, también se propone un nuevo diseño de rutas considerando los cinco camiones disponibles así como la capacidad de los mismos. De esta manera, en el problema se involucra la cantidad de residuos generados en cada punto de recolección. Lo anterior, propicia que el problema se resuelva como un CVRP asimétrico (ACVRP).

Para ambos casos, primero se obtuvo la ubicación de los puntos de recolección, que ya han sido mostrados en la Figura 5.1. Después, se obtuvieron las distancias mínimas entre cada par de puntos. Las cuales fueron calculadas a partir de los mapas proporcionados por la DGOyC.

En la siguiente sección se explica cómo, a partir de la ubicación de los puntos de recolección, se diseñó un grafo cuyos nodos representan los puntos de recolección, talleres o depósitos y la Estación de transferencia.

6.2 Representacion matemática

Como ya se mencionó, por una parte se van a rediseñar las rutas ya existentes y por otra, se propondrá un nuevo diseño de rutas. En ambos casos, se tienen problemas de ruteo los cuales son usualmente representados con un grafo. Además, como se explicó en el Capítulo 3, las hormigas de los algoritmos ACO se mueven en un *grafo de construcción*, por lo que es importante establecer las características del grafo con el que se va a trabajar. Aun cuando el grafo no difiere significativamente entre el caso ATSP y el ACVRP, se explicará por separado el diseño del mismo. Análogamente, se harán las formulaciones matemáticas por separado, las cuales sí muestran diferencias significativas.

6.2.1 Diseño de grafo: caso ATSP

El problema del agente viajero (TSP) consiste en un agente de ventas que tiene que visitar n ciudades una única vez, empezando y terminando el recorrido en la misma ciudad. Por otra parte, para el rediseño de rutas, en cada circuito un vehículo tiene que visitar todos los puntos de recolección una sola vez, empezando y terminando la ruta en el depósito. Dado esto, es fácil ver que el vehículo puede representar al agente viajero y las ciudades, a los puntos de recolección; mientras que el depósito sería la ciudad inicial y final. Cabe mencionar, que existe la variante asimétrica del problema del agente viajero, la cual hace referencia a que la distancia de la ciudad A a la ciudad B no es la misma que la distancia de la ciudad B a la ciudad A y dado que la distancia entre dos contenedores depende de la dirección, el TSP asimétrico es aplicable al problema real que se busca resolver en esta tesis.

El problema del agente viajero asimétrico se puede modelar mediante un grafo dirigido completo $G = (V, A)$, donde $V = \{0\} \cup \{1, 2, \dots, n\}$ es el conjunto de vértices o ciudades y $A = \{i, j : i, j \in V\}$ el conjunto de arcos o caminos entre ciudades. Para nuestro problema de recolección de residuos, los vértices $V' = \{1, \dots, n\}$ representan los puntos de recolección, mientras que el depósito o taller y la Estación de transferencia, ambos son representados por el vértice $D = \{0\}$, al que llamaremos depósito auxiliar. Esta representación doble es propuesta debido a la naturaleza del problema y a que lo hace más fácil de resolver. Pues, cuando se hace el rediseño de rutas y el problema queda formulado matemáticamente como un ATSP (como se ve más adelante), una vez que el vehículo ha recorrido todos los puntos de recolección de una ruta, debe ir forzosamente a la Estación de transferencia a descargar los residuos e, inmediatamente después, regresar al depósito. En este caso, se puede ver que entre la estación de transferencia y el depósito no se permite una parada intermedia, por lo que la unión de ambos puntos en uno solo supone una ventaja al plantear el problema.

En cuanto a los arcos, cada arco (i, j) representa el camino que debe seguirse del punto de recolección i para llegar al punto de recolección j , el cual no siempre es el mismo para ir de j a i . Por otra parte, cada arco tiene asociado un costo entero no negativo, el cual representa la distancia mínima entre cada par de puntos i, j . Cabe aclarar, que la distancia de cualquier punto de recolección al depósito auxiliar es la suma de la distancia de dicho punto de recolección a la Estación de transferencia, más la distancia de la Estación de transferencia al depósito o taller. Mientras que la distancia del depósito auxiliar a cada punto de recolección es simplemente la distancia mínima del depósito o taller a dicho punto de recolección.

Ahora bien, las hormigas que se mueven a través del grafo son la representación del vehículo, el cual no tiene una capacidad de carga, pues se supone que la demanda total de cada circuito es menor a la capacidad del vehículo. El objetivo es que cada hormiga construya una ruta y una ruta encontrada en G se define como un ciclo que inicia y termina en el mismo nodo, es decir, una sucesión de nodos y aristas, en donde el nodo inicial y final coinciden. Además, la distancia total de la ruta se calcula como la suma de las distancias entre cada par de nodos que pertenecen a dicha ruta.

A continuación, se presenta la formulación del problema de recolección de residuos para el rediseño de rutas.

6.2.2 Formulación para el rediseño de rutas

Para definir el modelo matemático y por ende la formulación, se identificaron las restricciones y características del problema, de tal manera que la elección fuera la adecuada. En el caso del rediseño de rutas, tenemos una separación previa de los 116 puntos de recolección en 5 circuitos. Suponiendo que cada vehículo es capaz de visitar todos los puntos de recolección de un circuito, sin rebasar su capacidad, el problema consiste en encontrar una ruta en cada circuito que minimice la distancia total recorrida, bajo la restricción de que el vehículo debe partir del depósito y regresar al mismo, después de haber descargado los residuos en la Estación de transferencia. Para este problema se eligió un ATSP, en donde el vehículo es el agente viajero que parte de un origen (depósito auxiliar) al que debe regresar y que puede visitar sólo una vez cada ciudad (punto de recolección).

Antes de pasar a la formulación matemática de este problema, conviene presentar la notación utilizada (Tabla 6.1).

Tabla 6.1: Notación utilizada en el rediseño de rutas (ATSP)

Notación	Descripción
Variable de Decisión	
x_{ij}	Variable binaria que indica si el vehículo atraviesa el arco (i, j)
Conjunto de Índices	
D	Depósito auxiliar $\{0\}$
V'	Conjunto de clientes $\{1, 2, \dots, n\}$
A	Conjunto de arcos
Parámetros	
c_{ij}	Costo por atravesar el arco (i, j)
n	Número de clientes

Teniendo en cuenta la notación mostrada en la Tabla 6.1, para la formulación del problema de rediseño de rutas, se define una variable x_{ij} binaria, que toma el valor de uno si el arco (i, j) está en la ruta solución. La formulación, es la siguiente:

$$\text{Min}_x \quad \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (6.1)$$

Sujeto a

$$\sum_{\substack{i \in V \\ i \neq j}} x_{ij} = 1 \quad \forall j \in V \quad (6.2)$$

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij} = 1 \quad \forall i \in V \quad (6.3)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \forall S \subseteq V \setminus \{1\}, |S| \geq 2 \quad (6.4)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in V \quad (6.5)$$

La función objetivo (6.1) consiste en minimizar el costo total, dado por la distancia de todos los arcos usados en la ruta. La restricción (6.2) señala que todos los puntos de recolección deben ser visitados exactamente una vez, mientras que la restricción (6.3) indica que el vehículo debe salir del punto de recolección que atendió. Por otra parte, la restricción (6.4) asegura la continuidad de la ruta, es decir, que la ruta sea conexa y evitar así, la generación de pequeñas subrutas. En (6.5) se especifica que la variable x_{ij} solo puede tomar los valores uno o cero.

Es importante aclarar que, a pesar de que el vehículo debe salir del depósito auxiliar, en la formulación se permite que el origen sea el depósito o un punto de recolección. Lo anterior no afecta al problema, pues una vez que se tiene una ruta, siempre se puede definir como origen el depósito auxiliar.

Por último, vale la pena mencionar que, si en el diseño del problema el depósito y la estación de transferencia fueran vértices separados, digamos el depósito el vértice 0 y la Estación el vértice $n + 1$, se necesitarían las restricciones adicionales 6.6, 6.7 y 6.8. La primera restricción señala que el vehículo no puede ir del depósito a la estación de transferencia directamente. La restricción 6.7 establece que después de que el vehículo haya ido a la estación de transferencia únicamente podrá ir al depósito. Finalmente, la restricción 6.8 no permite que después de visitar un punto de recolección i , el vehículo visite el depósito.

$$x_{0,n+1} = 0 \quad (6.6)$$

$$x_{n+1,0} = 1 \quad (6.7)$$

$$x_{i0} = 0 \quad \forall i \in V' \quad (6.8)$$

6.2.3 Diseño de grafo: caso ACVRP

En el segundo caso, se busca encontrar un conjunto de rutas de costo mínimo, de tal manera que cada punto de recolección sea visitado una sola vez y la capacidad del vehículo no sea rebasada por el total de los residuos recolectados, además de que cada ruta inicia y termina en el depósito. Dado lo anterior, se observa que el problema puede ser visto como un CVRP asimétrico (ACVRP), pues la distancia del punto i al punto j , no es la misma que la distancia del punto j al punto i .

Para este caso, otra vez se hace uso de grafo dirigido completo $G = (V, A)$, donde $V = \{0\} \cup \{1, 2, \dots, n\}$ es el conjunto de vértices y $A = \{i, j : i, j \in V\}$ el conjunto de arcos. Los vértices $V' = \{1, \dots, n\}$ representan los clientes, es decir, los puntos de recolección, mientras que el depósito o taller y la estación de transferencia, ambos son representados por el depósito auxiliar: $D = \{0\}$. En este caso, una vez que el vehículo ha llenado su capacidad, debe ir forzosamente a la estación de transferencia a descargar los residuos e inmediatamente después, regresar al depósito, de manera que, al igual que con el ATSP, se puede ver que entre la estación de transferencia y el depósito no se permite una parada intermedia, por lo que la unión de ambos nodos es benéfica.

También como en el ATSP, cada arco (i, j) representa el camino que debe seguirse desde el punto de recolección i para llegar al punto de recolección j , el cual no siempre es el mismo para ir de j a i . Por otra

parte, cada arco tiene asociado un costo entero no negativo, el cual representa la distancia mínima entre cada par de puntos i, j .

Adicionalmente, en el caso del nuevo diseño de rutas, cada cliente o punto de recolección i tiene asociada una demanda $q_i^k \in \mathbb{R}^+$ con $q_i^k = 0 \forall i \in V'$ no negativa, que será satisfecha por un solo vehículo k . Esta demanda no es más que el volumen de residuos reunidos en cada punto de recolección.

En este caso, las hormigas que se mueven a través del grafo son la representación de una flota de vehículos compuesta por $K \in \mathbb{N}$ unidades idénticas. Las cuales tienen una capacidad $Q \in \mathbb{R}^+$. Además se asume que $q_i^k < Q \forall i = 1, \dots, n$, es decir, que un sólo vehículo es capaz de recolectar los residuos del punto i .

A continuación, se presenta la formulación del problema de recolección de residuos para un nuevo diseño de rutas.

6.2.4 Formulación para un nuevo diseño de rutas

En el caso del nuevo diseño de rutas, los puntos de recolección situados en Ciudad Universitaria están divididos en dos zonas: Zona Cultural y Zona Escolar. Y, como ya se mencionó cuando se describió el Sistema de Recolección de CU, se cuenta con dos depósitos, los cuales sólo atienden a la zona asignada. En la Zona Escolar se atienden 85 puntos de recolección,¹ mientras que en la Zona Cultural 30 puntos. En ambos casos, un sólo vehículo no es capaz de recolectar en un sólo viaje todos los residuos generados en cada zona, por lo que es necesario hacer un diseño de rutas. De esta manera, tanto el problema asociado a la Zona Escolar como a la Zona Cultural, consiste en encontrar un conjunto de rutas que minimice la distancia total recorrida, bajo las siguientes restricciones:

- Para la Zona Escolar se dispone de un máximo de 4 vehículos, lo que se traduce en que máximo debe hacerse una partición en cuatro circuitos. En el caso de la Zona Cultural, sólo se cuenta con un camión, pero se hacen dos recorridos diferentes: uno por día o uno por la mañana y otro por la tarde. Al tomar en cuenta lo anterior, se puede asumir que se cuenta con dos vehículos, de tal manera que se diseñen dos rutas.
- En ambos casos, cada vehículo tiene una capacidad máxima de $16.5m^3$, es decir, en cada ruta el máximo de residuos recolectados sería de $16.5m^3$.
- En las dos zonas, cada vehículo o ruta inicia en el depósito y debe regresar al mismo después de haber descargado los residuos en la Estación de transferencia.

Para este problema se eligió un ACVRP, en donde cada ruta inicia en un origen (depósito auxiliar) al que debe regresar. Asimismo, el vehículo sólo puede visitar un punto de recolección, si no se viola la restricción de capacidad.

Antes de pasar a la formulación matemática de este problema, conviene presentar la notación utilizada (Tabla 6.2).

De acuerdo a la notación mostrada en la Tabla 6.2, en esta formulación, se definen dos variables binarias: x_{ij}^k que toma el valor de uno, si el arco (i, j) está en la ruta solución del vehículo k y y_i^k , la cual toma el valor de uno si el cliente o punto de recolección i es visitado por el vehículo k . Aunque la formulación con dos índices serviría para este problema, dado que la flota es homogénea, se presenta la formulación con tres índices para tener mayor generalidad. La formulación con tres índices se presenta a continuación.

¹En total se tenían 86 puntos, sin embargo uno de ellos se repite en dos circuitos, por lo que sólo es tomado en cuenta una vez.

Tabla 6.2: Notación utilizada en el rediseño de rutas (ACVRP)

Notación	Descripción
Variables de Decisión	
x_{ij}^k	Variable binaria que indica si el vehículo k atraviesa el arco (i, j)
y_i^k	Variable binaria que indica si el cliente i es atendido por el vehículo k
Conjunto de Índices	
D	Depósito auxiliar $\{0\}$
V'	Conjunto de clientes $\{1, 2, \dots, n\}$
A	Conjunto de arcos
K	Conjunto de vehículos
Parámetros	
c_{ij}	Costo por atravesar el arco (i, j)
n	Número de clientes
q_i	Demanda del cliente i
Q	Capacidad de los vehículos

$$\underset{x}{Min} \quad \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij} x_{ij}^k \quad (6.9)$$

Sujeto a

$$\sum_{k \in K} y_i^k = 1 \quad \forall i \in V' \quad (6.10)$$

$$\sum_{k \in K} y_0^k = K \quad (6.11)$$

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij}^k = \sum_{\substack{j \in V \\ j \neq i}} x_{ji}^k = y_i^k \quad \forall i \in V, k \in K \quad (6.12)$$

$$u_{ik} - u_{jk} + Qx_{ij}^k \leq Q - q_j \quad \forall i, j \in V', i \neq j, k \in K \quad (6.13)$$

$$q_i \leq u_i^k \leq Q \quad \forall i \in V', k \in K \quad (6.14)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall i, j \in V', k \in K \quad (6.15)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in V', k \in K \quad (6.16)$$

La función objetivo (Ecuación 6.9) consiste en minimizar el costo total, dado por la distancia de todos los arcos usados en el conjunto de rutas. Las restricciones (6.10) aseguran que todos los clientes sean visitados exactamente por un vehículo. La restricción (6.11) impone que todos los vehículos salgan del depósito; en otras palabras, que el depósito pertenezca a la *tour* de cada vehículo. La restricción (6.12) impone que el mismo número de vehículos visiten y dejen a cada cliente. La restricción (6.13) asegura que los subtours generados contengan al depósito y en conjunto con (6.14), se restringe la demanda total de un subtour a un máximo equivalente a la capacidad del vehículo. En estas restricciones, u_i es una variable que toma valores en los números reales y representa la carga que lleva el vehículo después de visitar el nodo i .

En este planteamiento, al igual que en el anterior, si el depósito y la estación de transferencia fueran vértices separados, se tendrían que agregar otras restricciones.

6.3 Método de Solución

El método de solución utilizado para resolver ambos problemas (ATSP y ACVRP) es Optimización por Colonia de Hormigas (ACO). Se eligió este tipo de algoritmos debido a que, por su funcionamiento, resulta natural su aplicación a problemas de ruteo. Además de que en los últimos años se ha observado su gran desempeño en dicho tipo de problemas.

En primer lugar, se implementaron los algoritmos *Ant System* (AS), *Elitist Ant System* (EAS), *Max-Min Ant System* (MMAS) y *Ant Colony System* (ACS) para hacer el rediseño de rutas en cada uno de los cinco circuitos preestablecidos. Mientras que, para el nuevo diseño de rutas sólo se utilizan los algoritmos MMAS y ACS, los cuales han mostrado un mejor desempeño en comparación con los otros dos.

6.3.1 Caso ATSP

En este primer caso, los algoritmos utilizados son los presentados en el Capítulo 3 (AS, EAS, MMAS y ACS) y no son modificados ya que su aplicación es directa al problema del agente viajero. No obstante, vale la pena recordar los detalles del funcionamiento de los algoritmos para este problema:

- Cada hormiga representa un vehículo y la solución que genera es la secuencia de visita de puntos de recolección en cada circuito.
- Las hormigas se ubican aleatoriamente ya sea en el depósito auxiliar o en un punto de recolección, de tal manera que cada una crea una ruta y después regresa a su nodo origen.
- La elección del siguiente nodo a visitar está en función del rastro de feromona y la información heurística.
- La información heurística está dada por el recíproco de la distancia o costo mínimo entre dos nodos, es decir $\frac{1}{d_{ij}}$. La función anterior define de manera simple la información heurística, además, entre dos nodos estén más cerca mayor será el valor de la información heurística.
- Cada hormiga tiene memoria de los nodos que ha visitado, por lo que no es posible que visite un nodo más de una vez. De esta manera cualquier ruta que crean es factible.
- La feromona inicial está en función de la distancia de un tour generado por medio del algoritmo del vecino más cercano.

Cabe mencionar que, si bien, el algoritmo ACS no es modificado, sí se le añade un algoritmo de búsqueda local. Esto se debe a que, en la mayoría de la literatura consultada, se encontró que el algoritmo ACS era aplicado en conjunto con una heurística. Dicha heurística generalmente es el algoritmo $k - opt$ (con k igual a 2 ó 3) o el algoritmo de Lin-Kernighan. Para nuestro problema, se implementó el algoritmo *3-opt reducido*, cuyo funcionamiento en conjunción con ACS se explica a continuación.

6.3.1.1 Algoritmo 3-opt

Como ya se mencionó, el algoritmo de optimización local elegido es el *3-opt reducido*. Este algoritmo opera de la siguiente manera: una vez que todas las hormigas han construido una solución o ruta, para cada una de estas rutas se elijen diferentes combinaciones de tres arcos que han de ser reemplazados por otros tres arcos para generar una nueva ruta.

A detalle, sean $(i, i + 1)$, $(j, j + 1)$ y $(k, k + 1)$ los arcos a ser removidos de una ruta llamada $r1$, con $i < j < k$ y de tal manera que se obtengan tres semirutas separadas (Fig 6.2a). En seguida, se considera

reconectar las semirutas, al agregar los arcos $(i, j + 1)$, $(j, k + 1)$ y $(k, i + 1)$ con lo cual se generaría una nueva ruta $r2$ (Fig 6.2b). Esta reconexión es realizada siempre y cuando la distancia total de la ruta $r1$ sea mayor que la distancia de la ruta $r2$ y entonces se cambia $r1$ por $r2$.² Este proceso se repite hasta que ya no sea posible encontrar un cambio que mejore la ruta. Esta ruta final se considera como un óptimo local.

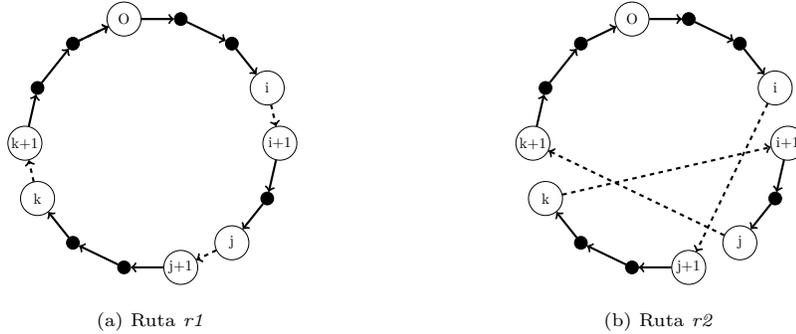


Figura 6.2: Reubicación de aristas en el algoritmo 3-opt

Ahora bien, en nuestro caso se considera una ruta dirigida, por lo que al tener elegidos los tres arcos a remover, sólo se puede reconectar la ruta de una manera posible para que la orientación de la ruta no se vea afectada. Por esta razón se le da el nombre de *reducido*, pues en otros casos se puede considerar cambiar la dirección de segmentos de la ruta.

Por otra parte, como el tiempo necesario para analizar todos los posibles movimientos es extremadamente alto, se han realizado algunas modificaciones a este algoritmo, entre las cuales se encuentran el uso de una Lista de Candidatos y una estrategia llamada *don't look bits*, las cuales fueron implementadas para hacer más rápida la búsqueda del óptimo local.

- **Lista de Candidatos:** como ya se mencionó, la única posibilidad de reconectar la ruta es remplazar los arcos removidos por los arcos: $(i, j + 1)$, $(j, k + 1)$ y $(k, i + 1)$, lo cual significa que el vértice $j + 1$ es el siguiente al vértice i , es decir, una vez elegido i la búsqueda se enfoca en un nodo $j + 1$. De esta manera, para elegir el arco $(j, j + 1)$, en lugar de fijarnos en todos los vértices $j + 1$, sólo tomaremos en cuenta aquellos que están más cerca de i , es decir, a los que están dentro de un Lista de Candidatos. Lo anterior se justifica en que es más posible disminuir la distancia de la ruta si el nodo siguiente a i es uno cercano al mismo. De manera análoga, se busca un nodo $k + 1$ que se encuentre en la Lista de Candidatos de j . La Lista de Candidatos tiene tamaño l y en ella se encuentran los l vértices más cercanos a i o j , ordenados de menor a mayor distancia.
- **Estrategia *don't look bits*:** inicialmente se crea una lista de *don't look bits* (DLB) que asocia un bit a cada nodo. En cada iteración, la búsqueda de nodos de corte (i y j) es restringida a aquellos cuyo bit está apagado, es decir tiene el valor de 0. En un principio, todos los bits están apagados, lo que indica que el movimiento de mejora que incluye a tal nodo debe ser explorado. El bit asociado a un nodo se enciende (toma el valor 1) cada vez que fracasa un intento por mejorar la ruta. Por el contrario, si un arco incidente en un nodo fue cambiado porque se mejora la ruta, el bit asociado a dichos nodos se apaga. Estos nodos son: $i + 1$, $j + 1$ y $k + 1$ (Stützle & Dorigo, 1999).

Estas dos variaciones introducidas al algoritmo original disminuyen en muy poco la calidad de la solución final, pero sí se tiene un gran ahorro del tiempo de procesamiento.

Como ya se mencionó, el algoritmo es aplicado a cada solución construida por cada hormiga. Una vez, que se han “mejorado” todas las soluciones, se continua con el algoritmo ACS es decir se elige la mejor solución y

²Nótese que para probar que $distancia(r1) > distancia(r2)$ basta probar que $c_{i,i+1} + c_{j,j+1} + c_{k,k+1} > c_{i,j+1} + c_{j,k+1} + c_{k,i+1}$

se actualizan los niveles de feromona con la regla de actualización global.

6.3.2 Caso ACVRP

En este caso, donde el problema de recolección de residuos sólidos es planteado como un ACVRP, los algoritmos utilizados son el ACS y MMAS, ambos presentados en el Capítulo 3. A diferencia del planteamiento del problema como un ATSP, en este caso hay una serie de modificaciones aplicadas al par de algoritmos, con el fin de encontrar una solución al problema. Sin embargo, también hay otras características compartidas entre ambos planteamientos. A continuación, se listan los detalles del funcionamiento de los algoritmos para este problema:

- La construcción de rutas es en paralelo, es decir, las hormigas diseñan el conjunto de rutas para los vehículos al mismo tiempo.³
- Las hormigas se ubican aleatoriamente en un punto de recolección, el cual sería el primer nodo a visitar después de salir del depósito.
- La elección del siguiente nodo a visitar está en función del rastro de feromona y la información heurística.
- La información heurística está dada por el recíproco de la distancia o costo mínimo entre dos nodos, es decir $\frac{1}{d_{ij}}$. La función anterior define de manera simple la información heurística, además, entre dos nodos estén más cerca mayor será el valor de la información heurística.
- Cada hormiga tiene memoria de los nodos que ha visitado, por lo que no es posible que visite un nodo más de una vez.
- La hormiga se mueve a un punto de recolección, siempre y cuando no se viole la restricción de capacidad. De esta manera, en conjunto con la característica anterior, cualquier conjunto de rutas que crean es factible.
- La hormiga regresa al depósito (termina una ruta) cuando se cumple con la restricción de capacidad o cuando todos los puntos de recolección han sido visitados.
- La feromona inicial está en función de la distancia de un tour generado por medio del algoritmo del vecino más cercano.
- Por último, se utiliza una Lista de Candidatos, la cual contiene a los l vecinos más cercanos a un punto de recolección. La función de la Lista de Candidatos es: por una parte reducir el tiempo de cómputo, por otra parte, elegir puntos de recolección que supongan ser mejores por estar más cercanos.

Además de las características antes mencionadas, también se implementaron diferentes heurísticas de mejora con el fin de obtener mejores soluciones. Pues si bien, se puede utilizar el algoritmo *3-opt reducido* explicado anteriormente, hay que tener en cuenta que, en el caso del ACVRP, cada solución obtenida es un conjunto de rutas y no una única ruta como se tenía en el caso del ATSP, además de que se tiene la restricción de capacidad máxima. Por esta razón, hay que aplicar dicha heurística de manera un poco diferente.

Antes de continuar con la explicación de las heurísticas implementadas, vale la pena dar un pequeño contexto sobre algoritmos de mejora para el VRP. Las heurísticas de mejora para el caso de los VRPs pueden ser clasificados en dos grupos: intra-ruta e inter-ruta, los cuales parten de una solución inicial y buscan una mejora al modificar las rutas. En concreto los intra-ruta serían los que modifican ruta por ruta dentro del

³Otra manera de implementar la construcción de rutas es de manera secuencial, la cual no causa diferencias en el algoritmo MMAS pero sí en el ACS. En este caso, se eligió la construcción en paralelo, pues se ha encontrado que otorga mejores resultados (Mazzeo & Loiseau, 2004)

conjunto de rutas, entendido tal conjunto como una solución. Dentro de este grupo encontramos los algoritmos k -opt y Lin-Kernighan. Los inter-ruta son algoritmos que modifican al menos dos rutas al mismo tiempo, ya sea mediante el intercambio de nodos o de arcos. Dentro del grupo de los inter-ruta estarían los operadores de Relocalización, de Intercambio y de Cruce.

- **Operador de Relocalización:** consiste en colocar un nodo i de una ruta $r1$ en otra ruta $r2$ (Fig 6.3), siempre y cuando con ello se minimice la distancia total recorrida y no se viole la restricción de capacidad.

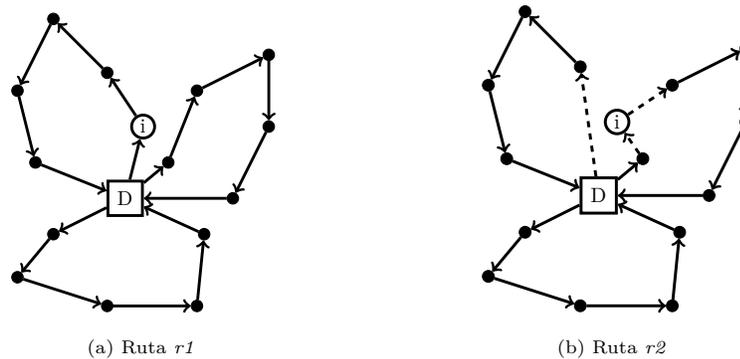


Figura 6.3: Relocalización de nodo

- **Operador de Intercambio:** consiste en que k nodos de dos rutas diferentes se intercambien entre sí (Fig 6.4), por lo que las rutas mantienen el mismo número de nodos (lo cual no ocurre con la reasignación vista en el apartado anterior). Este intercambio se realiza si la distancia total recorrida disminuye y si no se viola la restricción de capacidad.

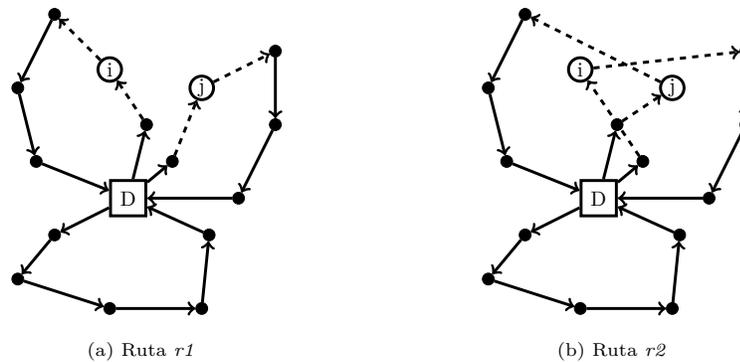


Figura 6.4: Intercambio de nodo con $k = 1$

- **Operador de Cruce:** intercambia arcos entre dos rutas siempre y cuando disminuya la distancia total recorrida y no se viole la restricción de capacidad (Fig 6.5).

Entre estas heurísticas se implementaron el algoritmo 3 -opt, el operador de relocalización y el operador de intercambio. El primero por tener un algoritmo que hiciera modificaciones en cada ruta de una solución completa. Los dos últimos, precisamente para hacer intercambios entre rutas; se excluyó el operador de cruce dado que en (Laporte, Gendreau, Potvin & Semet, 2000) se menciona que los dos primeros generan los mejores resultados. Hay que notar que en el caso del algoritmo 3 -opt, no es necesario revisar que se respete la restricción de capacidad, pues se parte de rutas factibles y no se agregan nodos extra a la ruta, que supongan un aumento en la carga del vehículo. Mientras que en el caso de los operadores, sí es necesario ver que se respete la restricción para que las soluciones modificadas sean factibles.

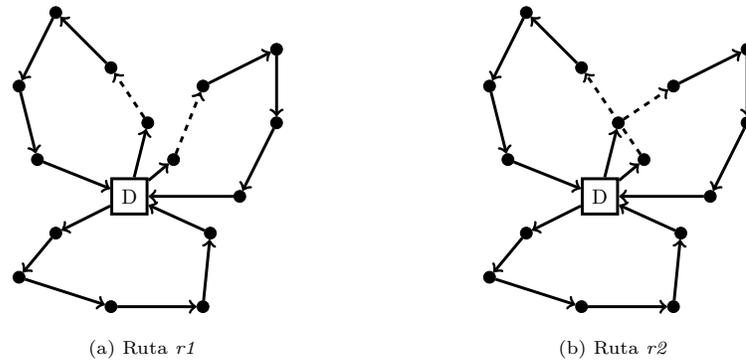


Figura 6.5: Operador de Cruce

Así como en el caso del ATSP, estos algoritmos son aplicados una vez que las hormigas han construido una solución. Sin embargo, en este caso primero se encuentra la mejor solución construida por las hormigas y una vez seleccionada, a tal solución se le aplican las heurísticas de búsqueda local. Ya que la solución ha sido “mejorada”, continúa la ejecución de los algoritmos ACS y MMAS, es decir se actualizan los niveles de feromona según la regla de actualización.

Adicionalmente, en la Figura 6.6 se muestra el diagrama de flujo general de los algoritmos de hormigas utilizados.

6.4 Resultados

En esta etapa de la metodología, se prueban los algoritmos en cada uno de los casos, con el fin de optimizar la recolección de residuos sólidos en CU. Dichos resultados se presentan a continuación, de manera aparte como un capítulo completo.

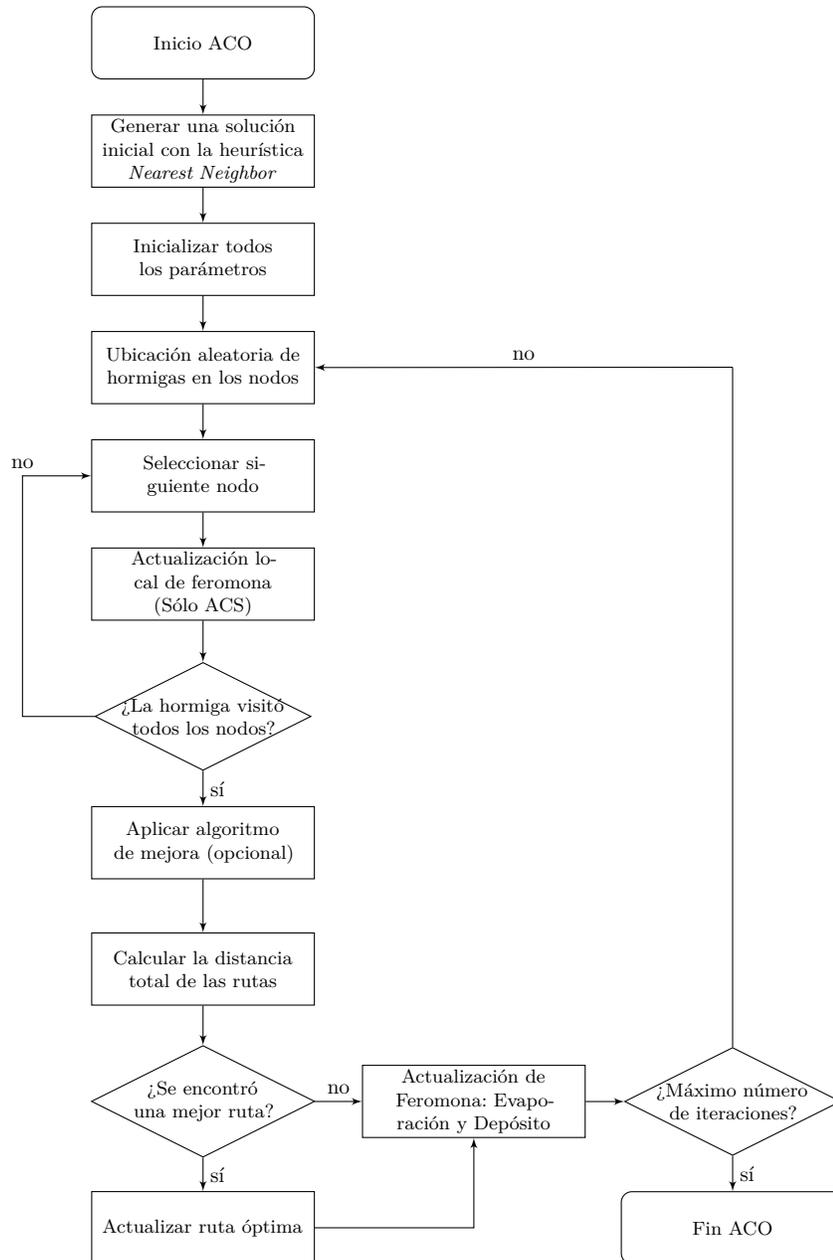


Figura 6.6: Diagrama de flujo de ACO

Capítulo 7

Experimentación y Análisis de Resultados

Este capítulo se divide en dos partes: en la primera, el problema de recolección de residuos en CU se verá como un ATSP, en la segunda, el problema será atacado desde un planteamiento de ACVRP. Para el ATSP se analizarán los resultados obtenidos en las experimentaciones realizadas por los cuatro algoritmos ACO: ACS, MMAS, EAS y AS; aplicados a los cinco circuitos previamente presentados. En el caso del ACVRP, se utilizarán únicamente los algoritmos ACS y MMAS para encontrar un nuevo conjunto de rutas. En ambos casos serán comparadas las rutas generadas con ACO con las ya establecidas y que actualmente operan en CU. Adicionalmente, se hará un análisis del efecto que tienen los parámetros al resolver el ATSP.

7.1 Resultados para ATSP

En esta sección se presentan los resultados obtenidos con los algoritmos AS, EAS, MMAS y ACS, utilizados para la optimización de los cinco circuitos descritos en el capítulo anterior. Los algoritmos fueron corridos 30 veces y cada uno con 3000 iteraciones. A continuación se muestra, en primer lugar, los puntos más importantes del estudio realizado a los parámetros de cada algoritmo (en el Anexo B se muestra más detalles del mismo) y después se hace una comparación entre las rutas actuales y las obtenidas con ACO.

7.1.1 Efecto de los parámetros en la ejecución de los algoritmos

En esta sección, se estudia el efecto que tienen algunos parámetros en el comportamiento de los algoritmos ACS, MMAS, EAS y AS, al ser utilizados para encontrar un tour óptimo en los cinco circuitos de recolección de residuos de CU. Por una parte, el estudio se enfoca en la evolución de la "mejor solución hasta el momento" (*best-so-far-solution*, *bfs*), durante la ejecución del algoritmo. Estos resultados se muestran en gráficas y se basan en la evolución del error relativo de la *best-so-far-solution* con respecto a la mejor solución encontrada con ACO para cada circuito. Todas las gráficas generadas se encuentran en el Anexo B. En cada gráfica, la curva muestra el promedio del error relativo de las 30 corridas con la misma configuración de parámetros. Adicionalmente, para cada algoritmo aplicado a cada circuito, se generó una tabla que incluye la mejor solución, peor solución, el promedio y la desviación estándar, las cuales también se expresan como el error relativo (%); además, se muestran el promedio y desviación estándar de iteraciones necesarias para alcanzar la mejor solución, así como la mínima y máxima iteración en la que se encuentra la mejor solución para el circuito, estas tablas se encuentran en el Anexo C. Como ya se mencionó en este capítulo, sólo se muestran los puntos más importantes del análisis realizado a los parámetros, por lo que al final de cada explicación del efecto de los parámetros, se presenta una tabla resumida que contiene los mejores y peores resultados (vistos como errores relativos) para cada algoritmo.

Los valores por defecto escogidos para los parámetros se muestran en la Tabla 7.1. Dichos valores fueron elegidos porque, de acuerdo a las experimentación de Dorigo y Stützle (2004), proporcionan los mejores resultados. En cada experimento se varía un parámetro a la vez, mientras que los demás permanecen constantes, de acuerdo al valor presentado en la Tabla 7.1. Además, en el caso del algoritmo MMAS se utiliza el mecanismo de reinicialización de feromona. Dicho mecanismo es activado si después de 250 iteraciones la mejor solución no mejora. También, para este algoritmo, se decidió utilizar la mejor solución local para actualizar la feromona. Lo anterior, debido a que en la literatura se menciona que para problemas pequeños es mejor utilizar esa solución en vez de la solución global. En el caso del algoritmo ACS, también se comparan los resultados al añadir la heurística de búsqueda local *3-opt*.

Tabla 7.1: Valores constantes de los parámetros

Algoritmo	α	β	ρ	m	q_0	ψ	e	τ_0
ACS	1	3	0.1	10	0.9	0.1	-	$1/nC^{nn}$
MMAS	1	3	0.02	n	-	-	-	$1/\rho C^{nn}$
EAS	1	3	0.5	n	-	-	n	$(e+m)/\rho C^{nn}$
AS	1	3	0.5	n	-	-	-	m/C^{nn}

Antes de pasar al estudio del efecto de los parámetros en los algoritmos ACO, se definen los siguientes términos utilizados para referirse a las soluciones:

- Solución A: mejor solución encontrada con los algoritmos ACO para el circuito Deportivo.
- Solución B: mejor solución encontrada con los algoritmos ACO para el circuito Exterior.
- Solución C: mejor solución encontrada con los algoritmos ACO para el circuito de la Investigación Científica.
- Solución D: mejor solución encontrada con los algoritmos ACO para el circuito Escolar.
- Solución E: mejor solución encontrada con los algoritmos ACO para el circuito CCU.

7.1.1.1 Pruebas a los parámetros del algoritmo AS

Los parámetros del algoritmo AS estudiados son: β , el cual determina la influencia de la información heurística; ρ , el factor de evaporación de feromona y m , el número de hormigas. Los valores propuestos para cada parámetro, para observar el comportamiento del algoritmo en cada circuito, se observan en la Tabla 7.2.

Tabla 7.2: Valores variables de los parámetros del algoritmo AS

Parámetros	Valores
β	{0.1, 1, 5, 8}
ρ	{0.01, 0.1, 0.3, 0.8}
m	Circuito Deportivo: {3, 7, 10, 14} Otros Circuitos: {5, 10, 15, 20}

- El parámetro más sensible fue β para todos los circuitos. Se observó que un valor muy pequeño de β arroja resultados muy distantes a los obtenidos con valores más grandes del mismo parámetro y con valores de otros parámetros. Adicionalmente, cuando β es igual a 1 se alcanzan las mejores soluciones para todos los circuitos. Lo anterior sugiere, que para este algoritmo es mejor no dar mayor peso a la información heurística y mantener el balance entre éste y la feromona.
- Tasas de evaporación de 0.01 y 0.1 producen una mejor evolución del algoritmo que tasa más grandes. Es decir que hacer que persista la feromona por más tiempo ayuda al comportamiento del algoritmo.
- En general, a excepción del C. Deportivo, entre más hormigas se utilicen (con límite igual al número de nodos), mejor la calidad de las soluciones generadas.
- Para el C. Deportivo solamente se alcanza la solución A cuando β toma el valor 1.
- Para el C. Exterior la solución B se alcanza con ρ igual a 0.01 y β igual a 1.
- En cuanto al C. de la Investigación Científica la solución C se alcanza en varios casos: con β igual a 1 y 3, con ρ igual a 0.01, 0.1 y 0.5 y, en el caso de m únicamente con n hormigas
- Para el C. Escolar la solución D también se alcanza con β igual a 1, 3, 5 y 8; con ρ igual a 0.01, 0.1, 0.3

y 0.5 y con m igual a 15 y 20.

- Para el C. CCU, la solución E no se encuentra con ninguna de las configuraciones propuestas para el conjunto de parámetros.

Tabla 7.3: Mejores resultados por cada parámetro del algoritmo AS aplicado a la Zona Escolar

Parámetro	C. Deportivo			C. Exterior			C. de la Inv. Científica			C. Escolar			C. CCU		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
β	0.137 $\beta=1$	7.821 $\beta=0.1$	0.000 $\beta=1$	0.305 $\beta=1$	34.300 $\beta=0.1$	0.000 $\beta=1$	0.688 $\beta=1$	24.865 $\beta=0.1$	0.000 $\beta=1, 3$	0.019 $\beta=5$	20.035 $\beta=0.1$	0.000 $\beta=1, \dots, 8$	1.894 $\beta=1$	29.982 $\beta=0.1$	0.003 $\beta=1$
ρ	0.134 $\rho=0.01$	0.707 $\rho=0.5$	0.003 $\rho=0.01$	0.190 $\rho=0.01$	0.382 $\rho=0.8$	0.000 $\rho=0.01$	2.090 $\rho=0.01$	3.889 $\rho=0.3$	0.000 $\rho=0.01, 0.1, 0.5$	0.0001 $\rho=0.01$	0.049 $\rho=0.8$	0.000 $\rho=0.01, \dots, 0.5$	0.225 $\rho=0.01$	2.641 $\rho=0.8$	0.003 $\rho=0.01, 0.1$
m	0.329 $m=14$	3.026 $m=3$	0.003 $m=3$	0.372 $m=n$	0.931 $m=5$	0.003 $m=20$	3.471 $m=5$	3.828 $m=20$	0.000 $m=n$	0.028 $m=n$	0.448 $m=5$	0.000 $m=15, 20, n$	2.188 $m=n$	3.338 $m=5$	0.066 $m=15, n$

A: Error relativo medio mínimo B: Error relativo medio máximo C: Error relativo mínimo

7.1.1.2 Pruebas a los parámetros del algoritmo EAS

El siguiente análisis corresponde a parámetros utilizados en el algoritmo EAS: β , ρ , m y e . Los valores propuestos de cada parámetro para observar el comportamiento del algoritmo en cada circuito, se observan en la Tabla 7.4.

Tabla 7.4: Valores variables de los parámetros del algoritmo EAS

Parámetros	Valores
β	{0.1, 1, 5, 8}
ρ	{0.01, 0.1, 0.3, 0.8}
m	Circuito Deportivo: {3, 7, 10, 14} Otros Circuitos: {5, 10, 15, 20}
e	Circuito Deportivo: {1, 5, 10, 25} Otros Circuitos: {1, 10, 20, 35}

- Los parámetros más sensibles fueron β y m . En el primer caso, se observó como un valor muy pequeño de β arroja resultados muy distantes a los obtenidos con otros valores (al igual que en el algoritmo AS). En el segundo caso, se observan cambios graduales al ir aumentando el número de hormigas.
- En general, los valores de β que proporcionan mejores resultados son mayores a los que en AS producían tales resultados. Esto se puede deber a que en el algoritmo EAS se deposita mayor cantidad de feromona por lo cual puede ser necesario una mayor influencia de la información heurística. Lo anterior se obtiene con valores de β mayores al de α , pues hay que recordar que este último parámetro está fijo en 1.
- En general, es mejor una adaptación lenta del algoritmo, es decir, cambios graduales en los niveles de feromona, los cuales se obtienen con menores tasas de evaporación. Lo anterior permite que lo aprendido en iteraciones anteriores persista.
- En general, entre más hormigas se utilicen (con límite igual al número de nodos), mejor la calidad de las soluciones generadas.
- Se observa que para el C. Escolar, explotar la mejor solución no es tan buena opción. Pues por una parte, en comparación con los otros circuitos, el parámetro e toma el valor más bajo para generar mejores soluciones. Por otra parte β toma el valor más alto para generar mejores soluciones en comparación con los otros circuitos, es decir que para este circuito es mejor explorar soluciones. Lo anterior, también se

observa para el C. Deportivo, donde el mejor promedio de soluciones se alcanza con β igual a 5 y e igual a 1.

