

**ENRUTAMIENTO MULTICAST MULTIOBJETIVO
BASADO EN COLONIA DE HORMIGAS**

Diego Pedro Pinto Roa

PROYECTO FINAL DE TESIS

**Orientador:
Prof. Benjamín Barán, D.Sc.**

**Ingeniería Electrónica
Facultad de Ciencias y Tecnología
Universidad Católica
“Nuestra Señora de la Asunción”**

**Asunción – Paraguay
Julio del 2005**

*Este trabajo está dedicado a mis padres, mi esposa y mi hijo.
Espero poder retribuirles en alguna medida su esfuerzo, su apoyo,
su comprensión y sobre todo el amor que he recibido de ellos en
todo momento.*

Agradecimientos

Le agradezco a nuestro Dios todo poderoso, por todas las bendiciones recibidas a lo largo de esta vida.

Al profesor D.Sc. Ing. Benjamín Barán, modelo de profesional y por sobre todo, excelente persona. Con su constante apoyo y orientación este proyecto ha llegado a su culminación. Su excepcional capacidad de superación frente a los problemas e inigualable visión positiva han inyectado vida y dinamismo al quehacer diario de las personas que tenemos la gran suerte de pertenecer a su grupo de investigación.

A la profesora Lic. Blanca de Trevisán y al profesor Lic. Jorge Cabañas, Director del Centro Nacional de Computación, por haber hecho posible que en dicha institución puedan ser llevados a cabo trabajos de investigación de esta naturaleza.

A los queridos compañeros del Centro Nacional de Computación, Ing. Jorge Crichigno, Lic. Joel Prieto, Lic. Ramón González, Pedro Gardel, M.Sc. Osvaldo Gómez, M.Sc. Hugo Estigarribia, Ing. Francisco Talavera, Ing. José Fernández y Marcos Villagra. Deseo agradecerles especialmente por su constante apoyo y colaboración en las diferentes instancias del desarrollo del presente trabajo, así también por el excelente ambiente de trabajo del que me han permitido ser parte.

Finalmente una mención especial a todos mis profesores, compañeros y amigos de la Facultad de Ciencias y Tecnología, con quienes he compartido estos años de formación. Mi deseo es que el conocimiento y la experiencia que hemos adquirido a través de nuestro paso por esta facultad nos sirvan para forjarnos una vida digna, honesta y plena, al servicio de nuestras familias y de nuestro país.

Índice de Contenidos

Índice de Contenidos	i
Índice de Figuras	iv
Índice de Tablas	vi
Lista de Acrónimos	xi
Lista de Símbolos	xii
1. Enrutamiento Multicast	1
1.1. Introducción	1
1.2. Ingeniería de Tráfico	4
1.3. Multicast.	5
1.4. Métricas de Optimización Multicast	6
1.4.1. Utilización Máxima de los Enlaces	6
1.4.2. Costo del Árbol Multicast	9
1.4.3. Retardo Medio y Retardo Máximo de Extremo a Extremo	10
1.5. Trabajos Relacionados	11
1.6. Enfoque Multiobjetivo.	15
1.7. Organización del presente trabajo	16
2. Optimización con Múltiples Objetivos	18
2.1. Introducción	18
2.2. Principios Básicos y Terminología	18
2.3. Búsqueda y Toma de Decisiones.	21
2.4. El Método de la Suma con Pesos	22
2.5. El Método del Orden Lexicográfico	24
2.6. Computación Evolutiva.	24
2.7. <i>Multiobjective Multicast Algorithm</i>	27
2.8. Optimización basada en Colonia de Hormigas	29
2.9. Resumen del Capítulo	30
3. Formulación Matemática.	31
3.1. Introducción	31
3.2. Modelo Matemático	31
3.3. Ejemplos de Problemas Multicast	34
3.4. Resumen del Capítulo	41

4. Colonia de Hormigas	42
4.1. Introducción	42
4.2. Comportamiento de Búsqueda de Alimentos de las Hormigas Reales	42
4.3. Optimización por Colonia de Hormigas	44
4.4. Enfoque Estándar de ACO	46
4.5. Enfoques Propuestos para Ingeniería de Tráfico Multicast	47
4.5.1. <i>Multiobjective Max-Min Ant System</i>	48
4.5.2. <i>Multiobjective Ómicron ACO</i>	51
4.5.3. <i>Multiobjective Ant Colony System</i>	53
4.6. Construcción basada en ACO del Árbol Multicast	55
4.7. Resumen del Capítulo	57
5. Pruebas Experimentales	58
5.1. Introducción	58
5.2. Ambiente Experimental	58
5.3. Procedimiento de Comparación	60
5.3.1. Obtención del Frente Pareto Aproximado	61
5.3.2. Métodos de Comparación	62
5.4. Pruebas Estáticas	64
5.4.1. Resultados para el Grupo Multicast 1 – Prueba Estática	65
5.4.2. Resultados para el Grupo Multicast 2 – Prueba Estática	65
5.4.3. Resultados para el Grupo Multicast 3 – Prueba Estática	66
5.4.4. Resultados para el Grupo Multicast 4 – Prueba Estática	67
5.4.5. Promedios Generales de las Pruebas Estáticas	67
5.5. Pruebas Semi-Dinámicas	68
5.5.1. Pruebas con Baja Carga	69
5.5.1.1. Resultados para el Grupo Multicast 1 – Baja Carga	69
5.5.1.2. Resultados para el Grupo Multicast 2 – Baja Carga	70
5.5.1.3. Resultados para el Grupo Multicast 3 – Baja Carga	70
5.5.1.4. Resultados para el Grupo Multicast 4 – Baja Carga	71
5.5.1.5. Promedios Generales – Baja Carga	72
5.5.2. Pruebas con Media Carga	72
5.5.2.1. Resultados para el Grupo Multicast 1 – Media Carga	72
5.5.2.2. Resultados para el Grupo Multicast 2 – Media Carga	73

5.5.2.3.	Resultados para el Grupo Multicast 3 – Media Carga.	73
5.5.2.4.	Resultados para el Grupo Multicast 4 – Media Carga.	74
5.5.2.5.	Promedios Generales – Media Carga.	75
5.5.3.	Pruebas en Saturación.	75
5.5.3.1.	Resultados para el Grupo Multicast 1 – Saturación. . .	75
5.5.3.2.	Resultados para el Grupo Multicast 2 – Saturación. . .	76
5.5.3.3.	Resultados para el Grupo Multicast 3 – Saturación. . .	77
5.5.3.4.	Resultados para el Grupo Multicast 4 – Saturación. . .	77
5.5.3.5.	Promedios Generales – Saturación.	78
5.5.4.	Promedios Globales de las Pruebas Semi-Dinámicas.	79
5.6.	Resumen del Capítulo	80
6.	Conclusiones Generales.	81
6.1.	Conclusiones Finales.	81
6.2.	Trabajos Futuros.	82
Anexo A. Tablas de Pruebas Estáticas.		84
Anexo B. Tablas de Pruebas Semi-Dinámicas.		90
Bibliografía.		106

Índice de Figuras

1.1	Dominio MPLS	3
1.2	Ejemplo de utilización de broadcast, unicast y un árbol de expansión para realizar multicast.	6
1.3	Ejemplo 1.1	7
1.4	Ejemplo 1.2	8
1.5	Árbol multicast de costo mínimo	9
1.6	Árbol multicast óptimo cuando el considerado el retardo máximo ..	10
2.1	Componentes de un algoritmo general de búsqueda estocástica. ...	25
2.2	Contorno de un algoritmo evolutivo general	27
2.3	Procedimiento general del MMA y relación entre Cromosoma, Genes y Tablas de Ruteo.	28
3.1	Problema ejemplo red NSF	34
3.2	Soluciones alternativas para el grupo multicast del Ejemplo 3.1.	35, 36
3.3	Red NSF para el Ejemplo 3.2 con nodo fuente 5 y destinos 0, 2, 6 y 13.	37
3.4	Soluciones alternativas para el grupo multicast del Ejemplo 3.2.	39
4.1	Comportamiento de las hormigas en la búsqueda de su alimento entre el nido y su fuente de alimentación.	44

4.2	Procedimiento general de un algoritmo ACO estándar.	47
4.3	Procedimiento general del Multiobjective MMAS (M3AS).	50
4.4	Procedimiento general del Multiobjective Ómicron ACO.	53
4.5	Procedimiento General del MOACS	54
4.6	Regla Pseudo-aleatoria. Procedimiento de selección de un nodo j del conjunto vecindario N_j	55
4.7	Procedimiento General de <i>Construir Árbol</i>	56
5.1.	Red de la Nipon Telegraph and Telephone, Co.	58

Índice de Tablas

3.1	Funciones Objetivos Calculados para el Ejemplo 3.2	38
3.2	Conjunto Óptimo Pareto del Ejemplo 3.2.	40
3.3	Vectores Objetivos ($\alpha_T, C_T, DA_T, DM_T$) del Ejemplo 3.2	40
5.1	Grupos Multicast utilizados en todas las pruebas	60
5.2	Parámetros de entradas utilizados en los diferentes algoritmos ACOs.	60
5.3	Pruebas Estáticas. Total de soluciones pertenecientes a Y_{apr} para cada grupo multicast.	61
5.4	Pruebas Semi-Dinámicas. Total de soluciones pertenecientes a Y_{apr} para cada grupo multicast (carga baja).	61
5.5	Pruebas Semi-Dinámicas. Total de soluciones pertenecientes a Y_{apr} para cada grupo multicast (carga media).	62
5.6	Pruebas Semi-Dinámicas. Total de soluciones pertenecientes a Y_{apr} para cada grupo multicast (carga alta).	62
5.7	Grupo Multicast 1 – Tiempo de corrida = 40 seg. – Pruebas Estáticas.	62
5.8	Grupo Multicast 1 – Promedios Parciales – Pruebas Estáticas. . .	65
5.9	Grupo Multicast 2 – Promedios Parciales – Pruebas Estáticas. . .	66
5.10	Grupo Multicast 3 – Promedios Parciales – Pruebas Estáticas. . .	67
5.11	Grupo Multicast 4 – Promedios Parciales – Pruebas Estáticas. . .	67
5.12	Promedios Generales de las Pruebas Estáticas.	68
5.13	Grupo Multicast 1 - Promedios Parciales a Baja Carga.	69
5.14	Grupo Multicast 2 - Promedios Parciales a Baja Carga.	70
5.15	Grupo Multicast 3 - Promedios Parciales a Baja Carga.	71
5.16	Grupo Multicast 4 - Promedios Parciales a Baja Carga.	71
5.17	Promedios Generales a Baja Carga.	72
5.18	Grupo Multicast 1 - Promedios Parciales a Media Carga.	73
5.19	Grupo Multicast 2 - Promedios Parciales a Media Carga.	73
5.20	Grupo Multicast 3 - Promedios Parciales a Media Carga.	74

5.21	Grupo Multicast 4 - Promedios Parciales a Media Carga	74
5.22	Promedios Generales a Media Carga	75
5.23	Grupo Multicast 1 - Promedios Parciales en Saturación	76
5.24	Grupo Multicast 2 - Promedios Parciales en Saturación	76
5.25	Grupo Multicast 3 - Promedios Parciales en Saturación	77
5.26	Grupo Multicast 4 - Promedios Parciales en Saturación	78
5.27	Promedios Generales en Saturación	78
5.28	Promedios Globales de las Pruebas Semi-Dinámicas.	79
A.1	Grupo Multicast 1 - Tiempo de corrida = 40 seg. – Pruebas Estáticas	85
A.2	Grupo Multicast 1 - Tiempo de corrida = 80 seg. – Pruebas Estáticas	85
A.3	Grupo Multicast 1 - Tiempo de corrida = 120 seg. – Pruebas Estáticas	85
A.4	Grupo Multicast 1 - Tiempo de corrida = 160 seg. – Pruebas Estáticas	85
A.5	Grupo Multicast 1 – Promedios Parciales – Pruebas Estáticas . .	85
A.6	Grupo Multicast 2 - Tiempo de corrida = 40 seg. – Pruebas Estáticas	86
A.7	Grupo Multicast 2 - Tiempo de corrida = 80 seg. – Pruebas Estáticas	86
A.8	Grupo Multicast 2 - Tiempo de corrida = 120 seg. – Pruebas Estáticas	86
A.9	Grupo Multicast 2 - Tiempo de corrida = 160 seg. – Pruebas Estáticas	86
A.10	Grupo Multicast 2 – Promedios Parciales – Pruebas Estáticas . .	86
A.11	Grupo Multicast 3 - Tiempo de corrida = 40 seg. – Pruebas Estáticas	87
A.12	Grupo Multicast 3 - Tiempo de corrida = 80 seg. – Pruebas Estáticas	87
A.13	Grupo Multicast 3 - Tiempo de corrida = 120 seg. – Pruebas Estáticas	87