- Para el C. de la Investigación Científica es bueno explotar la mejor solución, pues se observa que con valores más grandes de e la evolución del algoritmo es mejor. Para el C. CCU también es bueno explotar la mejor solución, sin embargo es mejor con valores intermedios de e . Por otra parte, para el C. Exterior no se observa una tendencia marcada hacia ciertos valores de e que produzcan los mejores resultados.
- Para cualquier circuito, la solución A, B, C, D o E, (según sea el caso) se alcanza con un mayor número de configuraciones en comparación con el algoritmo AS. Para todos los circuitos se logra encontrar su solución respectiva, con al menos una de las configuraciones; lo cual no sucedió en el caso del circuito CCU, al ser resuelto con el algoritmo AS. En la Tabla 7.5 se puede ver específicamente con qué valores se obtiene el resultado mencionado.

Tabla 7.5: Mejores resultados por cada parámetro del algoritmo EAS aplicado a la Zona Escolar

Parámetro	C. Deportivo			C. Exterior			C. de la Inv. Científica			C. Escolar			C. CCU		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
β	0.415 $\beta=5$	8.000 $\beta=0.1$	0.000 $\beta=1,3$	0.195 $\beta=3$	23.071 $\beta=0.1$	0.000 $\beta=1$	2.072 $\beta=3$	15.459 $\beta=0.1$	0.000 $\beta=1,\dots,8$	0.023 $\beta=8$	10.122 $\beta=0.1$	0.000 $\beta=5,8$	0.744 $\beta=3$	15.346 $\beta=0.1$	0.000 $\beta=3,5$
ρ	0.071 $\rho=0.01$	0.819 $\rho=0.5$	0.000 $\rho=x$	0.018 $\rho=0.01$	0.970 $\rho=0.8$	0.000 $\rho=0.01, 0.1, 0.8$	0.000 $\rho=0.01$	2.784 $\rho=0.3$	0.000 $\rho=x$	0.004 $\rho=0.01$	0.258 $\rho=0.8$	0.000 $\rho=.01, .1, .3, .8$	0.136 $\rho=0.01$	1.310 $\rho=0.8$	0.000 $\rho=x$
m	0.675 $m=14$	5.875 $m=3$	0.000 $m=n$	0.195 $m=n$	7.349 $m=5$	0.000 $m=20$	2.048 $m=15$	4.668 $m=5$	0.000 $m=10, 15, 20, n$	0.191 $m=n$	2.324 $m=5$	0.000 $m=20$	0.744 $m=n$	5.327 $m=5$	0.000 $m=x$
e	0.168 $e=1$	1.271 $e=25$	0.000 $e=10, 35$	0.195 $e=n$	0.572 $e=35$	0.000 $e=n$	1.985 $e=35$	3.881 $e=1$	0.000 $e=10, 20, n, 35$	0.003 $e=1$	0.314 $e=35$	0.000 $e=1, 10, 20, 35$	0.108 $e=10$	1.024 $e=1$	0.000 $e=10, 20, n, 35$

A: Error relativo medio mínimo B: Error relativo medio máximo C: Error relativo mínimo

7.1.1.3 Pruebas a los parámetros del algoritmo MMAS

Los parámetros involucrados en el algoritmo MMAS y estudiados en esta sección son: β , ρ y m . Además, también se analiza el comportamiento del algoritmo cuando se utiliza o no el mecanismo de reinicialización de feromona, así como cuando en lugar de utilizar la solución local para actualizar la feromona se utiliza la solución global. Los valores propuestos de cada parámetro, para observar el comportamiento del algoritmo en cada circuito, se observan en la Tabla 7.6.

Tabla 7.6: Valores variables de los parámetros del algoritmo MMAS

Parámetros	Valores
β	{0.1, 1, 5, 8}
ρ	{0.1, 0.3, 0.5 0.8}
m	Circuito Deportivo: {3, 7, 10, 14} Otros circuitos: {5, 10, 15, 20}

Aunado a la búsqueda de una solución en el rediseño de rutas, también se hizo un estudio del efecto de los parámetros en las soluciones generadas. Lo anterior es una de las aportaciones de este trabajo, pues se pudo analizar qué parámetros tienen mayor peso en el algoritmo, así como la influencia del valor que toman, al ser aplicados los algoritmos a un problema real. Dentro del análisis que se hizo, cabe destacar que:

- Para el algoritmo MMAS el parámetro ρ es el más sensible, pues cuando dicho parámetro toma el valor más alto se obtienen los peores resultados entre todas las configuraciones propuestas. Asimismo, las diferencias de los resultados obtenidos entre la menor tasa y la mayor tasa son más grandes en

comparación con otros parámetros.

- Para todos los circuitos se obtienen mejores resultados al utilizar n hormigas. Es decir que es benéfico permitir una mayor generación de soluciones en cada iteración.
- Para el C. de la Investigación Científica se observa una diferencia notable y favorable al utilizar el mecanismo de reinicialización de feromona. Cuando no se utiliza el mecanismo de reinicialización y en consecuencia se sesga la exploración, el algoritmo se estanca en malas soluciones. Para los demás circuitos la diferencia es prácticamente nula.
- Para el C. de la Investigación Científica no hay ninguna diferencia entre utilizar la solución global o la local. De hecho, en ambos casos en todas las corridas se encuentra la solución C. Por otra parte, para el C. CCU sí se nota una diferencia favorable al utilizar la solución local sobre la global, es decir, mantener cierta exploración en el algoritmo. Para los circuitos restantes las diferencias son casi nulas, aunque se nota una ligera mejora con la solución local para los circuitos Deportivo y Escolar y con la solución global para el C. Exterior.
- Por último, para los circuitos de la Investigación Científica y CCU, con todas las configuraciones se alcanza la solución C y E, respectivamente. Para el C. Deportivo la solución A no se alcanza con: β igual a 8, ρ igual a 5 o 8 y m igual a 3. Para el C. Exterior la solución B no se alcanza con: β igual a 8 y ρ igual a 0.8. Y para el C. Escolar la solución D no se alcanza con ρ igual a 0.8.
- Adicionalmente, en los casos del C. de la Investigación Científica y C. CCU, se observa en la Tabla 7.7 que con al menos un valor de cada parámetro, se encuentra la solución C y D, respectivamente, en todas las corridas.

Tabla 7.7: Mejores resultados por cada parámetro del algoritmo MMAS aplicado a la Zona Escolar

Parámetro	C. Deportivo			C. Exterior			C. de la Inv. Científica			C. Escolar			C. CCU		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
β	0.006 $\beta=1$	0.143 $\beta=8$	0.000 $\beta=0.1, \dots, 5$	0.001 $\beta=1$	0.812 $\beta=0.1$	0.000 $\beta=1,1,3,5$	0.000 $\beta=1,3$	2.165 $\beta=8$	0.000 $\beta=x$	0.000 $\beta=5$	0.582 $\beta=0.1$	0.000 $\beta=x$	0.000 $\beta=3$	1.100 $\beta=0.1$	0.000 $\beta=x$
ρ	0.012 $\rho=0.02$	1.476 $\rho=0.8$	0.000 $\rho=0.02, 0.1,0.3$	0.002 $\rho=0.02$	1.454 $\rho=0.8$	0.000 $\rho=0.02, .1, .3, .5$	0.000 $\rho=0.02, .01$	2.812 $\rho=0.8$	0.000 $\rho=x$	0.0002 $\rho=0.02$	1.609 $\rho=0.8$	0.000 $\rho=0.01, \dots, 0.5$	0.000 $\rho=0.02$	2.493 $\rho=0.8$	0.000 $\rho=x$
m	0.012 $m=n$	0.121 $m=3$	0.000 $m=7, 10,14,n$	0.002 $m=n$	0.106 $m=10$	0.000 $m=x$	0.000 $m=n$	1.888 $m=5$	0.000 $m=x$	0.0002 $m=n$	0.007 $m=5$	0.000 $m=x$	0.000 $m=n$	0.225 $m=5$	0.000 $m=x$
Reinicio de feromona	0.012 Sí	0.077 No	0.000 No/Sí	0.002 Sí	0.003 No	0.000 No/Sí	0.000 Sí	0.362 No	0.000 No/Sí	0.0002 Sí	0.0003 No	0.000 No/Sí	0.000 Sí	0.0002 No	0.000 No/Sí
Solución actualiza feromona	0.012 Local	0.033 Global	0.000 Local/Global	0.000 Global	0.002 Local	0.000 Local/Global	0.000 Local/Global	-	0.000 Local/Global	0.0002 Local	0.005 Global	0.000 Local/Global	0.000 Local	0.827 Global	0.000 Local/Global

A: Error relativo medio mínimo B: Error relativo medio máximo C: Error relativo mínimo

7.1.1.4 Pruebas a los parámetros del algoritmo ACS

En el caso del algoritmo ACS los parámetros que se estudian son: β , ρ , m , el número de hormigas; ψ , y q_0 . Además, dado que es muy utilizado agregar un algoritmo de búsqueda local, se hace la experimentación del algoritmo ACS en combinación con la heurística 3 – *opt*. Los valores propuestos de cada parámetro para observar el comportamiento del algoritmo en cada circuito se observan en la Tabla 7.8.

- En cuanto al algoritmo ACS, para los circuitos Deportivo, Exterior, Investigación Científica y Escolar el parámetro más sensible es β ; mientras que para el circuito CCU el parámetro es ρ .
- Para el C. Deportivo y de la Investigación Científica se debe dar un menor o igual peso a la información

Tabla 7.8: Valores variables de los parámetros del algoritmo ACS

Parámetros	Valores
β	{0.1, 1, 5, 8}
ρ	{0.01, 0.3, 0.5, 0.8}
m	Circuito Deportivo: {3, 7, 10, 14} Otros Circuitos: {5, 10, 15, 20}
ψ	{0.01, 0.3, 0.5, 0.8}
q_0	{0.3, 0.5, 0.7, 0.95}

heurística en comparación con el rastro de feromona (β igual 0.1 o 1 contra α igual a 1). Esto se puede deber a la estructura de ambos circuitos, ya que en estos circuitos hay nodos muy cercanos y si se da mucho peso a la distancia es más probable que estos queden en secuencia, sin embargo, es mejor romper dicha secuencia e ir primero a un punto más lejano.

- En contraposición al punto anterior para los circuitos Exterior, Escolar y CCU, un valor muy pequeño de β genera los peores resultados. Es decir que darle menor peso a la información heurística en comparación con el rastro de feromona es contraproducente para estos circuitos. En el caso del C. Exterior lo anterior se debe a que por la estructura del circuito las distancias entre nodos son buena fuente de información para elegir a que nodo moverse. Por otra parte para los circuitos Escolar y CCU una exploración del espacio de soluciones ayuda a elegir puntos que están más alejados pero que ayudan a obtener mejores soluciones globales
- Para los circuitos Exterior, Escolar y CCU se obtienen malos resultados con valores de ρ muy grandes. Es decir que para estos circuitos es contraproducente una evaporación muy alta. Asimismo con una tasa de evaporación pequeña, es decir exploración del espacio de búsqueda, el algoritmo funciona mejor.
- Para el C. Deportivo las tasa menores de feromona producen un peor comportamiento del algoritmo que las tasas más grande, es decir que intensificar la mejor solución encontrada ayuda al algoritmo.
- Para los circuitos Exterior, de la Investigación Científica y CCU donde una menor tasa de evaporación producía buenos resultados, el valor del factor de decaimiento que ayuda al algoritmo es el más grande propuesto. Para el C. Exterior lo anterior sugiere es mejor que menos feromona permanezca en las conexiones, probablemente porque al guiarse por las distancias es bueno para este circuito. En cuanto al C. de la Investigación Científica es mejor si hay exploración del espacio de soluciones.
- Por otra parte, para el C. Deportivo en donde la menor tasa de evaporación producía malos resultados, el valor del factor de decaimiento más pequeño es mejor que valores más grandes del mismo. Es decir, es mejor que dentro de una iteración el rastro no se evapore tanto, de esta manera el algoritmo no se guiará en demasía por las distancias, que como ya se mencionó, es mejor darle más peso a la feromona.
- Es mejor si más hormigas generan soluciones, en este caso el mayor número de hormigas fue igual al número de nodos del problema a resolver.
- Para todos los circuitos es contraproducente que la exploración sea muy poca ($q_0 > 0.95$) ya que se producen los peores resultados en comparación con valores menores del parámetro.
- Para los circuitos Exterior y de la Investigación Científica es mejor que haya mayor exploración.
- Para los circuitos Deportivo y Escolar parece que un balance equiprobable entre exploración y explotación es bueno.
- Para el C. CCU es mejor darle mayor importancia a la explotación ($q_0 = 0.9$), pero teniendo en cuenta que $q_0 = 0.95$ genera los peores resultados. Aunque esto parece ser contradictorio con algunos puntos mencionados anteriormente, hay que tener en cuenta que a mayores valores de q_0 , se elegirá aquél nodo cuyo producto entre feromona y distancia se maximice. Pero en este producto ya se está tomando en cuenta que es la distancia la que tiene un mayor peso que la feromona pues β es igual a 3 y α es igual a

1.

- Como se esperaba, agregar la búsqueda local permite que se alcancen mejores soluciones en menos iteraciones. Sin embargo, el tiempo computacional que requiere cada corrida al utilizar la búsqueda local.
- Una vez analizado el comportamiento del algoritmo ACS aplicado en cada circuito, se puede ver que este algoritmo tuvo la mayor variabilidad en resultados según el circuito al que estuviera aplicado, en comparación con los algoritmos AS, EAS y MMAS. Por otra parte, vale la pena mencionar que la solución A, B, C, D, o E, según sea el caso, se alcanza con todas las configuraciones de parámetros (C. Escolar y C. CCU) o casi todas las configuraciones (C. Deportivo, C. Exterior y C. de la Investigación Científica). En el caso del C. Deportivo la solución A no se encuentra con seis configuraciones: β igual a 5 y 8, con 7 hormigas, con ψ igual a 0.5 y 0.8 y cuando q_0 es igual 0.95. En el caso del C. Exterior la solución B no se encuentra con una configuración: β igual 5. En el caso del C. de la Investigación Científica la solución C no se encuentra con dos configuraciones: β igual 8 y con q_0 igual a 0.95.

Tabla 7.9: Mejores resultados por cada parámetro del algoritmo ACS aplicado a la Zona Escolar

Parámetro	C. Deportivo			C. Exterior			C. de la Inv. Científica			C. Escolar			C. CCU		
	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
β	0.000 $\beta=0.1$	4.263 $\beta=8$	0.000 $\beta=0.1, 1,3$	0.000 $\beta=1$	1.345 $\beta=0.1$	0.000 $\beta=0.1, \dots, 5$	0.000 $\beta=0.1,1$	3.949 $\beta=8$	0.000 $\beta=0.1, \dots, 5$	0.005 $\beta=5$	0.473 $\beta=0.1$	0.000 $\beta=x$	0.297 $\beta=3$	1.276 $\beta=0.1$	0.000 $\beta=x$
ρ	0.136 $\rho=0.3$	0.799 $\rho=0.01$	0.000 $\rho=x$	0.002 $\rho=0.01$	0.305 $\rho=0.8$	0.000 $\rho=x$	2.219 $\rho=0.01$	3.531 $\rho=0.1$	0.000 $\rho=x$	0.009 $\rho=0.1$	0.070 $\rho=0.8$	0.000 $\rho=x$	0.297 $\rho=0.1$	2.524 $\rho=0.8$	0.000 $\rho=x$
m	0.542 $m=n$	0.692 $m=7$	0.000 $m=3, 10,14,n$	0.006 $m=n$	0.203 $m=5$	0.000 $m=x$	1.043 $m=n$	3.531 $m=10$	0.000 $m=x$	0.0002 $m=15, 20,n$	0.037 $m=5$	0.000 $m=x$	0.018 $m=20$	1.913 $m=5$	0.000 $m=x$
ψ	0.246 $\psi=0.01$	2.689 $\psi=0.8$	0.000 $\psi=0.01, 0.1,0.3$	0.003 $\psi=0.3,5, 8$	0.203 $\psi=0.01$	0.000 $\psi=x$	1.450 $\psi=0.8$	3.531 $\psi=0.1$	0.000 $\psi=x$	0.004 $\psi=0.3$	0.039 $\psi=0.8$	0.000 $\psi=x$	0.228 $\psi=0.8$	1.455 $\psi=0.01$	0.000 $\psi=x$
q_0	0.019 $q_0=0.5$	1.382 $q_0=0.95$	0.000 $q_0=0.3, \dots,0.9$	0.0004 $q_0=0.3$	0.188 $q_0=0.95$	0.000 $q_0=x$	0.030 $q_0=0.3$	3.800 $q_0=0.95$	0.000 $q_0=0.3, \dots,0.9$	0.000 $q_0=0.5$	0.026 $q_0=0.95$	0.000 $q_0=x$	0.297 $q_0=0.9$	1.022 $q_0=0.95$	0.000 $q_0=x$
Búsqueda Local	0.075 Sí	0.609 No	0.000 No/Sí	0.000 Sí	0.040 No	0.000 No/Sí	0.083 Sí	3.531 No	0.000 No/Sí	0.001 Sí	0.009 No	0.000 No/Sí	0.0001 Sí	0.297 No	0.000 No/Sí

A: Error relativo medio mínimo B: Error relativo medio máximo C: Error relativo mínimo

Al comparar los cuatro algoritmos aplicados con las distintas configuraciones propuestas, se puede concluir lo siguiente:

- De las 13 configuraciones de parámetros propuestas para aplicar el algoritmo AS, con 7.7% de ellas se encuentra la mejor solución en el caso del C. Deportivo; mientras que para el C. Exterior es con 15.4%; para el C. de la Investigación Científica es con 30.8%; para el C. Escolar es con 69.2% y, para el C. CCU con ninguna configuración se logra encontrar la mejor solución.
- De las 17 configuraciones de parámetros propuestas para aplicar el algoritmo EAS, con 35.3% de ellas se encuentra la mejor solución en el caso del C. Deportivo; mientras que para el C. Exterior es con 41.2%; para el C. de la Investigación Científica es con 82.4%; para el C. Escolar es con 64.7% y, para el C. CCU con un 76.5% de las configuraciones.
- De las 15 configuraciones de parámetros propuestas para aplicar el algoritmo MMAS, con 73.3% de ellas se encuentra la mejor solución en el caso del C. Deportivo; mientras que para el C. Exterior es con 86.7%; para los circuitos de la Investigación Científica y CCU, con todas las configuraciones se encuentra la mejor solución; finalmente, para el C. Escolar con 93.3% de las configuraciones se logra el resultado mencionado.
- De las 22 configuraciones de parámetros propuestas para aplicar el algoritmo ACS, con 72.7% de ellas

se encuentra la mejor solución en el caso del C. Deportivo; mientras que para el C. Exterior es con 95.5%; para el C. de la Investigación Científica con 90.9% y para los circuitos Escolar y CCU con todas las configuraciones se logra encontrar la mejor solución.

- Dado los datos anteriores, el algoritmo MMAS funcionó mejor en los circuitos: Deportivo y de la Investigación Científica mientras que el algoritmo ACS funcionó mejor en los circuitos: Exterior y Escolar. En cuanto al C. CCU ambos algoritmos parecen arrojar buenos resultados.
- Al implementar los algoritmos es importante el problema en particular que se busca resolver. Es de llamar la atención que para el C. Deportivo con un menor número de configuraciones se logra encontrar la solución A, a pesar de ser el circuito con menos nodos. Lo anterior probablemente se deba a que en este circuito hay nodos muy cercanos entre ellos por lo que puede resultar difícil ir primero a un nodo lejano que, aunque inicialmente incrementa la distancia de la solución parcial, al obtener la solución completa la distancia total recorrida es menor que si se hubiera elegido el nodo cercano.
- Para el C. Escolar en todos los algoritmos se observó que es mejor la exploración del espacio de búsqueda. De hecho, sólo para este circuito la proporción de configuraciones con que se encuentra la solución D es mayor para el algoritmo AS que para el algoritmo EAS.
- En general el algoritmo AS generó los peores resultados. Lo anterior se puede deber a que todas las hormigas actualizan los niveles de feromona. De esta manera, sin importar qué tan bueno sea el camino, se puede acumular demasiada feromona, lo cual puede propiciar que las hormigas seleccionen caminos más largos.
- En contraposición al punto anterior, los algoritmos EAS, MMAS y ACS permiten que la acumulación de feromona sea mayor en las mejores rutas. Lo anterior permite que las mejores soluciones continúen siendo elegidas en próximas iteraciones.

De esta manera, son los algoritmos MMAS y ACS los que encuentran mejores soluciones en comparación con los algoritmos EAS y AS. Pues en promedio, de los cinco circuitos: con 24.6% de dichas configuraciones se encuentra la mejor solución, al aplicar el algoritmo AS; para el caso del algoritmo EAS es con 60.0% de las configuraciones que se logra dicho resultado; para el algoritmo MMAS, lo anterior ocurre con 90.7% de las configuraciones; mientras que para el algoritmo ACS es con 91.8%.

7.1.2 Rediseño de Rutas de Recolección en Ciudad Universitaria

En esta Sección se presentan el rediseño de las rutas de recolección para cada uno de los cinco circuitos. Así como la comparación con las rutas diseñadas por la DGOyC en cuanto a distancia recorrida.

En la Tabla 7.10 se muestra el orden a seguir en cada ruta encontrada a partir de los algoritmos ACO, recordando que la sucesión de puntos de recolección actual es en orden ascendente. En las Figuras A.2, A.4, A.6, A.8 y A.10 se presentan los mapas en los que se observa la ruta descrita en la Tabla 7.10. En algunos mapas no se muestra la ubicación exacta del depósito y/ o estación de transferencia debido a la lejanía de su ubicación.

En la Tabla 7.11 se compara la distancia total recorrida para cada ruta. En la columna Diseño actual, la distancia se calcula a partir del orden de visita de los puntos de recolección que siguen actualmente los camiones. En la columna Rediseño de rutas, la distancia es la obtenida con los algoritmos ACO. En comparación con las rutas actuales, la solución propuesta genera una reducción de 37.4 km, equivalente a 24.5%. Tal distancia no está distribuida uniformemente entre los cinco Circuitos, pues se observa una mayor reducción para el C. Deportivo (43.7%), seguido de los Circuitos Exterior y Escolar con 35.3% y 26.1%, respectivamente, mientras que para el C. de la Investigación Científica se reduce la distancia en 10.7%, por último la menor reducción que se tiene es para el C. CCU (7.5%).

Tabla 7.10: Secuencia de visita en los circuitos de CU obtenida con los algoritmos ACO

Circuito	Sucesión de puntos de recolección a visitar en la ruta
Circuito Deportivo	Depósito → 4 → 5 → 3 → 1 → 2 → 16 → 15 → 11 → 12 → 13 → 14 → 6 → 7 → 8 → 9 → 10 → Estación de Transferencia
Circuito Exterior	Depósito → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 11 → 10 → 9 → 12 → 13 → 14 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 22 → Estación de Transferencia
Circuito de la Investigación Científica	Depósito → 14 → 15 → 16 → 17 → 18 → 19 → 21 → 20 → 8 → 5 → 6 → 7 → 9 → 10 → 11 → 13 → 12 → 1 → 2 → 3 → 4 → 22 → 23 → Estación de Transferencia
Circuito Escolar	Depósito → 19 → 20 → 1 → 2 → 3 → 22 → 23 → 24 → 11 → 12 → 13 → 10 → 4 → 18 → 17 → 16 → 5 → 21 → 6 → 7 → 8 → 9 → 14 → 15 → Estación de Transferencia
Circuito CCU	Depósito → 1 → 7 → 20 → 19 → 18 → 17 → 16 → 15 → 14 → 12 → 13 → 11 → 10 → 8 → 9 → 3 → 4 → 6 → 5 → 2 → 28 → 29 → 30 → 24 → 27 → 26 → 25 → 22 → 21 → 23 → Estación de Transferencia

Tabla 7.11: Comparación entre rutas actuales y rediseño de rutas en Ciudad Universitaria

Circuito	Rediseño (ACO)	Diseño actual (DGOyC)
Circuito Deportivo	30,198 mts	43,407 mts
Circuito Exterior	28,596 mts	38,698 mts
Circuito de la Investigación Científica	28,993 mts	32,081 mts
Circuito Escolar	33,158 mts	41,808 mts
Circuito CCU	31,740 mts	34,134 mts
Total	152,685 mts	190,128 mts

7.2 Resultados para ACVRP

Como se mencionó en el capítulo anterior, para proponer un nuevo diseño de rutas de recolección en Ciudad Universitaria (CU), se utilizaron los algoritmos MMAS y ACS. Además de que, para dicho diseño, el problema se planteó como un ACVRP (*Asymmetric Capacitated Vehicle Routing Problem*) debido a que sus características son similares a las de nuestro caso de estudio. En esta sección se presentan los resultados obtenidos con dichos algoritmos, al ser aplicados al ACVRP propuesto para encontrar un conjunto de rutas de recolección en la Zona Escolar, así como en la Zona Cultural de CU. Adicionalmente, ambos algoritmos son aplicados con y sin las heurísticas: 3 – *opt*, operador de intercambio y operador de relocalización, que fueron presentadas en el capítulo anterior.

Es importante hacer notar que para el caso del ATSP, se encontraron valores propuestos en la literatura para los parámetros involucrados en cada algoritmo, de tal forma que ayuden a obtener mejores resultados. Lo anterior, no pasa en el caso del ACVRP, por lo que en primer lugar se hizo un pre-ajuste de parámetros utilizando un par de instancias cuyas características son parecidas a las del problema que en esta tesis se plantea. El par de instancias antes mencionado fue extraído del sitio <http://soa.iti.es/instancias-problemas>. En dicho sitio se encuentra una variedad de problemas con características particulares, que se basan en los factores que se listan en la Tabla 7.12. A continuación se explican de manera muy general tales factores y el porqué se escogieron las dos instancias que en el sitio pueden ser encontradas bajo el nombre de: *P_CA_1001* y *P_CA_0501*.¹

¹Una explicación a profundidad sobre la generación de los problemas se puede ver en Rodríguez y Ruiz, 2012.

Tabla 7.12: Características de las instancias

Factor	Explicación
Tipo de área geográfica	Factor que hace referencia a las distancias entre nodos, las cuales pueden ser cortas, medianas o grandes
Ubicación de los nodos en el territorio	Factor que determina la distribución de los nodos en el territorio. Dicha distribución puede ser aleatoria, radial o en cuadrícula.
Número de nodos	Cantidad de nodos en el territorio: 50, 100, 150, 200, 250,..., 500
Número de instancia	Se tiene un conjunto de 5 problemas por cada conjunto de características
Ubicación del depósito	Se refiere a en qué parte del territorio se encuentra el depósito: en el centro de todos los nodos, tiene una ubicación aleatoria o en la periferia, es decir en los límites del territorio.
Tipo de matriz	Factor que hace referencia a la distancia entre dos nodos, la cual puede ser simétrica ($d_{ij} = d_{ji}$) o asimétrica ($d_{ij} \neq d_{ji}$)
Demanda y Máxima capacidad del vehículo	Ambos factores pueden ser bajos o grandes; la demanda y la capacidad bajas son utilizadas para distancias cortas generalmente.

Para poder elegir una instancia que se asemejara más a nuestro caso de estudio, tomamos en cuenta lo siguiente: en el caso del primer factor, se eligieron distancias cortas, pues en CU los puntos de recolección no se encuentran muy alejados unos de otros. En relación al segundo factor, la ubicación de los nodos es aleatoria, pues en CU los puntos de recolección no se ubican ni de manera radial ni en cuadrícula. Dado que en la Zona Escolar se encuentran 85 puntos de recolección y en la Zona Cultural 30, se eligieron las instancias con 100 y 50 nodos. En relación al factor número de instancia, en ambos casos, se eligió la primera. Por otro lado, como el depósito en cada zona se ubica cerca de los límites de CU, se eligió una ubicación en la periferia. Adicionalmente, dado que en nuestro caso de estudio las distancias no son simétricas, el sexto factor se eligió como matriz asimétrica; y por último, la demanda y capacidad seleccionadas, son bajas, pues en los puntos de recolección la generación de residuos no es demasiado alta. De esta manera, los nombres de las instancias P_CA_1001 y P_CA_0501 tienen el siguiente significado: la primer letra P hace referencia a la localización del depósito; la segunda letra C está relacionada al tipo de distancia entre nodos; la tercer letra A se debe a la localización de los nodos; los tres números siguientes dicen qué número de nodos hay en la instancia y la última cifra se refiere al número de instancia.

En el caso de la instancia con 50 nodos, el problema será llamado P_CA_050 , mientras que en el caso con 100 nodos, el problema será referido como P_CA_100 .

Una vez seleccionadas las instancias, los algoritmos ACS y MMAS fueron corridos 30 veces, con 5000 iteraciones en cada corrida. Para el ajuste de parámetros se tomaron como base los valores que se muestran en la Tabla 7.13. Estos valores son los propuestos en la literatura, pero para el problema del agente viajero (TSP).

Tabla 7.13: Valores constantes de los parámetros

Algoritmo	α	β	ρ	m	q_0	ψ	τ_0
ACS	1	3	0.1	$n - 1$	0.9	0.1	$1/nC^{nn}$
MMAS	1	3	0.1	$n - 1$	-	-	$1/nC^{nn}$

Adicional a los valores mostrados en la Tabla 7.13, en el caso del algoritmo MMAS se utiliza el mecanismo

de reinicialización de feromona. Dicho mecanismo es activado, si después de 250 iteraciones la mejor solución se estanca en un óptimo local. También, para este algoritmo, se decidió utilizar la mejor solución global para actualizar la feromona. Lo anterior, debido a que en la literatura se menciona que, para problemas grandes, es mejor utilizar esa solución en vez de la solución local.

Por otra parte, tanto para el algoritmo MMAS como para el algoritmo ACS, se comparan los resultados al añadir: una heurística (β -opt, relocalización e intercambio); dos heurísticas (β -opt-relocalización, β -opt-intercambio y relocalización-intercambio) y las tres heurísticas (β -opt-relocalización-intercambio). Vale la pena mencionar que, en todos los casos, las heurísticas solamente son aplicadas a la mejor solución de la iteración. Adicionalmente cuando se añaden 2 o 3 heurísticas, éstas se aplican en el siguiente orden: intercambio, relocalización y β -opt. Por último, como ya se mencionó en el capítulo anterior, en la implementación de los algoritmos se utiliza una Lista de Candidatos. Dicha Lista tiene una cardinalidad de 15 vecinos, esto debido a los resultados obtenidos por Bell y McMullen, 2004. Sin embargo, también se hacen experimentos con una cardinalidad de 10 y 20 vecinos.

7.2.1 Ajuste de Parámetros

Para ajustar los parámetros se realizó un experimento por cada configuración de los mismos, en donde un parámetro es modificado a la vez, mientras que los demás permanecen constantes tomando los valores de la Tabla 7.13. Los diferentes valores que pueden tomar los parámetros de cada algoritmo se observan en la Tabla 7.14.

Tabla 7.14: Valores variables de los parámetros de los algoritmos MMAS y ACS

Algoritmo(s)	Parámetros	Valores
MMAS y ACS	α	{0.1, 3, 5, 8, 10}
MMAS y ACS	β	{0.1, 1, 5, 8, 10}
MMAS y ACS	ρ	{0.01, 0.3, 0.5 0.8, 0.99}
MMAS y ACS	m	100 nodos: {10, 20, 40, 60, 80} 50 nodos: {5, 10, 20, 30, 40}
ACS	ψ	{0.01, 0.3, 0.5 0.8, 0.99}
ACS	q_0	{0.01, 0.3, 0.5 0.8, 0.99}

Por otra parte, también se analiza el comportamiento del algoritmo MMAS: al utilizar o no el mecanismo de reinicialización de feromona y al utilizar la solución global para actualizar la feromona en lugar de la solución local. Adicionalmente, se hace la experimentación del desempeño de los algoritmos MMAS y ACS en combinación con una o más heurísticas y con diferentes tamaños de la Lista de Candidatos.

Como ya se mencionó, los algoritmos fueron corridos 30 veces para resolver ambas instancias. La mejor solución global final se registró para cada corrida y se calculó la media de las 30 soluciones finales generadas, así como su desviación estándar, mínimo y máximo. Estas cuatro estadísticas se presentan en las Tablas C.21, C.22, C.23 y C.24.

Para elegir las configuraciones de parámetros a utilizar el los algoritmos MMAs y ACS aplicados a nuestro caso de estudio, se hizo un análisis de varianza (ANOVA) por cada configuración de parámetros, con el fin de determinar si los menores promedios son significativamente mejores que los otros. La hipótesis nula está determinada como *no hay diferencia en las medias de las 30 soluciones para los diferentes valores de cada parámetro* y un $\alpha=0.05$ ha sido utilizado en la prueba ANOVA. Lo anterior, significa que si el p -value asociado a cada prueba es menor a dicha α , se rechaza la hipótesis nula. Lo cual, a su vez, nos dice que los resultados

obtenidos difieren significativamente entre al menos dos configuraciones de parámetros. De esta manera, si se obtienen diferencias significativas, basta ver qué promedio es menor para concluir cuáles son las mejores configuraciones de parámetros.

En la columna *Significancia* (Sig.) de las Tablas C.21, C.22, C.23 y C.24, se observan los resultados obtenidos: por ejemplo, A>B,C; quiere decir que la configuración de parámetros señalado en la fila A, es significativamente mejor que los de las filas B y C (lo mismo aplica para las letras minúsculas). Cabe mencionar que únicamente se indican las configuraciones que son mejores que otras.

Dado lo anterior, se eligieron las configuraciones de parámetros que mostraban ser significativamente mejores que otras. Por ejemplo, en el caso del algoritmo MMAS aplicado al problema P_CA_050 , se observa que al utilizar un α igual a 1 el promedio de soluciones es significativamente mejor que si α toma los valores de 0.1, 8 y 10 (Tabla C.21). Y, aunque no es significativamente mejor que si α es igual a 3 y 5, este par de valores no dan mejores resultados que 8 y 10. De esta manera, se eligió utilizar α igual a 1 para el algoritmo señalado. En las Tablas C.21 y C.23, se pueden observar los resultados obtenidos de todas las pruebas, a los cuales se les aplicó el mismo razonamiento antes descrito. De esta manera, las configuraciones que se muestran en la Tabla 7.15 son las que otorgan los mejores resultados tanto para el problema P_CA_050 , como para el problema P_CA_100 .

Tabla 7.15: Configuraciones de parámetros para el algoritmo MMAS

ID	Problema	Parámetros						Lista de Candidatos	Solución que actualiza feromona	Reinicialización de feromona
		α	β	ρ	m	τ_0				
a	P_CA_050	1	8	0.1	$n - 1$	$1/nC^{nn}$	15	Global	Sí	
b	P_CA_050	1	10	0.1	$n - 1$	$1/nC^{nn}$	15	Global	Sí	
c	P_CA_050	1	3	0.01	$n - 1$	$1/nC^{nn}$	15	Global	Sí	
d	P_CA_050	1	3	0.1	$n - 1$	$1/C^{nn}$	15	Global	Sí	
e	P_CA_050	1	3	0.1	$n - 1$	$1/nC^{nn}$	15	Local	Sí	
A	P_CA_100	1	8	0.1	$n - 1$	$1/nC^{nn}$	15	Global	Sí	
B	P_CA_100	1	10	0.1	$n - 1$	$1/nC^{nn}$	15	Global	Sí	
C	P_CA_100	1	3	0.01	$n - 1$	$1/nC^{nn}$	15	Global	Sí	
D	P_CA_100	1	3	0.1	$n - 1$	$1/nC^{nn}$	15	Global	Sí	
E	P_CA_100	1	3	0.3	$n - 1$	$1/nC^{nn}$	15	Global	Sí	
F	P_CA_100	1	3	0.1	$n - 1$	$1/C^{nn}$	15	Global	Sí	
G	P_CA_100	1	3	0.1	$n - 1$	$1/nC^{nn}$	10	Global	Sí	
H	P_CA_100	1	3	0.1	$n - 1$	$1/nC^{nn}$	15	Local	Sí	

En cuanto al algoritmo ACS, en las Tablas C.22 y C.24 se presentan los resultados obtenidos al aplicar la prueba ANOVA. En este caso, las configuraciones a ser aplicadas en el caso de estudio también fueron elegidas al ver cuáles eran significativamente mejores que otras. Es así que, las configuraciones de parámetros que generan mejores resultados se muestran en la Tabla 7.16.

Por último, en todos los casos se obtuvieron mejores resultados al aplicar las tres heurísticas en conjunto a la mejor solución. Por lo que, dichas heurísticas serán incorporadas en conjunto a cada configuración de parámetros.

Tabla 7.16: Configuraciones de parámetros para el algoritmo ACS

ID	Problema	Parámetros					Lista de Candida- tos	q_0	ψ
		α	β	ρ	m	τ_0			
f	P_CA_050	3	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
g	P_CA_050	5	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
h	P_CA_050	8	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
i	P_CA_050	1	10	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
j	P_CA_050	1	3	0.8	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
k	P_CA_050	1	3	0.99	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
l	P_CA_050	1	3	0.1	$n - 1$	$1/2nC^{nn}$	15	0.9	0.1
m	P_CA_050	1	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.01
n	P_CA_050	1	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
I	P_CA_100	3	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
J	P_CA_100	5	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
K	P_CA_100	8	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
L	P_CA_100	1	8	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
M	P_CA_100	1	10	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
N	P_CA_100	1	3	0.99	$n - 1$	$1/nC^{nn}$	15	0.9	0.1
O	P_CA_100	1	3	0.1	$n - 1$	$1/2nC^{nn}$	15	0.9	0.1
P	P_CA_100	1	3	0.1	$n - 1$	$1/nC^{nn}$	10	0.9	0.1
Q	P_CA_100	1	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.01
R	P_CA_100	1	3	0.1	$n - 1$	$1/nC^{nn}$	15	0.9	0.1

7.2.2 Diseño de Rutas de Recolección en Ciudad Universitaria

En esta sección se presentan los resultados obtenidos al aplicar los algoritmos MMAS y ACS al problema de recolección visto como un ACVRP. A su vez, los resultados se dividen en dos, pues primero se optimiza la Zona Escolar y después la Zona Cultural. En ambos casos, primero se muestran los resultados obtenidos con los algoritmos MMAS y ACS, después se hace una comparación con las rutas diseñadas por la DGOyC y, en el caso de la Zona Escolar, también se hace una comparación con las rutas diseñadas al plantear cada subproblema o circuito como un ATSP.²

Para la Zona Escolar se probaron 8 configuraciones de parámetros en el algoritmo MMAS y 10 configuraciones en el algoritmo ACS. Mientras que para la Zona Cultural se probaron 5 y 9 configuraciones para el algoritmo MMAS y ACS, respectivamente. Dichas configuraciones son las mostradas en las Tablas 7.15 y 7.16.

■ Zona Escolar

En las Tablas 7.17 y 7.18, se observan los resultados obtenidos con los algoritmos MMAS y ACS, según las configuraciones de parámetros. En primer lugar, se observa que con ambos algoritmos se encuentra una solución mínima (mín.) muy parecida: 120 746 mts con MMAS (configuración D) y 120 743 mts con ACS (configuración M). Ahora bien, a pesar de que la diferencia entre ambas soluciones es prácticamente nula, el promedio de las 30 soluciones finales generadas, así como la desviación estándar, es menor para la configuración

²Diremos algoritmos ACO para referirnos a los algoritmos utilizados para resolver el problema planteado como ATSP.

del algoritmo ACS. Lo anterior sugiere que con dicho algoritmo y configuración, se encuentra un conjunto de rutas de menor distancia en comparación con el diseño de rutas del algoritmo MMAS y la configuración respectiva.

Tabla 7.17: Resultados obtenidos con MMAS

ID	μ	σ	mín
A	122,681.70	622.79	121,292.00
B	122,231.07	682.45	120,847.00
C	124,322.70	1 208.95	121,325.00
D	123,768.90	1 232.12	120,746.00
E	124,274.97	1 174.86	122,235.00
F	124,126.67	1 285.22	121,903.00
G	123,667.17	1 169.26	121,646.00
H	122,117.10	525.42	120,795.00

μ : media σ : desviación estándar mín: valor mínimo

Tabla 7.18: Resultados obtenidos con ACS

ID	μ	σ	mín
I	125,279.63	904.23	123,202.00
J	125,069.83	955.17	123,182.00
K	137,138.57	1 117.85	134,384.00
L	122,732.70	418.81	122,060.00
M	122,336.13	514.82	120,743.00
N	124,824.23	947.29	122,025.00
O	125,531.80	832.54	124,023.00
P	125,214.37	753.73	123,096.00
Q	124,533.57	762.06	122,993.00
R	126,244.53	758.72	124,878.00

μ : media σ : desviación estándar mín: valor mínimo

Por otra parte, vale la pena mencionar que entre las 10 configuraciones propuestas para el algoritmo ACS, la que genera el menor promedio es la misma que genera la solución mínima (configuración M). El comportamiento antes descrito no se observa para el algoritmo MMAS pues la solución mínima (configuración D) está relacionada a un promedio mayor en comparación con las configuraciones A, B, G y H.

Sin tomar en cuenta la configuración M del algoritmo ACS, se pueden observar en general mejores resultados con el algoritmo MMAS, pues los promedios como las soluciones mínimas, son menores en comparación con las del algoritmo ACS. Sin embargo, como ya se mencionó, es con la configuración M del algoritmo ACS que se obtiene una mejor solución mínima.

Entre las 18 soluciones finales encontradas (8 con MMAS y 10 con ACS), se eligieron las dos mejores (configuraciones D y M) para hacer la comparación con el diseño de rutas que actualmente opera en Ciudad Universitaria. Cabe mencionar, que en las dos mejores soluciones finales el diseño de rutas consta de cuatro circuitos. Lo anterior es importante, dado que sólo se disponen de cuatro camiones para llevar a cabo la recolección de residuo inorgánico en la Zona Escolar de CU. A continuación, se presentan las características asociadas al diseño de rutas obtenido con MMAS y ACS. Después se comparan ambos diseños con las rutas trazadas por la DGOyC, así como el rediseño de rutas al plantearse el problema como un ATSP.

En la Tabla 7.19, se observan varias características relacionadas a cada una de las cuatro rutas generadas por cada uno de los dos algoritmos. La primer característica representa el número de puntos agrupados en las rutas. La información anterior sirve para analizar la distribución de puntos de recolección en cada ruta generada. Se puede decir que es mejor tener rutas con un número de puntos de recolección parecido entre ellas, pues así la diferencia entre visitas que hace cada camión no varía en gran cantidad de una ruta a otra. En el caso del algoritmo MMAS, los porcentajes nos dicen que el 20 % del total de puntos de recolección (85) se encuentran en la Ruta 2, 24.7 % en la Ruta 4, 27.1 % en la Ruta 3 y 28.2 % en la Ruta 1. Por otra parte, en el algoritmo ACS, 22.4 % del total de puntos de recolección pertenecen a la Ruta 2, 23.5 % a la Ruta 1, 25.9 % a la Ruta 3 y 28.2 % a la Ruta 4. Comparando ambos grupos de rutas, se puede decir que el algoritmo ACS distribuye mejor los puntos de recolección entre las cuatro rutas generadas, pues la diferencia entre la ruta de menor tamaño y la de mayor tamaño es de 5.8 %, mientras que para el MMAS la diferencia es de 8.2 %.

No sólo es importante la distribución de puntos de recolección, sino también la distancia recorrida en cada ruta. Los porcentajes de distancia recorrida, que se observan en la Tabla 7.19, están calculados con respecto a la distancia total recorrida entre las cuatro rutas. Dado lo anterior, para el algoritmo MMAS, la distancia

Tabla 7.19: Características de las rutas obtenidas con los algoritmos MMAS y ACS

	Ruta 1	Ruta 2	Ruta 3	Ruta 4	Total
Puntos de recolección	24	17	23	21	85
% Puntos de recolección	28.2	20.0	27.1	24.7	100
Distancia recorrida	33,265	28,566	27,711	31,204	120,746
M % Distancia recorrida	27.6	23.7	22.9	25.8	100
M Volumen recolectado	15.0	15.2	15.2	14.9	60.3
A % Volumen recolectado	24.9	25.2	25.2	24.7	100
S No. de puntos del C. Deportivo	16	0	0	0	16
No. de puntos del C. Exterior	6	5	7	4	22
No. de puntos del C. de la Inv. Científica	0	2	15	6	23
No. de puntos del C. Escolar	2	10	1	11	24
Puntos de recolección	20	19	22	24	85
% Puntos de recolección	23.5	22.4	25.9	28.2	100
Distancia recorrida	30,679	29,195	29,296	31,573	120,743
A % Distancia recorrida	25.4	24.2	24.3	26.1	100
C Volumen recolectado	15.1	15.2	15.2	14.8	60.3
S % Volumen recolectado	25.0	25.2	25.2	24.6	100
No. de puntos del C. Deportivo	8	0	0	8	16
No. de puntos del C. Exterior	11	0	7	4	22
No. de puntos del C. de la Inv. Científica	0	0	13	10	23
No. de puntos del C. Escolar	1	19	2	2	24

recorrida en la Ruta 3 corresponde a un 22.9% de la distancia total, para la Ruta 2 es un 23.7%, para la Ruta 4 un 25.8% y para la Ruta 1 un 27.6%. En cuanto al algoritmo ACS, la Ruta 2 recorre un 24.2% de la distancia total, la Ruta 3 un 24.3%, la Ruta 1 un 25.4% y la Ruta 4 un 26.1%. Aunque las diferencias entre los porcentajes del algoritmo MMAS y ACS son muy pequeñas, las cifras anteriores sugieren que los camiones de las rutas generadas con ACS recorren distancias más parecidas, en comparación con el algoritmo MMAS. De hecho, en el caso del ACS, la diferencia entre la ruta que recorre menor distancia y la que recorre una mayor distancia es de 1.9%; mientras que para el algoritmo MMAS la diferencia es de 4.7%.

Por último, para la comparación del algoritmo ACS y del MMAS, es importante analizar qué porcentaje de residuos es colectado en cada ruta, ya que es importante que haya una distribución equitativa del volumen recolectado. Lo anterior se debe a que es mejor que no haya camiones que se queden muy vacíos o que se llenen a su máxima capacidad. En este ámbito, analizando el algoritmo MMAS, la Ruta 4 recolecta 24.7% del total de residuos generados en la Zona Escolar, la Ruta 1 recolecta un 24.9% y las Rutas 2 y 3 colectan 25.2%, cada una. En el caso del algoritmo ACS, la Ruta 4 colecta 24.6%, la Ruta 1 un 25.0% y las Rutas 2 y 3 un 25.2%, cada una. Las diferencias de distribución de volumen recolectado, es prácticamente nulo entre ambos algoritmos. En general para los dos casos, no se tienen rutas en las cuales los camiones se llenen en exceso, ni rutas en donde los camiones vayan casi vacíos.

Lo anterior indica, que si bien la diferencia entre distancia total recorrida para los algoritmos MMAS y ACS es casi nula, sí hay diferencias en cuanto a distribución de puntos de recolección, así como distribución de distancia recorrida por cada ruta. Esto también se puede observar en las Figuras 7.1 y 7.2, en donde se ve que el área que cubre cada ruta es diferente según el algoritmo con el que se genera el conjunto de rutas. Cabe aclarar, que en el caso del algoritmo MMAS, (Figura 7.1) se ven divididas las Rutas 3 y 4, lo cual sólo se señaló así en el mapa, para no transponer áreas. Lo mismo sucede con la Ruta 1, en el caso del algoritmo

ACS. Por otra parte, vale la pena mencionar que en todas las rutas el depósito es el nodo con nombre *Talleres Centrales de Conservación*.

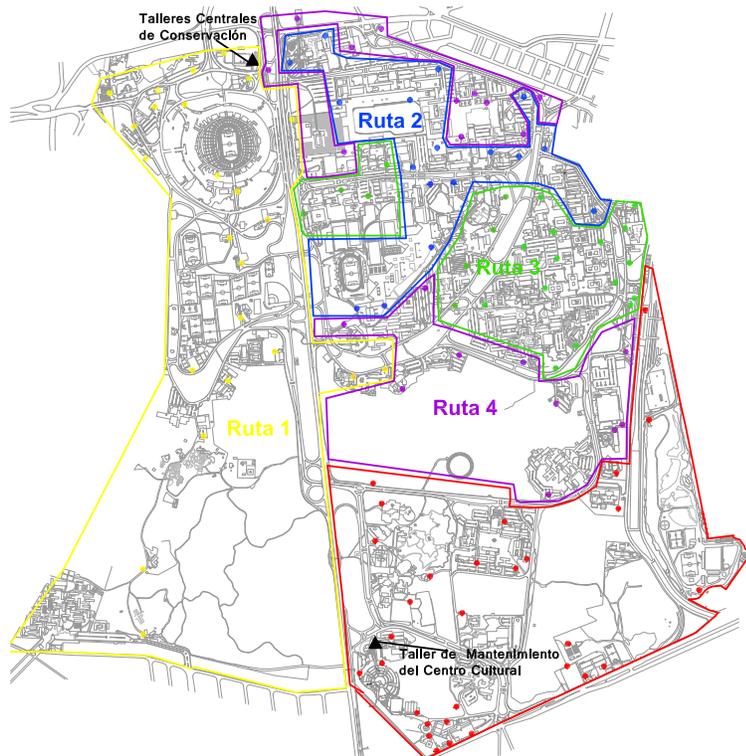


Figura 7.1: Área cubierta por cada una de las cuatro rutas en la Zona Escolar, diseñadas con el algoritmo MMAS

Particularmente, la secuencia de visita de puntos de recolección de cada una de las cuatro rutas generadas con los algoritmo MMAS y ACS se puede observar en la Tabla 7.20. La letra *N* hacer referencia al C. Deportivo, la letra *L* al C. Exterior, la letra *A* al C. de la Investigación Científica y la letra *R* al C. Escolar.

Una vez comparadas las mejores soluciones obtenidas con cada algoritmo, es importante también hacer la comparación con el diseño hecho por la DGOyC y con el rediseño, ya expuesto en una sección anterior, con ACO. En primer lugar, las rutas de la DGOyC, así como las rutas de ACO, tienen el mismo número de puntos de recolección, así como el mismo volumen de residuos recolectado (Tabla 7.21). Lo anterior debido a que con ACO sólo se optimizó el orden en el que los puntos de recolección de cada ruta son visitados. Ambos diseños sólo difieren en la distancia recorrida en cada ruta y, en consecuencia, en la distancia total. En estos diseños, los porcentajes de puntos de recolección agrupados en las rutas son: 18.8% para el C. Deportivo, 25.9% para el C. Exterior, 27.1% para el C. de la Investigación Científica y 28.2% para el C. Escolar. Estas cifras comparadas con las obtenidas con ACS y MMAS (al plantearse le problema como un ACVRP), sugieren que con ambos algoritmos la distribución de puntos de recolección es mejor en comparación con la distribución realizada por la DGOyC. Pues ya se mencionó que la diferencia entre la ruta de menor tamaño y la de mayor tamaño es de 5.8% para ACS y 8.2% para MMAS, mientras que la diferencia en el diseño de la DGOyC es de 9.4%.

En cuanto al volumen recolectado, en la Tabla 7.21 se observan los porcentajes de la DGOyC: 18.7% de los residuos se recolecta en el C. Deportivo, 23.7% en el C. de la Investigación Científica, 27.1% en el C. Exterior y 30.5% en el C. Escolar, de nuevo las diferencias entre estos porcentajes son mayores a las obtenidas con el nuevo diseño de rutas generado con los algoritmos MMAS y ACS. Lo anterior indica que, en el diseño de la DGOyC, los camiones asociados a los circuitos Exterior y Escolar recolectan una mayor cantidad de residuos

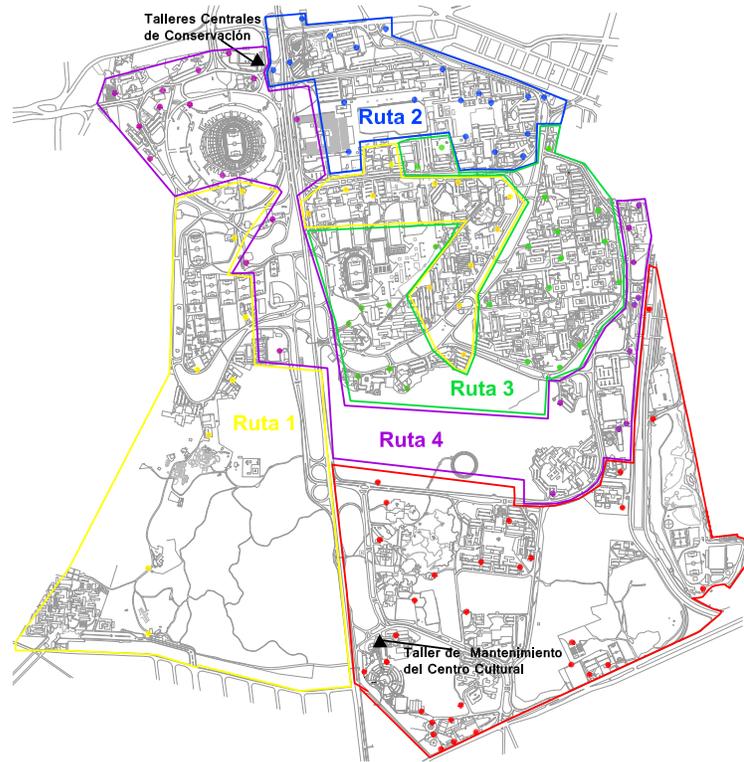


Figura 7.2: Área cubierta por cada una de las cuatro rutas en la Zona Escolar diseñadas con el algoritmo ACS

Tabla 7.20: Secuencia de visita de los puntos de recolección para la Zona Escolar (algoritmos ACS y MMAS)

Algoritmo	Ruta	Sucesión de puntos de recolección a visitar en la ruta
MMAS	Ruta 1	Depósito → L.3 → L.4 → N.4 → N.5 → N.3 → N.16 → R.20 → R.19 → L.2 → N.2 → N.1 → N.15 → L.5 → N.11 → N.12 → N.13 → N.14 → N.6 → N.7 → N.8 → N.9 → N.10 → L.20 → L.21 → Estación de Transferencia
	Ruta 2	Depósito → R.1 → R.4 → R.18 → R.17 → R.16 → R.5 → R.7 → R.8 → R.9 → R.14 → L.9 → L.13 → L.12 → L.10 → L.11 → A.21 → A.20 → Estación de Transferencia
	Ruta 3	Depósito → L.6 → L.7 → L.8 → R.15 → L.14 → L.15 → L.16 → L.17 → A.14 → A.15 → A.16 → A.17 → A.18 → A.19 → A.8 → A.5 → A.6 → A.7 → A.9 → A.10 → A.11 → A.4 → A.22 → Estación de Transferencia
	Ruta 4	Depósito → R.2 → R.22 → R.23 → R.24 → R.3 → R.21 → R.11 → R.12 → R.13 → R.10 → R.6 → L.18 → L.19 → L.22 → L.23 → A.13 → A.12 → A.1 → A.2 → A.3 → A.23 → Estación de Transferencia
ACS	Ruta 1	Depósito → N.8 → N.7 → N.6 → N.9 → N.11 → N.12 → N.13 → N.14 → L.6 → L.7 → L.8 → R.15 → L.9 → L.13 → L.14 → L.15 → L.16 → L.17 → L.18 → L.23 → Estación de Transferencia
	Ruta 2	Depósito → R.1 → R.2 → R.3 → R.22 → R.23 → R.24 → R.21 → R.11 → R.12 → R.13 → R.10 → R.18 → R.17 → R.16 → R.4 → R.5 → R.6 → R.7 → R.8 → Estación de Transferencia
	Ruta 3	Depósito → L.19 → L.20 → L.21 → L.22 → A.14 → A.15 → A.16 → A.17 → A.18 → A.19 → A.21 → R.9 → R.14 → L.12 → L.10 → L.11 → A.20 → A.8 → A.9 → A.10 → A.11 → A.13 → Estación de Transferencia
	Ruta 4	Depósito → N.2 → N.4 → N.5 → N.16 → R.20 → R.19 → L.2 → L.3 → L.4 → N.3 → N.1 → N.15 → L.5 → N.10 → A.1 → A.2 → A.3 → A.4 → A.22 → A.5 → A.6 → A.7 → A.12 → A.23 → Estación de Transferencia

que los circuitos restantes. Más aún, en ambos circuitos se recolecta una cantidad de basura mayor a la capacidad del camión, por lo que en realidad es necesario hacer dos viajes. Por otra parte, con el nuevo diseño de rutas obtenido con los algoritmos MMAS y ACS, se distribuye de manera más equitativa la recolección de residuos entre las cuatro rutas.

En la Tabla 7.21, también se puede observar el número de puntos asociados a los circuitos establecidos por la DGOyC, que se encuentran en cada ruta diseñada con los algoritmos ACS y MMAS. Se puede observar que todos los puntos del C. Deportivo se quedan agrupados en la Ruta 1 obtenida con el MMAS y además dicha ruta incluye otros 6 puntos del C. Exterior. Lo anterior nos dice que el C. Deportivo diseñado por la DGOyC es bastante buena, sin embargo todavía podrían añadirse otros puntos al circuito, ya que en el camión hay espacio para recolectar más residuos. Asimismo, 8 puntos del C. Deportivo quedan asignados a la Ruta 1 diseñada por el algoritmo ACS, y tal ruta incluye también 11 puntos del C. Exterior, es decir, tiene una composición parecida a la Ruta 1 del MMAS.

Los puntos del C. Escolar quedan en su mayoría asignados a la Ruta 2 del algoritmo ACS (19 puntos), es decir, la ruta Escolar también tiene una buena asignación, sin embargo, como ya se mencionó, el volumen del camión es sobrepasado por la cantidad de residuos recolectados en dicha ruta, por lo que es necesario hacer una reasignación. En cuanto a la Ruta 2 del MMAS, la mayoría de sus puntos pertenecen al C. Escolar (10), mientras que los restantes (11 puntos) se asignan a la Ruta 4. De nuevo, el C. Escolar queda dividido, seguramente por tener asociados los puntos de recolección donde más basura se genera y porque el camión asignado a esa ruta no es capaz de recolectar todos los desechos en un sólo viaje.

En cuanto a los puntos del C. de la Investigación Científica, la mayoría de éstos son asignados a la Ruta 3, diseñada tanto por el algoritmo MMAS, como por el algoritmo ACS. De nuevo, la mayoría de los puntos de recolección del C. de la Investigación Científica son asignados a una misma ruta del diseño, pero resulta mejor si se combina con algunos puntos del C. Exterior. De hecho en ambas Rutas 3, se incluyen 7 puntos del C. Exterior.

En cuanto a los puntos del C. Exterior, éstos quedan repartidos entre varias Rutas. En el caso del algoritmo MMAS, se observa que, de dicho circuito, hay puntos en las 4 rutas y el número es más o menos parecido. Lo anterior resulta lógico, pues si se observa la distribución de esos puntos en CU, se puede ver que la mayoría se encuentran en el centro de la Zona, por lo que es fácil que sean redistribuidos. En el caso del algoritmo ACS, los puntos que se agrupaban en el C. Exterior, se distribuyen entre las Rutas 1, 3 y 4, quedándose la mayoría en la Ruta 1. Como ya se comentó, en la Ruta 1 predominan los puntos del C. Deportivo, los cuales son muy cercanos a ciertos puntos del C. Exterior, por lo que es razonable que en esta Ruta 1 se agrupen tales puntos de ambos circuitos.

Finalmente, hay que comparar la distancia total recorrida con el diseño de rutas de la DGOyC, ACO, ACS y MMAS. La distancia total recorrida de las rutas de la Zona Escolar, asociada al diseño de la DGOyC es 155 994 mts, mientras que para ACO es de 120 945 mts, y para MMAS y ACS es 120 746 mts y 120 743 mts, respectivamente. La reducción obtenida con ACO es del 22.5 %, la reducción obtenida con ACS es 22.6 % y con MMAS es también 22.6 %. En primer lugar, como ya se había mencionado, al reordenar la secuencia de visita de los puntos de recolección con ACO, se obtiene una mejora. Lo mismo sucede si el problema es planteado como un ACVRP en donde se busca hacer una nueva asignación de rutas, siendo la mejora de 22.6 %, con respecto a la DGOyC. Ahora bien, la diferencia entre plantear el problema de recolección de residuos como un ATSP o como un ACVRP, es muy pequeña, pues en el primer caso la reducción es de 22.5 % mientras que en el segundo es de 22.6 %. Lo anterior sugiere, que la partición de la Zona Escolar hecha por la DGOyC es adecuada, pues al plantear el problema como un ATSP se respeta la partición de la zona y sólo se cambia el orden de visita. Sin embargo, al incluir la restricción de capacidad, en los circuitos Exterior y Escolar se está recolectando una cantidad de residuos que sobrepasa la capacidad del camión, por lo que en lugar de que se haga un solo recorrido en dicho par de rutas, en realidad es necesario hacer dos recorridos. Lo anterior implica que, al incluir la restricción de capacidad, la distancia asociada tanto al diseño de la DGOyC como al rediseño ACO es mayor a la presentada en la Tabla 7.21. De esta manera, esa diferencia de tan solo

0.1 %, en realidad es más grande, por lo que resulta mejor el diseño obtenido tanto con ACS como con MMAS, al ser planteado el problema como un ACVRP.

Tabla 7.21: Características del conjunto de rutas de la DGOyC y del rediseño obtenido con ACO

	Circuito Deportivo	Circuito Exterior	Circuito de la IC	Circuito Escolar	Total
D Puntos de recolección	16	22	23	24	85
G % Puntos de recolección	18.8	25.9	27.1	28.2	100
O Distancia recorrida	43,407	38,698	32,081	41,808	155,994
y % Distancia recorrida	27.8	24.8	20.6	26.8	100
C Volumen recolectado	11.3	16.3	14.3	18.4	60.3
% Volumen recolectado	18.7	27.1	23.7	30.5	100
Puntos de recolección	16	22	23	24	85
A % Puntos de recolección	18.8	25.9	27.1	28.2	100
C Distancia recorrida	30,198	28,596	28,993	33,158	120,945
O % Distancia recorrida	25.0	23.6	24.0	27.4	100
Volumen recolectado	11.3	16.3	14.3	18.4	60.3
% Volumen recolectado	18.8	27.0	23.7	30.5	100

■ Zona Cultural

Cuando se optimizó la ruta de recolección del circuito CCU, se trabajó bajo el supuesto de que un camión tenía la capacidad suficiente para recolectar los residuos que se generan en todos los puntos asociados a la Zona Cultural. No obstante, la cantidad de residuos generada es mayor a la capacidad del camión y de hecho con dos rutas se cubre toda la zona. Las dos rutas diseñadas por la DGOyC se realizan una por día, debido a que sólo se cuenta con un camión o también se puede realizar una por la mañana y otra por la tarde. Las características de cada una de estas rutas se pueden observar en la Tabla 7.22.

Tabla 7.22: Características de las rutas de recolección de la Zona Cultural

Ruta	Volumen Recolectado	Distancia Total Recorrida	Secuencia de puntos visitados
Ruta A	15.3 m ³	22 053 mts	Depósito → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → 10 → 11 → 12 → 13 → Estación de Transferencia
Ruta B	15.3 m ³	26 476 mts	Depósito → 14 → 15 → 16 → 17 → 18 → 19 → 20 → 21 → 22 → 23 → 24 → 25 → 26 → 27 → 28 → 29 → 30 → Estación de Transferencia

Al tomar en cuenta la cantidad de residuos generada en cada punto de recolección el problema es planteado como un ACVRP. Y, como ya ha sido mencionado, para optimizar las rutas de recolección se utilizaron los algoritmos ACS y MMAS. Las configuraciones de parámetros utilizadas son las que se muestran en las Tablas 7.15 y 7.15, asociadas al problema de 50 nodos.

Los resultados obtenidos con cada una de las configuraciones de parámetros propuestas, se muestran en las Tablas 7.23 y 7.24. En primer lugar, se puede observar que con ambos algoritmos se encuentra una solución mínima parecida, que apenas difiere en 90 mts: 46 076 mts con MMAS (configuración d) y 45 986 mts con ACS (configuración f). Ahora bien, a pesar de que la diferencia entre ambas soluciones es muy pequeña, el promedio de las 30 soluciones finales generadas, así como la desviación estándar, es menor para la configuración del algoritmo ACS. Lo anterior sugiere que, con dicho algoritmo y configuración, se encuentra un conjunto de rutas de menor distancia en comparación con las rutas del algoritmo MMAS y la configuración respectiva.

Tabla 7.23: Resultados obtenidos con MMAS

ID	μ	σ	mín
a	47,769.60	795.82	46,214.00
b	47,881.10	659.30	46,762.00
c	47,665.23	592.30	46,362.00
d	47,857.27	726.93	46,076.00
e	47,198.90	380.82	46,362.00

μ : media σ : desviación estándar mín: valor mínimo

Tabla 7.24: Resultados obtenidos con ACS

ID	μ	σ	mín
f	47,434.23	616.73	45,986.00
g	47,413.40	536.36	46,452.00
h	61,591.83	28.30	61,442.00
i	47,782.30	755.30	46,813.00
j	47,386.57	633.97	46,161.00
k	47,360.03	737.26	46,214.00
l	47,476.47	674.71	46,160.00
m	47,496.43	624.41	46,214.00
n	47,642.50	695.13	46,076.00

μ : media σ : desviación estándar mín: valor mínimo

Por otra parte, vale la pena mencionar que entre las 5 y 9 configuraciones propuestas para los algoritmos MMAS y ACS respectivamente, la configuración que genera el menor promedio no es la misma que genera la solución mínima. En el caso del algoritmo MMAS, la solución mínima (configuración d) está relacionada a un promedio mayor en comparación con las configuraciones a, c y e. Mientras que para el algoritmo ACS, el promedio de la configuración f es superado por las configuraciones g, j y k.

Adicionalmente, sin tomar en cuenta la configuración h del algoritmo ACS, se pueden observar en general mejores resultados con el algoritmo ACS, pues los promedios así como las soluciones mínimas son menores en comparación con las del algoritmo MMAS.

Entre las 14 soluciones finales encontradas (5 con MMAS y 9 con ACS) se eligieron las dos mejores (configuraciones d y f) para hacer la comparación con el diseño de rutas que actualmente opera en Ciudad Universitaria. En ambos casos, MMAS y ACS, el diseño de rutas consta de 2 recorridos, lo cual es importante, ya que es el mismo número de recorridos diseñados por la DGOyC. A continuación, se presentan las características asociadas al diseño de rutas obtenido con MMAS y ACS. Después se comparan ambos diseños con el realizado por la DGOyC.³

En la Tabla 7.25, se observa en primer instancia el número de puntos de recolección agrupados en cada una de las rutas obtenidas con MMAS y ACS. La información anterior sirve para analizar la distribución de puntos de recolección en cada ruta generada. Se puede decir que es mejor tener rutas con un número de puntos de recolección parecido entre ellas, pues así, la diferencia entre visitas que hace cada camión no varía en gran cantidad de una ruta a otra. En ambos casos, algoritmo MMAS y ACS, los porcentajes nos dicen que 50 % del total de puntos de recolección se encuentran en la Ruta A y por lo tanto, el otro 50 % se agrupan en la Ruta B. Es decir, que tanto para el algoritmo MMAS como para el algoritmo ACS, cada ruta agrupa la mitad de puntos de recolección. Más aún, para los dos algoritmos, cada ruta recolecta la mitad del total de residuos sólidos generados. Lo anterior se debe a que las Rutas A y B generadas con los dos algoritmos son prácticamente las mismas (Tabla 7.26), ya que la agrupación de puntos de recolección sólo difiere en uno, lo cual se refleja en la distancia recorrida.

Los porcentajes de distancia recorrida, que se observan en la Tabla 7.25, están calculados con respecto a la distancia total recorrida entre las dos rutas. Dado lo anterior, para el algoritmo MMAS, la distancia recorrida en la Ruta A corresponde a un 52.4 % de la distancia total y para la Ruta B 47.6 %. En cuanto al algoritmo ACS, la Ruta A recorre un 52.0 % de la distancia total y la Ruta B un 48.0 %. Aunque las diferencias entre los porcentajes del algoritmo MMAS y ACS son muy pequeñas, las cifras anteriores sugieren que los camiones de

³Para esta Zona, no se comparan los resultados con los obtenidos al plantearse el problema como un ATSP, pues sólo se utilizaba un solo camión para recolectar todos los residuos.

las rutas generadas con ACS recorren distancias más parecidas, en comparación con el algoritmo MMAS.

Una vez comparadas las mejores soluciones obtenidas con cada algoritmo, es importante también hacer la comparación con el diseño hecho por la DGOyC. En primer lugar, podemos observar que las rutas de la DGOyC agrupan 14 y 16 puntos, cada una, mientras que las rutas obtenidas con ACS y MMAS agrupan 15 puntos cada una. Si bien, la diferencia es prácticamente nula, con ambos algoritmos la distribución de puntos de recolección es mejor en comparación con la distribución realizada por la DGOyC.

En cuanto al volumen recolectado, en los tres casos, cada ruta recolecta la mitad del total de residuos generados en la Zona Cultural. Es decir, que tanto el diseño de la DGOyC, así como el diseño de los algoritmos, distribuye de manera equitativa la recolección de residuos entre las dos rutas.

Adicionalmente, en la Tabla 7.25, se muestra el número de puntos asociados a los circuitos establecidos por la DGOyC, que se encuentran en cada ruta diseñada con los algoritmos ACS y MMAS. Se puede observar que para optimizar la recolección, las rutas ya establecidas por la DGOyC se dividen. De esta manera, la Ruta A, generada con los algoritmos, agrupa 5 puntos de la ruta CCU1 diseñada por la DGOyC y 10 de la ruta CCU2 diseñada por tal dependencia. Mientras que la Ruta B, agrupa 8 puntos de la ruta CCU1 y 7 de la ruta CCU2.

Finalmente, hay que comparar la distancia total recorrida entre el diseño de rutas de la DGOyC y el obtenido con los algoritmos ACS y MMAS. La distancia total recorrida de las rutas de la Zona Cultural, asociada al diseño de la DGOyC, es 48 529 mts, mientras que para MMAS y ACS es 46 076 mts y 45 986 mts, respectivamente. La reducción obtenida con ACS es 5.3 % y con MMAS es 5.5 %. Es así que al plantear el problema como un ACVRP y resolverlo con los algoritmos ACS y MMAS se logra disminuir la distancia total recorrida. Dichos porcentajes son menores al obtenido en el rediseño de rutas (7.5 %). Sin embargo, en el caso del rediseño de rutas no se tomó en cuenta la restricción de capacidad, por lo que al plantear el problema como un ACVRP además de la mejora, se tiene una mejor representación del problema real.

Tabla 7.25: Características de las rutas obtenidas con los algoritmos MMAS y ACS para la Zona Cultural

	MMAS			ACS			DGOyC		
	Ruta A	Ruta B	Total	Ruta A	Ruta B	Total	Ruta CCU1	Ruta CCU2	Total
Puntos de recolección	15	15	30	15	15	30	14	16	30
% Puntos de recolección	50.0	50.0	100.0	50.0	50.0	100.0	46.7	53.3	100
Distancia (mts)	24,147	21,929	46,076	23,903	22,083	45,986	22,053	26,476	48,529
% Distancia recorrida	52.4	47.6	100.0	52.0	48.0	50.0	45.4	54.6	100.0
Volumen recolectado	15.3	15.3	30.6	15.3	15.3	30.6	15.3	15.3	30.6
% Volumen recolectado	50.0	50.0	100.0	50.0	50.0	100.0	50.0	50.0	100.0
No. de puntos de la Ruta CCU1	5	8	13	5	8	13	13	0	13
No. de puntos de la Ruta CCU2	10	7	17	10	7	17	0	17	17

Por último, en la Figura 7.4 se puede observar el área que cubren las Rutas A y B y, a su vez, se pueden comparar con las áreas cubiertas por las rutas CCU1 y CCU2 diseñadas por la DGOyC (Figura 7.3). Además, en la Tabla 7.26, se observa el orden en que se visitan los puntos de recolección, en el caso del diseño con ACS

y el diseño con MMAS.

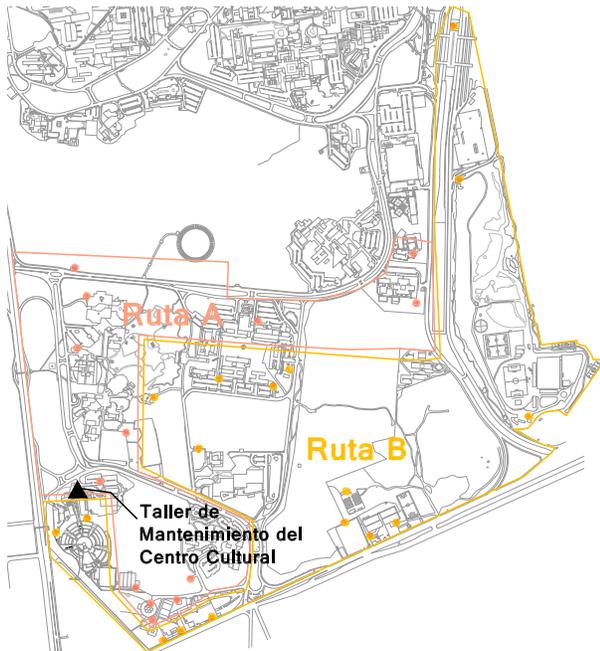


Figura 7.3: Área cubierta por cada una de las dos rutas diseñadas por la DGOyC

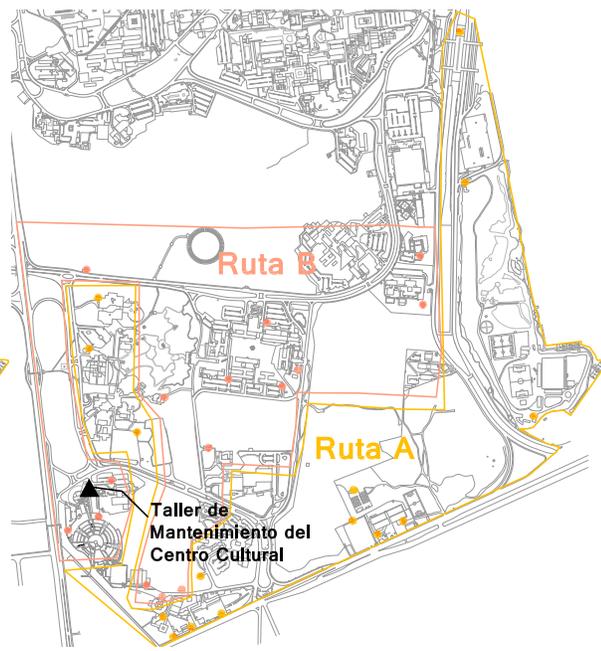


Figura 7.4: Área cubierta por cada una de las dos rutas diseñadas con el algoritmo ACS y el algoritmo MMAS

Tabla 7.26: Secuencia de visita de los puntos de recolección para la Zona Cultural (algoritmos ACS y MMAS)

Algoritmo	Ruta	Sucesión de puntos de recolección a visitar en la ruta
MMAS	Ruta A	Depósito → 7 → 8 → 9 → 5 → 2 → 28 → 29 → 30 → 24 → 25 → 26 → 27 → 22 → 21 → 23 → Estación de Transferencia
	Ruta B	Depósito → 18 → 17 → 16 → 15 → 14 → 12 → 13 → 11 → 10 → 20 → 19 → 1 → 3 → 4 → 6 → Estación de Transferencia
ACS	Ruta A	Depósito → 7 → 8 → 9 → 3 → 2 → 28 → 29 → 30 → 24 → 25 → 26 → 27 → 22 → 21 → 23 → Estación de Transferencia
	Ruta B	Depósito → 18 → 17 → 16 → 15 → 14 → 12 → 13 → 11 → 10 → 20 → 19 → 1 → 4 → 6 → 5 → Estación de Transferencia

Capítulo 8

Conclusiones

Es un hecho que el sistema actual de recolección de residuos sólidos en CU funciona, pues se mantienen limpias las instalaciones y los residuos sólidos son eliminados. No obstante lo anterior, es posible hacer mejoras a dicho sistema, que permitan un ahorro en distancias, tiempo y costos. En este trabajo, se modeló el sistema de recolección de residuos sólidos en CU como un problema de ruteo. En primer lugar se propuso un modelo basado en el problema del agente viajero asimétrico (ATSP) y en segundo lugar se propuso uno basado en el problema de ruteo de vehículos con capacidad asimétrico (ACVRP). Este último, como su nombre lo indica, incluye la capacidad de los vehículos, por lo que representa mejor el sistema real. En ambos casos (ATSP y ACVRP), primero se hizo un diseño de un grafo asimétrico, donde los nodos representan los puntos de recolección y los arcos las vialidades de CU, de tal manera que la dirección de dichas vialidades fue representada por arcos dirigidos.

Dada la complejidad y el costo computacional de los problemas propuestos (ATSP y ACVRP), se utilizaron métodos de solución heurísticos para resolverlos. Es así que, en primer instancia, se utilizaron los algoritmos AS, EAS, MMAS y ACS, para ser aplicados al ATSP. Mientras que para resolver el ACVRP se utilizaron los algoritmos MMAS y ACS, con una serie de modificaciones adecuadas al nuevo planteamiento. Por lo tanto, se considera que el objetivo general de esta tesis se logró, al modelar el problema de recolección de residuos sólidos en Ciudad Universitaria como un problema de ruteo de vehículos y resolverlo con algoritmos basados en ACO. Adicionalmente, se implementaron distintas heurísticas (*3-opt*, relocalización e intercambio), que también fueron adecuadas al problema que se quería resolver y que ayudaron a encontrar mejores soluciones. Dichas heurísticas fueron implementadas para ayudar a los algoritmos ACO a encontrar mejores soluciones y en un menor número de iteraciones. Lo anterior dio como resultado un rediseño de rutas y una nueva propuesta de diseño de rutas que, en ambos casos, minimiza la distancia recorrida por cada vehículo.

Dentro del rediseño de rutas (ATSP), se pudo constatar que los algoritmos MMAS y ACS, son los que encuentran mejores soluciones en comparación con los algoritmos EAS y AS. Pues se observó que, del total de configuraciones de parámetros propuestas para aplicar el algoritmo AS, al tomar en cuenta el promedio de los cinco circuitos, en un 24.6% de las configuraciones se encuentra la mejor solución; para el caso del algoritmo EAS es con 60.0% de las configuraciones que se logra dicho resultado; para el algoritmo MMAS, lo anterior ocurre con 90.7% de las configuraciones; mientras que para el algoritmo ACS es con 91.8%. Particularmente para cada circuito, también son los algoritmos ACS y MMAS los que encuentran la mejor solución con un mayor número de configuraciones de parámetros. Es así que, las diferencias entre estos porcentajes fueron un indicador de que los algoritmos ACS y MMAS encuentran mejores soluciones al problema tratado que el resto de los algoritmos ACO aquí implementados. Aunado a lo anterior, la convergencia a la mejor solución es más lenta con el algoritmo AS, mientras que el algoritmo MMAS muestra la convergencia más rápida a la mejor solución. Estas diferencias pudieron basarse en uno o más de los siguientes factores:

- En el algoritmo AS, todas las hormigas actualizan los niveles de feromona. De esta manera, sin importar qué tan bueno sea el camino, se puede acumular demasiada feromona, lo cual puede propiciar que las hormigas seleccionen caminos más largos.
- En contraposición al punto anterior, los algoritmos EAS, MMAS y ACS permiten que la acumulación de feromona sea mayor en las mejores rutas.
- Por otra parte, una diferencia entre EAS y ACS o MMAS, consiste en que, para los dos últimos, se busca que la cantidad de feromona no crezca desmesuradamente, propiciando así la diversificación. En el caso del MMAS se tienen límites en los niveles de feromona, mientras que para el ACS se tiene una regla de actualización de feromona local, que lleva a la disminución de feromona en los arcos.

Aunado a la búsqueda de una solución en el rediseño de rutas, también se hizo un estudio del efecto de los parámetros en las soluciones generadas. Lo anterior es una de las aportaciones de este trabajo, pues se pudo analizar qué parámetros tienen mayor peso en el algoritmo, así como la influencia del valor que toman, al ser aplicados los algoritmos a un problema real. Dentro del análisis que se hizo, cabe destacar que:

- En primer lugar, los valores de los parámetros que dieron mejores resultados para el AS, EAS y MMAS son cercanos a los propuestos en la literatura. Sin embargo, en el caso del algoritmo ACS, no pasa lo mismo. De hecho, se observó variabilidad en los valores que otorgaban mejores resultados según el circuito al que era aplicado el algoritmo.
- En el caso de los algoritmos EAS y AS los parámetros más sensibles fueron β y m . En el primer caso, se observó como un valor muy pequeño de β arroja resultados muy distantes de los obtenidos con otros valores. En el segundo caso, se observan cambios graduales al ir aumentando el número de hormigas.
- Para el algoritmo MMAS, el parámetro ρ es el más sensible, pues cuando dicho parámetro toma el valor más alto se obtienen los peores resultados entre todas las configuraciones propuestas.
- En cuanto al algoritmo ACS, para los circuitos Deportivo, Exterior, Investigación Científica y Escolar el parámetro más sensible es β ; mientras que para el circuito CCU el parámetro es ρ .
- En general, para los cuatro algoritmos ACO, es mejor si más hormigas generan soluciones, en este caso el mayor número de hormigas fue igual al número de nodos del problema a resolver.
- Se observó que es importante el balance entre la información heurística y el rastro de feromona, pues en casos donde $\alpha=1$ y $\beta=0.1$ los algoritmos AS, EAS y MMAS presentan un mal comportamiento y se estancan en malas soluciones.
- En general, es mejor una adaptación lenta de los algoritmos AS, EAS y MMAS, es decir, cambios graduales en los niveles de feromona, los cuales se obtienen con menores tasas de evaporación.
- Para el algoritmo ACS, es contraproducente que la exploración sea muy poca ($q_0 > 0.95$) ya que se producen los peores resultados en comparación con valores menores del parámetro.
- Es importante asignar valores adecuados a los parámetros según el problema a resolver, ya que esto permite una mejor ejecución del algoritmo.

En comparación con el sistema de recolección de residuos sólidos propuesto por la DGOyC, la distancia total de cada uno de los cinco circuitos establecidos se redujo entre 7.5% y 43.7% al proponer un rediseño de las rutas ya existentes en CU. Los cinco porcentajes, promedian una disminución de 24.5% en la distancia total recorrida en CU (Zona Escolar: 4 circuitos y Zona Cultural: 1 circuito). El porcentaje de disminución que corresponde únicamente a la Zona Escolar, es 22.5% mientras que para la Zona Cultural es de 7.5%.

En cuanto al nuevo diseño de rutas (ACVRP), también se propusieron diferentes configuraciones de parámetros para ser aplicados en los algoritmos ACS y MMAS. En primer lugar, se observó que al cambiar de un problema ATSP a un ACVRP la dificultad para encontrar la mejor solución es mayor. Pues, como ya se mencionó, al resolver el ATSP con ese par de algoritmos, el porcentaje de configuraciones con las que se obtuvo la mejor solución era muy alta para cada uno de los cinco circuitos (mayor al 70%); mientras que para la Zona Escolar al aplicar el algoritmo ACS al ACVRP, sólo con una de las 10 configuraciones ($\alpha=1$, $\beta=10$, $\rho=0.1$, $m = n - 1$, $\tau_0 = 1/nC^{mn}$, $q_0 = 0.9$ y $\psi = 0.1$), se alcanza la mejor solución y en el caso del algoritmo MMAS con ninguna configuración. Por otro lado, para la Zona Cultural, al aplicar el algoritmo ACS, con una de las 9 configuraciones propuestas se encuentra la mejor solución ($\alpha=5$, $\beta=3$, $\rho=0.1$, $m = n - 1$, $\tau_0 = 1/nC^{mn}$, $q_0 = 0.9$ y $\psi = 0.1$) y, de nuevo, con el algoritmo MMAS no se alcanza dicha solución.

Ahora bien, aunque en el nuevo diseño de rutas, el algoritmo ACS arrojó una mejor solución que la que generó el algoritmo MMAS, la diferencia entre ambos es muy pequeña (menor a 0.2%). Sin embargo, el diseño físico de las rutas varía bastante. Lo anterior, se debe a que en CU los puntos de recolección están muy cercanos unos de otros, por lo que al diseñar las rutas, es posible intercambiar uno o más puntos entre dos

rutas y aún así, en ambas no se aumente la distancia total recorrida. Lo anterior supone una ventaja, ya que dentro de una misma ruta se puede modificar el orden de visita en caso de que por alguna situación convenga hacerlo.

Particularmente, al proponer un nuevo diseño de rutas dentro de la Zona Escolar, la mejora que se obtiene con respecto al diseño de la DGOyC es de 22.6 %, porcentaje prácticamente igual al obtenido al proponer el rediseño de rutas (22.5 %). Si bien, esta diferencia es casi nula, hay que tener en cuenta que en el rediseño no se incluyó la restricción de capacidad y en realidad en los circuitos Exterior y Escolar se necesitan hacer dos viajes para cubrir la demanda, por lo que la distancia recorrida aumentaría para la Zona Escolar. En cuanto a la Zona Cultural, la disminución obtenida fue de 5.3 % con ACS y 5.5 % con MMAS, porcentajes menores a los obtenidos con el rediseño de rutas (7.5 %). Sin embargo, como ya se mencionó, en el caso de rediseño no se tomó en cuenta la restricción de capacidad.

En general, ya sea con el rediseño o el nuevo diseño de rutas, los resultados arrojados indican que se pueden hacer mejoras en el sistema de recolección de residuos sólidos que actualmente opera en CU. Pues principalmente se logró minimizar la distancia total recorrida y en consecuencia el costo derivado de la recolección. Se puede decir, que utilizar los algoritmos ACO para resolver este tipo de problemas es viable, ya que además de mejorar el sistema de recolección, la implementación es sencilla y robusta; ya que se pueden hacer modificaciones a los algoritmos de tal manera que exista un mapeo entre ellos y el problema real que se busque resolver.

8.1 Trabajos Futuros

Con los resultados obtenidos se puede mejorar el sistema de recolección de residuos sólidos en Ciudad Universitaria, sin embargo, la información que se obtuvo no es suficiente para conseguir la mejor solución al problema de recolección. Dado lo anterior, es necesario generar información fidedigna relacionada con la recolección, para que las soluciones que se propongan sean más aplicables. Aunado a lo anterior, también se puede introducir un mayor número de restricciones para que el problema planteado refleje de mejor manera la situación real, por ejemplo: ventanas de tiempo, demanda estocástica e información en tiempo real. Adicionalmente, se pueden aplicar los algoritmos presentados para optimizar las rutas de recolección de residuos orgánicos. Más aún, estos modelos y algoritmos pueden aplicarse a áreas de la Ciudad de México, donde también se enfrentan a problemáticas en la planeación de recolección de residuos, por lo que es posible hacer mejoras en los diseños de rutas de este tipo de problemas.

Trabajos publicados de la tesis

- Mancera, E., Garro, B. & Rodríguez, K. (2015). Optimización mediante algoritmo de hormigas aplicado a la recolección de residuos sólidos en UNAM CU. *Research in Computing Science*, 94, 163-177.