A.14	Grupo Multicast 3 - Tiempo de corrida = 160 seg. – Pruebas Estáticas	87
A.15	Grupo Multicast 3 – Promedios Parciales – Pruebas Estáticas	87
A.16	Grupo Multicast 4 - Tiempo de corrida = 40 seg. – Pruebas Estáticas	88
A.17	Grupo Multicast 4 - Tiempo de corrida = 80 seg. – Pruebas Estáticas	88
A.18	Grupo Multicast 4 - Tiempo de corrida = 120 seg. – Pruebas Estáticas	88
A.19	Grupo Multicast 4 - Tiempo de corrida = 160 seg. – Pruebas Estáticas	88
A.20	Grupo Multicast 4 – Promedios Parciales – Pruebas Estáticas	88
A.21	Promedios Generales de las Pruebas Estáticas.	89
B1.1	Grupo Multicast 1 - Tiempo de corrida = 40 seg. – Baja Carga	91
B1.2	Grupo Multicast 1 - Tiempo de corrida = 80 seg. – Baja Carga	91
B1.3	Grupo Multicast 1 - Tiempo de corrida = 120 seg. – Baja Carga	91
B1.4	Grupo Multicast 1 - Tiempo de corrida = 160 seg. – Baja Carga	92
B1.5	Grupo Multicast 1 - Promedios Parciales a Baja Carga	92
B1.6	Grupo Multicast 2 - Tiempo de corrida = 40 seg. – Baja Carga	92
B1.7	Grupo Multicast 2 - Tiempo de corrida = 80 seg. – Baja Carga	92
B1.8	Grupo Multicast 2 - Tiempo de corrida = 120 seg. – Baja Carga	92
B1.9	Grupo Multicast 2 - Tiempo de corrida = 160 seg. – Baja Carga	93
B1.10	Grupo Multicast 2 - Promedios Parciales a Baja Carga	93
B1.11	Grupo Multicast 3 - Tiempo de corrida = 40 seg. – Baja Carga.	93
B1.12	Grupo Multicast 3 - Tiempo de corrida = 80 seg. – Baja Carga.	93
B1.13	Grupo Multicast 3 - Tiempo de corrida = 120 seg. – Baja Carga.	93
B1.14	Grupo Multicast 3 - Tiempo de corrida = 160 seg. – Baja Carga.	93
B1.15	Grupo Multicast 3 - Promedios Parciales a Baja Carga.	94
B1.16	Grupo Multicast 4 - Tiempo de corrida = 40 seg. – Baja Carga.	94
B1.17	Grupo Multicast 4 - Tiempo de corrida = 80 seg. – Baja Carga.	94
B1.18	Grupo Multicast 4 - Tiempo de corrida = 120 seg. – Baja Carga.	94
B1.19	Grupo Multicast 4 - Tiempo de corrida = 160 seg. – Baja Carga.	95

B1.20	Grupo Multicast 4 - Promedios Parciales a Baja Carga.	95
B1.21	Promedios Generales a Baja Carga.	95
B2.1	Grupo Multicast 1 - Tiempo de corrida = 40 seg. – Media Carga. . .	95
B2.2	Grupo Multicast 1 - Tiempo de corrida = 80 seg. – Media Carga. . .	96
B2.3	Grupo Multicast 1 - Tiempo de corrida = 120 seg. – Media Carga. .	96
B2.4	Grupo Multicast 1 - Tiempo de corrida = 160 seg. – Media Carga. .	96
B2.5	Grupo Multicast 1 - Promedios Parciales a Media Carga.	96
B2.6	Grupo Multicast 2 - Tiempo de corrida = 40 seg. – Media Carga. . .	96
B2.7	Grupo Multicast 2 - Tiempo de corrida = 80 seg. – Media Carga. . .	97
B2.8	Grupo Multicast 2 - Tiempo de corrida = 120 seg. – Media Carga. .	97
B2.9	Grupo Multicast 2 - Tiempo de corrida = 160 seg. – Media Carga. .	97
B2.10	Grupo Multicast 2 - Promedios Parciales a Media Carga.	97
B2.11	Grupo Multicast 3 - Tiempo de corrida = 40 seg. – Media Carga. . .	97
B2.12	Grupo Multicast 3 - Tiempo de corrida = 80 seg. – Media Carga. . .	98
B2.13	Grupo Multicast 3 - Tiempo de corrida = 120 seg. – Media Carga. .	98
B2.14	Grupo Multicast 3 - Tiempo de corrida = 160 seg. – Media Carga. .	98
B2.15	Grupo Multicast 3 - Promedios Parciales a Media Carga.	98
B2.16	Grupo Multicast 4 - Tiempo de corrida = 40 seg. – Media Carga. . .	98
B2.17	Grupo Multicast 4 - Tiempo de corrida = 80 seg. – Media Carga. . .	99
B2.18	Grupo Multicast 4 - Tiempo de corrida = 120 seg. – Media Carga. .	99
B2.19	Grupo Multicast 4 - Tiempo de corrida = 160 seg. – Media Carga. .	99
B2.20	Grupo Multicast 4 - Promedios Parciales a Media Carga.	99
B2.21	Promedios Generales a Media Carga.	99
B3.1	Grupo Multicast 1 - Tiempo de corrida = 40 seg. – Saturación.	100
B3.2	Grupo Multicast 1 - Tiempo de corrida = 80 seg. – Saturación.	100
B3.3	Grupo Multicast 1 - Tiempo de corrida = 120 seg. – Saturación. . . .	100
B3.4	Grupo Multicast 1 - Tiempo de corrida = 160 seg. – Saturación. . . .	100
B3.5	Grupo Multicast 1 - Promedios Parciales en Saturación.	100
B3.6	Grupo Multicast 2 - Tiempo de corrida = 40 seg. – Saturación.	101
B3.7	Grupo Multicast 2 - Tiempo de corrida = 80 seg. – Saturación.	101
B3.8	Grupo Multicast 2 - Tiempo de corrida = 120 seg. – Saturación. . . .	101
B3.9	Grupo Multicast 2 - Tiempo de corrida = 160 seg. – Saturación. . . .	101

B3.10	Grupo Multicast 2 - Promedios Parciales en Saturación.	101
B3.11	Grupo Multicast 3 - Tiempo de corrida = 40 seg. – Saturación.	102
B3.12	Grupo Multicast 3 - Tiempo de corrida = 80 seg. – Saturación.	102
B3.13	Grupo Multicast 3 - Tiempo de corrida = 120 seg. – Saturación.	102
B3.14	Grupo Multicast 3 - Tiempo de corrida = 160 seg. – Saturación.	102
B3.15	Grupo Multicast 3 - Promedios Parciales en Saturación.	102
B3.16	Grupo Multicast 4 - Tiempo de corrida = 40 seg. – Saturación.	103
B3.17	Grupo Multicast 4 - Tiempo de corrida = 80 seg. – Saturación.	103
B3.18	Grupo Multicast 4 - Tiempo de corrida = 120 seg. – Saturación.	103
B3.19	Grupo Multicast 4 - Tiempo de corrida = 160 seg. – Saturación.	103
B3.20	Grupo Multicast 4 - Promedios Parciales en Saturación.	103
B3.21	Promedios Generales en Saturación.	104
B4	Promedios Globales de las Pruebas Semi-Dinámicas.	104

Lista de Acrónimos

ARPANET	Advanced Research Projects Administration Network
QoS	Quality of Service – Calidad de Servicio
MPLS	MultiProtocol Label Switching
LSR	Label Switch Routers
FEC	Forwarding Equivalent Class
AS	Autonomous System
ISP	Internet Service Provider
DiffServ	Differentiated Services
SLA	Service Level Agreement
VoD	Video on-demand
RSVP-TE	Resource Reservation Protocol – Traffic Engineering
SOP	Single objective Optimization Problem
MOP	Multiobjective Optimization Problem
ACO	Ant Colony Optimization
SPT	Shortest Path Tree
OSPF	Open Shortest Path First
KMB	Algoritmo propuesto por Kou, Markowsky y Berman
KPP	Algoritmo propuesto por Kompella, Pasquale y Polyzos
MMA	Multiobjective Multicast Algorithm
MMA2	Multiobjective Multicast Algorithm - 2
SPEA	Strength Pareto Evolutionary Algorithm
TSP	Traveling Salesman Problem
NSF	National Science Foundation
MMAS	Max-Min Ant System
MOACS	Multiobjective Ant Colony System
OA	Ómicron ACO
M3AS	Multiobjective Max-Min Ant System
MOA	Multiobjective Ómicron ACO
NTT	Nipon Telegraph and Telephone, Co.

Lista de Símbolos

T	Árbol multicast
T^*	Mejor árbol multicast en OA
T_{worst}	Peor árbol multicast en OA
T'	Conjunto de árboles en el estándar ACO y OA
T_k	Árbol multicast k-ésimo en construcción en el estándar ACO y OA
α_p	Utilización máxima de los enlaces de un camino
α_T	Utilización máxima de los enlaces de un árbol
C_T	El costo del árbol T
DA_T	Retardo Medio del árbol T
DM_T	Retardo Máximo de extremo a extremo del árbol T
G	Grafo que representa el modelo de una red de computadoras
G'	Grafo que representa el modelo de una red de computadoras
V	Conjunto de vértices de un grafo
E	Conjunto de enlaces de un grafo
$G(V,E)$	Grafo dirigido representado el modelo de una red de computadoras
h	índice de conjunto de parámetros en un problema MOP
k	Índice de conjunto de funciones objetivos en un problema MOP
m	Índice de conjunto de restricciones en un problema MOP
x	Variable de decisión en un problema MOP
y	Vector objetivo en un problema MOP
$f(x)$	Función objetivo en un problema MOP
$e(x)$	Conjunto de restricciones en un problema MOP
x_h	Elemento h-ésima de la variable x de decisión en un problema MOP
y_k	Elemento k-ésima del vector objetivo y en un problema MOP
$f_k(x)$	Elemento k-ésima del vector objetivo $f(x)$ en un problema MOP
$e_m(x)$	Elemento m-ésima del conjunto de restricciones $e(x)$ en un problema MOP
X	Espacio de decisión en MOP
Y	Espacio de objetivo en MOP
X_f	Espacio de soluciones factibles en MOP
Y_f	Conjunto de vectores factibles en MOP

u, v	Soluciones pertenecientes al espacio factible X_f
Q	Conjunto de soluciones pertenecientes a X_f
X_{true}	Conjunto Pareto óptimo en MOP
Y_{true}	Frente Pareto óptimo en MOP
Y_{know}	Frente Pareto conocido en MOP
F	Función objetivo en la optimización multiobjetivo por sumas de pesos
w_j	Peso usado para la j -ésima función objetivo en la optimización multiobjetivo por suma ponderada
P	Población de individuos en el algoritmo MMA
P_{nd}	Población de individuos no-dominados en el algoritmo MMA
g_i	i -ésima gene de un cromosoma en el algoritmo MMA
R	Conjunto de Rutas en la Tabla de Rutas del algoritmo MMA
s	Nodo fuente de un grupo multicast
N_r	Conjunto de nodos destinos de un grupo multicast
n_i	Nodo destino perteneciente a N_r
(i, j)	Enlace entre el nodo i y el nodo j
z_{ij}	Capacidad del enlace (i, j) , en Mbps
t_{ij}	Tráfico actual del enlace (i, j) , en Mbps
c_{ij}	Costo del enlace (i, j)
d_{ij}	Retardo del enlace (i, j) , en milisegundos
ϕ	Demanda de tráfico multicast, en bps
$T(s, N_r)$	Árbol multicast con nodo fuente en s y conjunto destino N_r
$p_T(s, n_i)$	Camino que conecta el nodo fuente s y el destino n_i
$d(p_T(s, n_i))$	Retardo del camino $p_T(s, n_i)$
S_x	Solución x -ésima del Conjunto Pareto óptimo
η_{ij}	Visibilidad para el enlace (i, j)
η_{cij}	Visibilidad por costo del enlace (i, j)
η_{dij}	Visibilidad por retardo del enlace (i, j)
η_{tij}	Visibilidad por tráfico del enlace (i, j)
λ^c	Variable de influencia relativa para la visibilidad por costo
λ^d	Variable de influencia relativa para la visibilidad por retardo
λ^t	Variable de influencia relativa para la visibilidad por tráfico
α	Parámetro de influencia relativa de los niveles de feromonas en las probabilidades de selección de un enlace

β	Parámetro de influencia relativa de los niveles de visibilidad en las probabilidades de selección de un enlace
τ	Matriz de feromonas
τ_{ij}	Nivel de feromonas para el enlace (i, j)
τ^0	Matriz de feromonas inicial en OA
τ_0	Nivel de feromonas inicial
τ_{min}	Nivel mínimo impuesto en los niveles de feromonas de los enlaces
τ_{max}	Nivel máximo impuesto en los niveles de feromonas de los enlaces
$\Delta\tau$	Diferencial τ utilizado en la actualización de los niveles de feromonas en los ACOs mono-objetivos
$\Delta\tau^k$	Diferencial τ de la k -ésima solución en la actualización de los niveles de feromonas en los ACOs multiobjetivo
ρ	Parámetro de evaporación de feromonas en los enlaces
θ	Número de hormigas utilizado en el enfoque estándar de ACO
N_i	Conjunto de nodos vecinos al nodo i
N	Conjunto de nodos de partida en la construcción de un árbol multicast
D_r	Conjunto de nodos destinos ya alcanzados durante la construcción de un árbol multicast
p_{ij}	Probabilidad de selección del enlace (i, j)
k	k -ésima hormiga en una iteración del estándar ACO
ω	Número de hormigas utilizados en los enfoques ACO multiobjetivo
a	Número de individuos en la población en el enfoque OA y número de frentes de soluciones en el enfoque MOA
q	Probabilidad de selección por explotación en la regla pseudo-aleatoria
q_0	Variable de comparación para la selección por explotación o exploración en la regla pseudo-aleatoria
Y_{SP}	Súper-población generada por la unión de varias poblaciones
Y_{M3AS}	Frente Pareto Obtenido con el M3AS
Y_{MOA}	Frente Pareto Obtenido con el MOA
Y_{MOACS}	Frente Pareto Obtenido con el MOACS
Y_{MMA}	Frente Pareto Obtenido con el MMA
Y'	Conjunto de soluciones generadas por la unión de Y_{M3AS} , Y_{MOA} , Y_{MOACS} y Y_{MMA}
Y_{apr}	Frente Pareto aproximado obtenido a partir de Y'

$\in Y_{apr}$	Promedio del número de soluciones que están en Y_{apr}
$Y_{apr} >$	Promedio del número de soluciones que son dominadas por Y_{apr}
$ Y_{apr} $	Promedio del número de soluciones halladas por un algoritmo
$\%Y_{apr}$	Porcentaje de soluciones halladas por un algoritmo dado, i.e. $100 \cdot (\in Y_{apr}) / (Y_{apr})$

Capítulo 1

Enrutamiento Multicast

1.1 Introducción

La fusión de las computadoras y las comunicaciones ha tenido una profunda influencia en la forma en que los sistemas de cómputo se organizan. El concepto de “centro de computo” como cuarto con una gran computadora a la cual los usuarios traían sus trabajos para procesar, es ahora totalmente obsoleto. El viejo modelo de una sola computadora que atendía todas las necesidades de computación de la organización ha sido reemplazado por uno en el cual un gran número de computadoras separadas pero interconectadas entre si, hacen el trabajo. Estos sistemas se llaman **redes de computadoras** [TAN03].

La Internet tiene sus orígenes en la ARPANET (*Advanced Research Projects Administration Network*), que era una red experimental de datos financiada por los Estados Unidos en los comienzos de la década de los 60. El objetivo principal del gobierno de los Estados Unidos era la creación de una red robusta con capacidad de mantenerse funcionando aún cuando parte de la misma esté sujeta a fallas. Con esta idea, la ARPANET fue construida basándose en un modelo de *datagramas*, en el cual cada paquete de datos es enviado en forma independiente a su destino. La red de *datagramas*, además de estar basada en una idea simple, tiene la habilidad de adaptarse automáticamente a cambios en la topología [TAN03].

Durante muchos años, Internet ha sido empleada por investigadores para realizar intercambio de información. Las aplicaciones más populares eran el acceso remoto, la transferencia de archivos y el correo electrónico (*e-mail*). Para este tipo de aplicaciones, el modelo de *datagramas* funciona muy bien. No obstante, con el tiempo se han creado un conjunto de nuevas aplicaciones con diferentes requerimientos de calidad de servicio (*Quality of Service – QoS*), distintos a aquellos para los cuales Internet fue creada, como ser un requerimiento de ancho de banda mínimo garantizado para que ciertas aplicaciones funcionen bien. El modelo de *datagramas* en el cual está basado Internet tiene muy poca capacidad de manejo de tráfico de datos dentro de la red y consecuentemente no puede garantizar una capacidad de ancho de banda mínima en *bps*

a los usuarios. Cuando una aplicación intenta alcanzar un sitio Web o hacer una llamada a través de Internet, algunas partes de la red pueden estar tan congestionadas que los paquetes no consiguen llegar a destino y simplemente son desechados. Por lo tanto, las aplicaciones no pueden funcionar en forma adecuada [WAN01].

Un problema que aparece en las redes de computadoras es el retardo que sufren los paquetes hasta alcanzar al destino. Cuando las aplicaciones como transmisiones de radio o TV tienen un retardo que sobrepasa ciertos niveles, estas carecen de utilidad. Entonces, para que las redes de computadoras actuales sean capaces de manejar tráfico de datos considerando parámetros de QoS, no sólo deben ser capaces de garantizar una cantidad mínima de ancho de banda a las aplicaciones, sino que deben proveer el servicio teniendo en cuenta el retardo a los nodos destinos [WAN01].

Las redes basadas en el modelo de datagramas históricamente han ofrecido un servicio conocido como servicio del mejor esfuerzo (*best-effort*). *Best-effort* simplemente representa el servicio más simple que una red puede proveer; no hay ninguna forma de asignación de recursos en la red. Este servicio consiste en que todos los paquetes son tratados de igual forma, sin niveles de servicio, requerimientos, reservas o garantías. Este servicio solo ofrece hacer el mejor esfuerzo para lograr la comunicación. Cuando un enlace está congestionado, los paquetes simplemente son descartados. Los objetivos principales de este modelo fueron la escalabilidad y el mantenimiento de la conexión bajo fallas. Los algoritmos encargados de seleccionar un camino (*enrutar*) entre dos nodos de la red, simplemente seleccionan el camino más corto considerando una métrica simple como número de saltos o retardo. Claramente este enfoque no es adecuado para soportar asignación de recursos como ser una capacidad mínima garantizada. Por ejemplo, para llevar a cabo una reserva de recursos, es necesario encontrar un camino desde el origen al destino con cierta cantidad de ancho de banda a lo largo del camino. Pero el protocolo *Internet Protocol* (IP) utilizado en las redes de datagramas no cuenta con dicha información. Por lo tanto, el uso de dichos algoritmos puede conllevar a una alta tasa de rechazos de solicitudes de conexión y a una mala distribución del tráfico sobre la red. Esto es, mucho tráfico fluyendo por aquellos enlaces que conforman los caminos más cortos, creando cuellos de botella, mientras otros enlaces son apenas utilizados [WAN01].

Entonces queda claro que la optimización de recursos requiere capacidades adicionales a aquellas proveídas por el modelo de datagramas. Si deseamos obtener una mejor utilización de la red, es necesario tener un control explícito sobre los caminos que

deben atravesar los datos. Entonces, el flujo total de datos sobre la red puede ser acomodado de forma a minimizar los recursos utilizados, maximizar la probabilidad de enrutamiento de toda la demanda de tráfico y minimizar el costo que ello implica, considerando los parámetros de QoS que cada demanda de tráfico solicita [STA01].

Para seleccionar explícitamente los caminos, se puede utilizar MPLS (*MultiProtocol Label Switching*). Una red MPLS consiste en un conjunto de nodos llamados *Label Switch Routers* (LSR), que tienen la capacidad de enrutar paquetes utilizando etiquetas, las cuales son adheridas a cada paquete IP. Las etiquetas definen un flujo de paquetes entre dos nodos de la red, o, en el caso de una transmisión punto a multipunto, entre el nodo origen y el conjunto de nodos destinos. Por cada flujo, denominado *Forwarding Equivalent Class* (FEC), un camino específico (*Label Switch Path - LSP*) a través de la red es definido. La Figura 1.1 muestra un dominio MPLS, el cual puede representar un sistema autónomo (*Autonomous System - AS*) administrado por alguna entidad privada, como ser un proveedor de servicios de Internet (*Internet Service Provider - ISP*) [STA01].

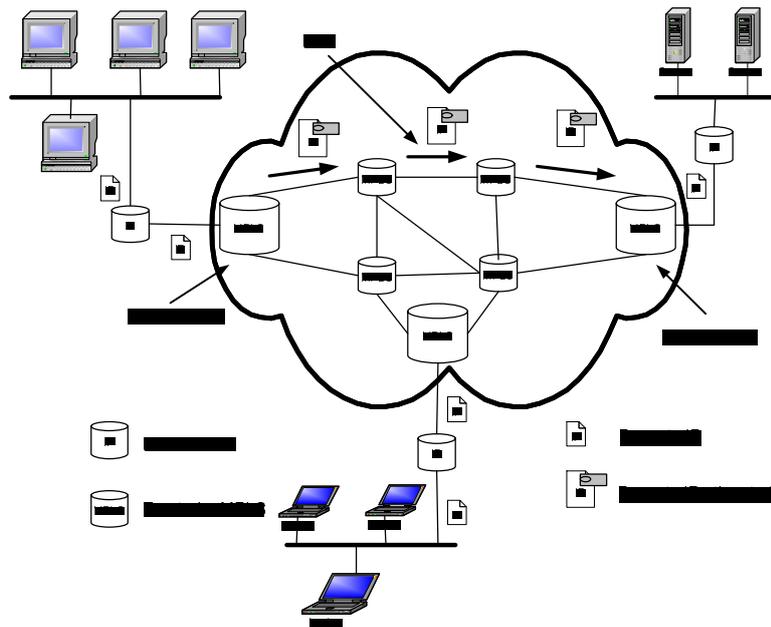


Figura 1.1. Dominio MPLS

El modelo DiffServ (*Differentiated Services*) en vez de distinguir flujos individuales LSP clasifica los paquetes en categorías (según el tipo de servicio solicitado). A cada categoría le corresponde un SLA (*Service Level Agreement*). Los usuarios pueden contratar o solicitar un determinado caudal en la categoría que deseen. Los routers tratan cada paquete según su categoría (que viene marcada en la cabecera del paquete).

1.2 Ingeniería de Tráfico

La elección dinámica de caminos para cada una de las demandas de tráfico en una red de computadoras, considerando el balanceo de carga de forma a evitar el congestionamiento de los enlaces, atendiendo los parámetros de QoS y optimizando los recursos de la red, es conocida como **Ingeniería de Tráfico** [STA01].

Existe un creciente número de aplicaciones punto a multipunto surgidas en Internet denominadas *multicast*, en las cuales se debe transmitir de un origen a varios destinos. Entre estas aplicaciones, se pueden mencionar como ejemplo el video bajo demanda (*Video on-demand - VoD*), las tele-conferencias, las transmisiones de radio y TV, y la educación a distancia. En consecuencia, se ha incrementado el interés en algoritmos de enrutamiento explícito punto a multipunto para ingeniería de tráfico. Esto se debe principalmente a que la pila de protocolos tradicionalmente usada en las redes de datagramas TCP/IP no puede hacer ingeniería de tráfico. Estos inconvenientes pueden ser superados utilizando por ejemplo MPLS y RSVP-TE (*Resource Reservation Protocol – Traffic Engineering*) [OOM02]. RSVP-TE es un protocolo de señalización utilizado para la reserva de recursos de *ancho de banda* a lo largo de un camino que conecta un nodo de ingreso y otro de egreso, lo cual permite garantizar servicios con QoS. De esta forma, la ingeniería de tráfico *multicast* en una red MPLS puede llevarse a cabo en dos pasos:

1. Encontrando el camino desde el nodo origen a cada uno de los nodos destinos utilizando un algoritmo apropiado para tal efecto.
2. Reservando los recursos a lo largo de los caminos hallados en el paso 1, utilizando RSVP-TE [DON02].

Los caminos deben ser elegidos de forma a acomodar todo el tráfico a un costo mínimo en términos de recursos, considerando parámetros de QoS como retardo máximo de extremo a extremo y retardo medio, y balanceando la carga de forma a evitar congestionamientos en la red. De este hecho se deriva que la ingeniería de tráfico *multicast* debe considerar más de una métrica, y por lo tanto, el problema de enrutamiento *multicast* en redes de computadora puede ser formulado como un Problema de Optimización Multiobjetivo (*Multi-objective Optimization Problem – MOP*) [VAN99].

Motivado por la necesidad creciente por parte de los administradores de AS de contar con herramientas informáticas eficaces para dicha tarea, y dado el éxito de los algoritmos de Optimización por Colonias de Hormigas (*Ant Colony Optimization - ACO*) [DOR91] en

varios problemas combinatoriales [SCH03, STÜ00, GOM04a], este trabajo propone por primera vez la resolución del problema de enrutamiento *multicast* utilizando los conceptos de ACO. De esta manera, una nueva alternativa de solución basada en ACO es propuesta al estado del Arte.

Habiendo introducido conceptos esenciales utilizados a lo largo del libro, el Capítulo 1 continúa de la siguiente manera: en la Sección 1.3 se da la definición formal de *multicast*, *unicast* y *broadcast*, y se ejemplifica la diferencia entre cada una de estas formas de comunicación. En la Sección 1.4 se definen las métricas de optimización consideradas en este trabajo. Posteriormente, en la Sección 1.5 se presentan los trabajos relacionados a problemas de enrutamiento *multicast* en redes de computadoras. Por último, en la Sección 1.6 se da una breve introducción al enfoque multiobjetivo considerado en este trabajo para la resolución del problema de enrutamiento *multicast* en redes de computadoras.

1.3 Multicast

La selección de una ruta entre dos nodos de una red es una de las tareas más importantes en las redes de computadoras. Esta es llevada a cabo por los algoritmos de enrutamientos, que se encargan de seleccionar la ruta más adecuada.

Las formas de comunicación tradicionales son básicamente tres y se definen a continuación [KOM93a].

1. *Unicast*: Transmisión de datos desde una fuente a un destino.
2. *Broadcast*: Transmisión de datos desde una fuente a todos los posibles destinos.
3. *Multicast*: Transmisión de datos desde una fuente a un conjunto de destinos.

Implementando *unicast* o *broadcast* se puede realizar una transmisión *multicast*. No obstante, estas soluciones no son eficientes, i.e. enviar datos a todos los nodos de la red para alcanzar a un subconjunto de nodos como es el caso de *broadcast* o utilizar el mismo enlace para transmitir el mismo flujo de datos a distintos nodos genera gran desperdicio de recursos. Como ejemplo de transmisión de *broadcast* en la Figura 1.2 (a) el nodo 0 desea transmitir a los nodos 2, 4 y 6, pero eventualmente también transmite a otros nodos que no desean la información, como el nodo 3 y 5. Por su parte, en la Figura 1.2 (b) puede verse la transmisión *multicast* utilizando varias transmisiones *unicast*. En

este caso nótese que el enlace entre el 0 y 1 es utilizado dos veces para transmitir el mismo dato a los nodos 2 y 4, generando una sobre utilización innecesaria.