Anexos

Anexo A

Mapas de las Rutas de RSU operantes en CU



Figura A.1: Ruta del Circuito Deportivo diseñada por la DGOyC

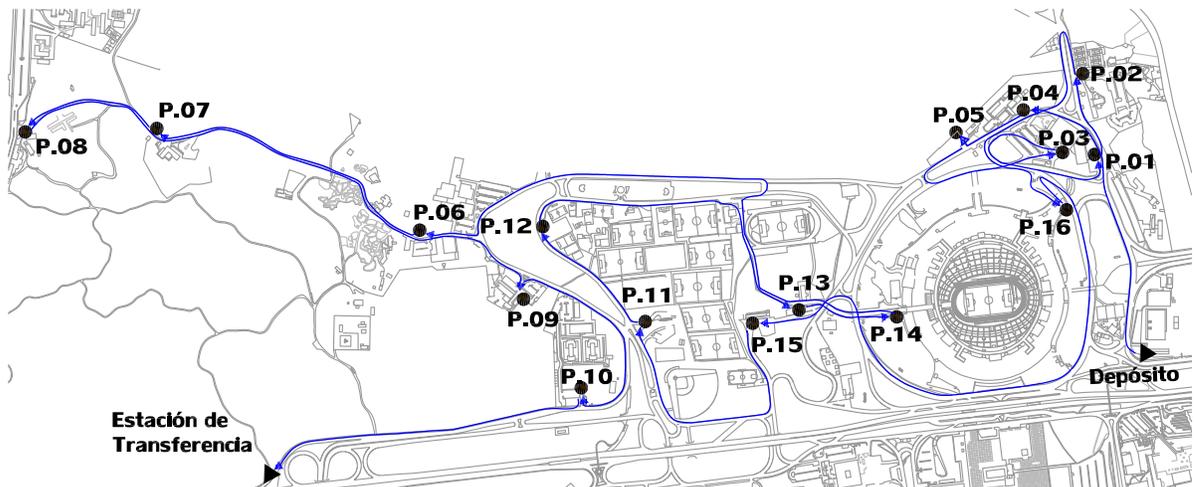


Figura A.2: Rediseño de ruta para el Circuito Deportivo con ACO

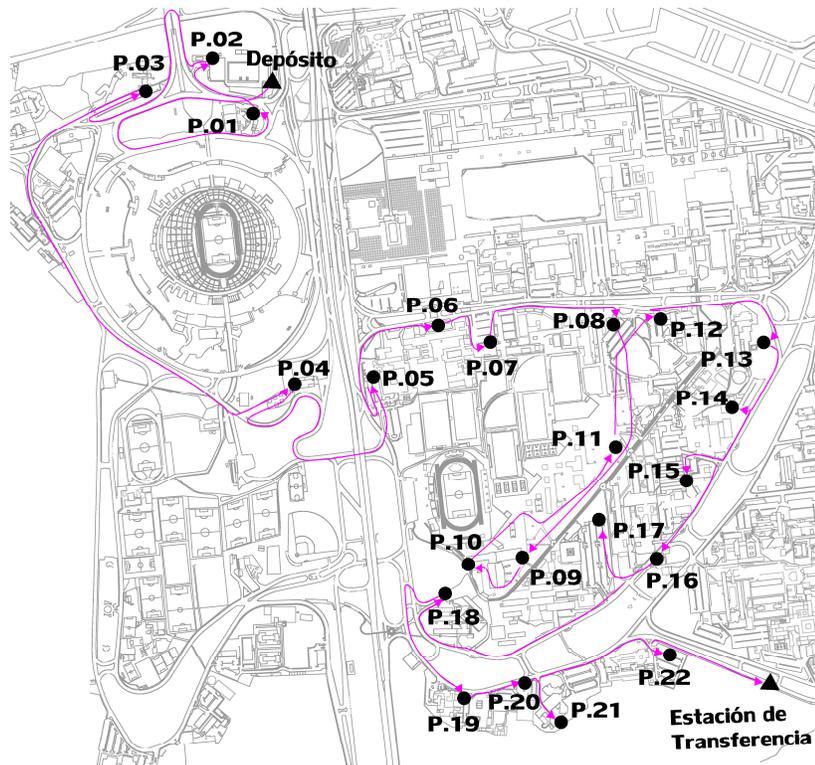


Figura A.3: Ruta del Circuito Exterior diseñada por la DGOyC

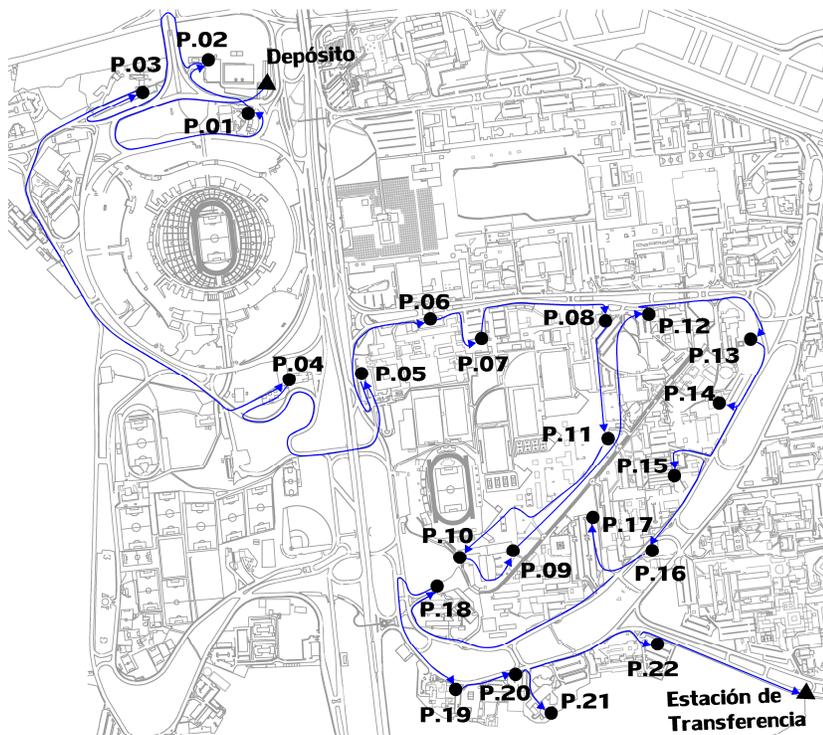


Figura A.4: Rediseño de ruta para el Circuito Exterior con ACO

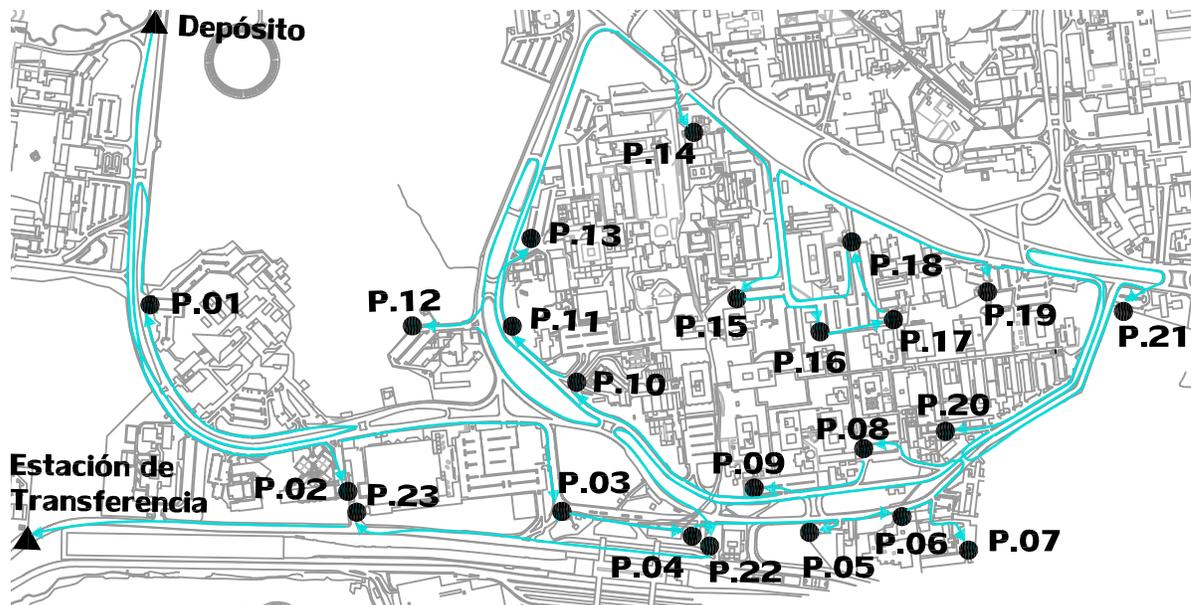


Figura A.5: Ruta del Circuito de la Investigación Científica diseñada por la DGOyC

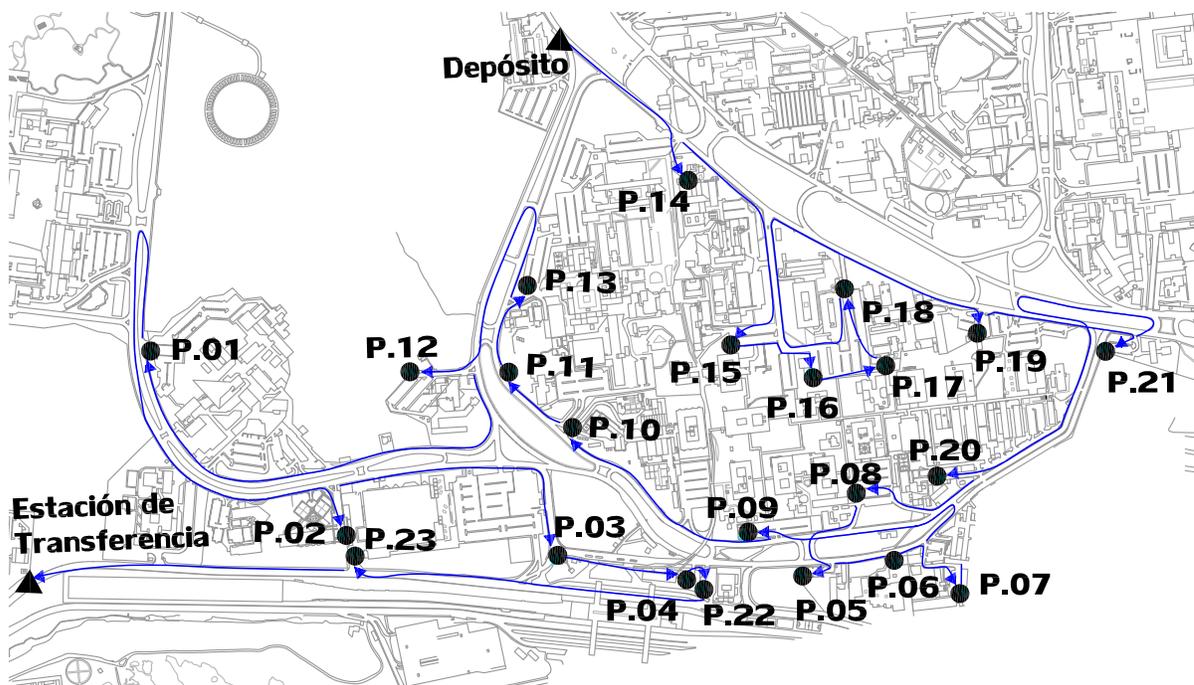


Figura A.6: Rediseño de ruta para el Circuito de la Investigación Científica con ACO

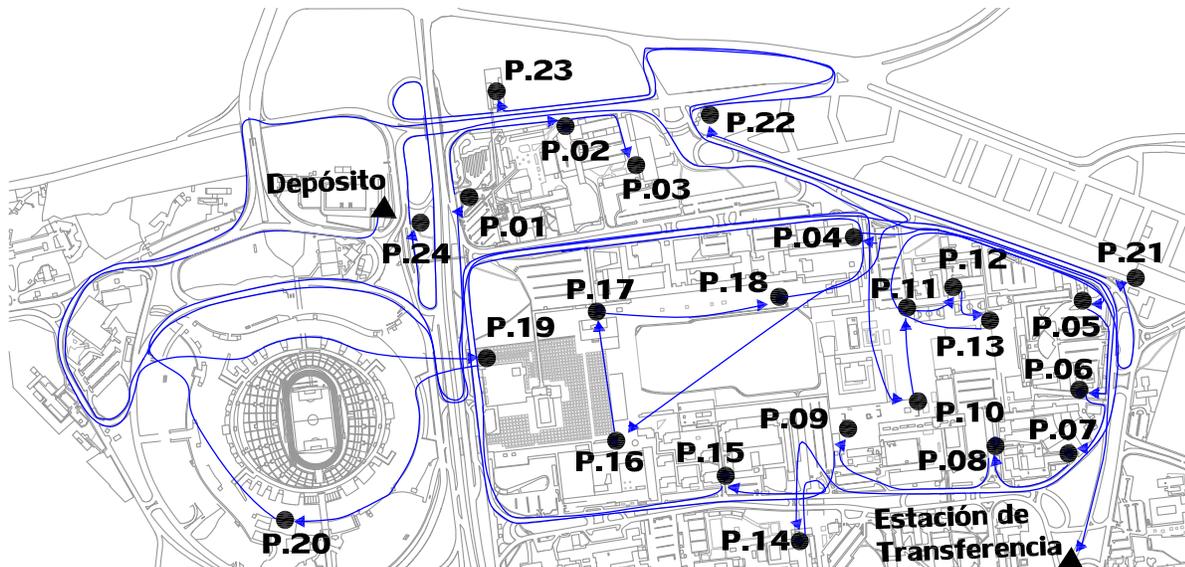


Figura A.7: Ruta del Circuito Escolar diseñada por la DGOyC

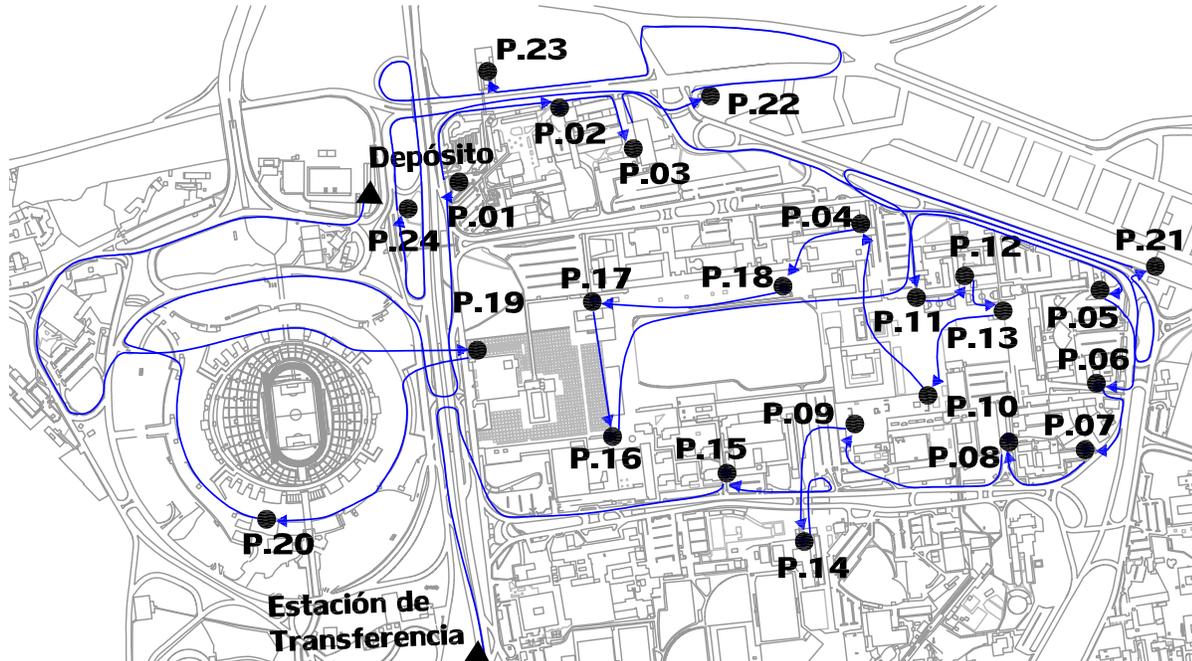


Figura A.8: Rediseño de ruta para el Circuito Escolar con ACO

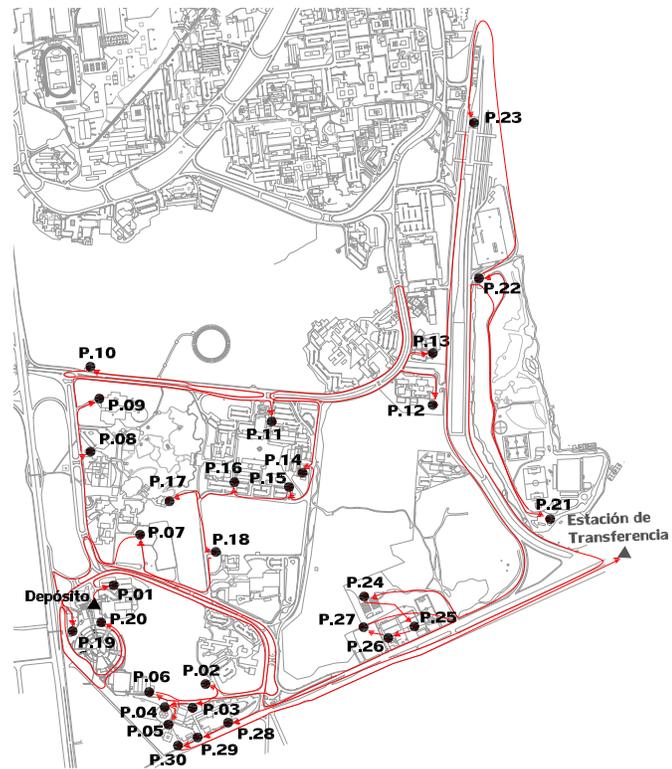


Figura A.9: Ruta del Circuito CCU diseñada por la DGOyC

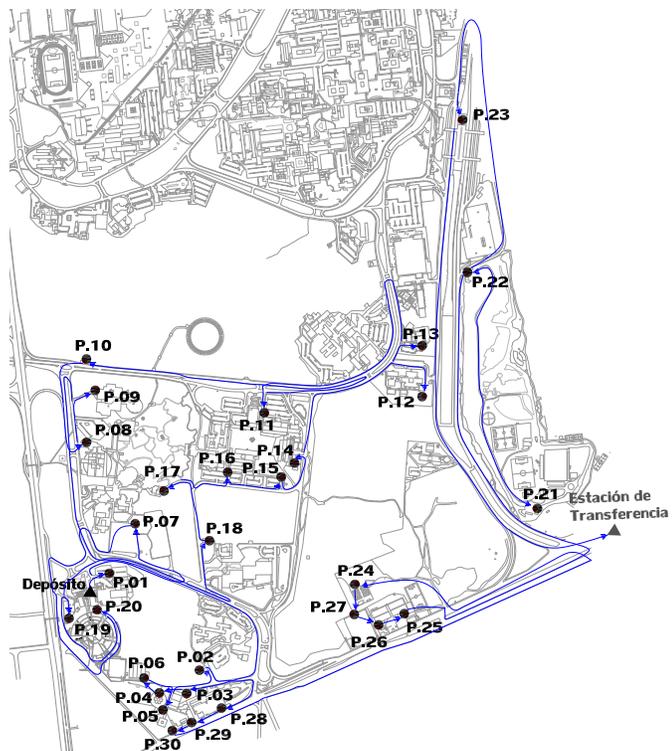


Figura A.10: Rediseño de ruta para el Circuito CCU con ACO

Anexo B

Análisis del efecto de los parámetros en los algoritmos ACO al ser aplicados al problema de recolección de residuos sólidos visto como un TSP

B.1 Algoritmo AS

Parámetro β : Para los cinco circuitos, el algoritmo AS presenta la peor ejecución cuando β toma el valor de 0.1, incluso la evolución del algoritmo no se alcanza a ver en las gráficas (Figuras B.1a, B.2a, B.3a, B.4a y B.5a). Lo anterior se refleja en la Tabla 7.3, pues el error relativo medio más grande (incluso entre todas las configuraciones propuestas para los parámetros), está asociado a β igual a 0.1. Además, para este valor de β se tiene la mayor variabilidad en las soluciones, en comparación con otros valores del parámetro (Tablas C.1, C.2, C.3, C.4 y C.5).

Por otra parte, para los circuitos Deportivo, de la Investigación Científica y CCU se observa que cuando β es igual a 1, el algoritmo presenta la mejor evolución y el menor error relativo medio (Tabla 7.3). Particularmente, en los dos primeros casos es muy claro que, con valores diferentes a 1, la evolución del algoritmo no es tan buena (Figuras B.1a, B.3a y B.5a). Además, vale la pena mencionar que en el caso del C. Deportivo, se tiene la menor variabilidad cuando β toma el valor de 1 (Tabla C.1).

En cuanto a los Circuitos Exterior y Escolar, con cualquier valor de β (excepto 0.1), la convergencia del algoritmo AS es muy rápida, incluso las diferencias que se observan son muy pequeñas. Adicionalmente, en la Tabla 7.3, se observa que el menor error relativo medio se obtiene con β igual a 1 para el C. Exterior y con 5 para el C. Escolar. Además, vale la pena mencionar que en el caso del C. Escolar, se tiene la menor variabilidad cuando β toma el valor de 5 (Tabla C.4).

Para los circuitos Deportivo, Exterior, de la Investigación Científica y Escolar se logra encontrar la solución A, B, C y D, respectivamente, con algún valor de β : con β igual 1 para los circuitos Deportivo y Exterior; con β igual a 3 para el C. de la Investigación Científica; mientras que para el C. Escolar, la solución D se obtiene con 3, 5 y 8. En el caso del C. CCU, la solución E no se alcanza con ningún valor de β .

Para todos los circuitos, excepto el C. CCU, la mayor media de iteraciones necesarias para encontrar la solución A, B, C o D, según sea el caso, corresponde a β igual 0.1. Además, con dicho valor se obtiene la peor evolución del algoritmo, por lo que se puede pensar que hay un estancamiento en malas soluciones (Figuras B.1d, C.2, B.3d, B.4d y C.5).

Parámetro ρ : Para todos los circuitos se observa que las tasas de evaporación más bajas (0.01 y 0.1) generan la mejor evolución del algoritmo, aún cuando en las primeras iteraciones la convergencia es más rápida para tasas más altas (Figuras B.1b, B.2b, B.3b, B.4b y B.5b). De hecho, el menor error relativo medio se obtiene con una tasa de evaporación igual a 0.01; y, para todos los circuitos, excepto el de la Investigación Científica, ese error relativo medio es el más bajo entre todas las configuraciones propuestas para los parámetros (Tabla 7.3). Al tener mejores resultados con tasas de evaporación bajas, se puede pensar que el funcionamiento del algoritmo es mejor al propiciar que la feromona persista más tiempo, es decir, que la adaptación del algoritmo sea lenta.

Adicionalmente, para los circuitos Deportivo, Escolar y CCU, la variabilidad de las soluciones generadas con ρ igual a 0.01 y 0.1 está por debajo de las asociadas a valores más grandes del parámetro, en contraposición con los circuitos Exterior y de la Investigación Científica, donde la mayor variabilidad está asociada a ese par de tasas de evaporación.

Para los circuitos Exterior, de la Investigación Científica y Escolar, sí se logra encontrar la solución B, C y D, respectivamente; en el C. Exterior con ρ igual a 0.01, en el C. de la Investigación Científica, la solución C se alcanza con ρ igual a 0.01, 0.1 y 0.5, mientras que en el C. Escolar, además de los valores mencionados, la solución D se alcanza con ρ igual a 0.3. En los circuitos Deportivo y CCU, no se logra alcanzar la solución A y D, respectivamente, con ningún valor de ρ .

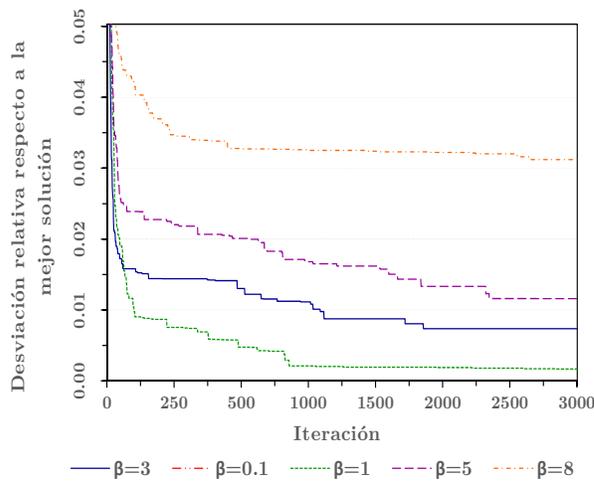
Finalmente, la media de iteraciones en los circuitos Deportivo, Exterior y de la Investigación Científica (excepto para ρ igual a 0.8 en el caso del C. Deportivo), es menor a la de los circuitos más grandes: Escolar y CCU (Figuras B.1d, B.2d, B.3d, B.4d y B.5d).

Parámetro m : En general, en todos los circuitos excepto el de la Investigación Científica, se observa que a mayor número de hormigas mejores son los resultados que se obtienen, en cuanto a calidad y rapidez de convergencia (Figuras B.1c, B.2c, B.3c, B.4c y B.5c). Pues al utilizar un menor número de hormigas, el desempeño del algoritmo se ve afectado debido a la menor diversificación de soluciones. Lo anterior también se ve en la Tabla 7.3, donde el error relativo medio más grande se obtiene con el menor número de hormigas. Así mismo, para el menor número de hormigas se tiene la mayor variabilidad en soluciones finales (Tablas C.1, C.2, C.4 y C.5). Adicionalmente, vale la pena mencionar que para el C. Exterior la convergencia es más rápida en comparación con los otros circuitos (Figura B.2d).

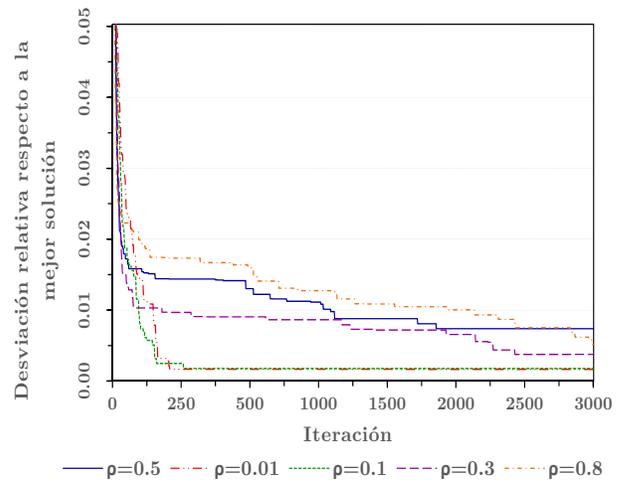
En cuanto al C. de la Investigación Científica, las diferencias en la evolución del algoritmo al variar la cantidad de hormigas son pequeñas. Aunque, el menor número hormigas genera ligeramente la mejor evolución (Figura B.3c) y el menor error relativo medio (Tabla C.3). No obstante lo anterior, con tal número de hormigas no se encuentra la solución C, lo cual sí sucede al utilizar más hormigas (Tabla 7.3), además, con el menor número de hormigas se tiene la mayor variabilidad en las soluciones finales (Tabla C.3).

Adicionalmente, para los circuitos Deportivo, Exterior y CCU no se logra encontrar la solución A, B y E, respectivamente, con ningún número de hormigas. En cuanto al C. Escolar, sí se encuentra la solución D al utilizar 15, 20 y n hormigas (Tabla 7.3).

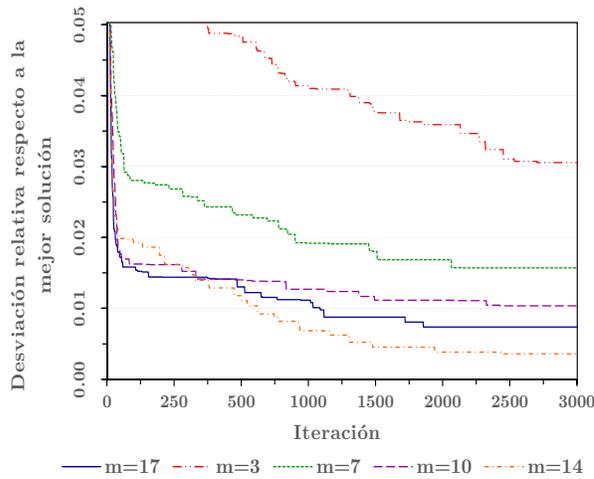
Finalmente, para los circuitos Deportivo, de la Investigación Científica y Escolar, la menor media de iteraciones necesarias para encontrar la mejor solución corresponde al mayor número de hormigas (n). Asimismo, la mayor media corresponde al menor número de hormigas, lo cual puede deberse a que al utilizar menos hormigas, el algoritmo queda estancado en las soluciones encontradas al inicio de la ejecución (Figuras B.1d, Figuras B.3d y B.4d). Por otra parte, la media de iteraciones es mayor para los circuitos Escolar y CCU, en comparación con los otros circuitos.



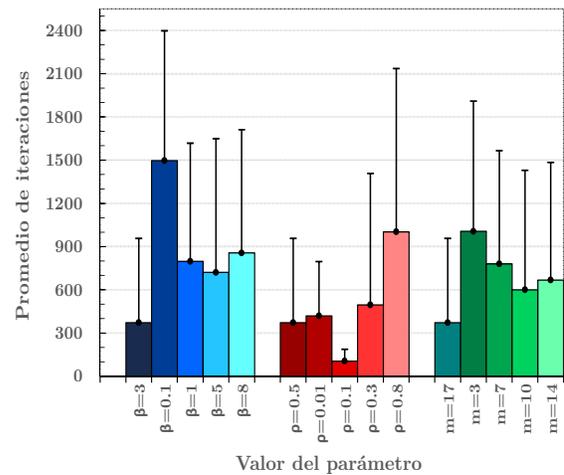
(a) AS con varios valores de β



(b) AS con varios valores de ρ

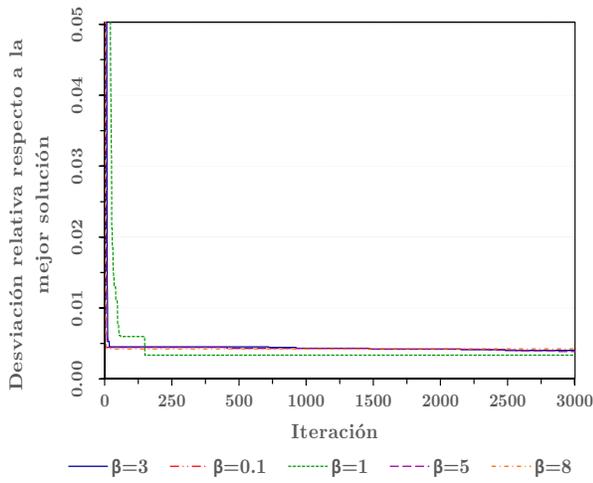


(c) AS con varios valores de m

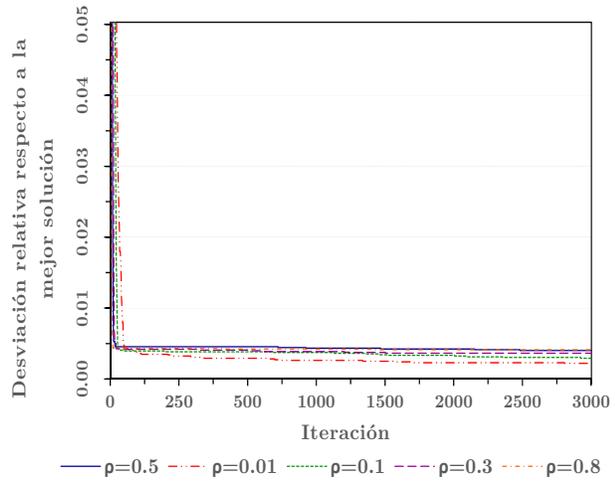


(d) Número de iteraciones promedio en encontrar la mejor solución al variar el valor de los parámetros. Las barras indican la desviación estándar

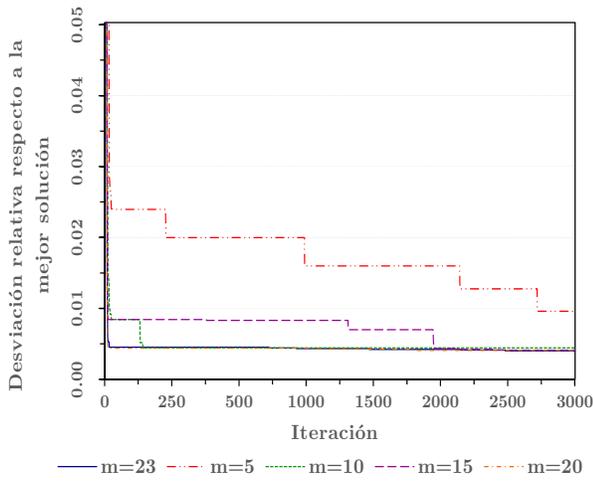
Figura B.1: Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito Deportivo



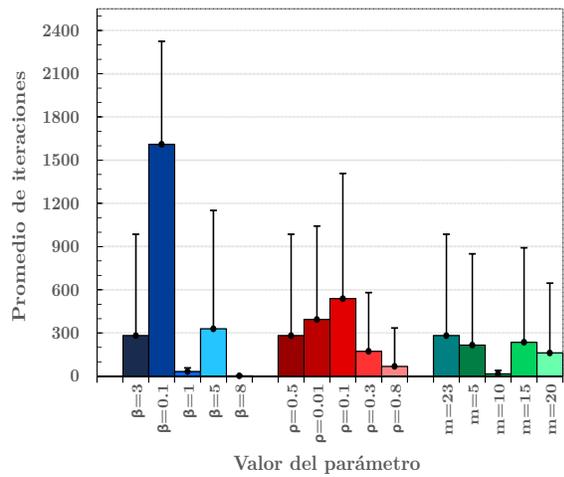
(a) AS con varios valores de β



(b) AS con varios valores de ρ

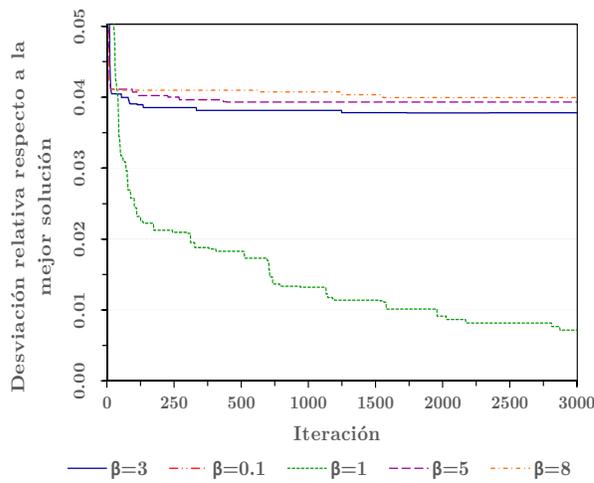


(c) AS con varios valores de m

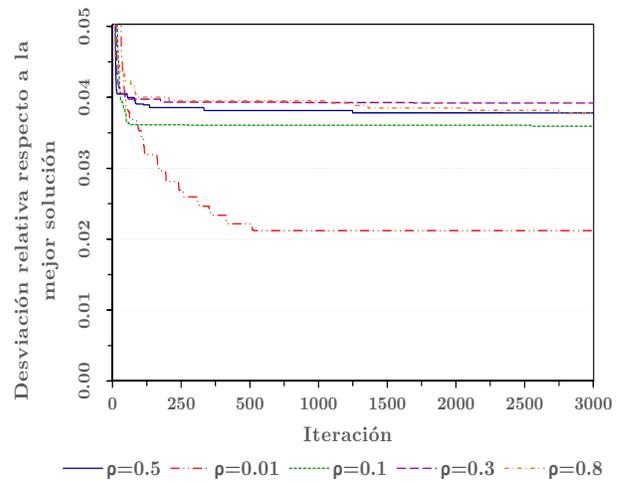


(d) Número de iteraciones promedio en encontrar la mejor solución al variar el valor de los parámetros. Las barras indican la desviación estándar

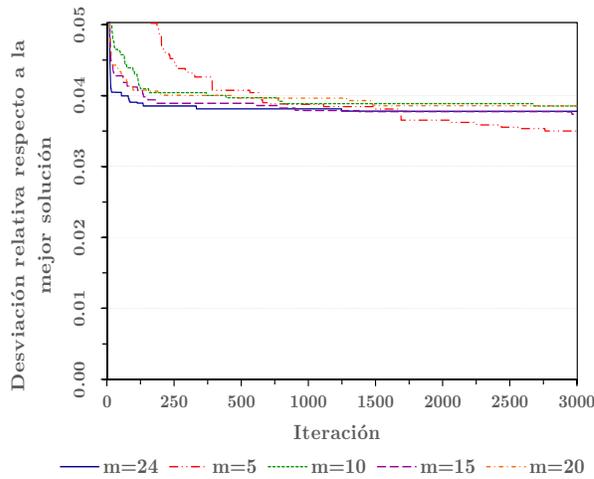
Figura B.2: Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito Exterior



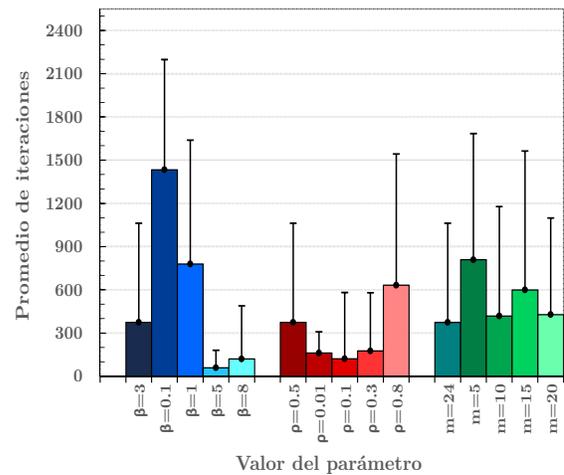
(a) AS con varios valores de β



(b) AS con varios valores de ρ

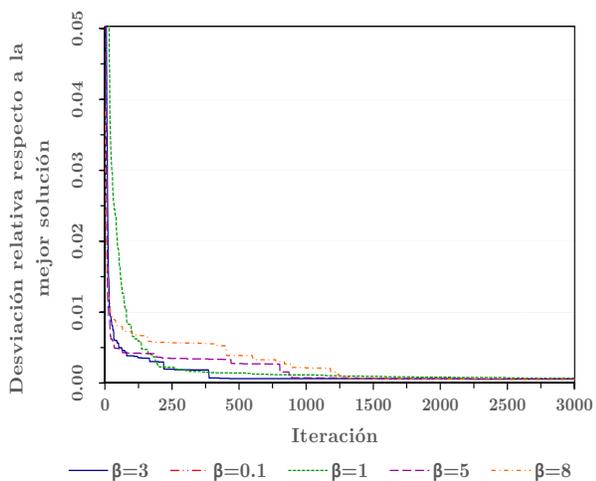


(c) AS con varios valores de m

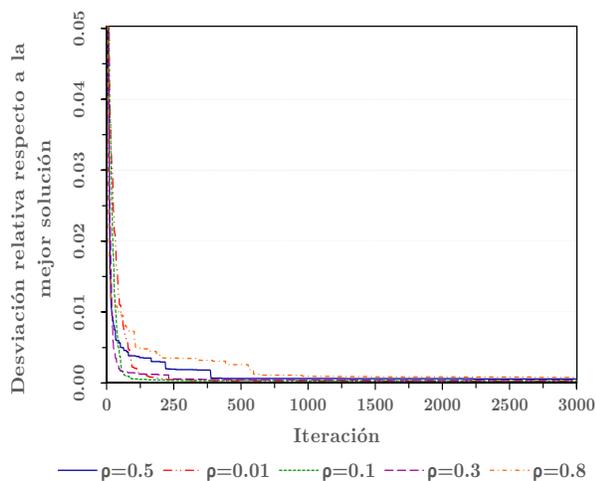


(d) Número de iteraciones promedio en encontrar la mejor solución al variar el valor de los parámetros. Las barras indican la desviación estándar

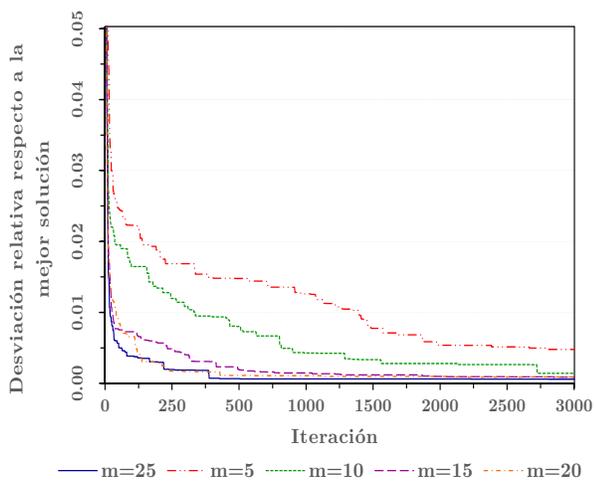
Figura B.3: Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito de la Investigación Científica



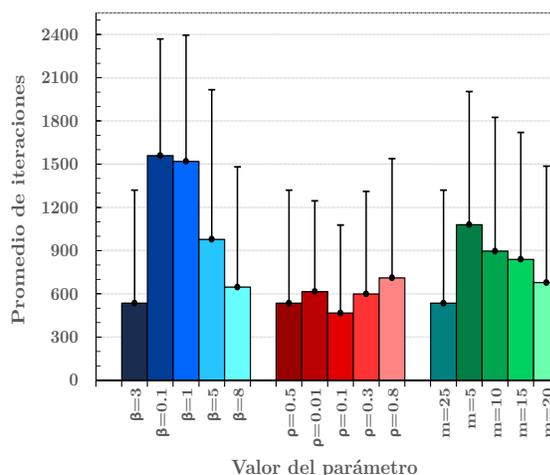
(a) AS con varios valores de β



(b) AS con varios valores de ρ



(c) AS con varios valores de m



(d) Número de iteraciones promedio en encontrar la mejor solución al variar el valor de los parámetros. Las barras indican la desviación estándar

Figura B.4: Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito Escolar

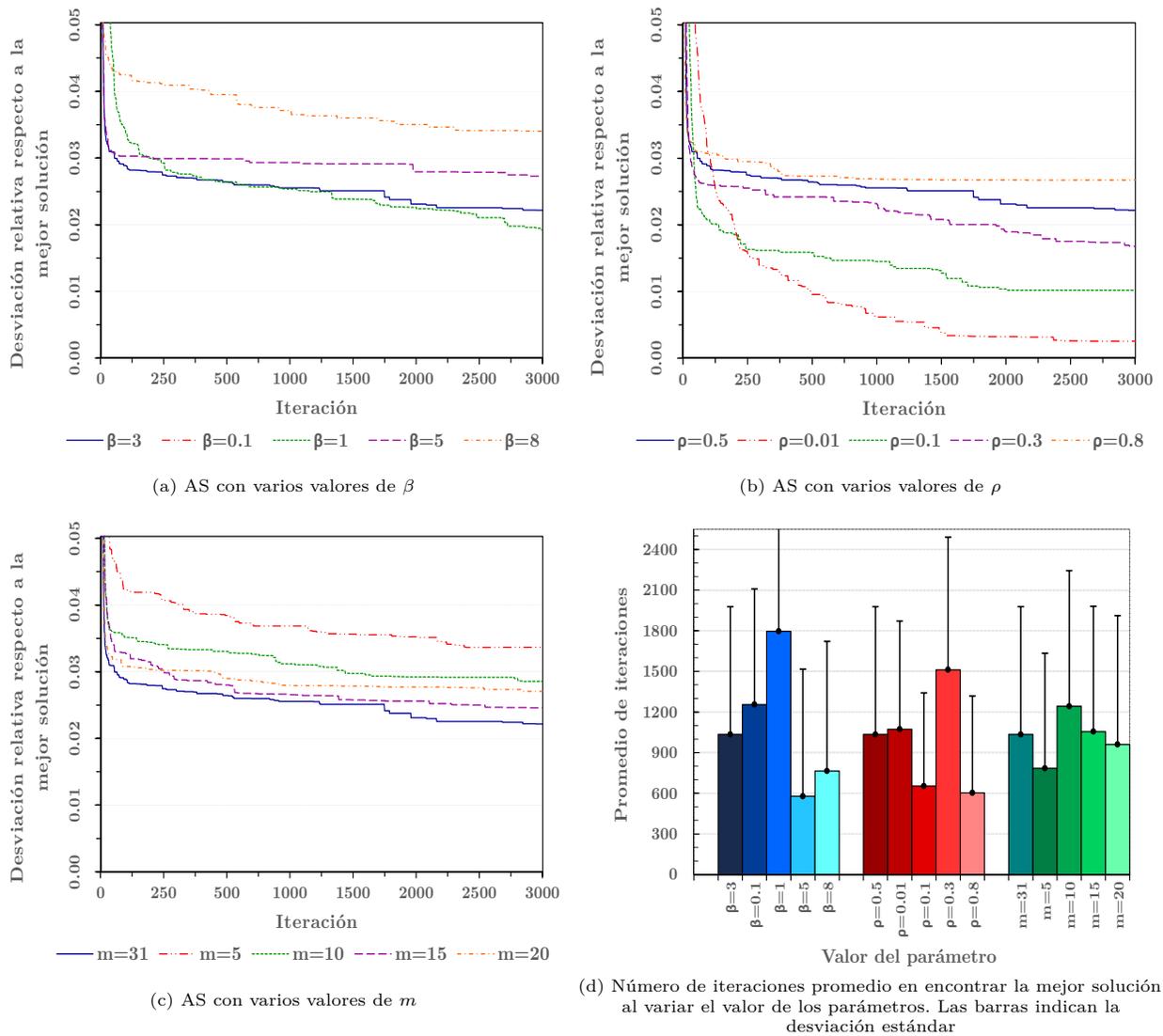


Figura B.5: Comportamiento del algoritmo AS para diversos parámetros, en el caso del Circuito CCU

B.2 Algoritmo EAS

Parámetro β : Al igual que el algoritmo AS presentó la peor evolución y el error relativo medio más grande, incluso entre todas las configuraciones propuestas, cuando β toma el valor más pequeño (0.1), el mismo comportamiento se observa en el algoritmo EAS (Tabla 7.5). Además, para este valor de β se tiene la mayor variabilidad en las soluciones en comparación con otros valores (Tablas C.6, C.7, C.8, C.9 y C.10). Sin embargo, para este algoritmo, también se observan malos resultados con β igual a 1.

Por otra parte, el valor del parámetro que produce la mejor evolución varía según el circuito: para el C. Deportivo se obtiene con β igual a 5; para los circuitos Exterior, CCU y de la Investigación Científica con β igual a 3, aunque, en los dos primeros casos, inicialmente el algoritmo converge más rápido con valores más grandes; por último, para el C. Escolar se observa que a mayor valor de β mejor la evolución del algoritmo.

Lo anterior se observa en la Tabla 7.5, pues el menor error relativo medio se obtiene con los valores de β mencionados para cada circuito.

Adicionalmente, para todos los circuitos, excepto el C. Exterior, se logra encontrar la solución A, C, D y E, según sea el caso, con más de un valor propuesto para β : para el C. Deportivo con 1 y 3; para el C. de la Investigación Científica con 1, 3, 5, y 8; para el C. Escolar con 5 y 8, y para el C. CCU con 3 y 5 (Tabla 7.5). En cuanto al C. Exterior el mínimo error relativo se alcanza sólo con β igual a 1.

Ahora bien, en general se observa que los valores de β que conducen a los mejores resultados son mayores que en el algoritmo AS. Lo anterior se puede deber a que al depositar más feromona con el algoritmo EAS, el equilibrio entre exploración y explotación se obtiene al aumentar la influencia relativa de la información heurística sobre la influencia relativa del rastro de feromona. Por otra parte, también se tiene la similitud entre ambos algoritmos si β toma un valor de 0.1, por lo que se puede pensar que con dicho valor hay un estancamiento en malas soluciones.

En cuanto al número de iteraciones necesarias para encontrar la mejor solución, la media asociada a β igual 0.1 está alrededor de las 50 iteraciones para todos los circuitos. Lo anterior sugiere que el algoritmo se estanca rápidamente en malas soluciones (Figuras B.6e, B.7e, B.8e, B.10e y B.10e).

Parámetro ρ : La evolución del algoritmo EAS al variar ρ (Figuras B.6b, B.7b, B.8b, B.9b y B.10b) es mejor durante las primeras iteraciones si ρ es mayor o igual a 0.1. No obstante, entre las 250 y 750 iteraciones, la tasa de evaporación de 0.01 genera una mejor evolución del algoritmo, comportamiento que se mantiene hasta las últimas iteraciones. De hecho, el menor error relativo medio se obtiene con ρ igual a 0.01 en los cinco circuitos (Tabla 7.5); además para los circuitos Deportivo, Exterior y de la Investigación Científica, tal error relativo es el menor entre todas las configuraciones propuestas. En general, se puede decir que es mejor si el algoritmo tiene una adaptación lenta, o bien, la feromona persiste más tiempo en los arcos.

Por otra parte, para el valor de ρ igual a 0.01 se tiene la menor variabilidad en las soluciones finales en comparación con otros valores (Tablas C.6, C.7, C.8, C.9 y C.10). Es importante mencionar, que en el caso del C. de la Investigación Científica, cuando la tasa de evaporación es de 0.01, en las 30 corridas se la solución C, por lo cual el error relativo medio es 0%.

En cuanto al peor error relativo medio, de acuerdo a la Tabla 7.5, se obtiene con ρ igual a 0.8 para los circuitos Exterior, Escolar y CCU, incluso la variabilidad en las soluciones finales es mayor en comparación con otros valores (Tablas C.7, C.9 y C.10). Mientras que para los circuitos Deportivo y de la Investigación Científica, es con 0.5 y 0.3 respectivamente, que se obtiene el peor error relativo medio.

Adicionalmente, para el C. Deportivo la solución A se encuentra con cualquier valor propuesto para ρ . De igual manera para los circuitos de la Investigación Científica y CCU, la solución C y E, respectivamente, se alcanza con todos los valores de ρ . Mientras que para el C. Exterior la solución B se alcanza con 0.01, 0.1 y 0.8; y para el C. Escolar la solución D se encuentra con 0.01, 0.1, 0.3 y 0.8.

Finalmente, en las Figuras B.6e, B.9e y B.9e, se observa que la media del número de iteraciones para encontrar la mejor solución va de 250 a 900, aproximadamente, en los circuitos Deportivo, Exterior y de la Investigación Científica. En cuanto al circuito más grande: CCU, la media de iteraciones es mayor, al oscilar entre las 900 y 1500 iteraciones (Figura B.10e).

Parámetro m : De manera general, en todos los circuitos se observa claramente que mientras más hormigas sean utilizadas, la evolución del algoritmo es mejor (Figuras B.6c, B.7c, B.8c, B.9c y B.10c). De hecho los resultados obtenidos con el menor valor de m , son los peores y por esta razón no se alcanza a ver la evolución respectiva del algoritmo. Lo anterior se refleja en la Tabla 7.5, donde el menor y mayor error relativo medio, se obtiene con mayores y menores números de hormigas utilizados, respectivamente. Asimismo, la variabilidad en las soluciones finales es menor cuando se utilizan n hormigas y, dicha variabilidad, es mayor con el menor

número de hormigas (Tablas C.6, C.7, C.8, C.9 y C.10).

Adicionalmente, en la Tabla 7.5, se puede ver que para los circuitos de la Investigación Científica y CCU, se logra encontrar la solución C y E, respectivamente, con dos o más valores propuestos para m : para el C. de la Investigación Científica con 10, 15, 20 y n hormigas; para el C. CCU con cualquier valor del parámetro. En cuanto a los circuitos Deportivo, Exterior y Escolar únicamente se alcanza la solución A, B y D, respectivamente, con n en el caso del C. deportivo y con 20 hormigas en los otros dos casos.

En cuanto al promedio de iteraciones necesarias para encontrar la mejor solución, se observa en los circuitos Exterior y de la Investigación Científica que, al utilizar más hormigas, el número de iteraciones necesarias para encontrar la mejor solución disminuye, excepto para m igual a n en el C. Exterior (Figuras B.7e y B.8e). Lo anterior debido a que al tener un mayor número de hormigas trabajando se generan más soluciones y con ello, una mayor exploración del espacio de búsqueda. Por otra parte también debe tenerse en cuenta el tiempo de cómputo es mayor al aumentar el número de hormigas. En el caso del C. CCU, se observa lo contrario, pues utilizar más hormigas provoca que el algoritmo tarde más iteraciones en llegar a la mejor solución (Figura B.10e).

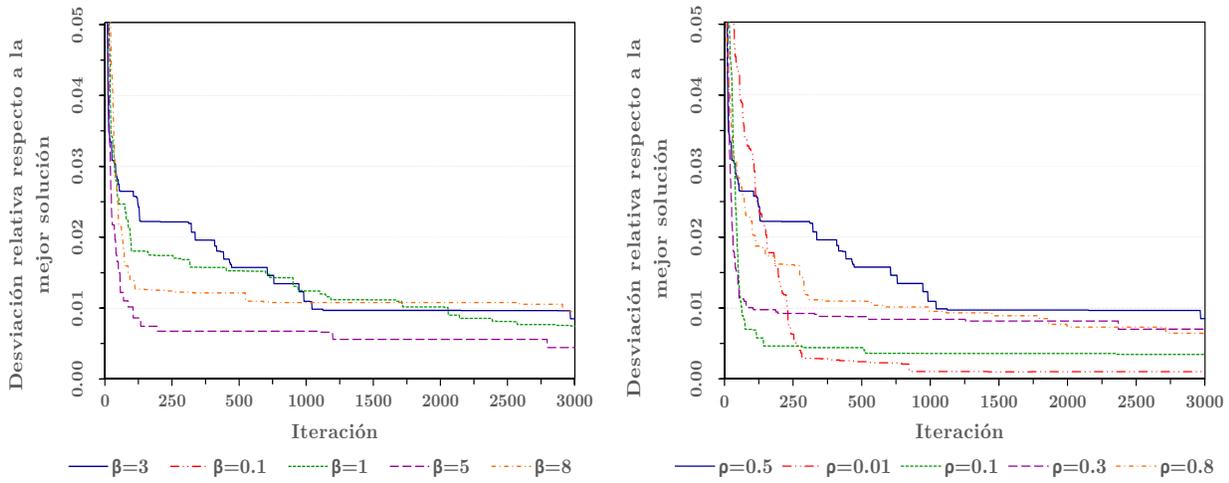
Parámetro e : La evolución del algoritmo EAS, al variar el parámetro e , difiere según el circuito que se busque optimizar. Por un lado, para los Circuitos Deportivo y Escolar, utilizar un valor de e igual al número de nodos resulta contraproducente para la ejecución del algoritmo. De hecho, mientras más grande sea el valor de e , más lenta es la convergencia del algoritmo y más pobre es la calidad de las soluciones generadas (Figuras B.6d y B.9d). Incluso, el peor error relativo medio está asociado al mayor valor propuesto para este parámetro (Tabla 7.5), lo cual sugiere que para tales circuitos una mayor explotación de la mejor solución global lleva a peores resultados. Asimismo, la variabilidad en las soluciones finales es menor cuando e es igual a 1 y es mayor cuando e toma el valor más grande (Tablas C.6 y C.9). Además, vale la pena mencionar que para el C. Escolar se obtiene el menor error relativo medio, entre todas las configuraciones propuestas, con un valor de e igual a 1.

En el caso del C. Exterior, se observa que el algoritmo EAS tiene un mejor comportamiento durante la mayoría de las iteraciones con e igual a 10, sin embargo hacia el final de la ejecución con un valor de e igual a n , genera la mejor solución e incluso el menor error relativo medio (Figura B.7d). Mientras que la ejecución más pobre del algoritmo está dada por valores de e iguales a 20 y 35, siendo este último el que proporciona el peor error relativo medio.

Por otro lado, para el C. de la Investigación Científica, el algoritmo EAS tiene un buen comportamiento cuando e es igual a n (Figura B.8d). De hecho, a menor valor del parámetro, más pobre el desempeño del algoritmo. Lo anterior se observa en la Tabla 7.5, donde el peor error relativo está asociado a e igual a 1, en ambos circuitos.

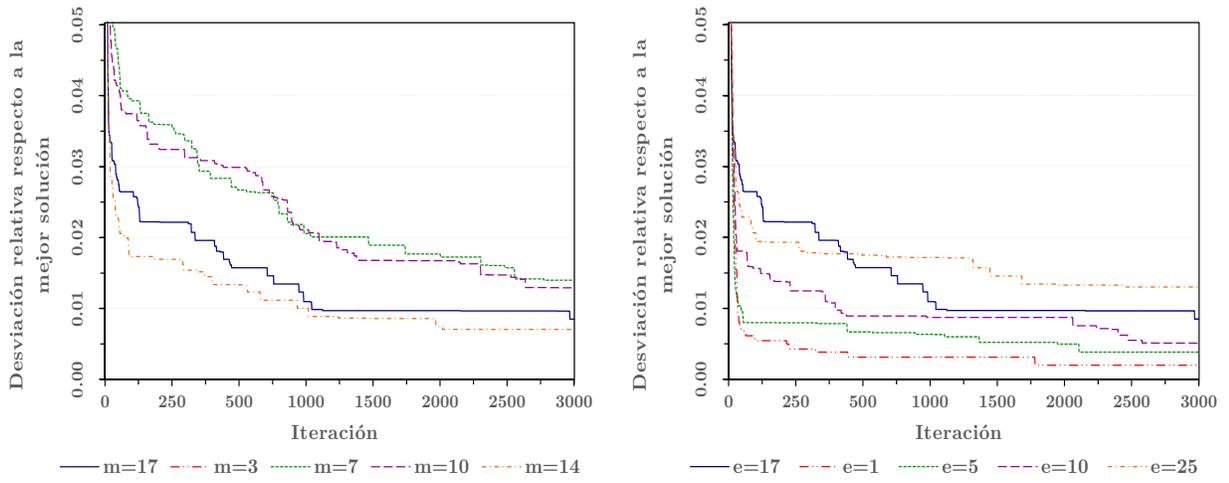
En el caso del C. CCU, por una parte se ve que el algoritmo EAS tiene un mejor comportamiento (Figura B.10d), así como menor variabilidad en soluciones finales y menor error relativo (Tabla C.10), esto, cuando e es igual a 10, seguido de e igual a 20. Por otra parte, cuando e es igual a n , se obtuvo un mayor error relativo, así como mayor variabilidad en soluciones finales, causando un mal comportamiento del algoritmo. Además, vale la pena mencionar que para el C. CCU se obtiene el menor error relativo medio, entre todas las configuraciones propuestas, con un valor de e igual a 10.

Finalmente, en la Tabla 7.5, se puede ver que para los circuitos de la Investigación Científica, Escolar y CCU, se encuentra la solución C, D y E, respectivamente, con casi todos los valores propuestos para e : para el C. Exterior con 10 y 35; para el C. de la Investigación Científica con 10, 20, n y 35; para el C. Escolar con 1, 10, 20 y 35; para el C. CCU con 10, 20, n y 35. En cuanto al C. Deportivo, la solución A se alcanza sólo con e igual a n .



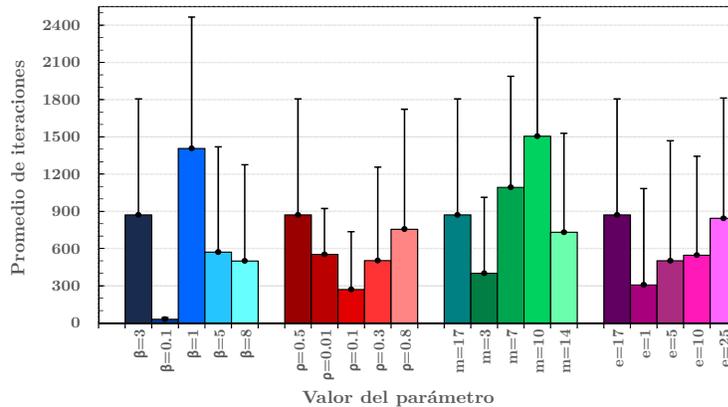
(a) EAS con varios valores de β

(b) EAS con varios valores de ρ



(c) EAS con varios valores de m

(d) EAS con varios valores de e



(e) Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo. Las barras indican la desviación estándar

Figura B.6: Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito Deportivo

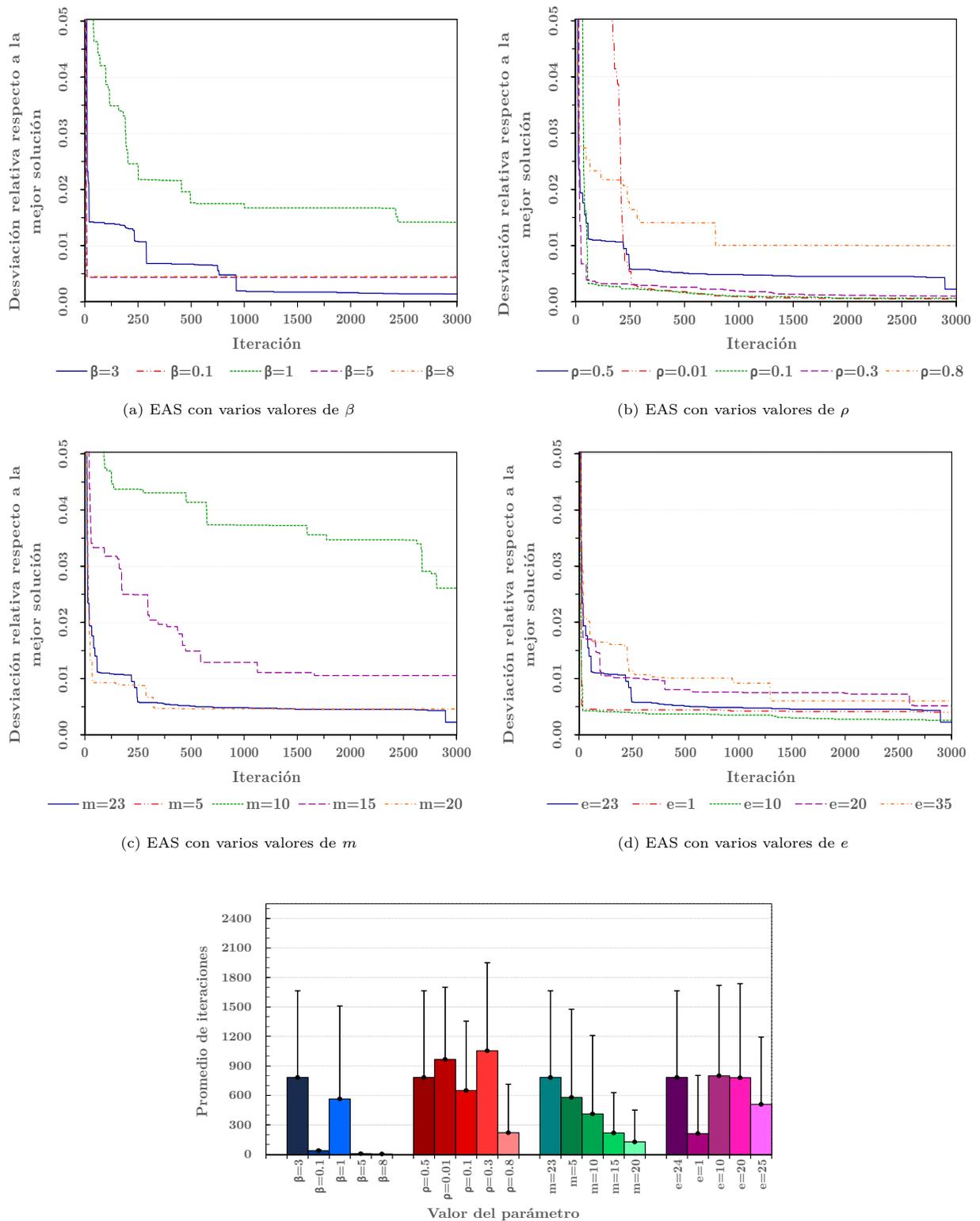
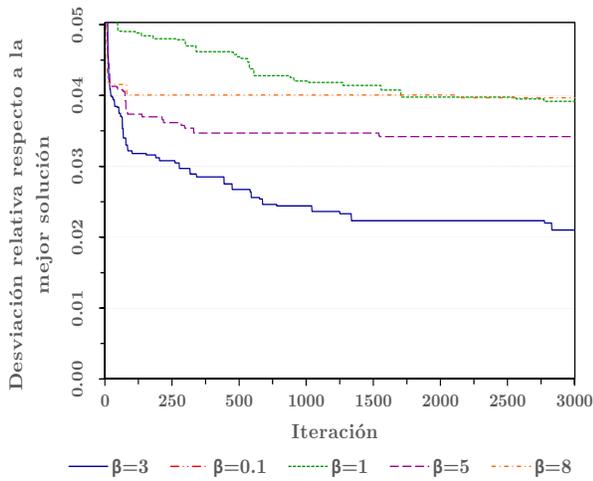
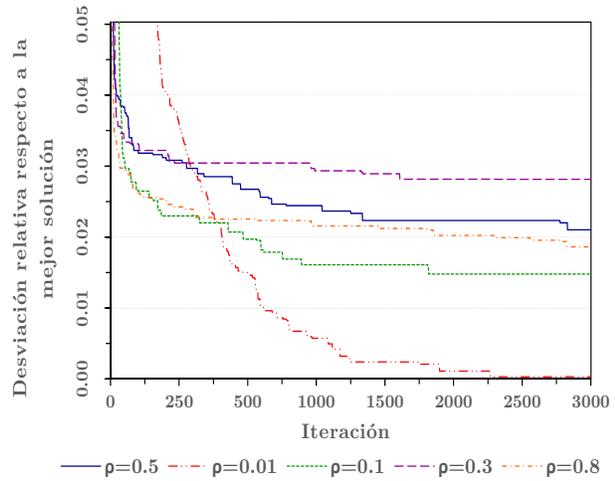


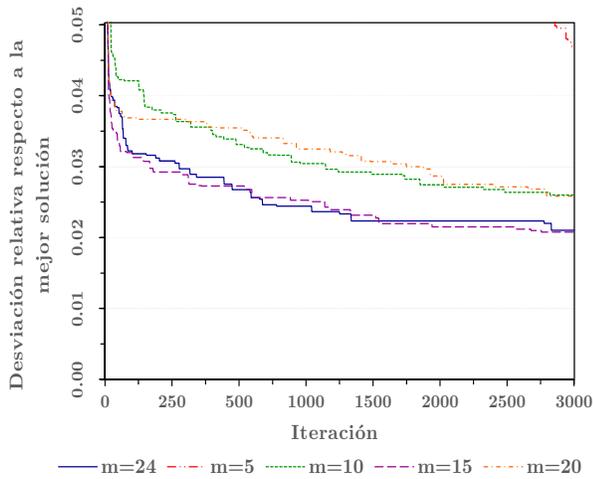
Figura B.7: Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito Exterior



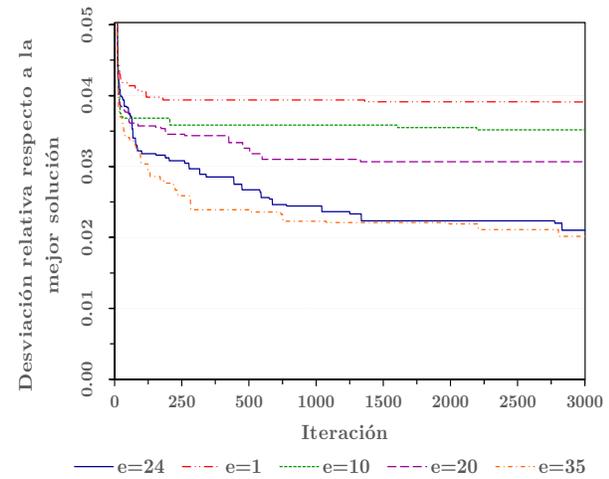
(a) EAS con varios valores de β



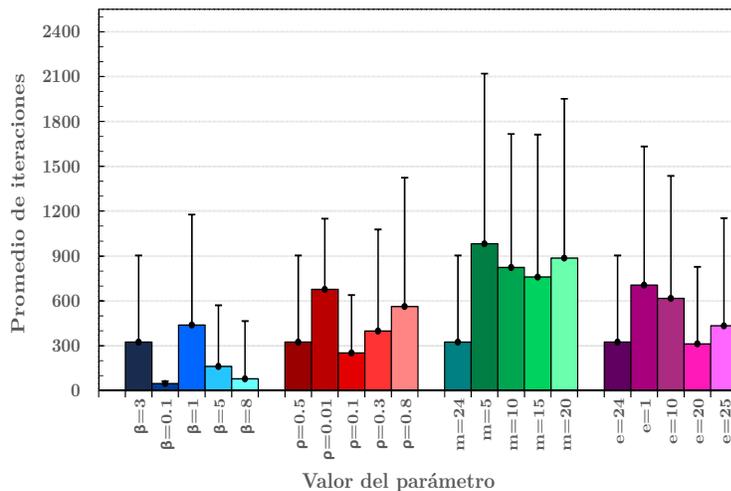
(b) EAS con varios valores de ρ



(c) EAS con varios valores de m



(d) EAS con varios valores de e



(e) Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo. Las barras indican la desviación estándar

Figura B.8: Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito de la Investigación Científica

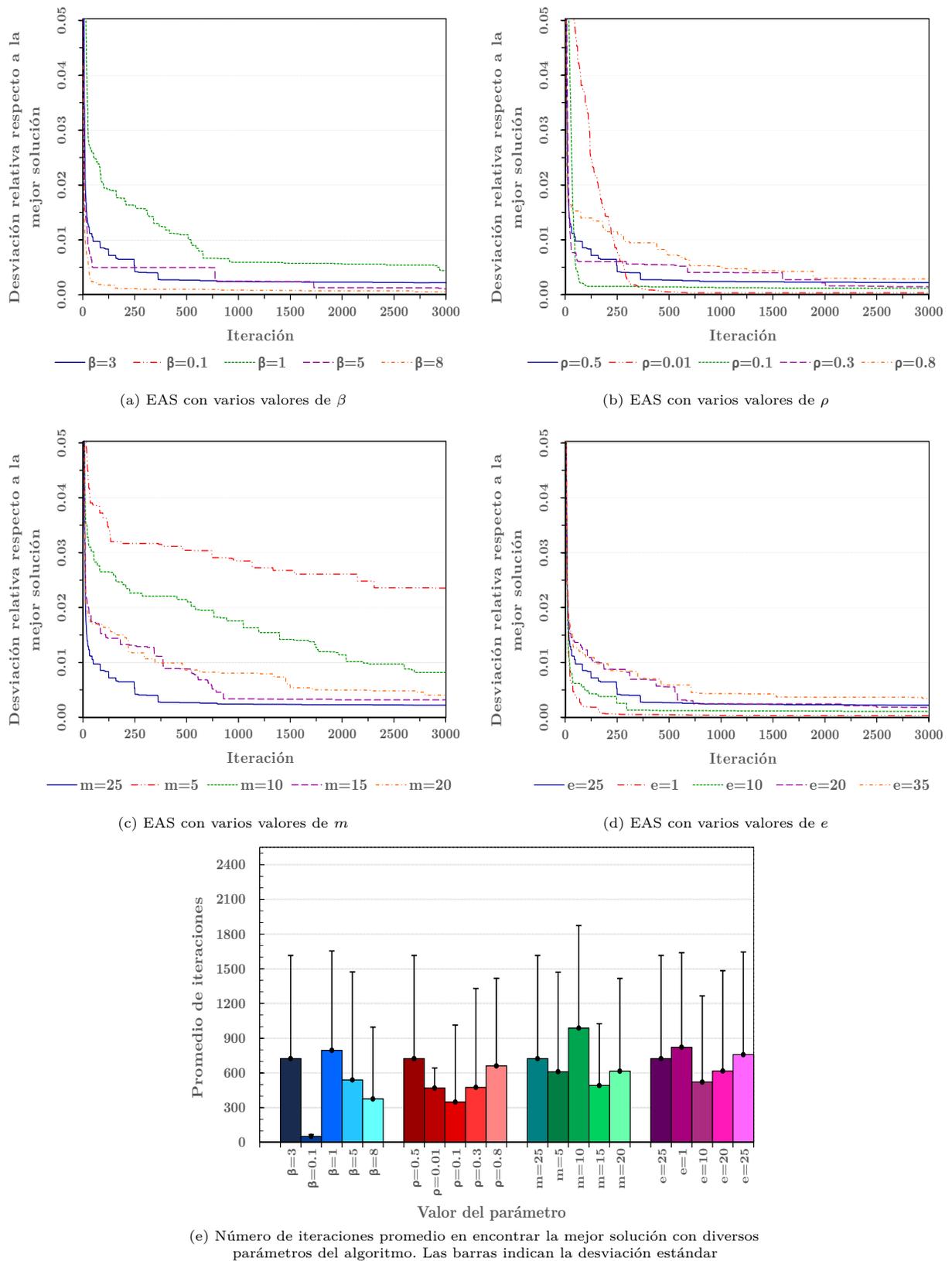
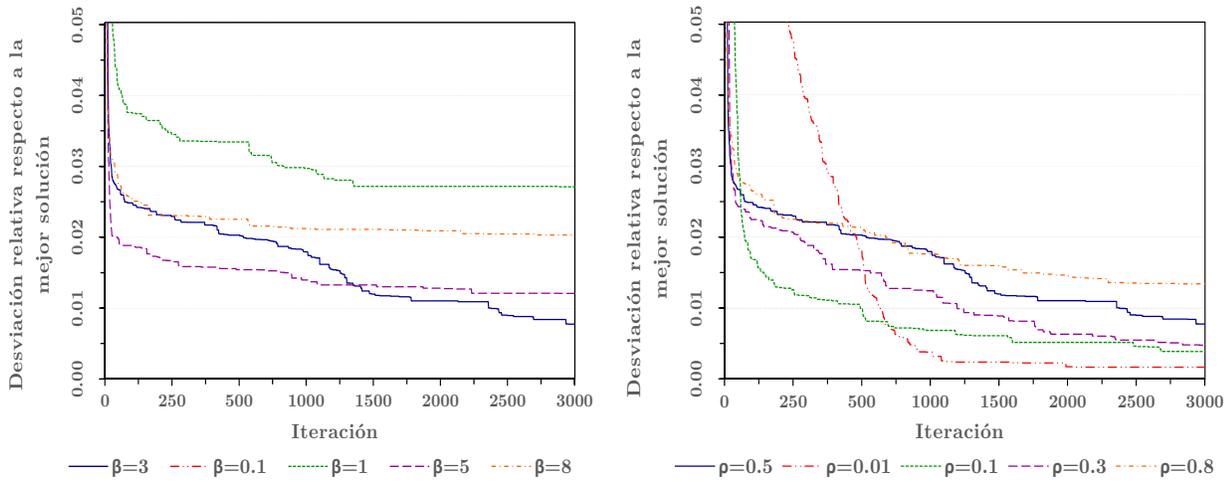
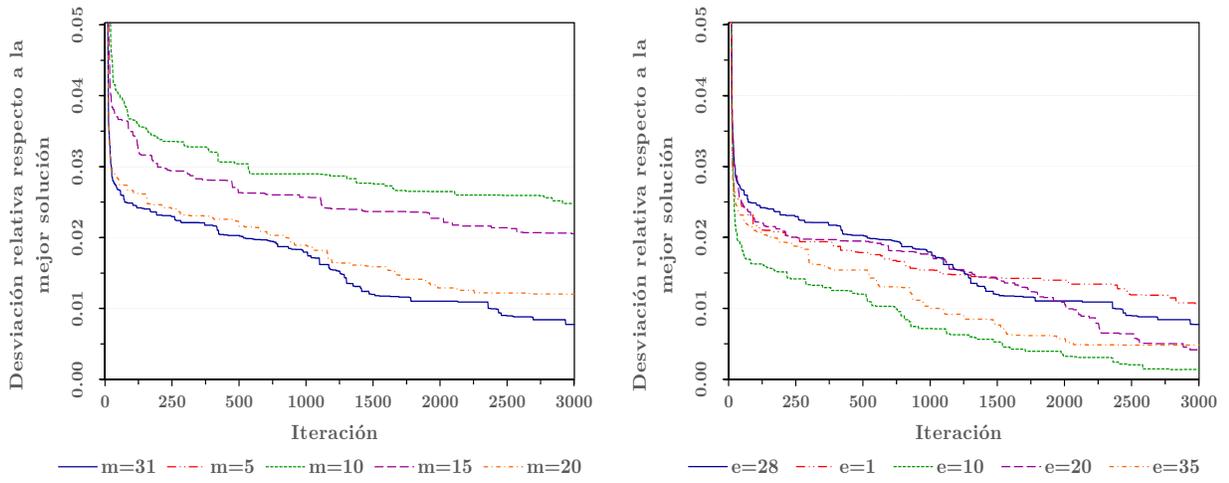


Figura B.9: Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito Escolar



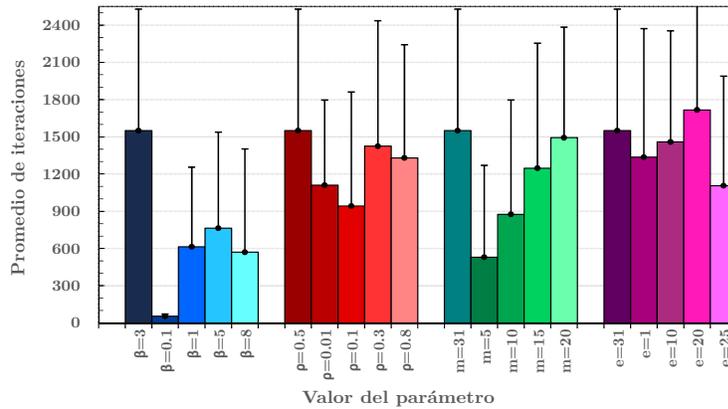
(a) EAS con varios valores de β

(b) EAS con varios valores de ρ



(c) EAS con varios valores de m

(d) EAS con varios valores de e



(e) Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo. Las barras indican la desviación estándar

Figura B.10: Comportamiento del algoritmo EAS para diversos parámetros, en el caso del Circuito CCU

B.3 Algoritmo MMAS

Parámetro β : La evolución del algoritmo MMAS es diferente para cada circuito al variar el parámetro β . Por un lado, para los circuitos Deportivo y de la Investigación Científica el algoritmo converge inicialmente más rápido para los valores de 3, 5 y 8. Sin embargo, los dos últimos valores son superados por 0.1 y 1 (Figuras B.11a y B.13a), de tal manera que el menor error relativo se obtiene con β igual a 1, para el primer circuito; y con β igual a 1 y 3, para el segundo circuito (Tabla 7.7). Aunado a lo anterior, la variabilidad de las soluciones finales es muy pequeña, ya que en todas (C. de la Investigación Científica) o casi todas (C. Deportivo) las corridas se alcanza la solución C o A, según sea el caso (Tablas C.11 y C.13). Además, en ambos casos, el peor error relativo está asociado al mayor valor de β y se observa que en el caso del C. Deportivo la variabilidad es igual a 0%, es decir, en todas las corridas se llega a la misma mala solución, mientras que en el otro circuito la variabilidad de las soluciones es mayor con respecto otros valores del parámetro (Tablas C.11 y C.13).

Por otro lado, en los circuitos Exterior, Escolar y CCU se observa que el algoritmo MMAS converge más rápido, durante las primeras iteraciones, mientras mayor sea el valor de β . Sin embargo, después de cierto número de iteraciones la evolución del algoritmo es muy similar para β mayor o igual a 1. De esta manera, el menor error relativo medio y las menores variabilidades se obtienen con β igual a 1, 5 y 3, para el circuito Exterior, Escolar y CCU respectivamente (Tabla 7.7). Específicamente, en el caso de β igual a 0.1, el comportamiento del algoritmo queda alejado del resultante con los demás valores. Además, con dicho valor se obtiene mayor variabilidad y el error relativo medio más grande (Tablas 7.7, C.12, C.14 y C.15). Lo anterior sugiere que con tal valor hay un estancamiento en malas soluciones.

Adicionalmente, para todos los circuitos, excepto el C. Deportivo y el C. Exterior, se logra encontrar la solución C, D o E, según sea el caso, con cualquier valor propuesto para β . En cuanto a los dos circuitos mencionados la solución A o B, dependiendo del circuito, se alcanza con β igual a 0.1, 1, 3 y 5 (Tabla 7.7).

Analizando el promedio de iteraciones, se observa que para el C. Deportivo se necesitan más iteraciones cuando β toma el valor de 1, a pesar de que con tal valor se alcanzan las mejores soluciones (Figura B.11f). Mientras que para los circuitos Exterior y de la Investigación Científica, con tal valor, que también genera los mejores resultados, el promedio es inferior al de otros valores. Para el C. CCU, también el promedio de iteraciones asociado al valor de β , que genera los mejores resultados (3), es menor al de otros valores del parámetro.

Parámetro ρ : Para todos los circuitos la ejecución más pobre del algoritmo está determinada por las tasas de evaporación 0.5 y 0.8 (Figuras B.11b, B.12b, B.13b, B.14b y B.15b). De hecho, en la Tabla 7.7, se puede observar que el error relativo medio más grande, incluso entre todas las configuraciones propuestas para los parámetros, se obtiene con un valor de ρ de 0.8.

Por otra parte, para los circuitos Deportivo y Exterior, muy al inicio del algoritmo la convergencia es más rápida cuando ρ es igual a 0.1, aunque después de cierto número de iteraciones se alcanzan resultados muy parecidos y diferencias casi imperceptibles, si la tasa de evaporación es igual a 0.02 0.1 ó 0.3.

Para los circuitos de la Investigación Científica, Escolar y CCU se ve que para valores medios de ρ (0.1, 0.3 y 0.5) la convergencia es más rápida durante las primeras 150 iteraciones aproximadamente. Sin embargo, después de ese número de iteraciones, los resultados generados por tasas iguales a 0.3 y 0.5 son superados por una tasa de evaporación equivalente a 0.02, la cual logra empatar el buen desempeño que se tiene desde un inicio con ρ igual a 0.1.

Aunado a lo anterior, para todos los circuitos se tiene que a mayor tasa de evaporación, mayor el error relativo medio: es así que el menor error relativo medio se obtiene con un tasa de evaporación de 0.02 y un mayor error relativo medio con una tasa de 0.8 (Tabla 7.7). De manera análoga, las tasas de evaporación

más grandes tienen asociada una mayor variabilidad (Tablas C.11, C.12, C.13, C.14 y C.15), así como mayor número de iteraciones promedio (Figuras B.11f, B.12f, B.13f, B.14f y B.15f). Lo anterior sugiere un mejor comportamiento del algoritmo con una adaptación lenta.

Finalmente, para los circuitos de la Investigación Científica y CCU se logra encontrar la solución C y E, respectivamente, con cualquier valor propuesto para ρ . En cuanto al C. Deportivo la solución A se alcanza con ρ igual a 0.02, 0.1 y 0.3, mientras que en para los circuitos Exterior y Escolar, la solución B y D, respectivamente, sólo no se alcanza con ρ igual a 0.8 (Tabla 7.7).

Parámetro m : En general, para cualquier circuito: a mayor número de hormigas, mejor el comportamiento del algoritmo y más rápida la convergencia hacia la mejor solución (Figuras B.11c, B.12c, B.13c, B.14c y B.15c). Vale la pena notar que en los experimentos el mayor número de hormigas utilizado es igual al número de nodos, por lo que no se puede decir hasta que número de hormigas el resultado anterior se mantiene. De hecho, el menor y mayor error relativo medio se obtienen con el mayor y menor número de hormigas, respectivamente; excepto para el C. Exterior, donde lo anterior sucede con 10 hormigas (Tabla 7.7). Además a mayor número de hormigas menor es la variabilidad de soluciones (Tablas C.11, C.12, C.13, C.14 y C.15). Adicionalmente, para todos los circuitos, excepto el C. Deportivo, se logra encontrar la solución B, D, C o E, según sea el caso, con cualquier valor propuesto para m . En cuanto al C. Deportivo únicamente con el menor número de hormigas no se logra alcanzar la solución A (Tabla 7.7).

Vale la pena mencionar que para los circuitos Escolar y Exterior las diferencias en el comportamiento del algoritmo para diferentes valores de m son muy pequeñas, además, la convergencia es más rápida en comparación con los otros circuitos. En cuanto a los tres circuitos restantes, son más apreciables las diferencias al variar el número de hormigas utilizadas.

Por último, de acuerdo a lo que se esperaría, el promedio de iteraciones necesarias para encontrar la solución es menor con n hormigas (Figuras B.11f, B.12f, B.13f, B.14f y B.15f).

Con mecanismo de reinicialización: Para los circuitos Exterior, Escolar y CCU no se observan diferencias significativas en la evolución del algoritmo al utilizar o no el mecanismo de reinicialización (Figuras B.12d, B.14d y B.15d). Por otra parte, para el C. Deportivo se observa una ligera mejora al utilizar el mecanismo de reinicialización (Figura B.11d), mientras que para el C. de la Investigación Científica la diferencia es muy clara, siendo mejor utilizarlo (Figura B.13d). Adicionalmente, para todos los circuitos, se logra encontrar la solución A, B, C, D o E, según sea el caso, al utilizar o no el mecanismo de reinicialización (Tabla 7.7).

Por otra parte, al comparar el promedio de iteraciones necesarias para alcanzar la mejor solución, se observa en la Figuras B.11f, B.12f, B.13f, B.14f y B.15f que al no utilizar este mecanismo, el tiempo promedio de convergencia es significativamente menor.

Utilizando la mejor solución global: Para todos los circuitos, excepto el C. CCU, la evolución del algoritmo hacia el final de las iteraciones es prácticamente la misma al utilizar la solución local o la global para actualizar los niveles de feromona (Figura B.11e, B.12e, B.13e, B.14e y B.15e). Apenas se nota una ligera convergencia más rápida con la solución local (global) para los circuitos Deportivo, Exterior y Escolar (de la Investigación Científica). Lo anterior también se observa en la Tabla 7.7, pues el menor error relativo medio para los circuitos Deportivo y Escolar se obtiene con la solución local (así como la menor variabilidad en las soluciones: Tablas C.11 y C.14), mientras que para el C. Exterior pasa lo contrario, más aún, con la solución global en todas las corridas se alcanza la solución B. En cuanto al C. de la Investigación Científica, el error relativo medio al utilizar la solución local o global es 0 %, es decir, en todas las corridas, en ambos casos, se alcanza la solución C.

En cuanto al C. CCU, las diferencias son significativas: al utilizar la solución local, el error relativo medio

es 0 % (en todas las corridas se alcanza la solución E); si se utiliza la solución global, el error relativo medio aumenta a 0.827 % (Tabla 7.7). El mismo comportamiento se observa en la variabilidad de las soluciones, pues es mayor si se utiliza la solución global (Tabla C.15).

Adicionalmente en la Tabla 7.7 se puede ver que para todos los circuitos, se logra encontrar la solución A, B, C, D o E, según corresponda, en al menos una corrida, al utilizar la solución local o global.

Finalmente, en relación al promedio de iteraciones necesarias para encontrar la mejor solución, se observa en las Figuras B.11f, B.12f y B.13f, que la diferencia es de 100 iteraciones aproximadamente entre utilizar la solución local o la global para los circuitos Deportivo, Exterior y de la Investigación Científica. Mientras que para los circuitos Escolar y CCU, se puede ver una diferencia notable y menor al utilizar la solución local.