Un árbol de expansión es la estructura apropiada para una comunicación *multicast*, como puede apreciarse en la Figura 1.2 (c). El árbol de expansión consta de un nodo fuente desde el cual cada uno de los nodos destinos puede ser alcanzado, a través de un solo camino. Es decir, consiste en un grafo sin ciclos [KOM93a]. El árbol de expansión *multicast* evita el desperdicio de recursos como lo hacen las técnicas mencionadas anteriormente.

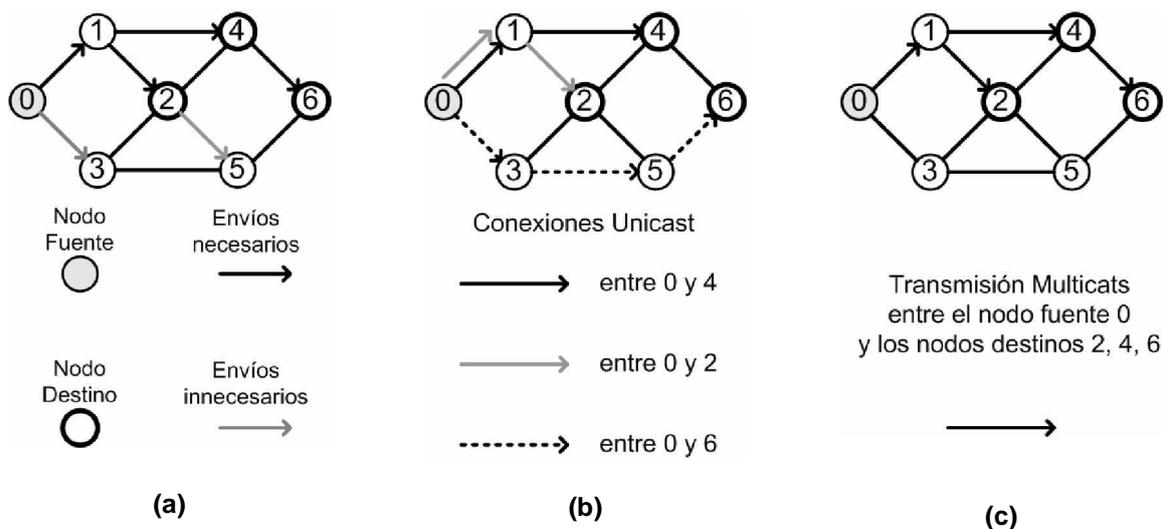


Figura 1.2. (a) *Multicast* utilizado *Broadcast* (b) *Multicast* utilizando *Unicast* (d) *Multicast* utilizando un árbol de expansión, forma apropiada que no desperdicia los recursos de la red.

1.4 Métricas de Optimización Multicast

Un algoritmo de enrutamiento *multicast* se encarga de construir un árbol *multicast* que conecta el nodo origen a cada uno de los destinos de la comunicación *multicast*. El mismo puede optimizar un conjunto de distintas métricas. Además, la aplicación para la cual se construye el árbol puede tener diversos requerimientos de QoS como retardo máximo de extremo a extremo limitado, retardo medio acotado o ancho de banda mínimo requerido. Seguidamente son presentadas las distintas métricas a ser consideradas en la ingeniería de tráfico *multicast*, tratada en este trabajo.

1.4.1 Utilización Máxima de los Enlaces

La utilización máxima de los enlaces es uno de los objetivos principales de la ingeniería de tráfico *multicast*, consiste en encontrar los caminos con una cierta

capacidad entre el nodo origen y el conjunto de nodos destinos, evitando el congestionamiento en la red y balanceando la carga en la misma. El enfoque tradicional utilizado en la práctica para lograr dicho objetivo es el enrutamiento a través de los caminos con menor utilización de los enlaces, donde la utilización de un enlace está definida como el tráfico que fluye a través de él, dividido su capacidad. Por su parte, la utilización máxima de un camino (α_p) se define como la utilización máxima de un enlace en el camino del origen al destino [SEO02]. El enrutamiento a través de dichos enlaces evita la congestión de una red sobrecargada, también reduce la pérdida de paquetes y el retardo total [DON04]. Además, disminuye la probabilidad de rechazos de posteriores solicitudes de tráfico [SEO02]. Dado que los algoritmos de enrutamiento *multicast* deben conectar el nodo origen con cada uno de los destinos, es deseable que el árbol *multicast* esté compuesto por aquellos enlaces con menor utilización. Esto es, si la utilización máxima de los enlaces de un árbol es definida como la utilización de su enlace más utilizado o utilización máxima de sus enlaces (α_T), se desea hallar el árbol que minimice α_T [SEO02]. A continuación se presenta un ejemplo en el cual se demuestra la utilidad del enrutamiento considerando la métrica α_T .

Ejemplo 1.1. Considere la Figura 1.3 (a), donde cada enlace tiene una capacidad de 10 Mbps y un tráfico actual asignado, en Mbps, dado en la Figura.

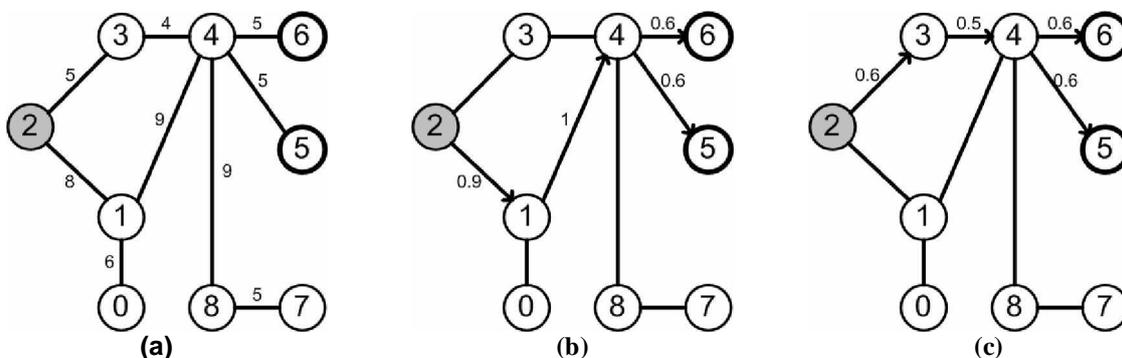


Figura 1.3. (a) Red de 9 nodos. Cada enlace tiene una capacidad de 10 Mbps y un tráfico actual fluyendo a través de él, en Mbps, dado en la misma Figura. (b) Árbol con origen en 2, conjunto destino {5, 6} y demanda de tráfico de 1 Mbps. La utilización de cada enlace es mostrada en la misma Figura. Para este árbol, $\alpha_T = 1$. (c) Árbol alternativo. Para esta opción, $\alpha_T = 0.6$

Suponga que el nodo 2 desea hacer una transmisión de 1 Mbps a los nodos 5 y 6, y que el árbol hallado para dicha transmisión es el mostrado en la Figura 1.3 (b). En dicha Figura también se muestra la utilización de cada enlace luego de haber sido enrutada la citada demanda *multicast*. Si bajo estas circunstancias es generada una nueva solicitud

multicast de 2 Mbps con origen en el nodo 2 y destinos en los nodos 0 y 1, dicha solicitud sería bloqueada por falta de recursos. Esta situación podría haber sido evitada si la transmisión de datos del primer grupo *multicast* se hiciera a través del árbol con menor α_T . Dicho árbol es mostrado en la Figura 1.3 (c).

Un problema de asignación de flujo fue mostrado en el Ejemplo 1.1, el mismo puede ser optimizado si todas las demandas *multicast* son conocidas *a priori*. Debido al dinamismo con que son generadas las demandas de tráfico en las redes de computadoras reales, es imposible conocer todas estas demandas de antemano. Por lo tanto, es razonable hacer optimizaciones locales teniendo en cuenta la utilización máxima de los enlaces del árbol en el momento en que cada grupo *multicast* solicita su demanda de tráfico, de forma a balancear el tráfico total sobre la red, de la mejor manera posible.

El enrutamiento a través de los enlaces de menor utilización es útil tanto para el balanceo de carga como para la reducción de la congestión, pero este podría desperdiciar recursos debido a la longitud de las rutas en término de saltos (i.e. suma de ancho de banda asignado en cada enlace a lo largo del camino) [SEO02]. Esta situación es mostrada en el Ejemplo 1.2.

Ejemplo 1.2. En la Figura 1.4 tenemos la red de la *National Science Foundation* (NSF) [BAR00]. Sobre cada enlace se tienen los valores de utilización de los enlaces (i.e. el enlace que conecta los nodos 7 y 8 tiene una utilización de 0.2). Suponga que el nodo 0 desea transmitir un flujo de datos al nodo 2. El camino de menor utilización de los enlaces está representado por un conjunto de flechas.

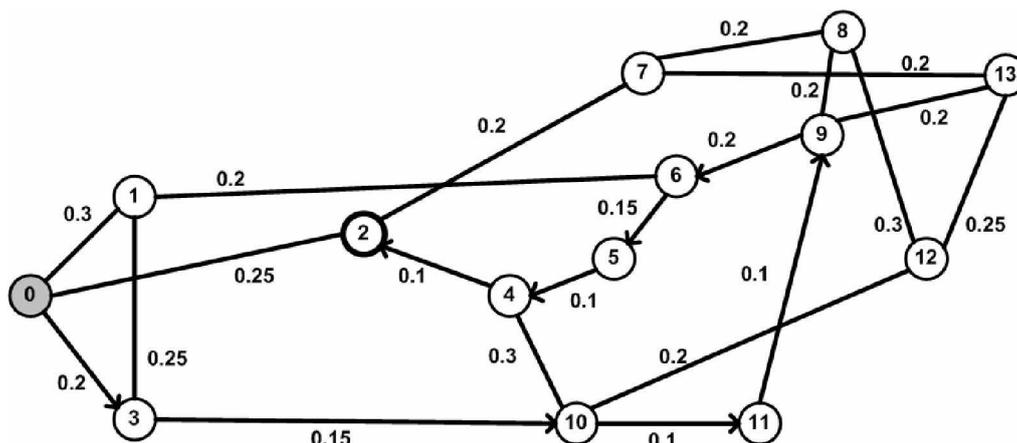


Figura 1.4. Red de la NSF, con la utilización de los enlaces como métrica expuestas sobre los enlaces. El camino de menor utilización de los enlaces, con $\alpha_p = 0.2$, es mostrado en la Figura.

Note que el camino directo 0-2 tiene $\alpha_p = 0.25$, pero un costo en términos de recursos utilizados mucho menor.

Note que la utilización máxima para este árbol *multicast* es $\alpha_p = 0.2$. Por otra parte, el camino directo desde 0 a 2 tiene una utilización máxima $\alpha_p = 0.25$, apenas superior a la alternativa mostrada en la Figura, pero un consumo de ancho de banda mucho menor. Si tenemos en cuenta el número de saltos, la solución mostrada en la Figura 1.4, no sería óptima. En dicho caso, elegiríamos la solución del camino directo 0 a 2, con un α_p apenas mayor. Entonces podemos concluir que una métrica a tener en cuenta es la utilización de los recursos de la red.

1.4.2 Costo del Árbol Multicast

Como fuera explicado en el Ejemplo 1.2, la utilización de los recursos también debería ser considerada como una métrica de optimización al construir un árbol *multicast*. Una forma general de optimizar dicha métrica es minimizando el costo del árbol. El costo del árbol C_T está dado por la suma de los costos de los enlaces que lo componen. El mismo puede tener diferentes significados: puede representar alguna cantidad monetaria, como también ser proporcional al consumo total de ancho de banda del grupo *multicast*. Cualquiera sea el significado, aquellas soluciones de menor costo siempre son deseables. Por ejemplo, en una red cuyos enlaces tienen costo unitario, la minimización del costo del árbol *multicast* conduce al árbol con menor número de enlaces, y por lo tanto, al de menor consumo de ancho de banda.

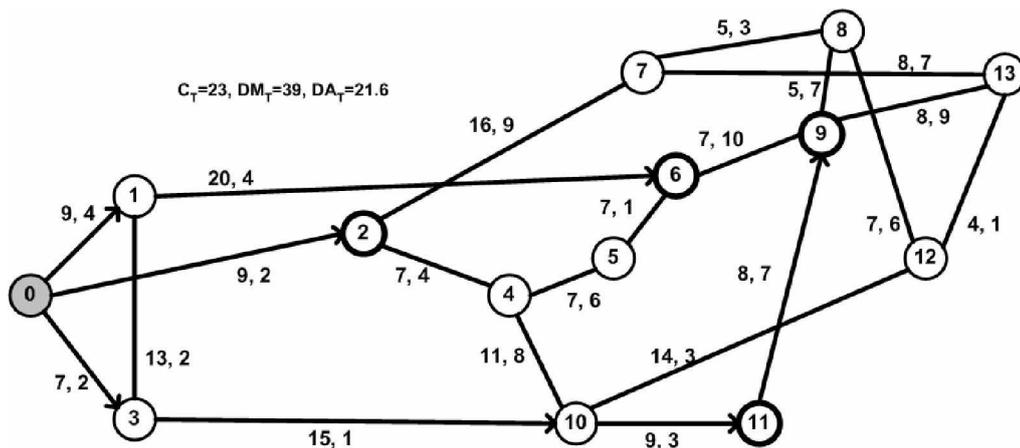


Figura 1.5. Árbol *multicast* de costo mínimo. Los números sobre los enlaces indican costo y retardo del mismo.