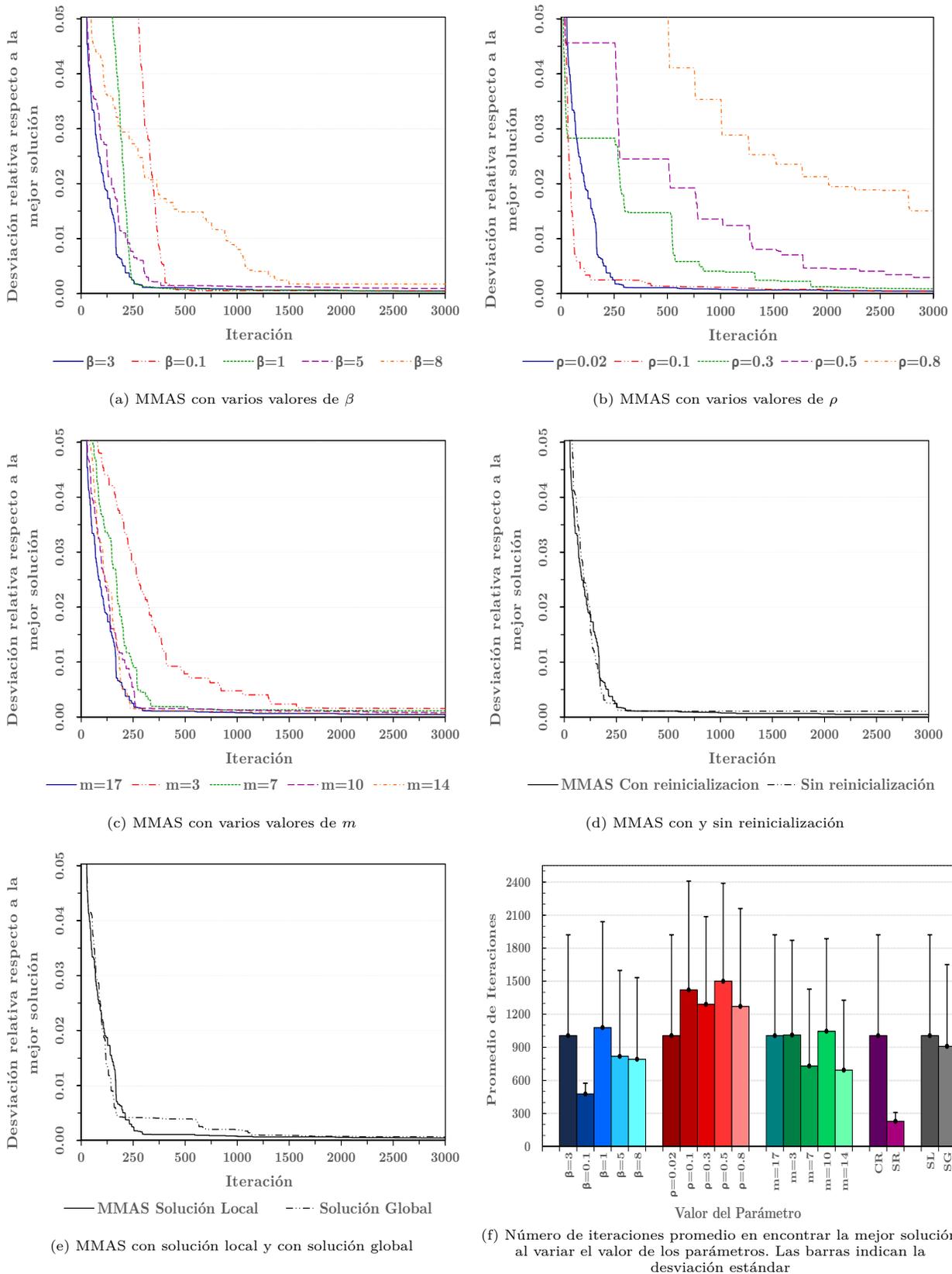
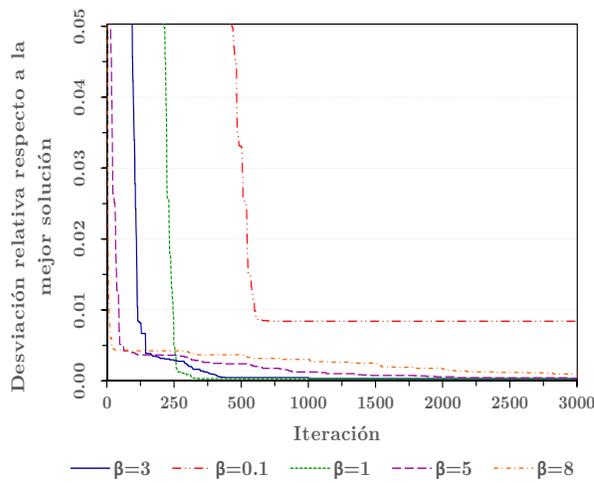
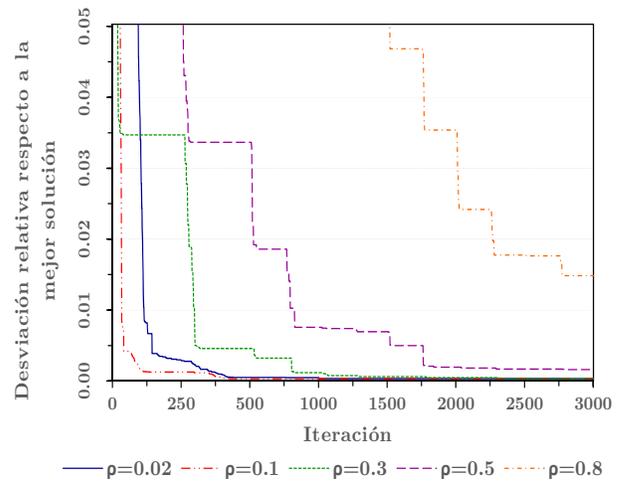


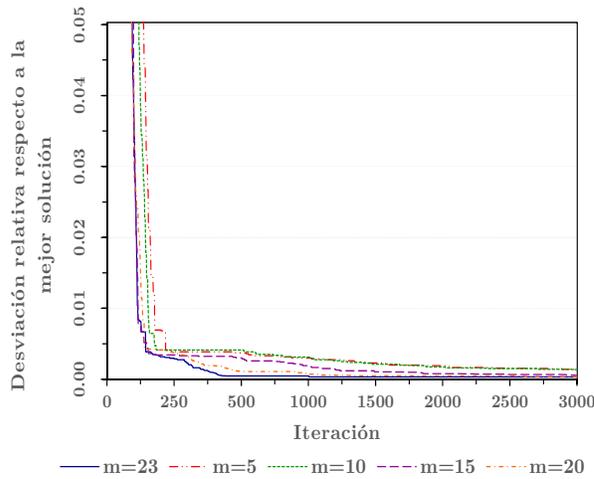
Figura B.11: Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito Deportivo



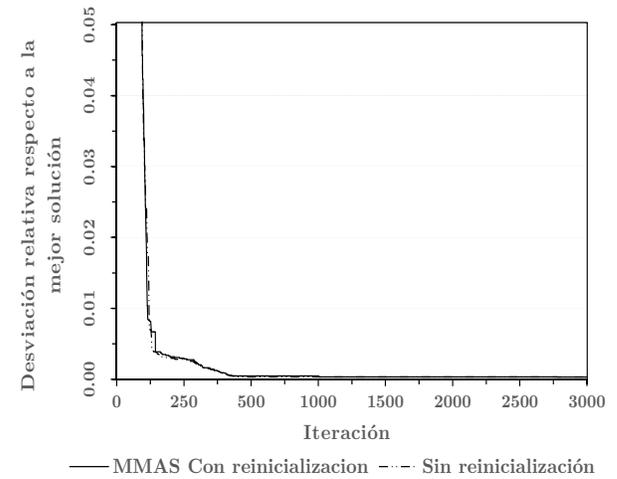
(a) MMAS con varios valores de β



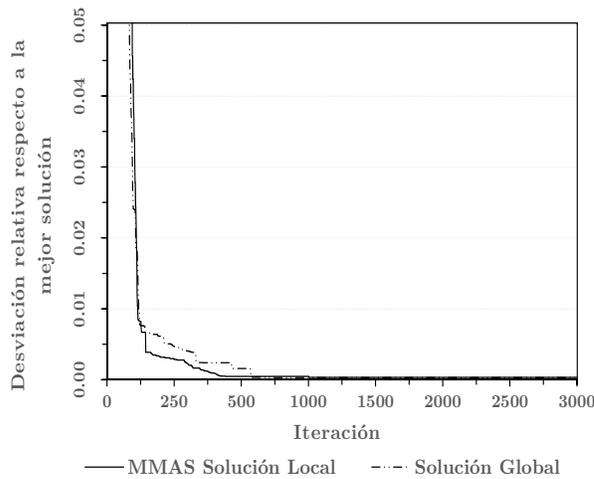
(b) MMAS con varios valores de ρ



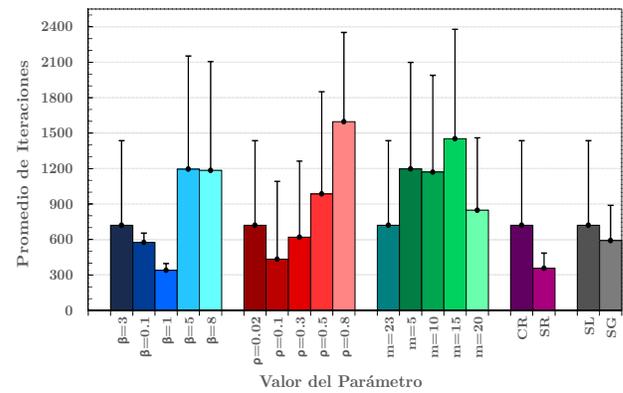
(c) MMAS con varios valores de m



(d) MMAS con y sin reinicialización



(e) MMAS con solución local y con solución global



(f) Número de iteraciones promedio en encontrar la mejor solución al variar el valor de los parámetros. Las barras indican la desviación estándar

Figura B.12: Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito Exterior

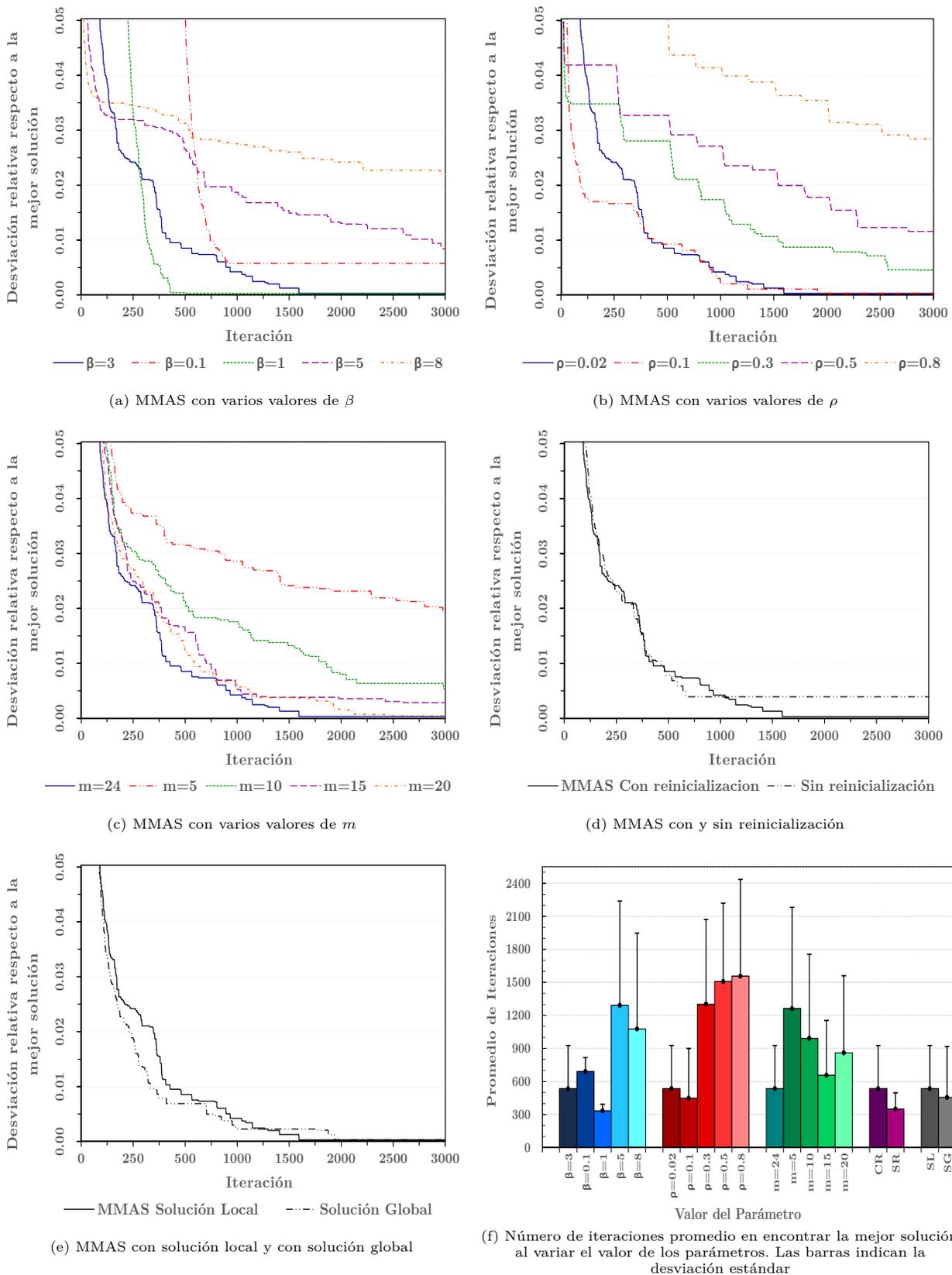
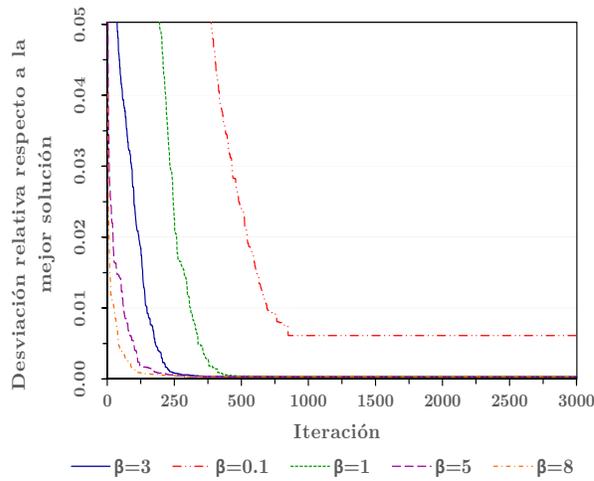
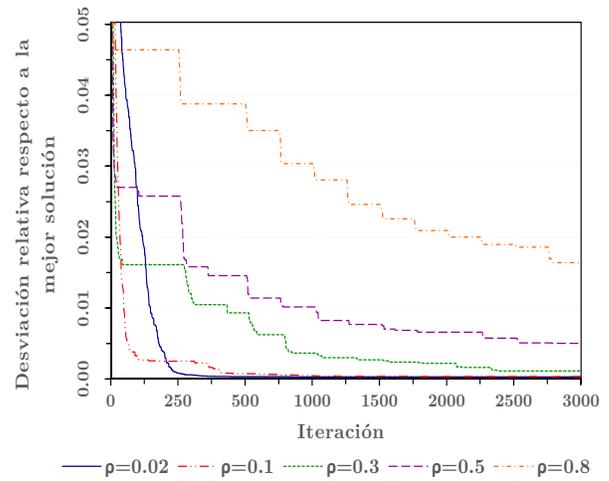


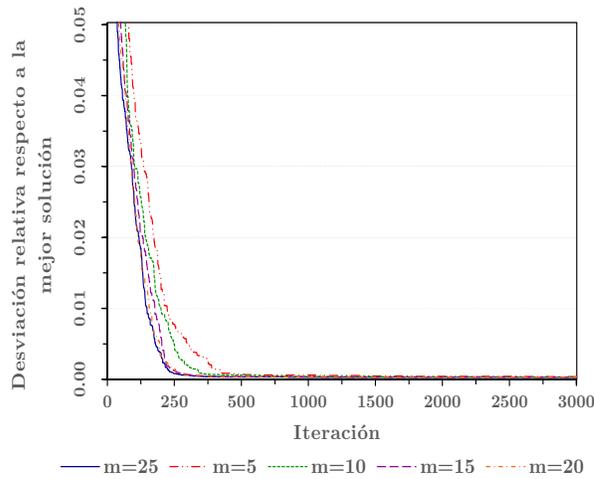
Figura B.13: Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito de la Investigación Científica



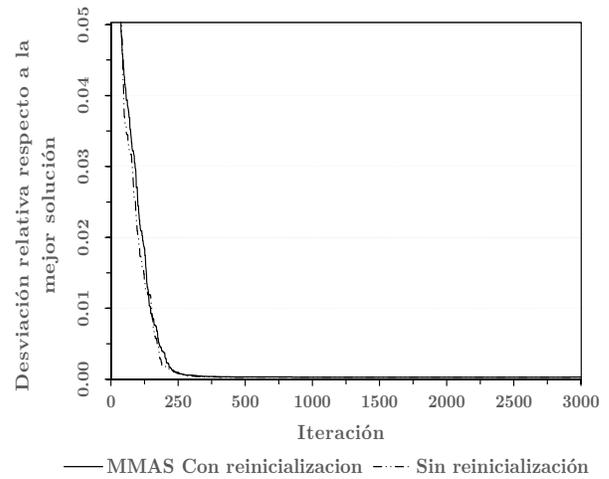
(a) MMAS con varios valores de β



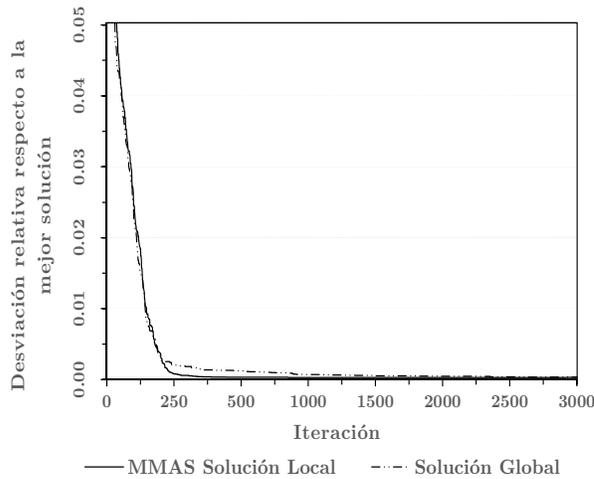
(b) MMAS con varios valores de ρ



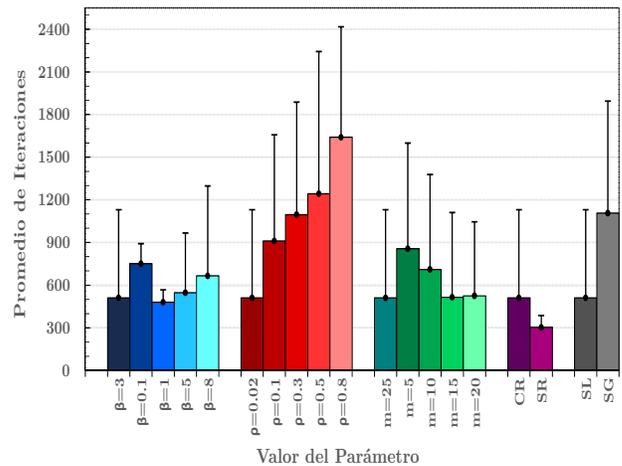
(c) MMAS con varios valores de m



(d) MMAS con y sin reinicialización



(e) MMAS con solución local y con solución global



(f) Número de iteraciones promedio en encontrar la mejor solución al variar el valor de los parámetros. Las barras indican la desviación estándar

Figura B.14: Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito Escolar

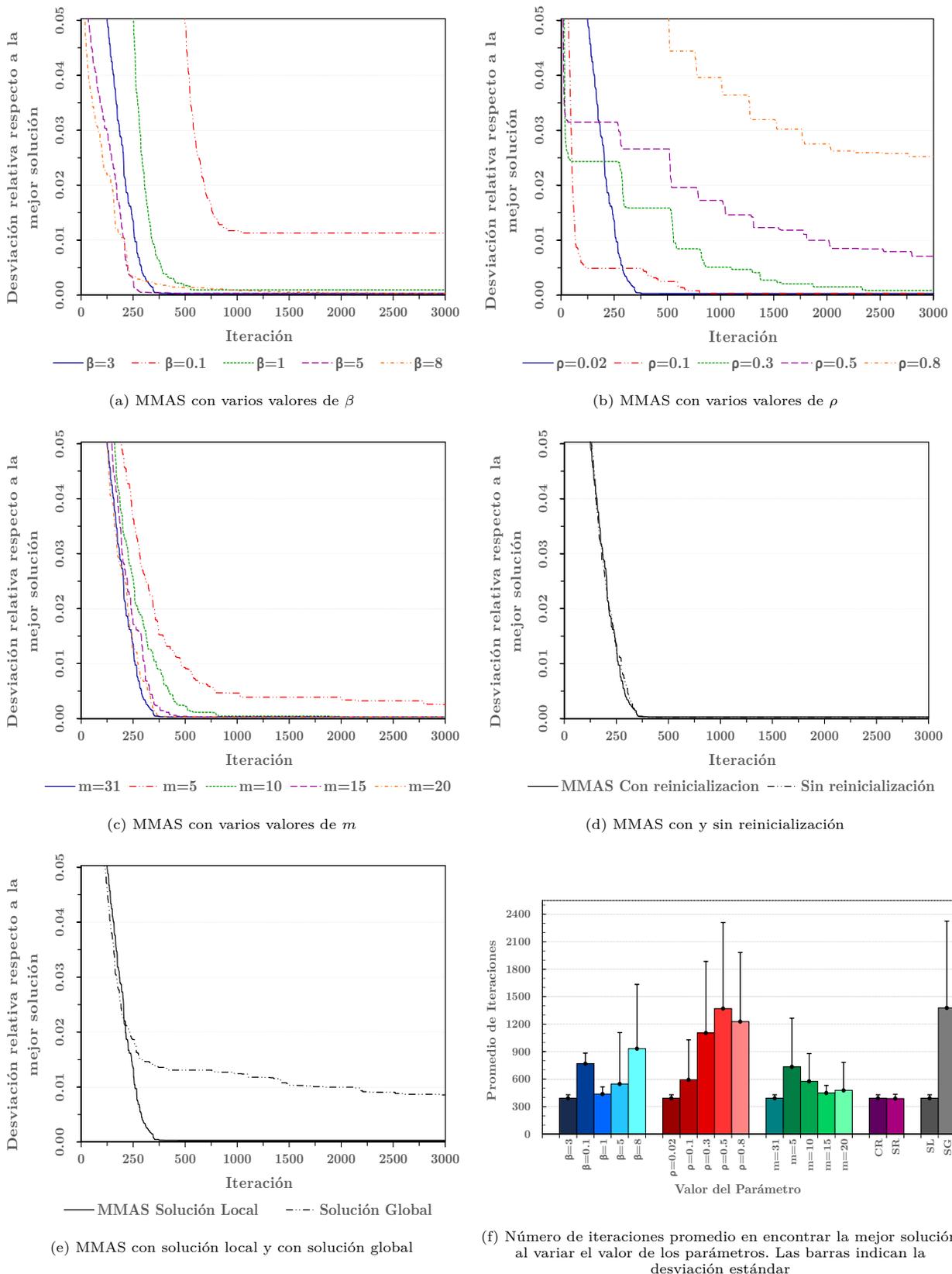


Figura B.15: Comportamiento del algoritmo MMAS para diversos parámetros, en el caso del Circuito CCU

B.4 Algoritmo ACS

Parámetro β : Se observa que la evolución del algoritmo ACS difiere al variar el parámetro β , según el circuito que se busque optimizar. Por un lado, para los Circuitos Deportivo y de la Investigación Científica los valores de β : 3, 5 y 8 proporcionan los peores resultados, en contraposición con los valores 0.1 y 1 que producen la mejor evolución del algoritmo (Figuras B.16a y B.18a). De hecho, el mayor error relativo medio, entre los valores del parámetro, así como entre las todas las configuraciones propuestas, se obtiene con β igual a 8. En cuanto al menor error relativo medio (incluso entre todas las configuraciones de parámetros), así como la menor variabilidad en soluciones, para el C. Deportivo se obtiene con β igual a 0.1 y para el C. de la Investigación Científica se obtiene con 0.1 y 1 (Tabla 7.9).

Por otro lado, en los Circuitos Exterior, Escolar y CCU se observa, en primer lugar, que cuando β es igual a 0.1 el comportamiento del algoritmo queda alejado del resultante con los demás valores (Figuras B.17a, B.19a y B.20a). De hecho, en la Tabla 7.9 se puede ver que el mayor error relativo medio se obtiene con tal valor del parámetro (incluso, es el peor entre todas las configuraciones propuestas, en el caso del C. Escolar y el C. Exterior) y en las Tablas C.17, C.19 y C.20 se puede ver que la mayor variabilidad en las soluciones se obtiene con dicho valor. En segundo lugar, el algoritmo ACS converge más rápido durante las primeras iteraciones, con los valores más grandes de β , sin embargo, después de cierto número de iteraciones y hasta el final de la ejecución del algoritmo la mejor evolución varía: para el C. Exterior se obtiene con β igual a 1; para el C. Escolar con β igual a 5, en cuanto al C. CCU la mejor evolución se observa con β igual a 3. De hecho, con tales valores se tiene el menor error relativo medio y la menor variabilidad en soluciones finales para los circuitos Exterior y Escolar. Además, vale la pena mencionar que, para el C. Exterior, ese error relativo es el menor entre todas las configuraciones de parámetros.

Adicionalmente, en la Tabla 7.9 se puede ver que para los circuitos Escolar y CCU se logra encontrar la solución D y E, respectivamente, con cualquier valor propuesto para β . En cuanto al C. Deportivo la solución A se alcanza con β igual a 0.1, 1 y 3, mientras que en los circuitos Exterior y de la Investigación Científica, la solución B y C, respectivamente, se alcanza con β igual a 0.1, 1, 3 y 5.

Por último, para los circuitos Deportivo y Exterior, el promedio de iteraciones es menor para β igual a 0.1 y 1, respectivamente (Figuras B.21 y B.22); valores, que también generan la mejor calidad en las soluciones. En contraposición, para los circuitos Escolar y CCU, el promedio de iteraciones es mayor para β igual a 0.1, valor que genera la peor calidad de soluciones en ambos circuitos (Figuras B.24 y B.25).

Parámetro ρ : La evolución del algoritmo ACS difiere al variar el parámetro ρ , según el circuito que se busque optimizar. Por un lado, para los circuitos Exterior, Escolar y CCU, la calidad de las soluciones con tasas de evaporación de 0.3, 0.5 y 0.8 es inferior a la de las tasas 0.01 y 0.1 (Figuras B.17b, B.19b y B.20b); también la convergencia es más lenta y la variabilidad de las soluciones es mayor para el primer grupo de valores (Tablas C.17, C.19 y C.20). Además, para el C. Exterior el menor error relativo medio se alcanza con una tasa de 0.01 y para los otros dos circuitos con una tasa de 0.1 (Tabla 7.9). En relación al peor error relativo medio, éste se obtiene con una tasa de 0.8, incluso en el caso del C. CCU tal error es el peor entre todas las configuraciones propuestas. Lo anterior sugiere que para estos circuitos es bueno agregar más feromona a la mejor solución global. Adicionalmente, vale la pena mencionar que en el caso de los circuitos Exterior y Escolar las diferencias de la evolución del algoritmo para diferentes tasas de evaporación son pequeñas.

En cuanto al C. Deportivo, las dos tasas de evaporación más pequeñas propician que el algoritmo tarde más tiempo en converger (Figura B.16b), así mismo la variabilidad en las soluciones finales es mayor y la calidad no es tan buena, en comparación con los tres valores más grandes. De hecho el peor error relativo medio se obtiene con la menor tasa de evaporación, mientras que el menor error relativo medio se obtiene con la tasa 0.3 (Tabla 7.9). Dado lo anterior, se puede pensar que para este circuito es mejor no explotar en exceso la mejor solución global.

Para el C. de la Investigación Científica la convergencia es ligeramente más rápida con ρ igual a 0.01 (Figura B.18b). Incluso con dicho valor se obtiene el menor error relativo medio (Tabla 7.9), sin embargo, también con una tasa de 0.5 se obtienen buenas soluciones. Por otra parte, el peor error relativo medio se obtiene con una tasa de 0.1 (Tabla 7.9).

En cuanto al promedio de iteraciones, para los circuitos de la Investigación Científica y Escolar se puede observar en las Figura B.23 y B.22, que para una tasa de 0.1, el promedio es menor en comparación con otros valores. Sin embargo, para el primer circuito significa estancamiento en malas soluciones y para el segundo circuito se puede decir que rápidamente se encuentran buenas soluciones.

Finalmente, en todos los circuitos se encuentra la solución A, B, C, D, o E, según sea el caso, con cualquier valor propuesto para ρ (Tabla 7.9).

Parámetro m : En general se observa que utilizar más hormigas propicia que la convergencia sea más rápida y que la calidad de las soluciones sea mejor durante la ejecución del algoritmo (Figuras B.16c), B.17c, B.18c, B.19c y B.20c). Este comportamiento es más claro en el C. de la Investigación Científica y en el C. CCU, pues en los demás circuitos las diferencias no son muy grandes. Adicionalmente, en los circuitos Exterior y Escolar, la convergencia es más rápida en comparación con los otros circuitos.

Por otra parte, la mayor variabilidad de las soluciones finales está asociado al menor número de hormigas, excepto para el C. de la Investigación Científica (Tablas C.16, C.17, C.18, C.19 y C.20).

Finalmente, en todos los circuitos, excepto el C. Deportivo, se logra encontrar la solución B, C, D, o E, según sea el caso, con cualquier valor propuesto para m .

Parámetro ψ : Para los circuitos Exterior, CCU y de la Investigación Científica, se observa que con un factor de decaimiento de 0.01 la convergencia del algoritmo es más lenta en comparación con otros valores (Figuras B.17d, B.20d y B.18d). De hecho el peor error relativo medio y la mayor variabilidad en soluciones, están asociados a tal valor de ψ , para los dos primeros circuitos; mientras que para el C. de la Investigación Científica es con ψ igual 0.1 Además, para estos tres circuitos, la mejor evolución del algoritmo se obtiene con ψ igual a 0.8; lo cual también se refleja en el mejor error relativo medio (Tabla 7.9). Dicho comportamiento sugiere que para estos circuitos es mejor una persistencia menor de feromona en los arcos. No obstante lo anterior, el número de iteraciones necesarias para encontrar la mejor solución, asociado a ψ igual a 0.8, está entre los dos promedios más grandes en los circuitos de la Investigación Científica y CCU (Figuras B.23 y B.25).

En cuanto a los circuitos Deportivo y Escolar, la peor evolución se presenta con ψ igual a 0.8, incluso el promedio de iteraciones necesarias para encontrar la mejor solución está entre los dos mayores para ambos circuitos. De igual manera, el peor error relativo medio está asociado al valor de ψ mencionado (Tabla 7.9). Por otra parte, la mejor evolución del algoritmo, el menor error relativo y medio y menor variabilidad en soluciones se obtiene con: ψ igual a 0.01 para el C. Deportivo (Figura B.16d, Tablas 7.9 y C.16 y con ψ igual a 0.3 para el C. Escolar (Figura B.19e, Tablas 7.9 y C.19). Dado lo anterior, se puede pensar que para ambos circuitos es mejor una mayor persistencia de la feromona en los arcos.

Adicionalmente, para los circuitos Exterior, de la Investigación Científica, Escolar y CCU se logra encontrar la solución B, C, D y E, respectivamente, con cualquier valor propuesto para ψ . En cuanto al C. Deportivo, la solución A no se encuentra al utilizar un factor de decaimiento igual a 0.5 u 0.8 (Tabla 7.9).

Por último, cabe señalar que, en los casos donde funciona mejor una valor cercano a 0 (o sea 0.01 y 0.3) es 0.8 el que da los peores resultados. Asimismo, en los circuitos donde es mejor 0.8, los valores más pequeños (0.01 y 0.1) producen los peores resultados.

Parámetro q_0 : Para todos los circuitos, se observa que el valor más cercano a 1 (0.95) lleva hacia la peor evolución del algoritmo (Figuras B.16e, B.17e B.18e, B.19d y B.20e) y a la mayor variabilidad de las soluciones finales. Incluso el peor error relativo medio está asociado a dicho valor (Tabla 7.9), es decir una exploración muy baja, es contraproducente para el algoritmo.

Por otra parte, para los circuitos Deportivo y de la Investigación Científica se observa que también un valor de 0.9 llevan hacia malos resultados (Figuras B.16e y B.18e). Mientras que los valores de 0.7, 0.5 y 0.3 permiten una convergencia más rápida y son los dos últimos que hacen que el algoritmo tenga un mejor desempeño, así como menor variabilidad en las soluciones finales (Tablas C.16 y C.18). En cuanto al menor error relativo medio, éste se asocia a un valor de q_0 igual a 0.5 y 0.3, para cada circuito respectivamente (Tabla 7.9).

Para el C. Exterior, valores cercanos a 1 (0.9 y 0.95) permiten que el algoritmo converja inicialmente más rápido a la mejor solución (Figura B.17e); sin embargo, esos resultados se ven superados después de cierto número de iteraciones por los valores más pequeños, en este caso 0.3 y 0.5, siendo éstos los que generan el menor error relativo medio (Tabla 7.9).

En cuanto al C. Escolar, con q_0 igual a 0.5 ó 0.7, el comportamiento de ACS es muy similar y la convergencia es más rápida, en comparación con los demás valores (Figura B.19d). De hecho con q_0 igual a 0.5, se tiene el menor error relativo medio, incluso entre todas las configuraciones propuestas para los parámetros (Tabla 7.9) .

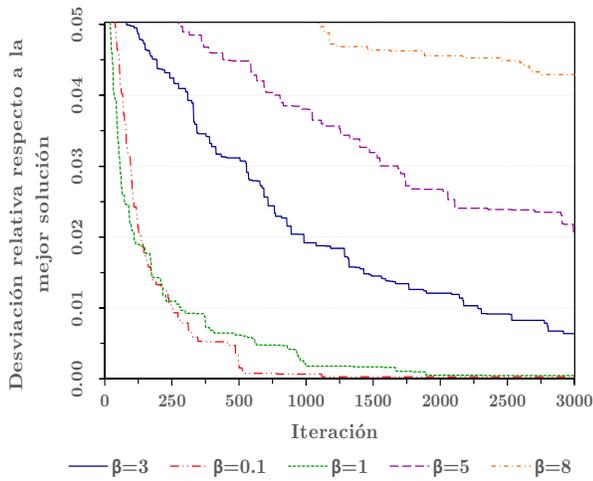
Para el C. CCU, se puede observar que el valor 0.9 hace que el algoritmo tenga la mejor evolución (Figura B.20e) y el menor error relativo medio (Tabla 7.9). Aunque, tal comportamiento es muy parecido al utilizar q_0 igual a 0.7. Adicionalmente, para los circuitos Escolar y CCU se logra encontrar la solución D y E, respectivamente, con cualquier valor propuesto para q_0 (Tabla 7.9). En cuanto a los tres circuitos restantes no se encuentra la solución A, B o C, según sea el caso, cuando q_0 es igual a 0.95.

Por último, para los circuitos Exterior y de la Investigación Científica se puede observar que los valores de q_0 que generan la mejor calidad en las soluciones, también son los valores para los que se necesita un mayor número de iteraciones para alcanzar la mejor solución (Figuras B.22 y B.23). Mientras que en el caso del C. Deportivo, el valor que produce las mejores soluciones, también es el valor para el que se necesita un menor número de iteraciones (Figura B.21)

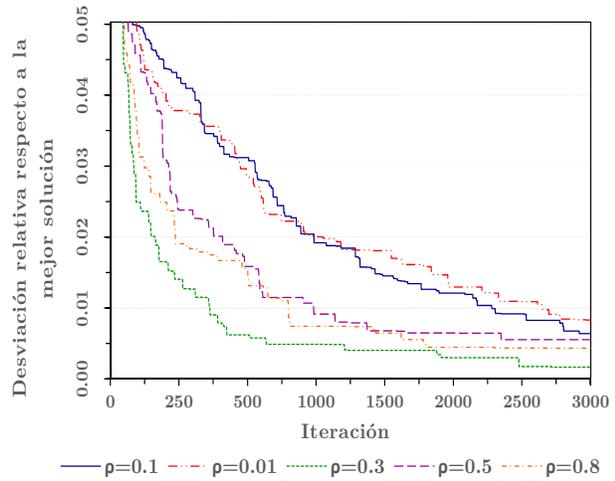
Búsqueda local: Al agregar la heurística de mejora (*3-opt*), el algoritmo ACS converge más rápido y el error relativo medio disminuye (Figuras B.16f, B.17f, B.18f, B.19f y B.20f). De hecho, se ve una diferencia notable en la evolución del algoritmo para los Circuitos Deportivo, de la Investigación Científica y CCU. Aunque para todos los circuitos se logra encontrar la solución A, B, C, D, o E, según sea el caso, con y sin la búsqueda local (Tabla 7.9).

Por otra parte, es importante mencionar que para los circuitos Exterior y CCU, el menor error relativo entre todas las configuraciones de los parámetros se genera al agregar la búsqueda local.

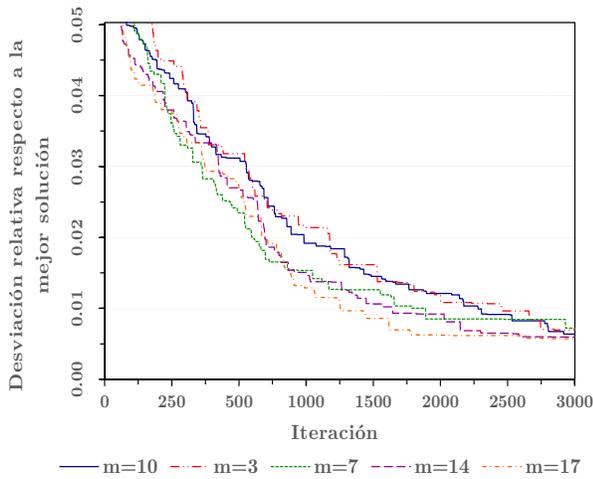
Por último, al utilizar la heurística de mejora, el promedio de iteraciones necesarias para encontrar la mejor solución disminuye en todos los casos (Figuras B.21, B.22, B.23, B.24 y B.25).



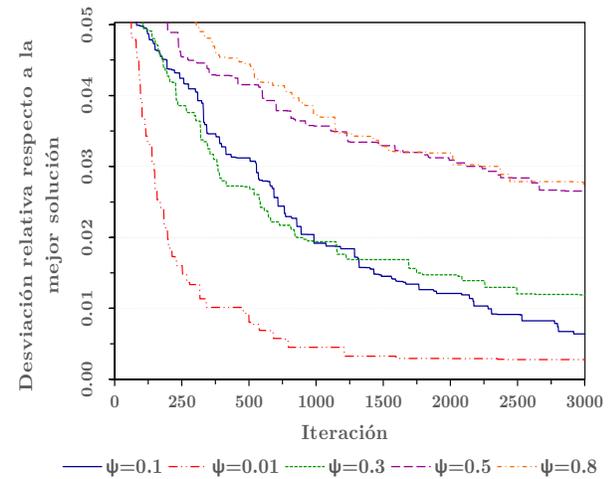
(a) ACS con varios valores de β



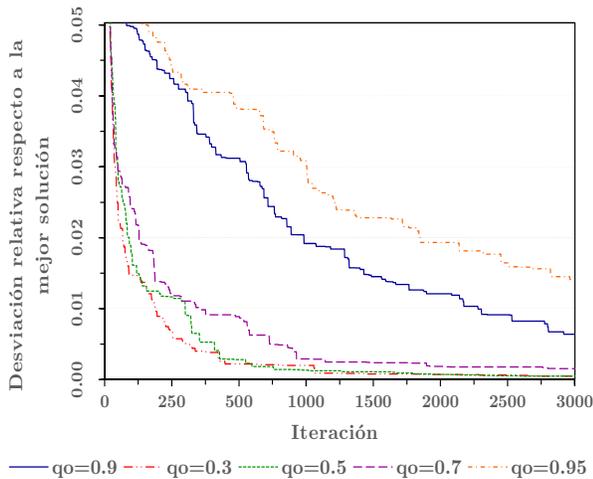
(b) ACS con varios valores de ρ



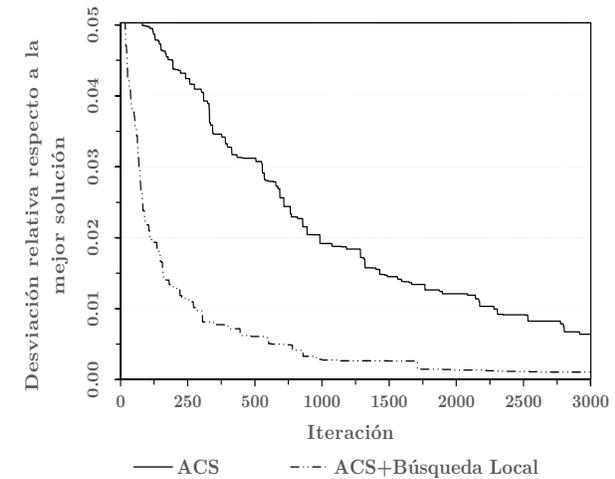
(c) ACS con varios valores de m



(d) ACS con varios valores de ψ



(e) ACS con varios valores de q_0



(f) ACS con y sin Búsqueda Local

Figura B.16: Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito Deportivo

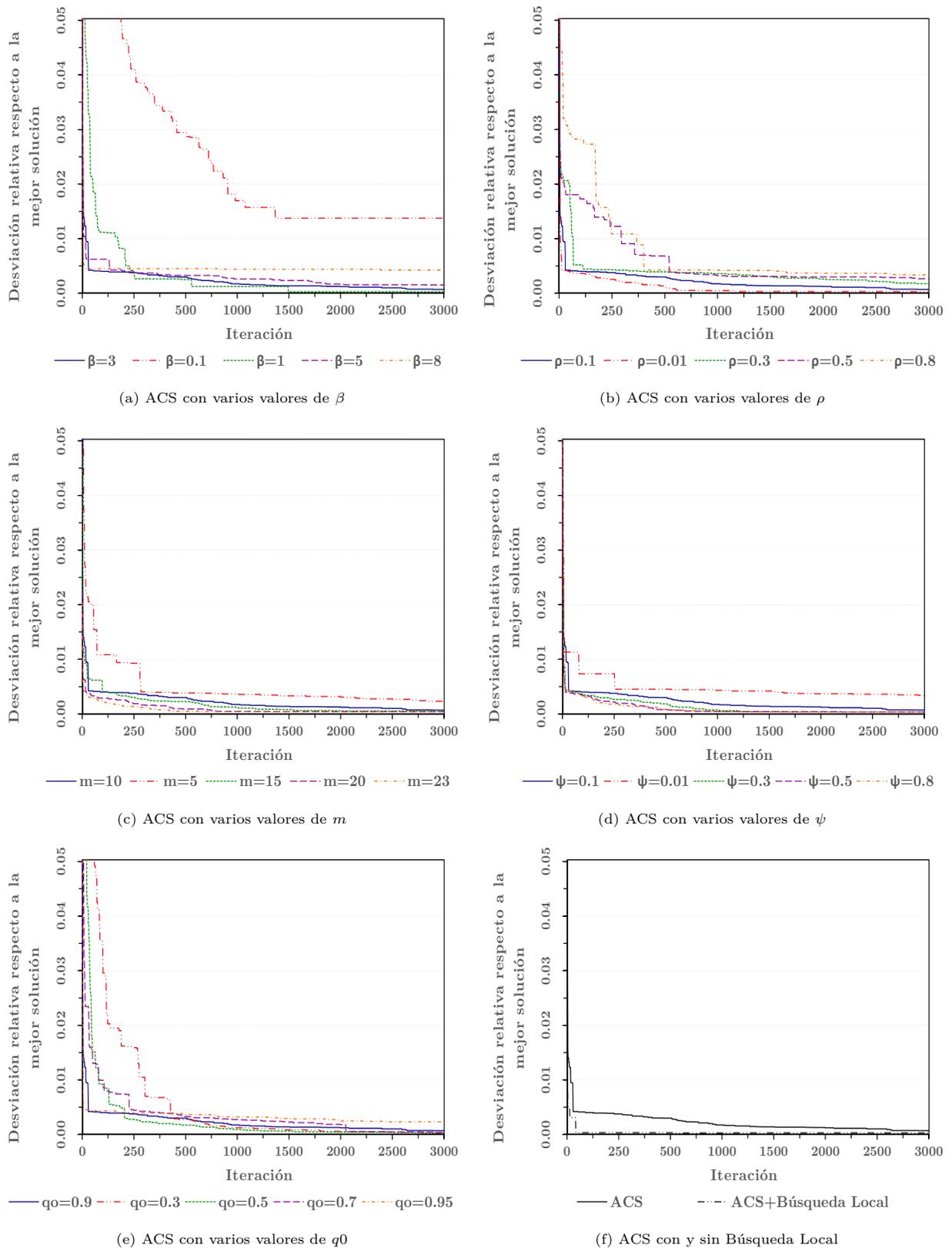


Figura B.17: Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito Exterior

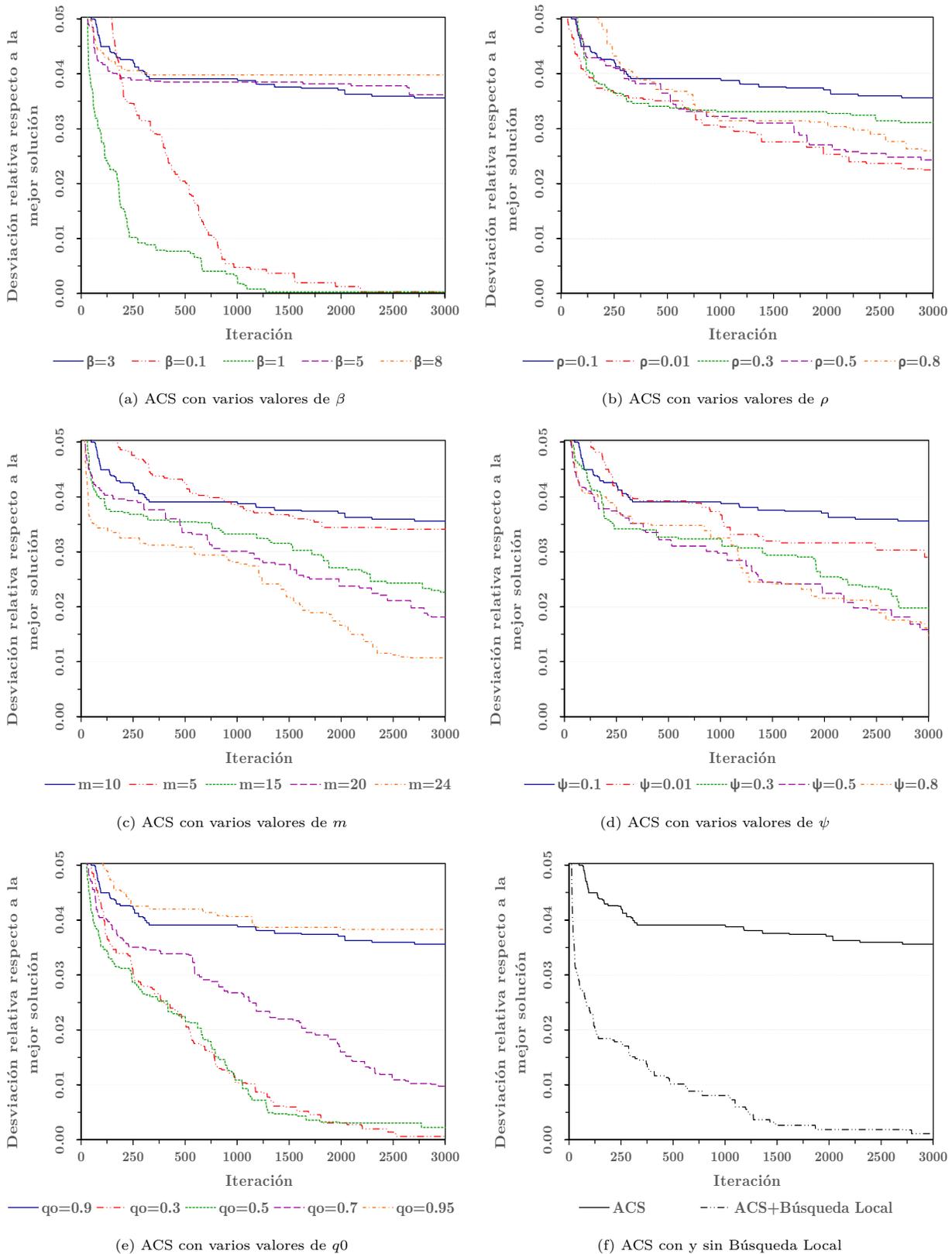


Figura B.18: Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito de la Investigación Científica