La minimización del costo del árbol optimiza una métrica muy importante asociada a los recursos de la red (i.e. ancho de banda consumido), pero en aplicaciones como e-learning, tele-conferencias, video bajo demanda y transmisiones de audio y/o video, se necesitan árboles de bajo retardo medio y/o bajo retardo máximo de extremo a extremo. La Figura 1.5 muestra el árbol de costo mínimo para el grupo *multicast* conformado por el nodo 0 como origen y el conjunto destino {2, 6, 9, 11}. El retardo y el costo de cada enlace están representados respectivamente por el valor próximo al mismo en la Figura 1.5.

La solución mostrada en la Figura 1.5 es óptima cuando se considera la métrica de costo. Sin embargo, esta puede no ser de utilidad para aquellas aplicaciones sensibles al retardo, como las citadas anteriormente.

1.4.3 Retardo Medio y Retardo Máximo de Extremo a Extremo

Enrutar a través del camino de menor retardo es la forma tradicional para optimizar las métricas de retardo (*delay*) medio (DA_T) y retardo máximo de extremo a extremo (DM_T). La extensión natural al caso *multicast* es la transmisión a través del árbol compuesto por los caminos de menor retardo desde el nodo fuente a los nodos destinos. El mismo es conocido como el árbol del camino más corto (*Shortest Path Tree - SPT*).

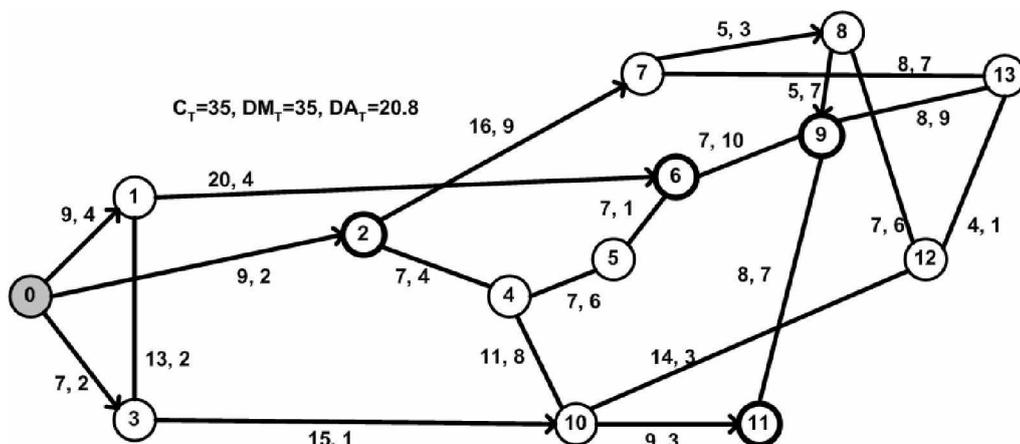


Figura 1.6. Árbol *multicast* óptimo cuando solo DM_T es considerado. Los números sobre los enlaces indican costo y retardo del mismo.

La Figura 1.6 ilustra dicha solución para el mismo grupo *multicast* mostrado en la Figura 1.5. El par de números sobre cada enlace corresponde al retardo y al costo del mismo. El retardo promedio DA_T está dado por la suma de los retardos de los caminos a cada uno de los nodos destino dividido por el número de nodos destino. Es importante destacar que, con estos valores de retardo, el árbol de costo mínimo de la Figura 1.5 tiene $DM_T = 39$ y $DA_T = 21.6$, mientras que el SPT tiene $DM_T = 35$ y $DA_T = 20.8$. El retardo

máximo y el retardo medio del árbol de costo mínimo son mayores que el obtenido por SPT y dicha solución podría no ser de utilidad para aquellas aplicaciones sensibles a estas métricas. En contrapartida, el costo del árbol de la Figura 1.5 es de apenas 23, mientras que el costo del SPT es 35. Claramente, existe una relación de compromiso entre estas métricas.

1.5 Trabajos Relacionados

En la Introducción se ha mencionado que el enfoque típico utilizado en Internet para enrutar paquetes es el enrutamiento a través del camino más corto, considerando la métrica de retardo o número de saltos. En redes de computadores, la mayor parte de los algoritmos utilizados son variantes del algoritmo de Dijkstra [DIJ59]. El mismo encuentra el camino de menor costo, en términos de retardo o saltos, desde un nodo fuente a un nodo destino. Si aplicamos el algoritmo a un grupo *multicast*, este produciría un árbol cuyos caminos desde el nodo origen al conjunto de nodos destinos son óptimos en términos de la métrica utilizada. En Internet el estándar más utilizado es el *Open Shortest Path First* (OSPF) [STA00], soportado por la mayoría de los *Routers* hoy comercializados.

Hallar el árbol de menor costo es conocido como Problema del árbol de Steiner, y es sabido que la complejidad de dicho problema es NP-completa [KOM93a]. La dificultad de hallar el árbol de costo mínimo radica en la existencia de un número de árboles que crece exponencialmente con el número de nodos de la red, y la solución óptima es obtenida por búsqueda exhaustiva [KOM93a]. Debido al elevado tiempo de cómputo que consume hallar la solución óptima, han sido propuestos varios algoritmos heurísticos para la resolución de dicho problema. Uno de los algoritmos más famosos fue propuesto por Kou, Markowsky y Berman (algoritmo KMB) [KOU81]. Dada una red de computadoras modelada como un grafo G y un grupo *multicast*, KMB transforma el grafo G a otro G' . G' consiste en un grafo completamente conexo, en el que cada enlace representa el camino de menor costo en G entre los dos nodos involucrados. El número de nodos de G' es igual al tamaño del grupo *multicast*. Utilizando G' , el algoritmo construye el árbol de expansión de menor costo. Luego, el árbol en la red original es obtenido transformando el árbol de expansión en G' a los caminos en G que ellos representan.

En [HWA00], Hwang et al. presentaron un algoritmo genético simple que minimiza el costo del árbol *multicast*. Los autores propusieron representar los cromosomas como cadenas de enteros, en los cuales cada elemento de la cadena es usado para indexar una tabla de rutas de un nodo destino. La longitud de los cromosomas es igual al número de

receptores *multicast*. La tabla de rutas para un nodo destino consiste en un conjunto de R posibles rutas que conectan el nodo origen y el destino, donde R es un parámetro del algoritmo. Cada elemento de la cadena (gen) puede tomar un valor entero entre 1 y R , el cual representa (indexa) una ruta dada de la tabla de rutas al nodo destino correspondiente. La adaptabilidad (o *fitness*) de un individuo es inversamente proporcional al costo del árbol que él representa. Los autores mencionan la manera de extender la propuesta a situaciones que requieran restricciones como retardo de extremo a extremo acotado y/o ancho de banda mínimo requerido para la transmisión de datos.

Los algoritmos que solo optimizan la función de costo, pueden proveer soluciones que violan las restricciones de retardo máximo impuestas por ciertas aplicaciones sensibles a las mismas. Algunos algoritmos añaden una restricción de retardo máximo al problema del árbol de costo mínimo para considerar la métrica de retardo. Este problema fue formulado por primera vez por Kompella, Pasquale y Polyzos [KOM93b]. Los autores propusieron un algoritmo heurístico denominado KPP basado en el algoritmo KMB. KPP extiende el alcance de KMB a problemas con restricciones de retardo, teniendo en cuenta el retardo a cada nodo destino en el momento de formar la red completamente conexas. El retardo máximo permitido es determinado *a priori*. La importancia de este trabajo radica en la formulación, por primera vez, de un problema de enrutamiento *multicast* en redes de computadoras en el cual más de una métrica independiente debe ser considerada.

Otras propuestas basadas en algoritmos genéticos fueron publicadas en [RAV98, ZHE01, ARA02]. El primero de ellos fue propuesto por Ravikumar et al. [RAV98], quienes presentaron un algoritmo genético simple en el cual un cromosoma consiste en el conjunto de enlaces que conforman el árbol que él representa. Para cada individuo el *fitness* es directamente proporcional a la calidad del individuo, siendo un buen individuo aquel árbol de bajo costo que cumple con la restricción de retardo máximo impuesto *a priori*. El operador de cruzamiento consiste en hallar los enlaces comunes de ambos padres y crear sub-árboles a partir de los enlaces comunes generando un hijo a través de la reconexión de los diferentes sub-árboles. La reconexión se lleva a cabo según el siguiente criterio: en caso que al menos uno de los padres cumpla con la restricción de retardo, la reconexión se hace a través del camino de menor costo entre los sub-árboles. En caso contrario, los sub-árboles se conectan a través del camino de menor retardo. El operador de mutación consiste en eliminar del árbol un enlace elegido al azar, y reconectar los sub-árboles de la misma forma que lo hace el operador de cruzamiento.

Este trabajo fue mejorado consecutivamente por Zhengying et al. [ZHE01] y por Araujo et al. [ARA02].

Puede notarse claramente que la resolución del problema de enrutamiento *multicast* fue ampliamente estudiada con algoritmos genéticos, y salvo algunos casos particulares, no fue tratado con algoritmos basados en colonias de hormigas. Liu y Wu [LIU03] proponen un algoritmo basado en colonias de hormigas para la construcción de un árbol *multicast*, donde solamente se minimiza el costo del árbol con restricciones en el grado de cada nodo, i.e. el grado de cada nodo es el número de enlaces que parten de él. Este trabajo sugiere construir una rama por vez entre el nodo fuente y un nodo destino, que es elegido al azar entre el conjunto de destinos. Donde cada rama es construida utilizando una colonia de hormigas.

En [CHU02], Chu et al. utilizan una colonia de hormigas en la construcción del árbol *multicast* minimizando el costo, considerando como restricciones múltiples métricas de calidad de servicio (QoS). El árbol se construye desde el nodo fuente adhiriendo enlaces seleccionados probabilísticamente entre aquellas que cumplan con las restricciones impuestas. Este proceso se repite hasta llegar a todos los destinos.

Das et al. sugieren resolver el problema del árbol de costo mínimo también con el enfoque de las colonia de hormigas [DAS02]. Para la construcción del árbol de costo mínimo, se ubica una hormiga en cada uno de los vértices terminales (nodo fuente y destinos) a ser conectados, y a cada iteración se mueven todas las hormigas a un nodo vecino. Cuando una hormiga llega a un nodo que ya fue alcanzado por otras hormigas esta deja de adherir enlaces y termina su tarea. De esta forma cuando ninguna de las hormigas pueda moverse, termina la búsqueda. Por las características del algoritmo, el grafo que representa la red debe ser *simétrico*, lo cuál representa una limitación significativa.

La imposición de un valor máximo *a priori* es la principal desventaja de los enfoques basados en restricciones de retardo máximo, dado que este mecanismo puede descartar soluciones de muy bajo costo, con retardo máximo solo apenas superior a la restricción dada *a priori*.

El progresivo aumento de redes MPLS y los protocolos de señalización como RSVP-TE, los cuales en conjunto permiten el encaminamiento explícito y la reserva de recursos a través de los caminos, ha motivado el estudio de nuevos algoritmos con esquemas alternativos de balanceo de carga y optimización de recursos. Para llevar a cabo dicha

tarea, el objetivo adoptado en recientes publicaciones [SEO02, DON04] es la minimización de la utilización máxima de los enlaces de la red (α). Para resolver el problema dinámico de ingeniería de tráfico *multicast*, en el cual las solicitudes de tráfico llegan una después de otra, Seok et al. [SEO02] propusieron un algoritmo heurístico basado en la extensión del algoritmo de Dijkstra [DIJ59] que minimiza α . Aunque esta propuesta es útil para minimizar la utilización máxima de los enlaces de la red y reducir el congestionamiento sobre la misma, el ancho de banda consumido puede ser desperdiciado debido a posibles rutas largas en términos de saltos. Por lo tanto, los autores restringen la longitud de las rutas a un valor máximo dado *a priori*, de forma a hallar un árbol *multicast* en el cual el número de saltos a los nodos destinos esté acotado. De esta manera, los autores intentan optimizar, además de la utilización máxima de los enlaces, el consumo de ancho de banda. El algoritmo propuesto en este trabajo puede ser dividido en dos pasos: 1- modificación del grafo original a uno en el cual el número de saltos desde el nodo origen a cualquier otro nodo de la red esté acotado por un valor dado *a priori*; 2- encontrar el árbol *multicast* que minimice la utilización máxima de los enlaces, utilizando para ello el algoritmo del camino de menor utilización máxima de los enlaces. Este trabajo demostró que el tráfico *multicast* puede ser mejor balanceado y por lo tanto la probabilidad de rechazos de solicitudes por falta de recursos puede ser disminuida cuando la métrica de enrutamiento utilizada es la utilización de los enlaces. La formulación del problema claramente demuestra la característica multiobjetivo de los problemas de enrutamiento *multicast*, pues existe más de un objetivo a optimizar.

En [DON04], Donoso et al. propusieron un esquema de ingeniería de tráfico *multicast* multi-árbol. Con este enfoque, el flujo de datos desde el nodo fuente a los nodos destinos es enviado a través de distintos árboles, de forma a balancear la carga sobre la red. El esquema no solamente tiene en cuenta la utilización máxima de los enlaces, sino también el número total de saltos, el consumo total de ancho de banda y el retardo total de extremo a extremo. El método propuesto minimiza una función objetivo compuesto por la suma ponderada de las cuatro métricas citadas. Debido a la naturaleza NP-completa del esquema propuesto, los autores también propusieron un algoritmo heurístico para la resolución del problema. Este consiste de dos etapas: 1- obtener un grafo modificado. En esta etapa, todos los caminos posibles desde el nodo fuente a cada uno de los nodos destinos son hallados. Luego, para cada nodo perteneciente al camino entre el nodo fuente un nodo destino, un valor de distancia basado en el número de saltos, ancho de banda consumido y retardo es calculado; 2- para cada destino *multicast*, hallar los

caminos requeridos para la transmisión de flujo en el grafo modificado. Es importante notar que este trabajo ya considera el estudio de algoritmos evolutivos como trabajo futuro, pues la característica multiobjetivo del problema y las relaciones de compromiso entre los distintos objetivos son resaltados en las pruebas de simulación hechas por los autores.

En [CRI04a], [CRI04b] y [CRI04c], Crichigno et al. presentan un algoritmo de ruteamiento *multicast* multiobjetivo, el *Multiobjective Multicast Algorithm* – MMA basado en el algoritmo evolutivo de segunda generación *Strength Pareto Evolutionary Algorithm* – SPEA [ZIT99], donde optimizan simultáneamente el costo del árbol, el retardo máximo extremo a extremo, el retardo promedio y la utilización máxima de los enlaces. Mas tarde, en [CRI04d], se propone otro algoritmo de ruteamiento *multicast* multiobjetivo llamado *Multiobjective Multicast Algorithm 2* – MMA2 que a diferencia de la versión anterior, codifica a los algoritmos como grafos en lugar de utilizar tablas de enrutamiento.

Debido a la gran cantidad de algoritmos evolutivos de segunda generación no quedaba claro cuál de ellos era el más adecuado para el problema de enrutamiento *multicast*, por lo cuál Talavera y Barán en [TAL04a] realizaron una comparación entre varios algoritmos evolutivos de segunda generación. Los resultados en pruebas estáticas indican que en promedio el SPEA es el más apropiado. Además tratan por primera vez el problema de selección de soluciones del conjunto Pareto, sugiriendo varias políticas de selección [TAL04b].

Dada la naturaleza multiobjetivo del problema de enrutamiento *multicast* en redes de computadoras que ya fuera estudiado por varios autores, la relación de compromiso entre las distintas métricas a optimizar (presentadas en la Sección 1.4), la imperiosa necesidad de optimizar cada una de ellas en forma simultánea y las pocas alternativas distintas a los algoritmos evolutivos, este trabajo propone formular el problema como puramente multiobjetivo. Por tal motivo, la siguiente Sección presenta una introducción al enfoque multiobjetivo utilizado en adelante.

1.6 Enfoque Multiobjetivo

En redes de computadoras la optimización de múltiples objetivos simultáneamente, proporciona un conjunto de soluciones y la imposibilidad de decidir cuál de ellas es mejor si se consideran todos los objetivos al mismo tiempo. Se dice que las soluciones de un problema con objetivos múltiples son óptimas porque ninguna otra solución, en todo el

espacio de búsqueda, es superior a ellas cuando se tienen en cuenta *todos* los objetivos al *mismo* tiempo, i.e. ningún objetivo puede mejorarse sin degradar al menos algún objetivo. Esto ha sido mostrado en la Sección 1.4, donde, con ejemplos, se ha demostrado que α_T , C_T , D_A y D_M pueden estar en conflicto entre si.

Al conjunto de estas soluciones óptimas se conoce como soluciones Pareto óptimas. Su nombre les fue dado en honor al ingeniero y economista Wilfredo Pareto, quien fue el primero en definir un nuevo criterio de optimalidad para los problemas en los que existen múltiples objetivos a optimizar, y existen conflictos al realizar la optimización simultánea de los mismos [PAR98]. A partir de este concepto se establece, como requisito para afirmar que una situación es mejor que otra, el que en ella no se empeore ningún objetivo, pero se mejore a alguno. En caso contrario, según Pareto [PAR98], para decidir se requiere un juicio de valor aportado por el tomador de decisiones y la ciencia no puede guiarnos.

La optimización simultánea de múltiples objetivos en los problemas de ingeniería de tráfico *multicast*, como la utilización máxima de los enlaces, el costo del árbol *multicast*, el retardo máximo de extremo a extremo y el retardo medio lleva a soluciones en las que los objetivos presentan conflictos entre si; es decir, la mejora en un objetivo conduce a un deterioro en algún otro objetivo. Aunque la mayoría de los problemas de enrutamiento en redes de computadoras involucran este tipo de situaciones, las propuestas computacionales presentadas hasta la fecha se limitan a convertir el problema de objetivos múltiples en uno en que existe un solo objetivo. Esta reducción es debida a los modelos matemáticos empleados y puede realizarse de varias maneras. Por ejemplo, priorizando uno de los objetivos y utilizando los restantes como restricciones, o generando un objetivo compuesto, otorgando pesos a cada uno de ellos, de forma a optimizar la suma ponderada de los mismos. De todos modos, ninguna de estas reducciones refleja fielmente al problema y, por lo tanto, tampoco otorga soluciones completamente satisfactorias para este contexto multiobjetivo.

1.7 Organización del Presente Trabajo.

En el presente capítulo se ha introducido al problema de enrutamiento *multicast* en redes de computadoras. A continuación, en el Capítulo 2 se da una definición formal de un Problema de Optimización Multiobjetivo (MOP) y se definen términos relevantes referentes a dichos problemas. Luego, en el Capítulo 3 se presenta la formulación

matemática del problema de enrutamiento *multicast* en redes de computadoras como un MOP, y se definen los objetivos. Posteriormente, en el Capítulo 4 se presentan los Algoritmos basados en *Ant Colony Optimization* (ACO) que serán considerados en el presente trabajo. Por su parte, el Capítulo 5, se presentan las simulaciones en un ambiente estático así como también los semi-dinámicos de los Algoritmos ACO propuestos y el MMA. Las Conclusiones y Trabajos Futuros finalmente son presentados en el Capítulo 6.

Capítulo 2

Optimización con Múltiples Objetivos

2.1 Introducción

El problema de enrutamiento *multicast* en redes de computadoras es considerado en este trabajo como un Problema de Optimización Multiobjetivo (MOP), con las siguientes funciones objetivos a optimizar:

- 1- utilización máxima de los enlaces del árbol (α_T),
- 2- costo del árbol (C_T),
- 3- retardo máximo de extremo a extremo (DM_T) y,
- 4- retardo medio a los nodos destinos (DA_T), conforme serán definidos en el Capítulo 3.

De manera a abordar el Problema de Optimización Multiobjetivo (MOP), a continuación se presenta la definición general de un problema multiobjetivo. También se menciona algunas técnicas tradicionales de búsqueda utilizadas en la actualidad para la resolución de problemas con más de un objetivo a optimizar. Posteriormente, se propone a los algoritmos ACO como herramientas que poseen cualidades deseables para la resolución de MOPs.

2.2 Principios Básicos y Terminologías

A continuación se presenta la definición formal de un problema de optimización multiobjetivo (MOP) adoptada en este trabajo [VAN99, ZIT99].

Definición 1: Problema de Optimización Multiobjetivo (*Multiobjective Optimization Problem: MOP*). Un MOP general incluye un conjunto de h parámetros (variables de decisión), un conjunto de k funciones objetivo, y un conjunto de m restricciones. Las funciones objetivo y las restricciones son funciones de las variables de decisión. Luego, el MOP puede expresarse como:

$$\begin{array}{ll}
 \textit{Optimizar} & \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \\
 \textit{sujeto a} & \mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_m(\mathbf{x})) \geq \mathbf{0} \\
 \textit{donde} & \mathbf{x} = (x_1, x_2, \dots, x_h) \in \mathbf{X} \\
 & \mathbf{y} = (y_1, y_2, \dots, y_k) \in \mathbf{Y}
 \end{array} \tag{2.1}$$

siendo x el vector de decisión e y el vector objetivo. El espacio de decisión se denota por X , y el espacio objetivo por Y . Optimizar, dependiendo del problema, puede significar igualmente, minimizar o maximizar. El conjunto de restricciones $e(x) \geq 0$ determina el conjunto de soluciones factibles X_f y su correspondiente conjunto de vectores objetivo factibles Y_f .

Definición 2: Conjunto de soluciones factibles. El conjunto de soluciones factibles X_f se define como el conjunto de vectores de decisión x que satisface $e(x)$:

$$X_f = \{x \in X / e(x) \geq 0\} \quad (2.2)$$

La imagen de X_f , es decir, la región factible del espacio objetivo, se denota por

$$Y_f = f(X_f) = \bigcup_{x \in X_f} \{y = f(x)\} \quad (2.3)$$

De estas definiciones se tiene que cada solución del MOP en cuestión consiste de una h -tupla $x=(x_1, x_2, \dots, x_h)$, que conduce a un vector objetivo $y = (f_1(x), f_2(x), \dots, f_k(x))$, donde cada x debe cumplir con el conjunto de restricciones $e(x) \geq 0$. El problema de optimización mono-objetivo consiste en hallar la x que tenga el “mejor valor” de $f(x)$, pero en un contexto multiobjetivo no existe un único mejor valor, sino un conjunto de “mejores soluciones”. Entre éstas, ninguna se puede considerar mejor que las demás si se tienen en cuenta todos los objetivos simultáneamente. De este hecho se deriva que pueden existir –y generalmente existen– conflictos entre los diferentes objetivos que componen el problema, i.e., la optimización de algún objetivo implica la degradación de algún otro objetivo. Por ende, al tratar con MOPs se precisa de un nuevo concepto de “óptimo”.

En la optimización de un solo objetivo, el conjunto de variables de decisión factibles está completamente ordenado mediante una función objetivo f . Es decir, dadas dos soluciones $u, v \in X_f$, se cumple una sola de las siguientes proposiciones: $f(u) > f(v)$, $f(u) = f(v)$ o $f(v) > f(u)$. El problema de optimización consiste en hallar la solución (o soluciones) que tengan los valores óptimos (máximos o mínimos) de f . Cuando se trata de varios objetivos, sin embargo, la situación cambia. X_f , en general, no está totalmente ordenada por los objetivos. El orden que se da generalmente es parcial (i.e. pueden existir dos vectores de decisión diferentes u y v tal que $f(u)$ no puede considerarse mejor que $f(v)$ ni $f(v)$ puede considerarse mejor que $f(u)$ a pesar de que $f(u) \neq f(v)$). Para expresar esta situación matemáticamente, las relaciones $=$, \leq y \geq se deben extender. Esto se puede realizar de la siguiente manera:

Definición 3: Extensión de las relaciones $=$, \leq y \geq a MOPs. Dados 2 vectores de decisión $u, v \in X$,

$$\begin{aligned} f(u) = f(v) & \quad \text{si y solo si} \quad \forall i \in \{1, 2, \dots, k\}: f_i(u) = f_i(v) \\ f(u) \leq f(v) & \quad \text{si y solo si} \quad \forall i \in \{1, 2, \dots, k\}: f_i(u) \leq f_i(v) \\ f(u) < f(v) & \quad \text{si y solo si} \quad f(u) \leq f(v) \wedge f(u) \neq f(v) \end{aligned} \quad (2.4)$$

Las relaciones \geq y $>$ se definen de manera similar.

Definición 4: Dominancia Pareto en un contexto de minimización. Para dos vectores de decisión u y $v \in X$,

$$\begin{aligned} u \succ v \text{ (} u \text{ domina a } v\text{)} & \quad \text{si y solo si} \quad f(u) < f(v) \\ v \succ u \text{ (} v \text{ domina a } u\text{)} & \quad \text{si y solo si} \quad f(v) < f(u) \\ u \sim v \text{ (} u \text{ y } v \text{ no son comparables)} & \quad \text{si y solo si} \quad f(u) \not< f(v) \wedge f(v) \not< f(u) \end{aligned} \quad (2.5)$$

Definición 5: Dominancia Pareto en un contexto de maximización. Para dos vectores de decisión u y $v \in X$,

$$\begin{aligned} u \succ v \text{ (} u \text{ domina a } v\text{)} & \quad \text{si y solo si} \quad f(u) > f(v) \\ v \succ u \text{ (} v \text{ domina a } u\text{)} & \quad \text{si y solo si} \quad f(v) > f(u) \\ u \sim v \text{ (} u \text{ y } v \text{ no son comparables)} & \quad \text{si y solo si} \quad f(u) \not> f(v) \wedge f(v) \not> f(u) \end{aligned} \quad (2.6)$$

Alternativamente, $u \triangleright v$ denota que u domina o es no comparable con v .

Luego, de aquí en adelante, ya no será necesario diferenciar el tipo de optimización a realizar (minimización o maximización), al punto que un objetivo puede ser maximizado, mientras que otro puede ser minimizado.