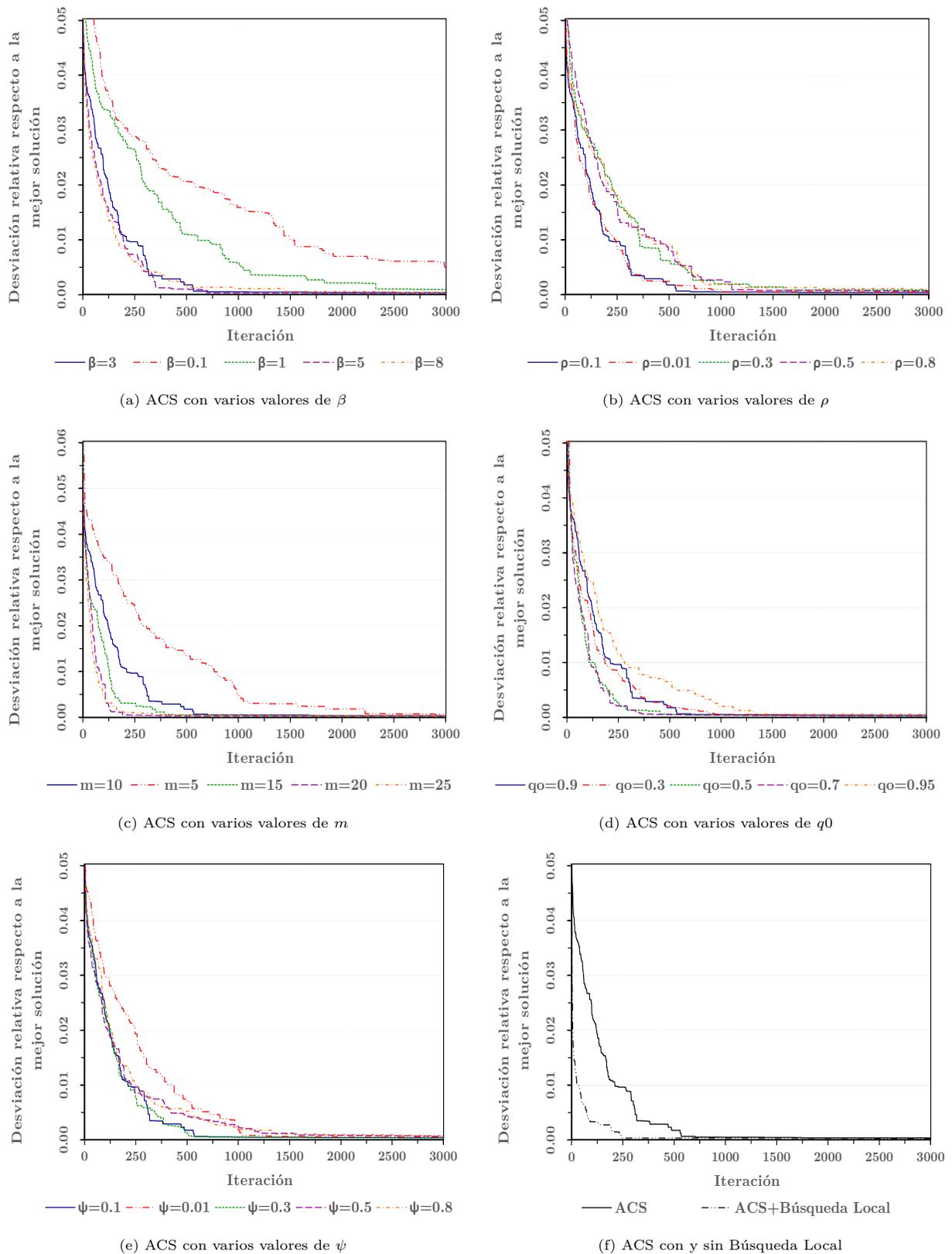
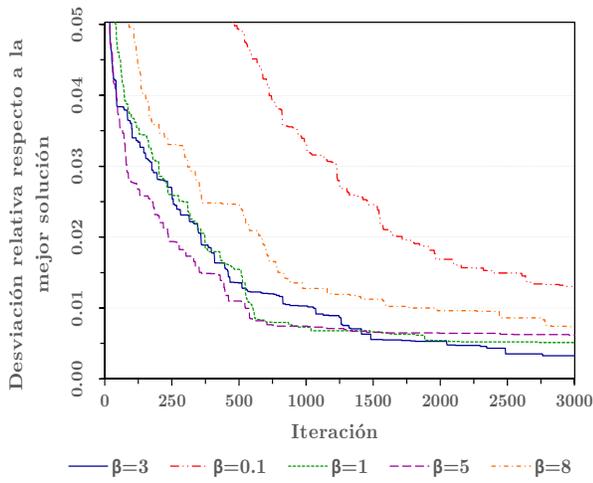
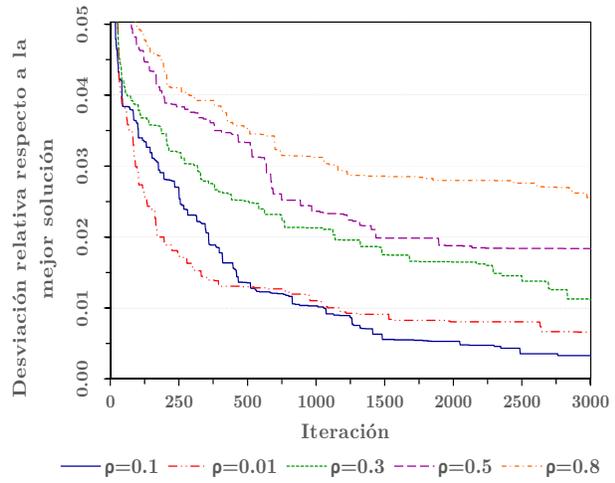


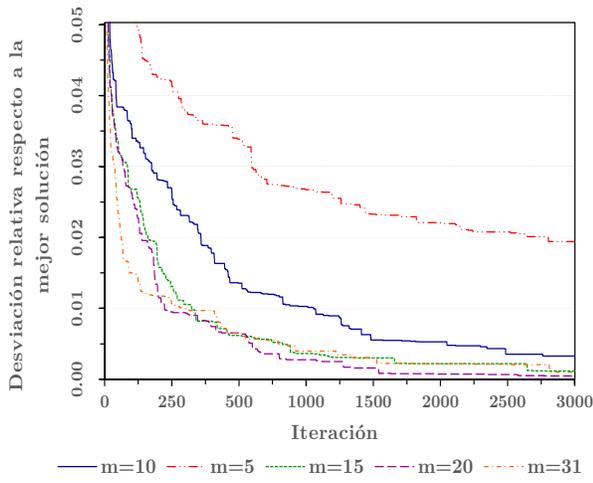
Figura B.19: Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito Escolar



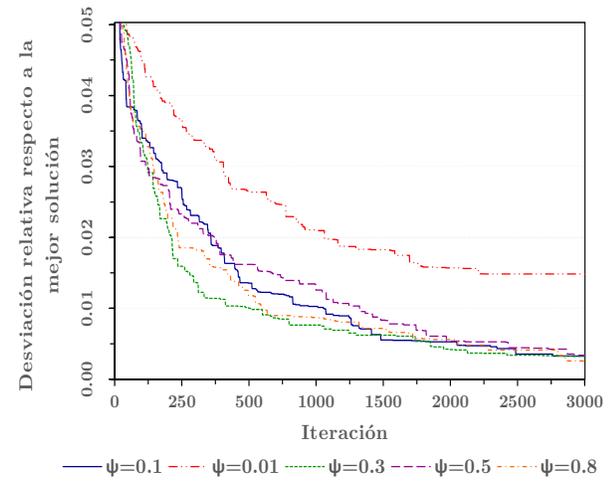
(a) ACS con varios valores de β



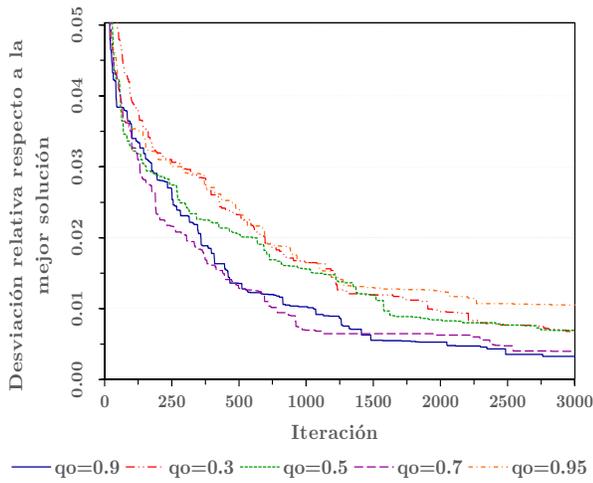
(b) ACS con varios valores de ρ



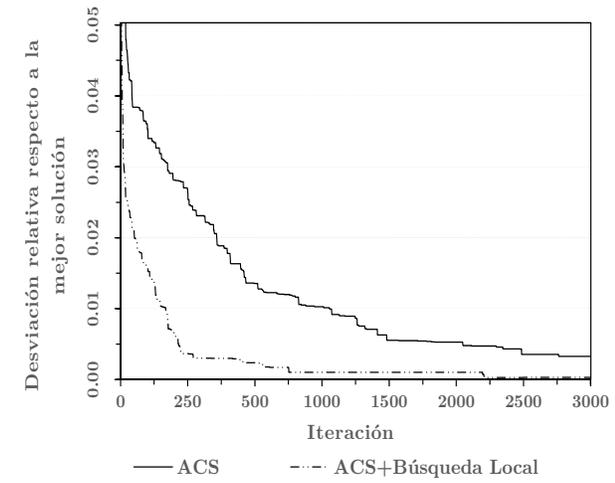
(c) ACS con varios valores de m



(d) ACS con varios valores de ψ



(e) ACS con varios valores de q_0



(f) ACS con y sin Búsqueda Local

Figura B.20: Comportamiento del algoritmo ACS para diversos parámetros, en el caso del Circuito CCU

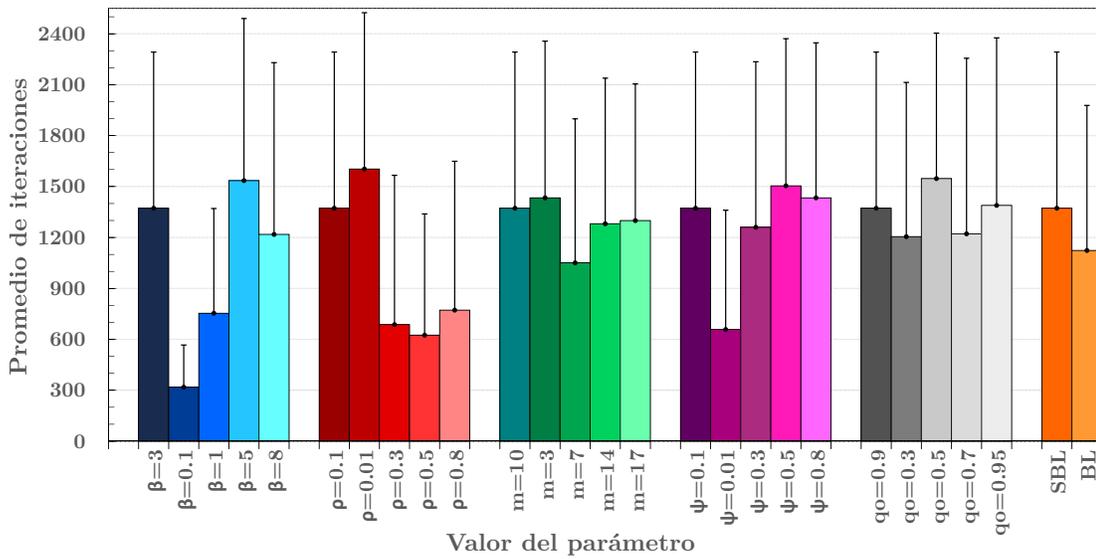


Figura B.21: Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Circuito Deportivo). Las barras indican la desviación estándar

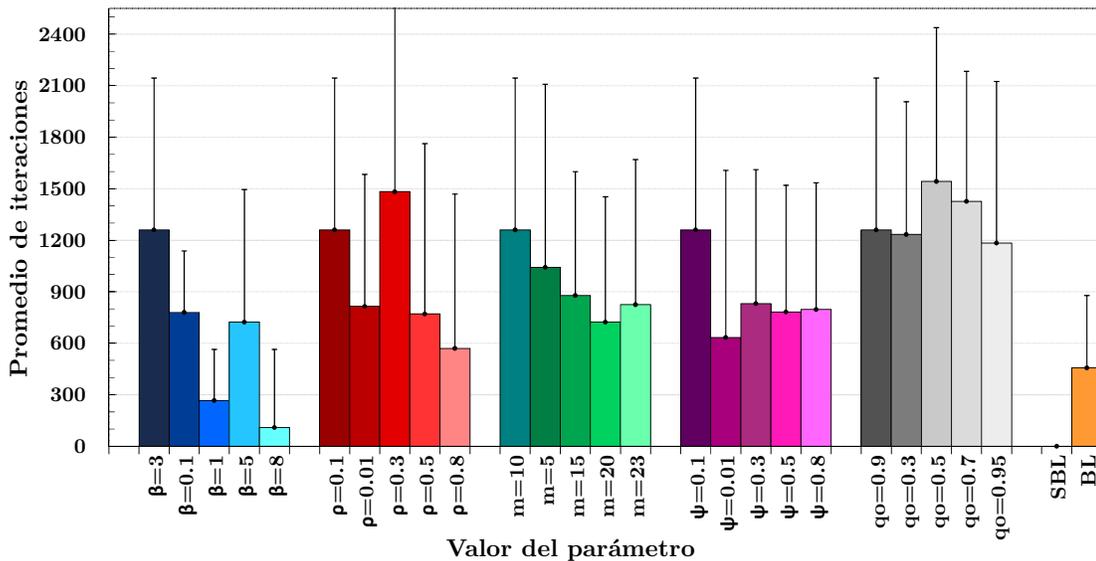


Figura B.22: Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Circuito Exterior). Las barras indican la desviación estándar

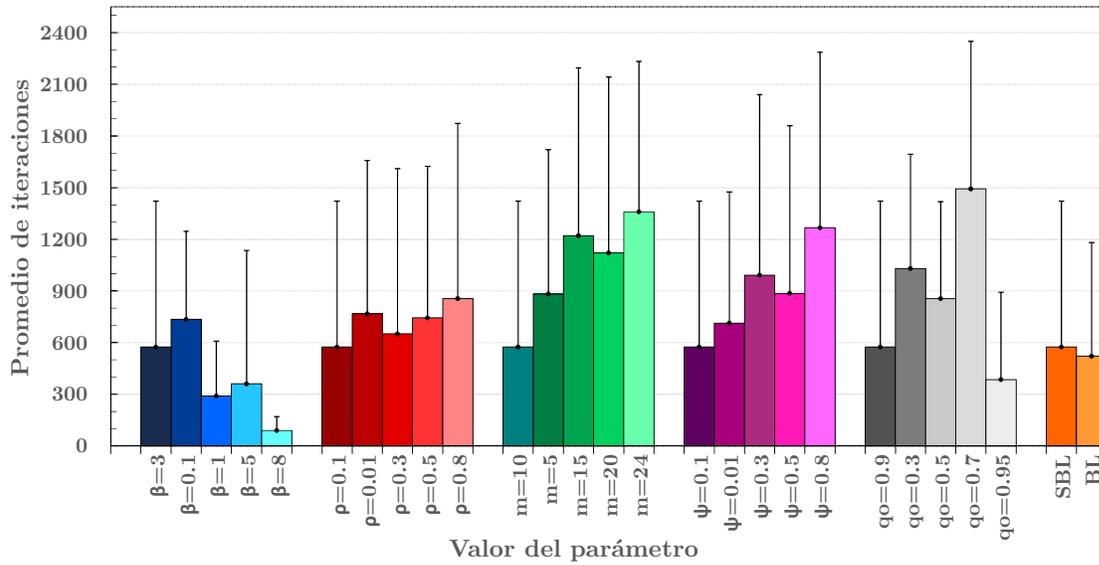


Figura B.23: Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Círculo de la Investigación Científica). Las barras indican la desviación estándar

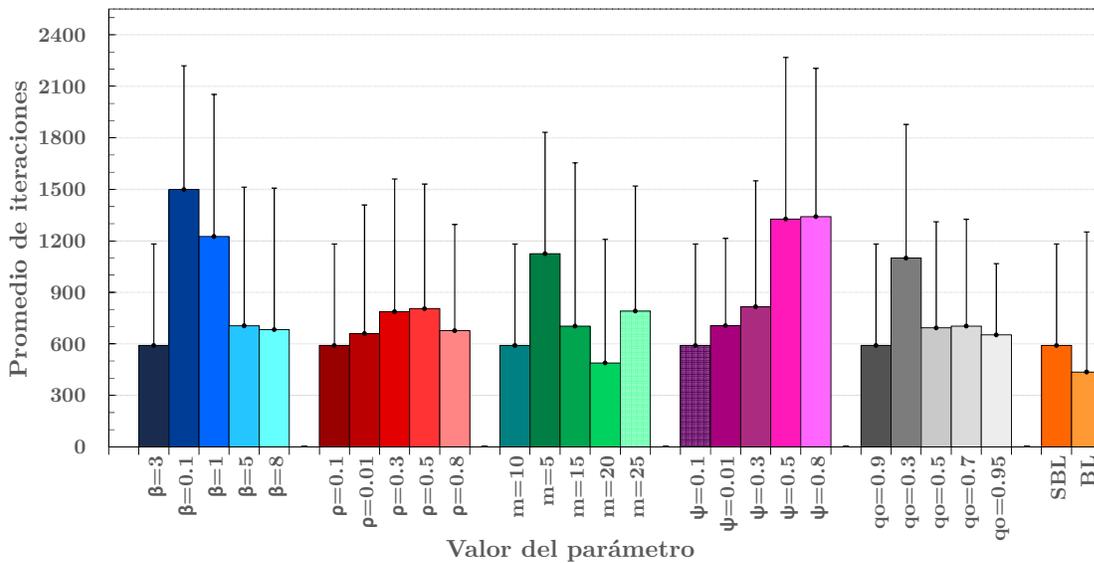


Figura B.24: Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Círculo Escolar). Las barras indican la desviación estándar

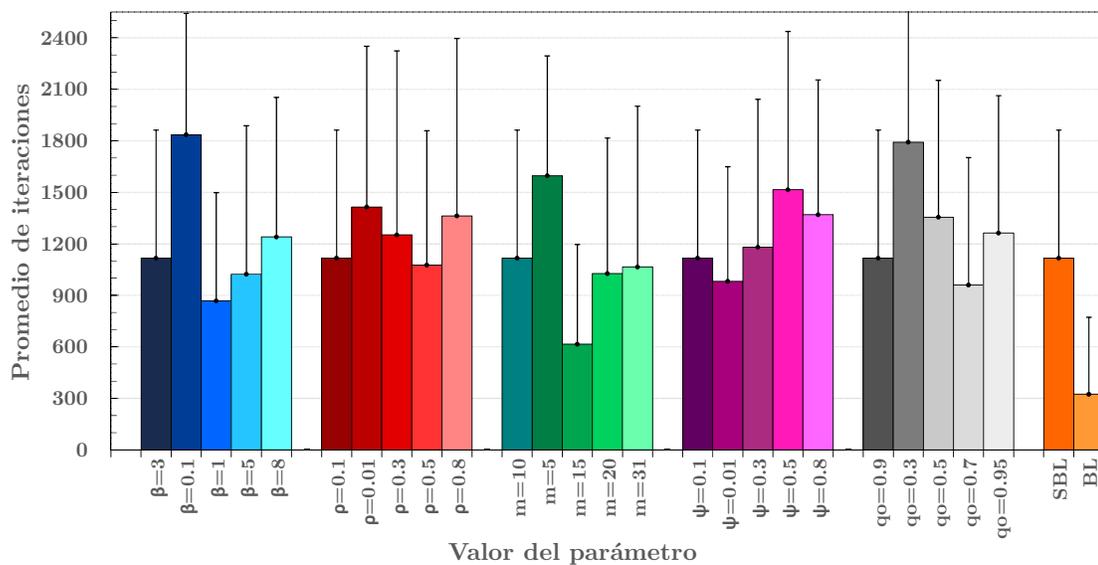


Figura B.25: Número de iteraciones promedio en encontrar la mejor solución con diversos parámetros del algoritmo ACS (Circuito CCU). Las barras indican la desviación estándar

Anexo C

Resultados estadísticos de los algoritmos ACO

Tabla C.1: Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito Deportivo

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	30 411.5	342.93	30241	31356	0.71	1.14	0.14	3.83	371.7	585.45	8	1856
$\beta = 0.1$	32 559.9	658.89	31265	33682	7.82	2.18	3.53	11.54	1496.6	901.34	69	2976
$\beta = 1$	30 239.5	32.24	30198	30319	0.14	0.11	0.00	0.40	797.2	820.60	27	2844
$\beta = 5$	30 539.3	362.34	30241	31356	1.13	1.20	0.14	3.83	720.8	928.26	9	2695
$\beta = 8$	31 131.3	323.91	30241	31326	3.09	1.07	0.14	3.74	856.2	855.50	11	2656
$\rho = 0.01$	30 238.4	10.71	30199	30242	0.13	0.04	0.00	0.15	419.0	376.95	25	1579
$\rho = 0.1$	30 241.9	0.43	30241	30243	0.15	0.00	0.14	0.15	104.8	82.40	26	430
$\rho = 0.3$	30 302.3	160.15	30241	30880	0.35	0.53	0.14	2.26	495.3	912.18	9	2826
$\rho = 0.8$	30 345.1	186.79	30241	31044	0.49	0.62	0.14	2.80	1002.0	1134.02	8	2988
$m = 3$	31 111.8	486.95	30199	31841	3.03	1.61	0.00	5.44	1006.0	904.35	13	2702
$m = 7$	30 663.5	473.69	30241	31357	1.54	1.57	0.14	3.84	780.1	785.72	2	2959
$m = 10$	30 501.6	417.13	30241	31341	1.01	1.38	0.14	3.79	600.0	828.93	5	2595
$m = 14$	30 297.4	80.67	30241	30674	0.33	0.27	0.14	1.58	667.3	816.85	7	2538

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.2: Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito Exterior

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	28 702.3	33.51	28628	28717	0.37	0.12	0.11	0.42	282.2	703.36	5	2654
$\beta = 0.1$	38 404.4	1151.77	35088	39726	34.30	4.03	22.70	38.92	1609.3	715.39	303	2877
$\beta = 1$	28 683.1	49.38	28596	28717	0.30	0.17	0.00	0.42	32.4	25.38	11	149
$\beta = 5$	28 698.2	38.61	28597	28717	0.36	0.14	0.00	0.42	329.3	822.29	2	2878
$\beta = 8$	28 709.0	30.44	28597	28717	0.40	0.11	0.00	0.42	2.6	1.96	1	11
$\rho = 0.01$	28 650.3	49.33	28596	28717	0.19	0.17	0.00	0.42	393.8	648.66	10	2806
$\rho = 0.1$	28 670.8	47.61	28597	28717	0.26	0.17	0.00	0.42	538.6	868.78	5	2923
$\rho = 0.3$	28 691.3	43.94	28597	28717	0.33	0.15	0.00	0.42	173.3	407.21	4	1504
$\rho = 0.8$	28 705.2	30.60	28628	28717	0.38	0.11	0.11	0.42	68.1	266.59	4	1398
$m = 5$	28 862.2	470.56	28717	31122	0.93	1.65	0.42	8.83	216.2	633.23	6	2720
$m = 10$	28 714.1	16.07	28629	28717	0.41	0.06	0.12	0.42	15.9	24.11	6	142
$m = 15$	28 705.3	30.25	28629	28717	0.38	0.11	0.12	0.42	235.9	655.97	5	2400
$m = 20$	28 704.2	33.59	28597	28717	0.38	0.12	0.00	0.42	162.3	483.76	3	1808

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.3: Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito de la Investigación Científica

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	30 080.2	218.12	28993	30138	3.75	0.75	0.00	3.95	373.4	689.21	7	2344
$\beta = 0.1$	36 202.0	883.95	33520	37661	24.86	3.05	15.61	29.90	1432.9	765.35	38	2842
$\beta = 1$	29 192.4	349.87	28993	29847	0.69	1.21	0.00	2.95	778.9	860.17	17	2988
$\beta = 5$	30 124.4	56.48	29848	30139	3.90	0.19	2.95	3.95	58.2	121.89	3	445
$\beta = 8$	30 142.5	98.44	29848	30485	3.96	0.34	2.95	5.15	119.8	369.17	2	1554
$\rho = 0.01$	29 598.9	451.40	28993	30138	2.09	1.56	0.00	3.95	160.8	147.64	15	526
$\rho = 0.1$	30 026.1	310.35	28993	30138	3.56	1.07	0.00	3.95	121.1	460.12	13	2545
$\rho = 0.3$	30 120.5	57.50	29847	30138	3.89	0.20	2.95	3.95	175.7	404.40	11	1692
$\rho = 0.8$	30 081.3	110.13	29847	30138	3.75	0.38	2.95	3.95	631.7	911.54	4	2748
$m = 5$	29 999.3	267.51	29162	30507	3.47	0.92	0.58	5.22	809.1	875.20	9	2762
$m = 10$	30 101.3	89.92	29847	30138	3.82	0.31	2.95	3.95	417.5	760.19	6	2679
$m = 15$	30 068.1	124.33	29847	30138	3.71	0.43	2.95	3.95	598.7	964.55	5	2961
$m = 20$	30 102.8	89.27	29847	30138	3.83	0.31	2.95	3.95	427.4	670.91	5	2115

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.4: Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito Escolar

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	33 167.3	14.17	33158	33200	0.03	0.04	0.00	0.13	535.3	784.00	11	2808
$\beta = 0.1$	39 801.4	612.95	38495	40951	20.04	1.85	16.10	23.50	1558.6	810.24	60	2945
$\beta = 1$	33 171.0	12.73	33158	33195	0.04	0.04	0.00	0.11	1519.3	876.27	49	2860
$\beta = 5$	33 164.4	13.89	33158	33218	0.02	0.04	0.00	0.18	977.8	1039.30	4	2986
$\beta = 8$	33 178.1	99.04	33158	33701	0.06	0.30	0.00	1.64	646.3	834.99	4	2763
$\rho = 0.01$	33 158.0	0.18	33158	33159	0.00	0.00	0.00	0.00	615.3	630.22	56	2683
$\rho = 0.1$	33 158.6	0.50	33158	33159	0.00	0.00	0.00	0.00	466.8	610.70	49	2673
$\rho = 0.3$	33 159.6	3.35	33158	33177	0.00	0.01	0.00	0.06	598.4	712.25	9	2613
$\rho = 0.8$	33 174.2	17.69	33159	33200	0.05	0.05	0.00	0.13	710.5	827.97	8	2691
$m = 5$	33 306.4	161.39	33159	33848	0.45	0.49	0.00	2.08	1079.1	925.17	16	2786
$m = 10$	33 194.2	22.92	33159	33271	0.11	0.07	0.00	0.34	896.0	929.00	9	2987
$m = 15$	33 176.8	18.74	33158	33218	0.06	0.06	0.00	0.18	838.8	881.17	8	2842
$m = 20$	33 177.4	17.96	33158	33200	0.06	0.05	0.00	0.13	677.8	808.55	6	2627

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.5: Estadísticas básicas de los parámetros evaluados del algoritmo AS aplicado al Circuito CCU

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	32 434.4	245.08	31761	32719	2.19	0.77	0.07	3.08	1034.3	943.57	9	2947
$\beta = 0.1$	41 256.4	1074.33	38765	42939	29.98	3.38	22.13	35.28	1254.5	854.52	82	2882
$\beta = 1$	32 341.0	268.78	31741	32663	1.89	0.85	0.00	2.91	1795.8	933.12	150	2985
$\beta = 5$	32 596.7	255.18	31899	32954	2.70	0.80	0.50	3.82	577.8	939.05	9	2894
$\beta = 8$	32 810.3	263.23	32465	33671	3.37	0.83	2.28	6.08	764.6	956.99	3	2919
$\rho = 0.01$	31 811.4	114.93	31741	32222	0.22	0.36	0.00	1.52	1072.5	798.97	106	2644
$\rho = 0.1$	32 053.9	266.28	31741	32449	0.99	0.84	0.00	2.23	653.7	687.38	32	2007
$\rho = 0.3$	32 261.9	299.66	31760	32674	1.64	0.94	0.06	2.94	1511.3	979.81	15	2996
$\rho = 0.8$	32 578.2	231.34	31815	32719	2.64	0.73	0.24	3.08	602.7	716.43	5	2499
$m = 5$	32 799.4	294.48	31797	33412	3.34	0.93	0.18	5.27	784.6	849.21	17	2666
$m = 10$	32 637.0	222.85	31797	32902	2.83	0.70	0.18	3.66	1242.8	1001.21	12	2874
$m = 15$	32 511.1	278.08	31761	32763	2.43	0.88	0.07	3.22	1055.0	925.62	8	2888
$m = 20$	32 589.5	177.74	31880	32720	2.68	0.56	0.44	3.09	959.6	952.74	9	2834

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.6: Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito Deportivo

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	30 445.5	359.88	30198	31356	0.82	1.19	0.00	3.83	870.6	935.03	16	2968
$\beta = 0.1$	32 613.7	1284.42	30277	35937	8.00	4.25	0.26	19.00	29.9	13.50	11	62
$\beta = 1$	30 413.5	389.84	30198	31355	0.71	1.29	0.00	3.83	1406.6	1058.92	24	2985
$\beta = 5$	30 323.2	281.84	30241	31357	0.41	0.93	0.14	3.84	570.5	849.04	10	2795
$\beta = 8$	30 476.9	424.85	30241	31356	0.92	1.41	0.14	3.83	498.7	776.81	24	2910
$\rho = 0.01$	30 219.5	21.87	30198	30241	0.07	0.07	0.00	0.14	552.6	369.96	198	1744
$\rho = 0.1$	30 292.9	207.54	30198	31057	0.31	0.69	0.00	2.84	269.2	466.13	33	2391
$\rho = 0.3$	30 400.6	351.87	30198	31340	0.67	1.17	0.00	3.78	502.6	753.42	12	2595
$\rho = 0.8$	30 383.2	301.47	30198	31057	0.61	1.00	0.00	2.84	755.2	967.35	10	2952
$m = 3$	31 972.1	707.20	30243	33430	5.88	2.34	0.15	10.70	399.7	613.20	3	2849
$m = 7$	30 611.2	446.31	30241	31356	1.37	1.48	0.14	3.83	1093.0	895.13	10	2781
$m = 10$	30 579.0	378.92	30199	31059	1.26	1.25	0.00	2.85	1504.6	955.85	17	2879
$m = 14$	30 402.0	323.20	30241	31356	0.68	1.07	0.14	3.83	730.5	798.12	12	2403
$e = 1$	30 248.7	23.49	30241	30318	0.17	0.08	0.14	0.40	306.3	448.45	12	2274
$e = 5$	30 304.7	202.30	30241	31043	0.35	0.67	0.14	2.80	501.1	780.91	16	2807
$e = 10$	30 343.4	241.15	30241	31057	0.48	0.80	0.14	2.84	545.8	803.14	6	2662
$e = 25$	30 581.9	405.93	30241	31356	1.27	1.34	0.14	3.83	843.2	969.96	12	2922

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.7: Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito Exterior

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	28 651.6	44.50	28597	28717	0.19	0.16	0.00	0.42	782.7	881.07	6	2896
$\beta = 0.1$	35 193.4	2136.23	31003	39152	23.07	7.47	8.42	36.91	37.2	8.56	24	62
$\beta = 1$	28 992.9	859.05	28596	31482	1.39	3.00	0.00	10.09	563.6	945.44	12	2915
$\beta = 5$	28 713.0	21.91	28597	28717	0.41	0.08	0.00	0.42	5.9	2.41	1	11
$\beta = 8$	28 717.0	0.00	28717	28717	0.42	0.00	0.42	0.42	3.2	1.50	1	6
$\rho = 0.01$	28 601.1	10.84	28596	28629	0.02	0.04	0.00	0.12	966.0	734.46	135	2789
$\rho = 0.1$	28 606.1	24.00	28596	28717	0.04	0.08	0.00	0.42	650.1	705.38	35	2634
$\rho = 0.3$	28 616.5	24.52	28597	28717	0.07	0.09	0.00	0.42	1053.5	896.42	8	2818
$\rho = 0.8$	28 873.5	667.76	28596	31516	0.97	2.34	0.00	10.21	220.5	493.25	4	2139
$m = 5$	30 697.6	1608.84	28717	33107	7.35	5.63	0.42	15.77	579.2	897.16	7	2918
$m = 10$	29 333.8	1028.13	28717	32061	2.58	3.60	0.42	12.12	411.2	798.48	7	2811
$m = 15$	28 889.1	609.64	28597	31122	1.02	2.13	0.00	8.83	217.4	410.96	5	1659
$m = 20$	28 718.7	190.44	28596	29694	0.43	0.67	0.00	3.84	126.7	323.63	6	1703
$e = 1$	28 702.2	33.58	28628	28717	0.37	0.12	0.11	0.42	211.4	592.07	7	2798
$e = 10$	28 660.8	47.61	28596	28717	0.23	0.17	0.00	0.42	799.3	920.22	6	2759
$e = 20$	28 734.8	436.62	28597	31034	0.49	1.53	0.00	8.53	780.3	956.55	4	2646
$e = 35$	28 759.6	359.88	28596	30334	0.57	1.26	0.00	6.08	291.3	419.72	5	1569

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.8: Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito de la Investigación Científica

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	29 593.7	573.92	28993	30138	2.07	1.98	0.00	3.95	325.2	578.78	10	2830
$\beta = 0.1$	33 475.1	1710.48	30153	37327	15.46	5.90	4.00	28.74	47.5	16.02	22	89
$\beta = 1$	30 120.1	579.36	28993	31109	3.89	2.00	0.00	7.30	438.6	738.55	11	2777
$\beta = 5$	29 975.6	395.55	28993	30138	3.39	1.36	0.00	3.95	162.6	408.46	3	1714
$\beta = 8$	30 134.5	240.11	28993	30485	3.94	0.83	0.00	5.15	79.0	386.13	1	2122
$\rho = 0.01$	28 993.0	0.00	28993	28993	0.00	0.00	0.00	0.00	677.1	472.76	250	2265
$\rho = 0.1$	29 412.7	561.07	28993	30138	1.45	1.94	0.00	3.95	251.9	387.50	24	1817
$\rho = 0.3$	29 800.1	509.70	28993	30138	2.78	1.76	0.00	3.95	398.7	679.03	8	2317
$\rho = 0.8$	29 524.2	559.46	28993	30537	1.83	1.93	0.00	5.33	562.7	861.52	7	2827
$m = 5$	30 346.3	730.10	29123	32264	4.67	2.52	0.45	11.28	982.5	1137.29	12	2985
$m = 10$	29 737.9	435.84	28993	30637	2.57	1.50	0.00	5.67	823.9	892.23	10	2819
$m = 15$	29 586.6	496.81	28993	30268	2.05	1.71	0.00	4.40	759.8	952.50	7	2755
$m = 20$	29 726.5	489.24	28993	30138	2.53	1.69	0.00	3.95	886.3	1065.04	9	2998
$e = 1$	30 118.2	73.58	29847	30138	3.88	0.25	2.95	3.95	705.9	926.53	8	2754
$e = 10$	30 003.6	350.41	28993	30138	3.49	1.21	0.00	3.95	617.4	818.57	8	2284
$e = 20$	29 873.7	460.77	28993	30138	3.04	1.59	0.00	3.95	312.5	515.41	7	1915
$e = 35$	29 568.5	557.39	28993	30138	1.98	1.92	0.00	3.95	433.6	720.11	7	2803

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.9: Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito Escolar

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	33 221.2	50.39	33159	33381	0.19	0.15	0.00	0.67	724.3	891.19	6	2840
$\beta = 0.1$	36 514.3	1176.62	34339	39516	10.12	3.55	3.56	19.17	51.6	15.41	26	79
$\beta = 1$	33 294.5	167.64	33159	34100	0.41	0.51	0.00	2.84	795.3	859.11	21	2951
$\beta = 5$	33 183.5	37.21	33158	33306	0.08	0.11	0.00	0.45	539.7	933.89	5	2959
$\beta = 8$	33 165.5	19.61	33158	33264	0.02	0.06	0.00	0.32	375.8	620.14	6	2691
$\rho = 0.01$	33 159.4	3.06	33158	33171	0.00	0.01	0.00	0.04	470.4	173.04	130	970
$\rho = 0.1$	33 187.5	36.63	33158	33271	0.09	0.11	0.00	0.34	349.2	664.19	27	2643
$\rho = 0.3$	33 196.7	44.45	33158	33318	0.12	0.13	0.00	0.48	475.4	854.44	16	2875
$\rho = 0.8$	33 243.5	102.12	33158	33678	0.26	0.31	0.00	1.57	661.8	755.87	5	2521
$m = 5$	33 928.4	648.39	33194	35464	2.32	1.96	0.11	6.95	610.6	859.81	5	2899
$m = 10$	33 418.8	378.16	33159	34602	0.79	1.14	0.00	4.35	987.3	887.03	6	2696
$m = 15$	33 253.5	101.91	33159	33713	0.29	0.31	0.00	1.67	492.0	533.33	11	1959
$m = 20$	33 282.1	248.06	33158	34464	0.37	0.75	0.00	3.94	615.5	801.18	11	2800
$e = 1$	33 159.1	2.13	33158	33170	0.00	0.01	0.00	0.04	821.8	817.61	10	2416
$e = 10$	33 183.3	26.69	33158	33253	0.08	0.08	0.00	0.29	522.1	744.19	9	2981
$e = 20$	33 208.4	59.09	33158	33348	0.15	0.18	0.00	0.57	616.6	866.81	9	2743
$e = 35$	33 262.0	232.15	33158	34446	0.31	0.70	0.00	3.88	758.5	887.16	12	2942

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.10: Estadísticas básicas de los parámetros evaluados del algoritmo EAS aplicado al Circuito CCU

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	31 976.1	316.50	31740	32674	0.74	1.00	0.00	2.94	1549.8	979.50	17	2957
$\beta = 0.1$	36 610.7	1786.63	33292	41100	15.35	5.63	4.89	29.49	54.2	16.74	28	85
$\beta = 1$	32 591.4	485.02	31741	33688	2.68	1.53	0.00	6.14	613.6	641.60	17	2885
$\beta = 5$	32 114.0	423.33	31740	33047	1.18	1.33	0.00	4.12	763.5	774.15	3	2886
$\beta = 8$	32 375.1	438.86	31741	33186	2.00	1.38	0.00	4.56	570.0	832.67	6	2726
$\rho = 0.01$	31 783.0	159.51	31740	32414	0.14	0.50	0.00	2.12	1110.2	686.28	452	2957
$\rho = 0.1$	31 852.8	222.49	31740	32414	0.36	0.70	0.00	2.12	942.6	918.43	49	2867
$\rho = 0.3$	31 870.4	275.65	31740	32854	0.41	0.87	0.00	3.51	1425.0	1010.63	40	2990
$\rho = 0.8$	32 155.9	412.79	31740	32990	1.31	1.30	0.00	3.94	1329.9	912.43	25	2952
$m = 5$	33 430.8	690.66	31740	34844	5.33	2.18	0.00	9.78	529.3	740.43	10	2406
$m = 10$	32 517.8	512.21	31740	33964	2.45	1.61	0.00	7.01	875.0	921.59	18	2925
$m = 15$	32 382.6	407.35	31740	33047	2.02	1.28	0.00	4.12	1247.1	1007.05	18	2980
$m = 20$	32 111.9	379.30	31740	32911	1.17	1.20	0.00	3.69	1493.5	890.21	88	2922
$e = 1$	32 065.1	421.54	31741	32674	1.02	1.33	0.00	2.94	1336.6	1035.48	32	2974
$e = 10$	31 774.3	167.25	31740	32658	0.11	0.53	0.00	2.89	1458.8	895.32	17	2887
$e = 20$	31 862.1	234.01	31740	32663	0.38	0.74	0.00	2.91	1716.1	966.42	37	2991
$e = 35$	31 883.9	270.96	31740	32713	0.45	0.85	0.00	3.07	1106.0	882.87	13	2972

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.11: Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito Deportivo

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	30 201.5	10.75	30198	30241	0.01	0.04	0.00	0.14	1006.2	915.16	127	2991
$\beta = 0.1$	30 204.8	14.55	30198	30242	0.02	0.05	0.00	0.15	476.7	97.96	266	740
$\beta = 1$	30 199.7	7.81	30198	30241	0.01	0.03	0.00	0.14	1078.3	962.32	248	2988
$\beta = 5$	30 217.1	21.29	30198	30241	0.06	0.07	0.00	0.14	818.4	780.06	29	2605
$\beta = 8$	30 241.3	0.48	30241	30242	0.14	0.00	0.14	0.15	791.6	741.33	51	2793
$\rho = 0.1$	30 202.8	12.97	30198	30241	0.02	0.04	0.00	0.14	1420.9	988.48	34	2985
$\rho = 0.3$	30 215.8	20.94	30198	30241	0.06	0.07	0.00	0.14	1291.0	795.67	23	2823
$\rho = 0.5$	30 278.2	198.55	30199	31051	0.27	0.66	0.00	2.82	1499.9	889.45	11	2814
$\rho = 0.8$	30 643.9	469.75	30200	31968	1.48	1.56	0.01	5.86	1271.6	887.91	5	2775
$m = 3$	30 234.5	15.96	30199	30242	0.12	0.05	0.00	0.15	1012.2	859.45	160	2996
$m = 7$	30 222.7	21.40	30198	30242	0.08	0.07	0.00	0.15	730.8	697.30	62	2431
$m = 10$	30 212.8	20.29	30198	30241	0.05	0.07	0.00	0.14	1045.8	840.85	186	2934
$m = 14$	30 208.4	18.28	30198	30241	0.03	0.06	0.00	0.14	693.0	634.41	183	2396
Sin Reini- cialización	30 221.3	21.39	30198	30241	0.08	0.07	0.00	0.14	228.4	80.10	40	387
Solución Global	30 208.0	18.50	30198	30241	0.03	0.06	0.00	0.14	908.9	742.31	135	2994

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.12: Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito Exterior

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	28 596.7	0.48	28596	28597	0.00	0.00	0.00	0.00	720.2	716.08	104	2776
$\beta = 0.1$	28 828.2	459.45	28596	30326	0.81	1.61	0.00	6.05	575.1	78.98	431	787
$\beta = 1$	28 596.3	0.45	28596	28597	0.00	0.00	0.00	0.00	339.2	57.17	247	501
$\beta = 5$	28 597.8	5.91	28596	28629	0.01	0.02	0.00	0.12	1195.8	956.89	43	2948
$\beta = 8$	28 615.1	41.42	28597	28717	0.07	0.14	0.00	0.42	1184.7	920.98	2	2823
$\rho = 0.1$	28 596.8	0.41	28596	28597	0.00	0.00	0.00	0.00	433.4	658.68	32	2688
$\rho = 0.3$	28 596.9	0.35	28596	28597	0.00	0.00	0.00	0.00	619.9	643.33	10	2311
$\rho = 0.5$	28 633.1	50.67	28596	28717	0.13	0.18	0.00	0.42	985.7	865.03	5	2817
$\rho = 0.8$	29 011.8	694.51	28597	31680	1.45	2.43	0.00	10.78	1597.2	755.11	258	2776
$m = 5$	28 626.0	47.44	28596	28717	0.10	0.17	0.00	0.42	1197.0	902.11	108	2985
$m = 10$	28 626.3	41.27	28596	28717	0.11	0.14	0.00	0.42	1171.1	818.24	100	2964
$m = 15$	28 603.1	12.86	28596	28629	0.02	0.04	0.00	0.12	1453.3	925.62	93	2985
$m = 20$	28 597.9	5.89	28596	28629	0.01	0.02	0.00	0.12	848.2	612.24	180	2605
Sin Reini- cialización	28 596.9	0.31	28596	28597	0.00	0.00	0.00	0.00	356.7	128.30	88	609
Solución Global	28 596.0	0.00	28596	28596	0.00	0.00	0.00	0.00	591.0	297.95	139	1276

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.13: Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito de la Investigación Científica

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	28 993.0	0.00	28993	28993	0.00	0.00	0.00	0.00	535.5	391.24	107	1594
$\beta = 0.1$	29 151.7	193.35	28993	29710	0.55	0.67	0.00	2.47	691.2	125.16	434	902
$\beta = 1$	28 993.0	0.00	28993	28993	0.00	0.00	0.00	0.00	333.7	59.18	246	504
$\beta = 5$	29 228.2	334.08	28993	29848	0.81	1.15	0.00	2.95	1291.5	946.92	48	2959
$\beta = 8$	29 620.8	333.75	28993	29848	2.17	1.15	0.00	2.95	1076.2	871.07	37	2959
$\rho = 0.1$	28 993.0	0.00	28993	28993	0.00	0.00	0.00	0.00	448.6	451.37	31	1912
$\rho = 0.3$	29 116.5	249.08	28993	29803	0.43	0.86	0.00	2.79	1301.1	770.76	280	2852
$\rho = 0.5$	29 320.3	392.09	28993	30005	1.13	1.35	0.00	3.49	1507.9	711.17	271	2763
$\rho = 0.8$	29 808.2	430.40	28993	30486	2.81	1.48	0.00	5.15	1556.6	878.18	5	2778
$m = 5$	29 540.5	367.07	28993	30137	1.89	1.27	0.00	3.95	1261.9	919.84	167	2978
$m = 10$	29 139.9	274.15	28993	29847	0.51	0.95	0.00	2.95	991.7	763.25	95	2983
$m = 15$	29 066.7	217.19	28993	29846	0.25	0.75	0.00	2.94	657.1	496.66	131	2595
$m = 20$	28 997.3	23.73	28993	29123	0.01	0.08	0.00	0.45	860.3	700.15	180	2480
Sin Reini- cialización	29 097.9	275.40	28993	29847	0.36	0.95	0.00	2.95	349.3	147.37	118	669
Solución Global	28 993.0	0.00	28993	28993	0.00	0.00	0.00	0.00	454.8	462.54	37	1936

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.14: Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito Escolar

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	33 158.1	0.25	33158	33159	0.00	0.00	0.00	0.00	510.3	620.42	194	2919
$\beta = 0.1$	33 351.0	351.13	33158	34429	0.58	1.06	0.00	3.83	751.1	140.70	374	1006
$\beta = 1$	33 159.7	4.12	33158	33170	0.01	0.01	0.00	0.04	479.7	87.54	328	711
$\beta = 5$	33 158.0	0.00	33158	33158	0.00	0.00	0.00	0.00	546.3	420.48	113	1702
$\beta = 8$	33 158.1	0.35	33158	33159	0.00	0.00	0.00	0.00	665.5	632.27	103	2924
$\rho = 0.1$	33 158.2	0.38	33158	33159	0.00	0.00	0.00	0.00	910.8	747.39	48	2807
$\rho = 0.3$	33 185.6	28.47	33158	33270	0.08	0.09	0.00	0.34	1095.7	791.72	15	2387
$\rho = 0.5$	33 314.6	197.22	33158	33760	0.47	0.59	0.00	1.82	1242.8	1000.76	8	2786
$\rho = 0.8$	33 691.5	372.77	33216	34488	1.61	1.12	0.17	4.01	1640.8	776.57	5	2789
$m = 5$	33 160.3	4.46	33158	33177	0.01	0.01	0.00	0.06	856.7	742.86	181	2807
$m = 10$	33 158.7	2.37	33158	33171	0.00	0.01	0.00	0.04	710.0	668.22	233	2634
$m = 15$	33 158.2	0.38	33158	33159	0.00	0.00	0.00	0.00	514.5	596.13	197	2998
$m = 20$	33 158.2	0.41	33158	33159	0.00	0.00	0.00	0.00	524.6	520.91	140	2527
Sin Reini- cialización	33 158.1	0.31	33158	33159	0.00	0.00	0.00	0.00	302.5	83.76	135	494
Solución Global	33 159.7	7.25	33158	33198	0.01	0.02	0.00	0.12	1106.6	788.51	173	2656