Definido el concepto de dominancia Pareto, puede ser introducido el criterio de optimalidad Pareto de la siguiente manera:

Definición 6: Optimalidad Pareto. Dado un vector de decisión $x \in X_f$, se dice que x es no dominado respecto a un conjunto $Q \subseteq X_f$ si y solo si

$$\forall v \in Q : (x \triangleright v) \quad (2.7)$$

En caso que x sea no dominado respecto a todo el conjunto X_f , y solo en ese caso, se dice que x es una **solución Pareto óptima**. Por lo tanto, el conjunto Pareto óptimo X_{true} puede ser definido formalmente como:

Definición 7: Conjunto Pareto óptimo X_{true} . Dado el conjunto de vectores de decisión factibles X_f , se denomina conjunto Pareto óptimo X_{true} al conjunto de vectores de decisión no dominados que pertenecen a X_f , es decir:

$$X_{true} = \{ x \in X_f \mid x \text{ es no dominado con respecto a } X_f \} \quad (2.8)$$

El correspondiente conjunto de vectores objetivos $Y_{true} = f(X_{true})$ constituye el **frente Pareto óptimo**.

2.3 Búsqueda y Toma de Decisiones

Se pueden distinguir 2 niveles de dificultad al resolver un problema de optimización multiobjetivo: *la búsqueda y la toma de decisiones* [HOR97]. El primer aspecto se refiere al proceso de optimización, durante el cual se explora el espacio de soluciones factibles buscando las soluciones Pareto óptimas. En esta fase, como ocurre en la optimización de objetivo único, el espacio de búsqueda puede ser lo suficientemente grande y complejo como para impedir el uso de técnicas de optimización que den resultados exactos [STE86]. El segundo aspecto es equivalente a seleccionar un conjunto o una única solución de compromiso del conjunto Pareto óptimo ya encontrado. Este paso generalmente tiene que ver con cuestiones inherentes a las condiciones del problema y los recursos disponibles.

Los métodos de optimización multiobjetivo pueden clasificarse en tres categorías diferentes dependiendo de la manera en que se combinan ambas fases [COH78]:

- a) *Toma de decisión previa a la búsqueda (decidir, luego buscar)*: los objetivos del MOP se combinan en un único objetivo que implícitamente incluye información de preferencia obtenida del responsable de la toma de decisiones. Estos son métodos de toma de decisión *a priori*, tales como la suma con pesos, orden lexicográfico, etc. Gran parte de las técnicas tradicionales para resolución de MOPs utilizan esta estrategia [COH78, STE86].
- b) *Búsqueda previa a la toma de decisión (buscar, luego decidir)*: se realiza la optimización sin incluir información de preferencia. El responsable de la toma de decisiones escoge del conjunto de soluciones obtenidas (que idealmente son todas del conjunto Pareto óptimo). A éstos se conocen como métodos de toma de decisión *a posteriori*. Ejemplos: la optimización con *algoritmos evolutivos* y *colonia de hormigas*.

c) *Toma de decisión durante la búsqueda (decidir mientras se busca)*: esto permite que quien toma las decisiones pueda establecer preferencias durante un proceso de *optimización interactivo*. Luego de llevarse a cabo cada paso del proceso de optimización, se presenta al responsable de decisiones los conflictos existentes y se le permite especificar sus preferencias o compromisos. De esta manera se desarrolla una búsqueda guiada. Estos métodos reciben el nombre de *progresivos o interactivos*.

Combinar múltiples objetivos en un único objetivo a optimizar, posee la ventaja de que a la función resultante se puede aplicar cualquiera de los métodos de optimización de un único objetivo, ya disponibles en la literatura [COH78, STE86], sin mayores modificaciones. No obstante, tal combinación requiere conocimiento del dominio del problema que no siempre está disponible. En los problemas de diseño de diversas áreas de ingeniería, se realiza la búsqueda justamente para tener un mejor conocimiento del espacio de búsqueda del problema y de las soluciones alternativas. Al realizar la búsqueda antes de la toma de decisión no se requiere este conocimiento, pero se impide que quien toma la decisión exprese previamente sus preferencias, lo que probablemente conduce a una reducción de la complejidad del espacio de búsqueda. Un problema con las técnicas de toma de decisión interactivas o *a posteriori* es la dificultad de presentar al responsable de la toma de decisiones las diferencias entre las soluciones de un MOP de muchas dimensiones. Aún así, la integración de la búsqueda y la toma de decisiones que proponen las técnicas interactivas parece una alternativa promisorias para aprovechar las ventajas de los dos primeros tipos de técnicas.

El presente trabajo está enfocado en técnicas que han demostrado ser capaces de explorar espacios de búsquedas prácticamente ilimitados y altamente complejos y generar aproximaciones al conjunto Pareto óptimo.

2.4 El Método de Suma con Pesos

En redes de computadoras, el método tradicional de enrutamiento ha sido basado en esquemas que optimizan una sola métrica. Recientemente se han propuestos nuevos esquemas y algoritmos que consideran más de una métrica a optimizar [SEO02, DON04]. El método utilizado es el de la suma ponderada utilizando pesos [COH78]. Este método combina las diferentes funciones objetivo en una sola función F , de la siguiente manera:

Optimizar
$$F = \sum_{j=1}^k w_j f_j(x) \tag{2.9}$$

donde:

$x \in X_f$, y w_j es el peso usado para ponderar la j -ésima función objetivo.

Usualmente, y sin pérdida de generalidad, se escogen pesos fraccionarios y diferentes de cero, de manera a cumplir la condición de normalización: $\sum_{j=1}^k w_j = 1$, $w_j > 0$ [DON04].

El procedimiento es el siguiente: se escoge una combinación de pesos y se optimiza la función F para obtener una solución óptima. Otras soluciones surgen a partir de optimizaciones realizadas sobre una combinación diferente de pesos [DON04].

En el presente contexto, optimizar implica maximizar todas las funciones objetivo, o minimizarlas. Esto es, o se maximiza o se minimiza todas las funciones $f_j(x)$; no se admite la maximización de algunos objetivos y la minimización de otros.

En [DEB99] se demuestra que, si se utiliza un algoritmo de optimización que obtiene resultados exactos siempre y los pesos escogidos son siempre positivos, el método genera soluciones que siempre pertenecen al conjunto Pareto óptimo. La interpretación de este método es la siguiente. Alterar el vector de pesos y optimizar la ecuación implica encontrar un hiperplano (una línea para el caso en que se tengan dos objetivos) con una orientación fija en el espacio de la función objetivo. La solución óptima es el punto donde un hiperplano con esta orientación tiene una tangente común con el espacio de búsqueda factible. De aquí deducimos que este método no puede usarse para encontrar soluciones Pareto óptimas en problemas de optimización multicriterio que tienen un frente Pareto óptimo cóncavo. Para una discusión más detallada de esta característica se refiere al lector a [DEB99].

Si bien no se adecuan a la naturaleza del problema, la ventaja principal del método de suma con pesos es que permite la utilización de algoritmos desarrollados para la resolución de problemas de optimización mono-objetivos (*Single-objective Optimization Problems - SOPs*) bien conocidos y de probada eficacia, aún cuando se trate de problemas reales, de gran tamaño. Para problemas de gran escala, muy pocas técnicas de optimización multiobjetivo reales se han presentado [HOR97], a diferencia de técnicas de optimización de un solo objetivo que han existido desde hace bastante tiempo y han

sido probadas en diferentes situaciones. Sin embargo, este método cuenta con las siguientes limitaciones [DEB99]:

- a) El algoritmo de optimización se debe aplicar varias veces para encontrar las múltiples soluciones Pareto óptimas. Como cada corrida es independiente de las demás, generalmente no se obtiene un *efecto sinérgico*. Por tanto, delinear el frente Pareto óptimo resulta computacionalmente muy caro.
- b) Requiere conocimiento previo del problema a resolver y es sensible a los pesos utilizados.
- c) Algunos algoritmos son sensibles a la forma del frente Pareto (problemas con curvas cóncavas).
- d) La variación entre las diferentes soluciones encontradas depende de la eficiencia del optimizador de un solo objetivo. Podría darse el caso de encontrar siempre la misma solución o soluciones muy parecidas, en corridas múltiples.
- e) Como los optimizadores de un solo objetivo no son eficientes en búsquedas de universos discretos [DEB99], tampoco serán eficientes para optimizaciones multiobjetivo en espacios discretos.

2.5 El método del Orden Lexicográfico.

El método de Orden Lexicográfico se basa en la asignación de prioridades *a priori* a cada uno de los Objetivos a optimizar. El objetivo que posee la mayor prioridad es utilizado para ordenar las soluciones. Si ocurre un empate, el siguiente objetivo de mayor prioridad es comparado, y así sucesivamente. Todos los objetivos f_1, \dots, f_k son ordenados en orden creciente de prioridad. Esta técnica es recomendable cuando se conoce perfectamente el orden de importancia de los objetivos, porque el desempeño del algoritmo depende en gran medida de dicho ordenamiento [LAN02].

2.6 Computación Evolutiva

La generación del conjunto Pareto puede ser computacionalmente expansiva y esto a menudo resulta no factible, porque la complejidad de la aplicación subyacente impide a los métodos exactos ser aplicables [ZIT99]. Por esta razón, varias estrategias de búsqueda estocásticas como son los algoritmos evolutivos (Evolutionary Algorithms- EAs), búsqueda tabú, el recocido simulado y optimización por colonia de hormigas, fueron desarrolladas. Estos métodos usualmente no garantizan la identificación de las soluciones óptimas pero tratan de hallar una buena aproximación, i.e., un conjunto entero de vectores

soluciones objetivas que no están demasiados lejos del vector objetivo óptimo o Conjunto Pareto.

En general, un algoritmo de búsqueda estocástica consiste de tres partes:

1. una *memoria activa* que contiene la solución actualmente considerada
2. un *módulo de selección*, y
3. un *módulo de variación*, estos tres módulos son presentado en la Figura 2.1.

Acercas de la *selección*, puede distinguirse entre selección de *grupos* y *medioambiental*. La selección de grupos apunta a escoger las soluciones prometedoras para la variación y normalmente se realiza en una forma aleatoria. En contraste, selección medioambiental determina cuál de las previas soluciones guardadas y las nuevamente creadas son guardadas en la memoria interna. El *módulo de variación* toma un conjunto de soluciones y sistemáticamente o aleatoriamente modifica estas soluciones con el fin de generar potencialmente mejores soluciones. En resumen, una iteración de un optimizador estocástico incluye los consecutivos pasos: *selección de grupos*, *variación*, y *selección medioambiental*; este ciclo puede ser repetido hasta que un cierto criterio de parada sea satisfecho.

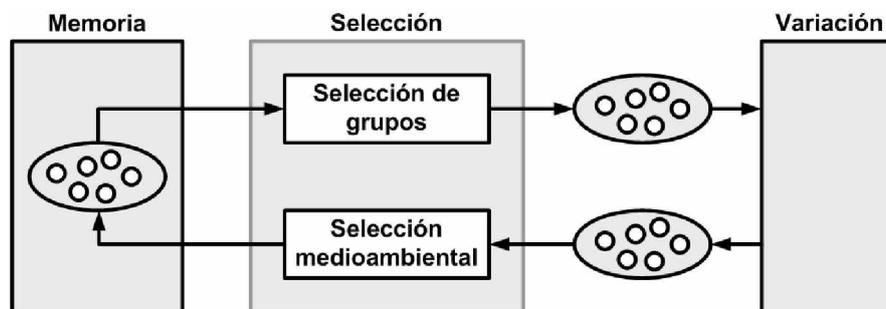


Figura 2.1. Componentes de un algoritmo general de búsqueda estocástica.

Muchas estrategias de búsqueda estocástica han sido desarrolladas originalmente para optimización de un solo objetivo y por lo tanto consideran solo una solución en un dado momento, i.e., la memoria activa contiene solo una simple solución. Como consecuencia, la selección por grupos no es necesaria y la variación es hecha por modificación de la solución candidata actual.

En contraste, un algoritmo evolutivo es caracterizado por tres rasgos:

1. un conjunto de soluciones candidatas es mantenido,

2. un proceso de selección de grupos es implementado sobre este conjunto, y
3. varias soluciones pueden ser combinadas en términos de re-combinación para generar nuevas soluciones.

Por analogía a la evolución natural, las soluciones candidatas son llamadas *individuos* y el conjunto de soluciones candidatas es llamado *población*. Cada individuo representa una posible solución, i.e., un vector de decisión, al problema que es manejado.

El proceso de selección de grupos usualmente consiste en dos estados: asignación de fitness y muestreo. Donde el fitness representa la capacidad de un individuo de adaptarse el medio en que se encuentra (el medio en que se encuentra es el problema a resolver) [ZIT99]. En el primer estado, cada individuo de la población actual es evaluado en el espacio objetivo y asignado su fitness, reflejando su calidad. Luego, un conjunto de grupos es creado por muestreo aleatorio de la población de acuerdo a los valores de fitness. Por ejemplo, dos individuos son aleatoriamente elegidos desde la población, y el que tenga mejor valor de fitness, es copiado al conjunto de grupos. Este proceso es repetido hasta que el conjunto de grupos sea completado.

Cuando el conjunto de pares es obtenido, los operadores de variación son aplicados a dicho conjunto. Con EAs, normalmente existen dos operadores de variación, a saber, el operador de cruce y el de mutación. El operador de cruce toma un cierto número de padres y crea un número predefinido de hijos por combinación de partes de los padres. Para imitar la naturaleza estocástica de la evolución, una probabilidad de cruce es asociada con este operador. Por contraste, el operador de mutación modifica individuos por cambios de pequeñas partes en el vector asociado, de acuerdo a una tasa de mutación dada. Note que debido a los efectos aleatorios, algunos individuos en el conjunto de grupos pueden no ser afectados por la variación.