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.15: Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado al Circuito CCU

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	31 740.0	0.00	31740	31740	0.00	0.00	0.00	0.00	391.0	37.85	303	463
$\beta = 0.1$	32 089.1	300.48	31740	32819	1.10	0.95	0.00	3.40	768.1	115.76	581	1037
$\beta = 1$	31 760.9	110.89	31740	32348	0.07	0.35	0.00	1.92	436.6	78.52	287	573
$\beta = 5$	31 740.4	0.50	31740	31741	0.00	0.00	0.00	0.00	547.0	560.99	160	2469
$\beta = 8$	31 740.9	0.35	31740	31741	0.00	0.00	0.00	0.00	932.7	702.18	159	2656
$\rho = 0.1$	31 740.0	0.18	31740	31741	0.00	0.00	0.00	0.00	592.7	435.99	64	1518
$\rho = 0.3$	31 758.0	51.38	31740	31969	0.06	0.16	0.00	0.72	1103.1	782.15	19	2362
$\rho = 0.5$	31 955.7	246.53	31740	32531	0.68	0.78	0.00	2.49	1370.3	939.32	9	2826
$\rho = 0.8$	32 531.3	467.34	31740	33265	2.49	1.47	0.00	4.80	1228.1	754.77	7	2763
$m = 5$	31 811.3	177.13	31740	32393	0.22	0.56	0.00	2.06	733.8	530.94	356	2918
$m = 10$	31 740.4	0.49	31740	31741	0.00	0.00	0.00	0.00	574.6	304.39	337	2060
$m = 15$	31 740.3	0.47	31740	31741	0.00	0.00	0.00	0.00	448.3	82.85	308	656
$m = 20$	31 740.1	0.25	31740	31741	0.00	0.00	0.00	0.00	477.2	305.33	354	2081
Sin Reini- cialización	31 740.1	0.25	31740	31741	0.00	0.00	0.00	0.00	387.2	47.17	288	488
Solución Global	32 002.6	385.83	31740	32705	0.83	1.22	0.00	3.04	1377.5	948.39	207	2952

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.16: Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito Deportivo

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	30 381.8	338.19	30198	31356	0.61	1.12	0.00	3.83	1372.4	919.89	31	2918
$\beta = 0.1$	30 198.0	0.00	30198	30198	0.00	0.00	0.00	0.00	318.5	248.05	91	1116
$\beta = 1$	30 203.1	19.54	30198	30275	0.02	0.06	0.00	0.25	753.6	616.95	48	2320
$\beta = 5$	30 814.7	622.81	30199	31665	2.04	2.06	0.00	4.86	1534.9	954.71	72	2992
$\beta = 8$	31 485.4	403.27	30242	32369	4.26	1.34	0.15	7.19	1218.0	1011.79	38	2942
$\rho = 0.01$	30 439.2	393.83	30198	31358	0.80	1.30	0.00	3.84	1603.1	920.83	125	2968
$\rho = 0.3$	30 239.0	26.79	30198	30318	0.14	0.09	0.00	0.40	687.0	879.34	25	2948
$\rho = 0.5$	30 355.9	302.14	30198	31314	0.52	1.00	0.00	3.70	624.7	714.08	27	2491
$\rho = 0.8$	30 319.8	284.22	30198	31403	0.40	0.94	0.00	3.99	771.9	876.88	33	2741
$m = 3$	30 399.2	385.30	30198	31313	0.67	1.28	0.00	3.69	1433.4	923.15	141	2848
$m = 7$	30 407.0	373.99	30199	31348	0.69	1.24	0.00	3.81	1050.8	847.90	16	2932
$m = 14$	30 369.2	321.58	30198	31313	0.57	1.06	0.00	3.69	1280.9	857.56	4	2819
$m = 17$	30 361.7	339.09	30198	31357	0.54	1.12	0.00	3.84	1299.4	805.28	37	2834
$\psi = 0.01$	30 272.4	196.97	30198	31313	0.25	0.65	0.00	3.69	658.0	702.76	26	2417
$\psi = 0.3$	30 547.0	454.31	30198	31341	1.16	1.50	0.00	3.79	1260.2	974.82	37	2994
$\psi = 0.5$	30 990.4	443.39	30199	31669	2.62	1.47	0.00	4.87	1504.1	866.80	20	2854
$\psi = 0.8$	31 010.0	402.02	30242	31384	2.69	1.33	0.15	3.93	1433.3	913.08	97	2991
$q_0 = 0.3$	30 203.8	14.84	30198	30241	0.02	0.05	0.00	0.14	1204.9	908.30	24	2954
$q_0 = 0.5$	30 203.8	14.84	30198	30241	0.02	0.05	0.00	0.14	1546.9	856.73	31	2982
$q_0 = 0.7$	30 235.3	64.91	30198	30553	0.12	0.21	0.00	1.18	1221.4	1034.16	12	2929
$q_0 = 0.95$	30 615.2	490.51	30199	31358	1.38	1.62	0.00	3.84	1389.0	986.64	73	2963
Búsqueda Local	30 220.7	35.28	30198	30318	0.08	0.12	0.00	0.40	1123.4	854.33	126	2620

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.17: Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito Exterior

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	28 607.4	24.90	28596	28717	0.04	0.09	0.00	0.42	1260.1	884.30	2	2895
$\beta = 0.1$	28 980.7	997.48	28596	31481	1.35	3.49	0.00	10.09	779.4	358.19	148	1379
$\beta = 1$	28 596.0	0.00	28596	28596	0.00	0.00	0.00	0.00	267.0	297.43	39	1492
$\beta = 5$	28 631.1	53.32	28596	28717	0.12	0.19	0.00	0.42	723.8	771.28	1	2818
$\beta = 8$	28 709.0	30.44	28597	28717	0.40	0.11	0.00	0.42	109.2	455.09	1	2387
$\rho = 0.01$	28 596.6	0.49	28596	28597	0.00	0.00	0.00	0.00	815.2	768.12	5	2722
$\rho = 0.3$	28 637.5	50.77	28596	28717	0.15	0.18	0.00	0.42	1483.0	1089.80	1	2880
$\rho = 0.5$	28 663.5	55.76	28596	28717	0.24	0.19	0.00	0.42	770.8	991.63	1	2871
$\rho = 0.8$	28 683.1	53.20	28596	28717	0.30	0.19	0.00	0.42	570.0	898.56	1	2862
$m = 5$	28 654.0	50.54	28596	28717	0.20	0.18	0.00	0.42	1042.5	1065.91	1	2835
$m = 15$	28 602.6	13.36	28596	28632	0.02	0.05	0.00	0.13	878.5	720.91	2	2817
$m = 20$	28 598.8	9.03	28596	28632	0.01	0.03	0.00	0.13	723.5	729.59	5	2527
$m = 23$	28 597.7	6.50	28596	28632	0.01	0.02	0.00	0.13	824.8	845.07	22	2703
$\psi = 0.01$	28 684.4	47.71	28596	28717	0.31	0.17	0.00	0.42	633.5	973.09	1	2942
$\psi = 0.3$	28 596.8	0.43	28596	28597	0.00	0.00	0.00	0.00	830.9	780.32	19	2657
$\psi = 0.5$	28 596.8	0.43	28596	28597	0.00	0.00	0.00	0.00	782.1	737.91	5	2351
$\psi = 0.8$	28 596.8	0.43	28596	28597	0.00	0.00	0.00	0.00	642.7	644.35	9	2201
$q_0 = 0.3$	28 596.1	0.35	28596	28597	0.00	0.00	0.00	0.00	1773.7	819.84	277	2971
$q_0 = 0.5$	28 596.1	0.35	28596	28597	0.00	0.00	0.00	0.00	1439.1	941.52	87	2993
$q_0 = 0.7$	28 598.7	8.68	28596	28632	0.01	0.03	0.00	0.13	1677.1	747.70	162	2980
$q_0 = 0.95$	28 649.8	53.76	28596	28717	0.19	0.19	0.00	0.42	914.1	1045.68	1	2991
Búsqueda Local	28 596.0	0.00	28596	28596	0.00	0.00	0.00	0.00	555.3	502.31	32	2292

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.18: Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito de la Investigación Científica

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	30 016.8	240.08	28993	30138	3.53	0.83	0.00	3.95	574.5	847.48	5	2706
$\beta = 0.1$	28 993.0	0.00	28993	28993	0.00	0.00	0.00	0.00	735.4	511.28	50	2185
$\beta = 1$	28 993.0	0.00	28993	28993	0.00	0.00	0.00	0.00	289.8	318.38	21	1275
$\beta = 5$	30 033.4	295.60	28993	30138	3.59	1.02	0.00	3.95	361.1	774.30	2	2655
$\beta = 8$	30 138.0	0.00	30138	30138	3.95	0.00	3.95	3.95	90.1	80.30	1	317
$\rho = 0.01$	29 636.3	542.58	28993	30138	2.22	1.87	0.00	3.95	768.0	889.52	12	2896
$\rho = 0.3$	29 886.3	395.23	28993	30138	3.08	1.36	0.00	3.95	651.1	959.56	5	2932
$\rho = 0.5$	29 689.0	545.46	28993	30138	2.40	1.88	0.00	3.95	744.2	880.02	3	2888
$\rho = 0.8$	29 737.6	553.51	28993	30507	2.57	1.91	0.00	5.22	855.8	1017.16	2	2902
$m = 5$	29 973.7	314.91	28993	30138	3.38	1.09	0.00	3.95	882.5	838.24	6	2432
$m = 15$	29 641.4	451.94	28993	30138	2.24	1.56	0.00	3.95	1221.4	973.86	1	2954
$m = 20$	29 510.7	506.00	28993	30138	1.79	1.75	0.00	3.95	1120.7	1021.91	2	2865
$m = 24$	29 295.5	431.37	28993	30138	1.04	1.49	0.00	3.95	1359.8	873.13	2	2668
$\psi = 0.01$	29 825.4	498.16	28993	30696	2.87	1.72	0.00	5.87	712.7	761.70	25	2961
$\psi = 0.3$	29 557.5	539.15	28993	30138	1.95	1.86	0.00	3.95	991.9	1048.25	8	2713
$\psi = 0.5$	29 443.7	555.35	28993	30138	1.55	1.92	0.00	3.95	886.1	973.51	2	2918
$\psi = 0.8$	29 413.4	511.71	28993	30138	1.45	1.76	0.00	3.95	1267.1	1019.07	13	2992
$q_0 = 0.3$	29 001.7	32.98	28993	29123	0.03	0.11	0.00	0.45	1030.0	664.20	114	2524
$q_0 = 0.5$	29 049.9	216.67	28993	29847	0.20	0.75	0.00	2.95	855.6	563.00	123	2772
$q_0 = 0.7$	29 267.3	403.06	28993	30138	0.95	1.39	0.00	3.95	1492.7	856.72	139	2934
$q_0 = 0.95$	30 094.7	146.91	29422	30138	3.80	0.51	1.48	3.95	385.6	507.03	9	2010
Búsqueda Local	29 017.1	132.00	28993	29716	0.08	0.46	0.00	2.49	521.8	659.19	15	2791

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.19: Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito Escolar

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	33 160.9	9.73	33158	33199	0.01	0.03	0.00	0.12	590.9	590.91	61	2476
$\beta = 0.1$	33 314.8	404.11	33158	34411	0.47	1.22	0.00	3.78	1499.1	720.45	336	2985
$\beta = 1$	33 179.2	28.70	33158	33251	0.06	0.09	0.00	0.28	1225.6	827.38	171	2884
$\beta = 5$	33 159.5	7.46	33158	33199	0.00	0.02	0.00	0.12	705.2	807.92	67	2942
$\beta = 8$	33 162.3	10.93	33158	33199	0.01	0.03	0.00	0.12	682.9	823.86	16	2687
$\rho = 0.01$	33 164.2	12.41	33158	33216	0.02	0.04	0.00	0.17	660.3	748.32	56	2950
$\rho = 0.3$	33 173.8	45.74	33158	33389	0.05	0.14	0.00	0.70	787.5	772.78	23	2979
$\rho = 0.5$	33 170.2	25.70	33158	33251	0.04	0.08	0.00	0.28	805.5	725.08	33	2686
$\rho = 0.8$	33 181.2	32.72	33158	33252	0.07	0.10	0.00	0.28	677.3	618.49	42	2882
$m = 5$	33 170.3	28.10	33158	33270	0.04	0.08	0.00	0.34	1125.1	707.06	280	2878
$m = 15$	33 158.1	0.25	33158	33159	0.00	0.00	0.00	0.00	703.6	950.83	31	2957
$m = 20$	33 158.1	0.25	33158	33159	0.00	0.00	0.00	0.00	488.9	719.77	58	2963
$m = 25$	33 158.1	0.25	33158	33159	0.00	0.00	0.00	0.00	791.0	728.26	14	2482
$\psi = 0.01$	33 166.2	24.28	33158	33252	0.02	0.07	0.00	0.28	706.3	508.59	98	1956
$\psi = 0.3$	33 159.2	4.56	33158	33176	0.00	0.01	0.00	0.05	816.3	733.49	57	2551
$\psi = 0.5$	33 168.1	15.09	33158	33216	0.03	0.05	0.00	0.17	1327.1	940.88	111	2995
$\psi = 0.8$	33 171.1	15.08	33158	33216	0.04	0.05	0.00	0.17	1341.3	863.67	257	2957
$q_0 = 0.3$	33 158.3	0.55	33158	33160	0.00	0.00	0.00	0.01	1100.0	777.79	122	2844
$q_0 = 0.5$	33 158.0	0.00	33158	33158	0.00	0.00	0.00	0.00	693.5	617.70	40	2080
$q_0 = 0.7$	33 158.1	0.31	33158	33159	0.00	0.00	0.00	0.00	703.4	622.08	26	2016
$q_0 = 0.95$	33 166.8	24.44	33158	33271	0.03	0.07	0.00	0.34	652.6	414.67	84	1453
Búsqueda Local	33 158.2	0.38	33158	33159	0.00	0.00	0.00	0.00	436.2	815.63	2	294

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.20: Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado al Circuito CCU

Parámetro	Mejor solución				Error Relativo (%)				Iteración donde se encuentra a la mejor solución			
	μ	σ	mín	máx	μ	σ	mín	máx	μ	σ	mín	máx
Base	31 834.4	341.95	31740	33181	0.30	1.08	0.00	4.54	1116.8	745.83	84	2761
$\beta = 0.1$	32 145.0	417.35	31740	32860	1.28	1.31	0.00	3.53	1835.3	706.81	730	2941
$\beta = 1$	31 893.2	285.34	31740	32774	0.48	0.90	0.00	3.26	868.0	630.37	99	2737
$\beta = 5$	31 926.2	312.07	31740	32529	0.59	0.98	0.00	2.49	1023.2	864.33	6	2960
$\beta = 8$	31 964.2	320.86	31740	32677	0.71	1.01	0.00	2.95	1240.2	812.45	263	2968
$\rho = 0.01$	31 939.4	314.62	31740	32529	0.63	0.99	0.00	2.49	1414.4	936.30	22	2997
$\rho = 0.3$	32 084.6	464.70	31740	33114	1.09	1.46	0.00	4.33	1252.1	1071.35	11	2994
$\rho = 0.5$	32 313.4	552.69	31740	33171	1.81	1.74	0.00	4.51	1076.5	782.01	84	2801
$\rho = 0.8$	32 541.2	443.42	31740	33353	2.52	1.40	0.00	5.08	1362.8	1033.27	61	2973
$m = 5$	32 347.2	581.87	31740	33413	1.91	1.83	0.00	5.27	1597.0	697.17	289	2804
$m = 15$	31 767.6	150.80	31740	32566	0.09	0.48	0.00	2.60	615.8	580.40	30	2759
$m = 20$	31 745.7	17.18	31740	31797	0.02	0.05	0.00	0.18	1027.2	788.84	129	2775
$m = 28$	31 762.5	28.03	31740	31797	0.07	0.09	0.00	0.18	1065.6	935.87	117	2812
$\psi = 0.01$	32 201.7	414.23	31740	32801	1.45	1.31	0.00	3.34	981.2	668.24	158	2250
$\psi = 0.3$	31 832.9	231.74	31740	32528	0.29	0.73	0.00	2.48	1180.5	861.41	213	2917
$\psi = 0.5$	31 837.8	229.62	31740	32529	0.31	0.72	0.00	2.49	1515.4	921.49	21	2922
$\psi = 0.8$	31 812.5	182.46	31740	32472	0.23	0.57	0.00	2.31	1370.0	784.87	44	2876
$q_0 = 0.3$	31 945.8	364.86	31740	32803	0.65	1.15	0.00	3.35	1792.3	927.37	98	2936
$q_0 = 0.5$	31 950.7	314.68	31740	32565	0.66	0.99	0.00	2.60	1355.4	796.97	41	2822
$q_0 = 0.7$	31 856.8	268.50	31740	32634	0.37	0.85	0.00	2.82	959.9	742.16	122	2825
$q_0 = 0.95$	32 064.5	562.23	31740	33414	1.02	1.77	0.00	5.27	1262.8	800.45	71	2821
Búsqueda Local	31 740.0	0.18	31740	31741	0.00	0.00	0.00	0.00	324.5	447.96	17	2499

μ : media σ : desviación estándar mín: valor mínimo máx: valor máximo

Tabla C.21: Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado a la instancia P_CA_050

	Parámetro	μ	Sig.	σ	mín	máx
1	$\alpha = 0.1$	514.0		9.9	485	528
2	$\alpha = 1$	489.2	>1,5,6	13.1	465	514
3	$\alpha = 3$	495.6	>1	15.7	466	524
4	$\alpha = 5$	499.1	>1	16.6	470	529
5	$\alpha = 8$	505.0		18.0	463	540
6	$\alpha = 10$	503.3	>1	15.6	474	530
7	$\beta = 0.1$	591.1		21.1	552	642
8	$\beta = 1$	532.4		22.3	485	581
9	$\beta = 3$	489.2	>7,8	13.1	465	514
10	$\beta = 5$	472.5	>7,8,9	11.4	448	493
11	$\beta = 8$	460.4	>7,...,10	9.6	446	479
12	$\beta = 10$	457.6	>7,...,10	7.0	445	471
13	$\rho = 0.01$	489.2	>17,18	13.8	459	513
14	$\rho = 0.1$	489.3	>17,18	13.1	465	514
15	$\rho = 0.3$	498.0	>18	14.0	462	522
16	$\rho = 0.5$	500.2	>18	16.3	458	524
17	$\rho = 0.8$	510.3		15.8	481	543
18	$\rho = 0.99$	516.4		19.3	478	562
19	$m = 5$	507.1		15.9	480	539
20	$m = 10$	503.7		19.3	472	535
21	$m = 20$	494.8	>19	12.6	464	525
22	$m = 30$	494.3	>19	15.4	471	523
23	$m = 40$	488.6	>19,20	13.4	457	518
24	$m = 49$	489.2	>19,20	13.1	465	514
25	τ_0 menor	493.9		14.6	468	518
26	τ_0 base	489.2		13.1	465	514
27	τ_0 mayor	480.4	>25,26	10.0	461	499
28	LC=10	486.2		12.9	462	511
29	LC=15	489.2		13.1	465	514
30	LC=20	490.6		17.0	461	528
31	Solución Global	489.2		13.1	465	514
32	Solución Local	451.0	>31	5.0	442	460
33	Con Reincialización	489.2	>34	13.1	465	514
34	Sin Reincialización	514.2		20.2	467	546
35	Sin Heurísticas	489.2		13.1	465	514
36	MMAS+ β -opt	470.2	>35	10.91	449	497
37	MMAS+Exch.	478.1		13.4	453	514
38	MMAS+Rel.	463.8	>35,37	12.2	438	493
39	Sin Heurísticas	489.2		13.1	465	514
40	MMAS+ β -opt+ Exch.	463.1	>39	11.4	437	490
41	MMAS+ β -opt+ Rel.	456.4	>39	8.1	441	472
42	MMAS+Exch.+ Rel.	460.2	>39	9.2	445	476
43	Sin Heurísticas	489.2		13.1	465	514
44	MMAS+ β -opt+Exch.+Rel.	451.0	>43	9.5	435	472

 μ : media σ : desviación estándar mín: valor mínimo μ : media σ : desviación estándar mín: valor mínimo

Tabla C.22: Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado a la instancia P_CA_050

	Parámetro	μ	Sig.	σ	mín	máx
1	$\alpha = 0.1$	518.7	>6	8.1	495	531
2	$\alpha = 1$	501.7	>1,6	10.6	466	518
3	$\alpha = 3$	482.9	>1,2,6	10.3	463	504
4	$\alpha = 5$	481.1	>1,2,6	12.8	458	509
5	$\alpha = 8$	480.3	>1,2,6	18.7	441	526
6	$\alpha = 10$	1034.3		8.9	1014	1052
7	$\beta = 0.1$	654.2		12.5	620	675
8	$\beta = 1$	565.1	>7	26.4	518	595
9	$\beta = 3$	501.7	>7,8	10.6	466	518
10	$\beta = 5$	473.1	>7,8,9	6.6	461	489
11	$\beta = 8$	458.6	>7,...,10	5.9	449	470
12	$\beta = 10$	451.9	>7,...,11	5.4	443	470
13	$\rho = 0.01$	514.4		6.3	502	524
14	$\rho = 0.1$	501.7	>13	10.6	466	518
15	$\rho = 0.3$	491.2	>13,14	8.0	478	507
16	$\rho = 0.5$	487.9	>13,14	10.3	465	506
17	$\rho = 0.8$	481.7	>13,14,15	9.1	464	496
18	$\rho = 0.99$	480.6	>13,14,15	10.3	453	495
19	$m = 5$	506.4		13.9	481	535
20	$m = 10$	502.0		9.9	478	516
21	$m = 20$	499.6		10.7	471	516
22	$m = 30$	500.7		8.1	485	518
23	$m = 40$	504.1		7.9	487	517
24	$m = 49$	501.7		10.6	466	518
25	$\psi = 0.01$	488.43	>26,...,30	11.24	464	513
26	$\psi = 0.1$	501.67	>27,...,30	10.56	466	518
27	$\psi = 0.3$	512.30		6.52	499	526
28	$\psi = 0.5$	513.83		7.99	492	528
29	$\psi = 0.8$	516.57		8.41	490	526
30	$\psi = 0.99$	515.70		8.21	496	529

	Parámetro	μ	Sig.	σ	mín	máx
31	$q0 = 0.01$	504.6		7.6	489	518
32	$q0 = 0.1$	500.0		8.8	482	517
33	$q0 = 0.3$	501.2		15.4	458	524
34	$q0 = 0.5$	503.6		7.4	484	522
35	$q0 = 0.8$	501.7		10.6	466	518
36	$q0 = 0.99$	501.1		8.9	483	519
37	τ_0 menor	497.2	>39	10.0	473	513
38	τ_0 base	501.7	>39	10.6	466	518
39	τ_0 mayor	519.0		7.2	502	532
40	LC=10	487.0	>41	12.7	462	511
41	LC=15	501.7		10.6	466	518
42	LC=20	491.5	>41	18.0	461	528
43	Sin Heurísticas	501.7		10.6	466	518
44	ACS+3-opt	494.1	>43	5.9	480	508
45	ACS+Exch.	490.8	>43	4.8	481	499
46	ACS+Rel.	478.6	>43,44,45	6.9	451	487
47	Sin Heurísticas	501.7		10.6	466	518
48	ACS+3-opt+Exch.	480.6	>47	4.8	470	489
49	ACS+3-opt+Rel.	471.2	>47,48	5.5	455	477
50	ACS+Exch.+Rel.	467.4	>47,48	7.3	450	480
51	Sin Heurísticas	501.7		10.6	466	518
52	ACS+3-opt+Exch.+Rel.	464.1	>51	6.9	450	474

 μ : media σ : desviación estándar mín: valor mínimo μ : media σ : desviación estándar mín: valor mínimo

Tabla C.23: Estadísticas básicas de los parámetros evaluados del algoritmo MMAS aplicado a la instancia P_CA_100

Parámetro	μ	Sig.	σ	mín	máx
1 $\alpha = 0.1$	958.1		9.3	932	974
2 $\alpha = 1$	938.8	>1,4,5,6	17.2	904	969
3 $\alpha = 3$	950.6		15.2	907	974
4 $\alpha = 5$	953.8		17.8	914	977
5 $\alpha = 8$	956.8		13.1	936	982
6 $\alpha = 10$	959.6		17.7	924	992
7 $\beta = 0.1$	1079.2		24.6	1013	1122
8 $\beta = 1$	1018.6	>7	25.3	970	1072
9 $\beta = 3$	938.8	>7,8	17.2	904	969
10 $\beta = 5$	910.0	>7,8,9	12.1	887	933
11 $\beta = 8$	888.7	>7, ...,10	13.0	855	914
12 $\beta = 10$	889.1	>7, ...,10	10.9	863	907
13 $\rho = 0.01$	931.0	>16,17,18	13.3	894	959
14 $\rho = 0.1$	938.8	>16,17,18	17.2	904	969
15 $\rho = 0.3$	942.9	>16,17,18	16.8	900	967
16 $\rho = 0.5$	958.0	>18	13.3	924	979
17 $\rho = 0.8$	961.3	>18	15.4	929	994
18 $\rho = 0.99$	976.1		11.5	953	996
19 $m = 10$	964.8		12.6	939	992
20 $m = 20$	955.7		16.2	918	982
21 $m = 40$	946.6	>19	12.9	914	969
22 $m = 60$	945.8	>19	15.0	903	970
23 $m = 80$	940.8	>19	16.9	915	977
24 $m = 99$	938.8	>19,20	17.2	904	969

Parámetro	μ	Sig.	σ	mín	máx
25 τ_0 menor	938.9		12.3	911	958
26 τ_0 base	938.8		17.2	904	969
27 τ_0 mayor	929.7	>25,26	16.2	890	961
28 LC=10	917.9	>29,30	16.1	866	939
29 LC=15	938.8	>30	17.2	904	969
30 LC=20	952.5		11.1	925	974
31 Solución Global	938.8		17.2	904	969
32 Solución Local	874.6	>31	6.9	860	889
33 Con Reincialización	938.8	>34	17.2	904	969
34 Sin Reincialización	968.5		23.8	934	1011
35 Sin Heurísticas	938.8		17.2	904	969
36 MMAS+ β -opt	913.5	>35	17.3	873	948
37 MMAS+Exch.	908.3	>35	11.1	886	931
38 MMAS+Rel.	910.4	>35	13.0	886	940
39 Sin Heurísticas	938.8		17.2	904	969
40 MMAS+ β -opt+ Exch.	886.8	>39	10.9	866	911
41 MMAS+ β -opt+ Rel.	882.8	>39	12.9	862	904
42 MMAS+ Exch.+ Rel.	888.5	>39	11.2	867	911
43 Sin Heurísticas	938.8		17.2	904	969
44 MMAS+ β -opt +Exch.+Rel.	872.4	>43	10.9	853	891

μ : media σ : desviación estándar mín: valor mínimo

μ : media σ : desviación estándar mín: valor mínimo

Tabla C.24: Estadísticas básicas de los parámetros evaluados del algoritmo ACS aplicado a la instancia P_CA_100

	Parámetro	μ	Sig.	σ	mín	máx
1	$\alpha = 0.1$	960.0	>6	6.8	944	973
2	$\alpha = 1$	947.7	>1,6	11.0	924	966
3	$\alpha = 3$	941.9	>1,6	10.7	911	958
4	$\alpha = 5$	939.4	>1,6	9.4	924	953
5	$\alpha = 8$	943.3	>1,6	11.0	915	965
6	$\alpha = 10$	1707.1		8.8	1684	1722
7	$\beta = 0.1$	1153.9		14.0	1116	1176
8	$\beta = 1$	1056.8	>7	8.5	1036	1073
9	$\beta = 3$	947.7	>7,8	11.0	924	966
10	$\beta = 5$	914.0	>7,8,9	8.9	899	927
11	$\beta = 8$	894.1	>7,...,10	8.7	872	905
12	$\beta = 10$	894.3	>7,...,10	8.0	865	904
13	$\rho = 0.01$	960.8		8.0	929	972
14	$\rho = 0.1$	947.7	>13	11.0	924	966
15	$\rho = 0.3$	944.7	>13	9.7	920	961
16	$\rho = 0.5$	944.8	>13	12.6	912	966
17	$\rho = 0.8$	940.8	>13	9.7	922	957
18	$\rho = 0.99$	938.7	>13,14	11.7	918	957
19	$m = 5$	954.1		13.0	924	981
20	$m = 10$	953.2		12.0	922	977
21	$m = 20$	951.3		9.2	934	972
22	$m = 30$	950.1		9.6	933	967
23	$m = 40$	946.9		10.8	921	963
24	$m = 49$	947.7		11.0	924	966
25	$\psi = 0.01$	933.90	>26,...,30	12.04	908	955
26	$\psi = 0.1$	947.73	>27,...,30	11.04	924	966
27	$\psi = 0.3$	956.27		7.78	940	975
28	$\psi = 0.5$	957.50		7.92	940	972
29	$\psi = 0.8$	960.03		7.35	944	971
30	$\psi = 0.99$	960.23		8.48	941	969

	Parámetro	μ	Sig.	σ	mín	máx
31	$q0 = 0.01$	948.5		10.0	923	968
32	$q0 = 0.1$	948.0		9.0	929	963
33	$q0 = 0.3$	948.4		11.1	918	963
34	$q0 = 0.5$	948.6		9.8	921	963
35	$q0 = 0.8$	947.7		11.0	924	966
36	$q0 = 0.99$	950.4		9.2	920	964
37	τ_0 menor	948.0	>39	7.7	932	963
38	τ_0 base	947.7	>39	11.0	924	966
39	τ_0 mayor	959.1		9.2	932	976
40	LC=10	927.1	>41,42	10.3	909	945
41	LC=15	947.7		11.0	924	966
42	LC=20	965.3		9.8	942	980
43	Sin Heurísticas	947.7		11.0	924	966
44	ACS+3-opt	930.0	>43	8.5	912	946
45	ACS+Exch.	920.5	>43,44	8.1	898	931
46	ACS+Rel.	925.9	>43	9.9	898	946
47	Sin Heurísticas	947.7		11.0	924	966
48	ACS+ 3-opt+ Exch.	903.0	>47,49	6.3	885	914
49	ACS+ 3-opt+ Rel.	909.1	>47	8.4	895	927
50	ACS+ Exch.+ Rel.	904.1	>47	8.1	885	916
51	Sin Heurísticas	947.7		11.0	924	966
52	ACS+3-opt +Exch.+Rel.	890.5	>51	8.1	869	906

 μ : media σ : desviación estándar mín: valor mínimo μ : media σ : desviación estándar mín: valor mínimo

Bibliografía

- Archetti, C., Bianchessi, N. & Speranza, M. G. (2014). Branch-and-cut algorithms for the split delivery vehicle routing problem. *European Journal of Operational Research*.
- Arias-Rojas, J. S., Jiménez, F. J. & Montoya-Torres, J. R. (2013). Solving of school bus routing problem by Ant Colony Optimization. *Revista EIA*, 9(17), 193-208.
- Baldacci, R. & Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2), 347-380.
- Bautista, J., Fernández, E. & Pereira, J. (2008). Solving an urban waste collection problem using ant heuristics. *Computers & Operations Research*, 35(9), 3020-3033.
- Bautista, J. & Pereira, J. (2004). Ant Algorithms for Urban Waste Collection Routing. En *Ant Colony Optimization and Swarm Intelligence* (pp. 302-309). Springer.
- Beliën, J., De Boeck, L. & Van Ackere, J. (2012). Municipal Solid Waste Collection and Management Problems: A Literature Review. *Transportation Science*, 48(1), 78-102.
- Bell, K. E. & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18, 41-48.
- Bonomo, F., Durán, G., Larumbe, F. & Marenco, J. (2012). A method for optimizing waste collection using mathematical programming: a Buenos Aires case study. *Waste Management & Research*, 30(3), 311-324.
- Bullnheimer, B., Hartl, R. F. & Strauss, C. (1999). An improved ant System algorithm for the vehicle Routing Problem. *Annals of operations research*, 89, 319-328.
- Chaitin, G. J., da Costa, N. C. A. & Doria, F. A. (2012). *Gödel's way: Exploits into an undecidable world*. CRC Press.
- Chen, C. H. & Ting, C. J. (2006). An Improved Ant Colony System Algorithm For The Vehicle Routing Problem. *Journal of the Chinese Institute of Industrial Engineers*, 23(2), 115-126.
- Chen, X., Tan, Z., Yang, G. & Cheng, W. (2012). A hybrid algorithm to solve traveling salesman problem. En *Advances in Electronic Engineering, Communication and Management Vol. 1* (pp. 99-105). Springer.
- Christiansen, C. H. & Lysgaard, J. (2007). A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research Letters*, 35(6), 773-781.
- Clarke, G. & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4), 568-581.
- Dantzig, G. B. & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80-91.
- Deneubourg, J. L., Aron, S., Goss, S. & Pasteels, J. M. (1990). The self-organizing exploratory pattern of the argentine ant. *Journal of insect behavior*, 3(2), 159-168.
- Desrochers, M., Desrosiers, J. & Solomon, M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations research*, 40(2), 342-354.
- Di Caro, G. & Dorigo, M. (1998). Antnet: distributed stigmergetic control for communications networks. *Journal of Artificial Intelligence Research*, 9, 317-365.
- Di Caro, G., Ducatelle, F. & Gambardella, L. M. (2008). Theory and practice of ant colony optimization for routing in dynamic telecommunications networks.
- Dorigo, M., Birattari, M. & Stützle, T. (2006). Ant colony optimization: artificial ants as a computational intelligence technique. *IEEE Computational Intelligence Magazine*, 1, 28-39.
- Dorigo, M., Di Caro, G. & Gambardella, L. M. (1999, abril). Ant algorithms for discrete optimization. *Artif. Life*, 5(2), 137-172.
- Dorigo, M. & Gambardella, L. M. [Luca M]. (1997). Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1(1), 53-66.
- Dorigo, M., Maniezzo, V. & Colorni, A. (1991). *Ant system: an autocatalytic optimizing process*. Université Libre de Bruxelles.
- Dorigo, M., Maniezzo, V. & Colorni, A. (1996). The ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1), 29-41.
- Dorigo, M. & Stützle, T. (2004). *Ant Colony Optimization*. MIT Press, Cambridge, MA.
- Dror, M. & Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science*, 141-145.

- Enríquez, C. A. (2015). *Diseño de rutas de recolección en ciudad universitaria* (Tesis de maestría, Universidad Nacional Autónoma de México).
- Favaretto, D., Moretti, E. & Pellegrini, P. (2004). Ant colony system for variants of traveling salesman problem with time windows. *Ca Foscari University, Venice, Tech. Rep.*
- Favaretto, D., Moretti, E. & Pellegrini, P. (2007). Ant colony system for a VRP with multiple time windows and multiple visits. *Journal of Interdisciplinary Mathematics*, 10(2), 263-284.
- Filipiak, K. A., Abdel-Malek, L., Hsieh, H.-N. & Meegoda, J. N. (2009). Optimization of municipal solid waste collection system: case study. *Practice Periodical of Hazardous, Toxic, and Radioactive Waste Management*, 13(3), 210-216.
- Fuellerer, G., Doerner, K. F., Hartl, R. F. & Iori, M. (2009). Ant Colony Optimization for the two-dimensional Loading Vehicle Routing Problem. *Computers & Operations Research*, 36(3), 655-673.
- Gajpal, Y. & Abad, P. (2009). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research*, 36(12), 3215-3223.
- Gambardella, L. M. [Luca M.], Taillard, É. & Agazzi, G. (1999). MACS-VRPTW: A Multiple Colony System For Vehicle Routing Problems With Time Windows. En *New Ideas in Optimization* (pp. 63-76). McGraw-Hill.
- Gambardella, L. M. [Luca M.], Taillard, E. & Dorigo, M. (1999). Ant colonies for the quadratic assignment problem. *Journal of the operational research society*, 167-176.
- Gharehchopogh, F., Maleki, I. & Farahmandian, M. (2012). New Approach for Solving Dynamic Traveling Salesman Problem with Hybrid Genetic Algorithms and Ant Colony Optimization. *International Journal of Computer Applications (IJCA)*, 53(1), 39-44.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5), 533-549.
- Golden, B. L. & Wong, R. T. (1981). Capacitated arc routing problems. *Networks*, 11(3), 305-315.
- Hemmelmayer, V., Doerner, K. F., Hartl, R. F. & Rath, S. (2013). A heuristic solution method for node routing based solid waste collection problems. *Journal of Heuristics*, 19(2), 129-156.
- Hlaing, Z. C. S. S. & Khine, M. A. (2011). Solving traveling salesman problem by using improved ant colony optimization algorithm. *International Journal of Information and Education Technology*, 1(5), 404-409.
- Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press.
- Hornig, E. S. & Fuentealba, N. R. (2009). Modelo ACO para la recolección de residuos por contenedores ACO model applied to the waste collection by containers. *Ingeniare. Revista chilena de ingeniería*, 17(2), 236-243.
- Hromkovic, J. (2010). *Algorithmics for hard problems: introduction to combinatorial optimization, randomization, approximation, and heuristics*. Springer-Verlag.
- Ismail, Z. & Loh, S. L. (2009). Ant colony optimization for solving solid waste collection scheduling problems. *Journal of mathematics and statistics*, 5(3), 199.
- Karadimas, N. V., Doukas, N., Kolokathi, M. & Defteraiou, G. (2008). Routing Optimization Heuristic Algorithms for Urban Solid Waste Transportation Management. *WSEAS Transactions on Computers*, 7(12), 2022-2031.
- Karadimas, N. V., Kouzas, G., Anagnostopoulos, I. & Loumos, V. (2005). Urban Solid Waste Collection and Routing: the Ant Colony Strategic Approach. *International Journal of Simulation: Systems, Science & Technology*, 6(12-13), 45-53.
- Karadimas, N. V., Papatzelou, K. & Loumos, V. G. (2007). Optimal solid waste collection routes identified by the ant colony system algorithm. *Waste Management & Research*, 25(2), 139-147.
- Katiyar, V. (2013). An Enhanced Ant Colony System for Solving Vehicle Routing Problem with Time Window. *International Journal of Computer Applications*, 73(12), 27-31.
- Katkar, A. (2012). Improvement of solid waste collection by using optimization technique. *International Journal of Multidisciplinary Research*, 2(4).
- Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, 220(4598), 671-680.
- Kulcar, T. (1996). Optimizing solid waste collection in brussels. *European Journal of Operational Research*, 90(1), 71-77.

- Kulkarni, R. V. & Bhavé, P. R. (1985). Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, 20(1), 58-67.
- Kwan, M. K. (1962). Graphic programming using odd or even points. *Chinese Mathematics*, 1, 237-277.
- Laporte, G., Gendreau, M., Potvin, J.-Y. & Semet, F. (2000). Classical and modern heuristics for the vehicle routing problem. *International Transactions in Operational Research*, 7(4), 285-300.
- Lazo, P. A. (2005). *Aplicación de un Procedimiento de Ruteo para la Recolección de Contenedores de Residuos Domiciliarios en San Pedro de la Paz* (Tesis de Licenciatura, Universidad de Concepción, Chile).
- Lenstra, J. K. & Kan, A. H. G. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2), 221-227.
- Mansini, R. & Speranza, M. G. (1998). A linear programming model for the separate refuse collection service. *Computers & Operations Research*, 25(7), 659-673.
- Marković, D., Janošević, D., Jovanović, M. L. & Nikolić, V. (2010). Application method for optimization in solid waste management system in the city of Niš. *Facta universitatis-series: Mechanical Engineering*, 8(1), 63-76.
- Martello, S. & Vigo, D. (1998). Exact solution of the two-dimensional finite bin packing problem. *Management Science*, 44(3), 388-399.
- Mazzeo, S. & Loiseau, I. (2004). An ant colony algorithm for the capacitated vehicle routing. *Electronic Notes in Discrete Mathematics*, 18, 181-186.
- Min, Hokey. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research Part A: General*, 23(5), 377-386.
- Montemanni, R., Gambardella, L. M., Rizzoli, A. E. & Donati, A. V. (2005). Ant colony system for a dynamic vehicle routing problem. *Journal of Combinatorial Optimization*, 10(4), 327-343.
- Morrison, D. R., Jacobson, S. H., Sauppe, J. J. & Sewell, E. C. (2016). Branch-and-bound algorithms: a survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19, 79-102.
- Mourao, M. & Almeida, M. T. (2000). Lower-bounding and heuristic methods for a refuse collection vehicle routing problem. *European Journal of Operational Research*, 121(2), 420-434.
- Neissi, N. A. & Mazloom, M. (2009). GLS Optimization Algorithm for Solving Traveling Salesman Problem. En *Second International Conference on Computer and Electrical Engineering, (ICCEE '09)* (Vol. 1, pp. 291-294). IEEE.
- Ogwueleka, T. C. (2009). Route optimization for solid waste collection: Onitsha (Nigeria) case study. *Journal of Applied Sciences and Environmental Management*, 13(2), 37-40.
- Osman, I. H. & Laporte, G. (1996). Metaheuristics: A bibliography. *Annals of Operations Research*, 63(5), 511-623.
- Pellegrini, P., Favaretto, D. & Moretti, E. (2007). Multiple ant colony optimization for a rich vehicle routing problem: a case study. En *Knowledge-Based Intelligent Information and Engineering Systems* (pp. 627-634). Springer.
- Psaraftis, H. N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2), 130-154.
- Pullen, H. G. M. & Webb, M. H. J. (1967). A computer application to a transport scheduling problem. *The Computer Journal*, 10(1), 10-13.
- Reimann, M., Doerner, K. & Hartl, R. F. (2004). D-ants: savings based ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, 31(4), 563-591.
- Rizzoli, A. E., Oliverio, F., Montemanni, R. & Gambardella, L. M. (2004). Ant Colony Optimisation for vehicle routing problems: from theory to applications. *Galleria Rassegna Bimestrale Di Cultura*, 9(1), 1-50.
- Rodríguez, A. & Ruiz, R. (2012). A study on the effect of the asymmetry on real capacitated vehicle routing problems. *Computers & Operations Research*, 39(9), 2142-2151.
- Russell, R. & Igo, W. (1979). An assignment routing problem. *Networks*, 9(1), 1-17.
- Salari, E. & Eshghi, K. (2008). An aco algorithm for the graph coloring problem. (Vol. 3, 6, pp. 293-304). Citeseer.
- Solnon, C. (2013). *Ant Colony Optimization and Constraint Programming*. John Wiley & Sons.

- Sörensen, K. (2013). Metaheuristics—the metaphor exposed. *International Transactions in Operational Research*.
- Stützle, T. & Dorigo, M. (1999). Aco algorithms for the traveling salesman problem. *Evolutionary Algorithms in Engineering and Computer Science*, 163-183.
- Stützle, T. & Hoos, H. H. (2000). MAX-MIN Ant System. *Future Generation Computer Systems*, 16(8), 889-914.
- Taha, H. A. (2004). *Investigación de operaciones, 7a. edición*. PEARSON EDUCACIÓN.
- Talbi, E. G. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.
- Tan, X., Zhuo, X. & Zhang, J. (2006). Ant colony system for optimizing vehicle routing problem with time windows (VRPTW). En *Computational Intelligence and Bioinformatics* (pp. 33-38). Springer.
- Tillman, F. A. (1969). The multiple terminal delivery problem with probabilistic demands. *Transportation Science*, 3(3), 192-204.
- Toth, P. & Vigo, D. (Eds.). (2001). *The Vehicle Routing Problem*. Philadelphia, PA, USA: Society for Industrial y Applied Mathematics.
- Vance, P. H. (1998). Branch-and-price algorithms for the one-dimensional cutting stock problem. *Computational Optimization and Applications*, 9(3), 211-228.
- Venkata Narasimha, K., Kivelevitch, E., Sharma, B. & Kumar, M. (2013). An ant colony optimization technique for solving min-max Multi-Depot Vehicle Routing Problem. *Swarm and Evolutionary Computation*, 13, 63-73.
- Wang, Y. P. (2013). Adaptive Ant Colony Algorithm for the VRP Solution of Logistics Distribution. *Research Journal of Applied Sciences, Engineering and Technology*, 6(5), 807-811.
- Wilson, N. H. M. & Colvin, N. J. (1977). *Computer control of the rochester dial-a-ride system*. Department of Civil Engineering, Massachusetts Institute of Technology. Center for Transportation Studies.
- Yang, L., Lin, K., Lin, S., Gao, X. & Ye, S. (2013). Application of Intelligence Algorithm on TSP. *International Conference on Advanced Computer Science and Electronics Information(ICACSEI 2013)*, 41, 206-209.
- Yoshikawa, M. & Otani, K. (2010). Ant Colony Optimization Routing Algorithm with Tabu Search. En *Proceedings of the International MultiConference of Engineers and Computer Scientists (IMECS)* (Vol. 3, pp. 17-19).
- Yu, B., Yang, Z. Z. & Yao, B. Z. (2009). An improved ant colony optimization for vehicle routing problem. *European Journal of Operational Research*, 196(1), 171-176.