Basado en estos conceptos, la evolución natural es simulada por un proceso computacional iterativo como el mostrado en la Figura 2.2. Primeramente, una población inicial es creada aleatoriamente (o de acuerdo a un esquema predefinido), el cuál es el punto de partida del proceso de evolución. Entonces, un ciclo consistente de pasos de evaluación (asignación de fitness), selección, recombinación, y/o mutación es ejecutado un cierto número de iteraciones. Cada iteración del ciclo es llamada *generación*, y a menudo un número máximo predefinido de generaciones sirve como criterio de finalización del ciclo. Sin embargo, otras condiciones pueden también ser consideradas, como la estancación en la población o la existencia de un individuo con suficiente calidad.

Al final del ciclo, los mejores individuos en la población final representan el resultado del EA [ZIT99].

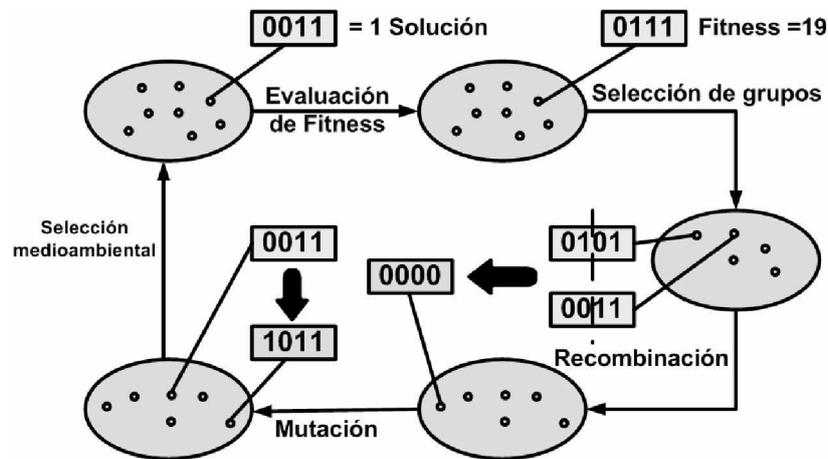


Figura 2.2 Contorno de un algoritmo evolutivo general para un problema con variable de decisión de cuatro bits.

Dada la introducción presentada sobre los conceptos claves de algoritmos basados en los EAs, se procede a presentar en la siguiente Sección una breve descripción del *Multiobjective Multicast Algorithm* (MMA), el cuál es un algoritmo evolutivo diseñado reciente y exclusivamente para la optimización del problema de enrutamiento *multicast*.

2.7 Multiobjective Multicast Algorithm

El *Multiobjective Multicast Algorithm* (MMA) fue presentado por J. Crichigno y B. Barán en el año 2004 [CRI04a, CRI04b, CRI04c]. A la fecha, es uno de los mejores algoritmos diseñados para el problema de enrutamiento *multicast* en un contexto multiobjetivo. Este algoritmo está basado en el esquema del algoritmo evolutivo de segunda generación "*Strength Pareto Evolutionary Algorithm*" (SPEA) [ZIT99], y es un representante del estado del Arte en Ingeniería de Tráfico Multicast en un contexto multiobjetivo. Por esta razón, el MMA fue tomado de la literatura para confirmar la calidad de las nuevas propuestas en ACO, mediante las comparaciones de sus resultados.

Entre las características más resaltantes del MMA puede mencionarse que optimiza cuatro objetivos simultáneamente, i.e., obtiene un conjunto frente Pareto en una sola corrida. En la Figura 2.3(a) puede observarse el algoritmo principal del MMA y brevemente se procederá a explicar cada rutina del algoritmo principal.

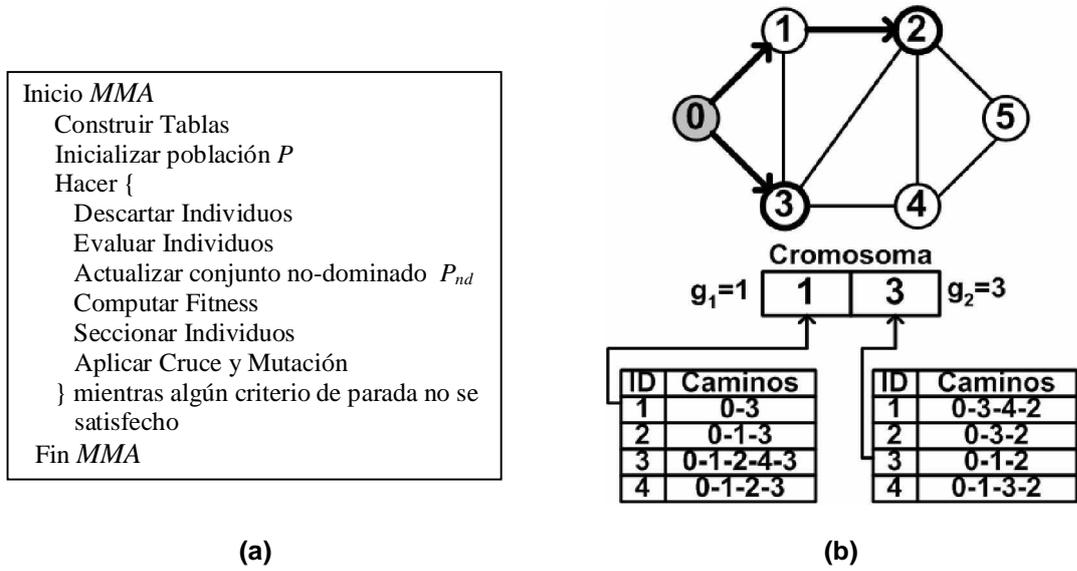


Figura 2.3. (a) Procedimiento General del MMA. (b) Relación entre Cromosoma, Genes y Tablas de Enrutamiento para el nodo fuente $s=0$ y el conjunto destino $N_r = \{2,3\}$.

En general el MMA mantiene una población P y un conjunto externo de soluciones Pareto P_{nd} . El procedimiento general se inicia con una población de individuos aleatorios y se evolucionan estos individuos a soluciones Pareto, las cuales son incluidas en P_{nd} . Las principales sub-rutinas de la Figura 2.3(a), son las siguientes:

Construir Tablas. Dado el conjunto de nodos destinos N_r , para cada nodo destino $n_i \in N_r$, una tabla de enrutamiento es construida. Esta Tabla consiste de R caminos cortos, R caminos económicos y R caminos de baja utilización de carga. Nótese que R es un parámetro del algoritmo. Cada cromosoma que representa una solución es conformada por una cadena de longitud $|N_r|$, en el cuál el elemento (*gen*) g_i representa el camino entre el nodo fuente “ s ” y un nodo destino n_i del grupo *multicast*. Como ejemplo de relación entre un cromosoma, gene y tablas de enrutamiento es mostrado en la Figura 2.3 (b) para un nodo fuentes $s = 0$ y un conjunto de destinos $N_r = \{2,3\}$.

Descartar individuos. En cada nueva población P , pueden generarse soluciones idénticas. La aplicación del operador del cruce en dos soluciones idénticas genera las mismas soluciones, por lo que la habilidad de buscar nuevas soluciones queda reducida. En consecuencia, las soluciones duplicadas son remplazadas por nuevas generadas aleatoriamente.

Evaluar individuos. Se calcula el vector objetivo de cada cromosoma de P usando las funciones de costos, retardo medio, retardo máximo y máxima utilización de carga.

Actualizar conjunto no-dominado P_{nd} . Cada solución no-dominada de P es comparada con los individuos de P_{nd} . Aquellas soluciones de P no-dominadas por alguna solución de P_{nd} son copiadas en P_{nd} . Además, si alguna solución de P_{nd} es dominada por alguna solución de P , ésta es removida de P_{nd} .

Computar Fitness. El fitness de cada individuo es obtenido usando el procedimiento del SPEA, y el operador de *Selección de Individuos* es aplicado sobre la unión del conjunto de P y P_{nd} para seleccionar buenos individuos y generar una nueva población P .

Al final de cada generación los operadores de *Cruce* y *Mutación* son aplicados a cada par de individuos. Todo el conjunto de procedimientos anteriormente explicados, se aplica hasta que algún criterio de parada se cumpla.

2.8 Optimización basada en Colonia de Hormigas

En la década de los 90 surgió un nuevo método, conocido como Optimización por Colonia de Hormigas o *Ant Colony Optimization (ACO)*, exitosamente utilizado en la resolución de varios problemas combinatoriales mono-objetivo. ACO es una metaheurística basada en algunos comportamientos de la colonia de hormigas reales [DOR91]. Ya en años más recientes se ha dado los primeros pasos en la resolución de problemas multiobjetivo (*MOPs*) basados en estos conceptos, las cuáles también tuvieron gran éxito en varios problemas combinatorios como es el caso del Problema de Cajero Viajante Bi-objetivo [GAR04].

Debido al reciente desarrollo de ACO, no existe en la literatura un gran desarrollo teórico sobre el éxito de éstas metaheurísticas, como es reconocido por Dorigo y Stützle [DOR01, STÜ02], i.e., aunque ACO ha mostrado experimentalmente ser altamente efectiva, sólo limitado conocimiento está disponible para explicar por qué la metaheurística ACO es tan exitosa [DOR01]. Recién en el 2004 Gómez y Barán realizaron uno de los primeros estudios teóricos basados en pruebas experimentales y dan las razones del éxito de ACO en el Problema del Cajero Viajante (*Traveling Salesman Problem - TSP*) en un contexto mono-objetivo [GOM04a].

Es importante destacar que en el presente estado del arte, la resolución del problema de enrutamiento *multicast* multiobjetivo aplicando ACO no ha sido tratada aún. Solo

existen algunos trabajos que proponen la resolución del problema de enrutamiento *multicast* mono-objetivo sujeto a varias restricciones [CHU02, LIU03, DAS02].

Debido a lo mencionado, este trabajo tiene un aporte significativo al estado del arte al proponer por primera vez la resolución del problema de enrutamiento *multicast* multiobjetivo aplicando ACO. Además, cada objetivo es optimizado simultáneamente, i.e. no se utilizan los métodos de suma de pesos ni orden lexicográfico.

Por otra parte, en este Capítulo no se entra en mayores detalles sobre ACO debido a que esto será tratado en un Capítulo posterior por ser unos de los temas de mayor interés de este trabajo, teniendo en cuenta que la presente propuesta de solución del problema de enrutamiento *multicast* multiobjetivo es basada justamente en ACO.

2.9 Resumen del Capítulo

En el presente Capítulo se ha dado la definición formal de un problema de optimización multiobjetivo y se han definido términos y conceptos que serán utilizados en los posteriores Capítulos. Luego, se ha presentado brevemente los conceptos de optimización basada en algunos métodos tradicionales (con toma de decisión *a priori*) como también el de Computación Evolutiva (con toma de decisión *a posteriori*), exitosamente utilizado en la resolución de MOPs. Además se dio una breve introducción a la novedosa metaheurística llamada *Ant Colony Optimization* – ACO (con toma de decisión *a posteriori*). Esta ha demostrado experimentalmente en varios problemas combinatorios (SOPs, MOPs) ser una poderosa herramienta, teniendo cualidades deseables para la resolución de MOPs combinatorios. Habiendo introducido estos conceptos, en el Capítulo 3 se formula el problema de enrutamiento *multicast* como un MOP.

Capítulo 3

Formulación Matemática

3.1 Introducción

Este Capítulo propone el Problema de Enrutamiento Multicast en un contexto MOP, en el cual debe ser construido un árbol *multicast* para un grupo (nodo fuente y destinos) dado, minimizando:

- 1- la utilización máxima de los enlaces del árbol,
- 2- el costo del árbol,
- 3- el retardo máximo de extremo a extremo, y
- 4- el retardo medio.

En consecuencia, el presente Capítulo queda organizado de la siguiente manera: en la Sección 3.2 se da el modelo matemático del problema de enrutamiento *multicast* en redes de computadoras. Posteriormente, de forma a aclarar el significado de la notación matemática, la Sección 3.3 presenta un ejemplo de un problema de enrutamiento *multicast* multiobjetivo.

3.2 Modelo Matemático

Una red de computadoras puede ser modelada como un grafo dirigido $G = (V, E)$, donde V es el conjunto de vértices y E el conjunto de arcos. Los vértices del grafo representan nodos de la red, y los arcos representan los enlaces entre los nodos.

Sea:

- $(i,j) \in E$: enlace entre los nodos i y j ; $i, j \in V$.
- $z_{ij} \in \mathfrak{R}^+$: capacidad del enlace (i,j) , medido en Mbps.
- $c_{ij} \in \mathfrak{R}^+$: costo del enlace (i,j) .
- $d_{ij} \in \mathfrak{R}^+$: retardo (delay) del enlace (i,j) .
- $t_{ij} \in \mathfrak{R}^+$: tráfico actual fluyendo a través del enlace (i,j) , medido en Mbps.
- $s \in V$: nodo origen del grupo *multicast*.
- $N_r \subseteq V - \{s\}$: conjunto de nodos destinos del grupo *multicast*.
- $n_i \in N_r$: uno de los $|N_r|$ destinos, donde $|\cdot|$ indica cardinalidad.

- $\phi \in \mathfrak{R}^+$: demanda de tráfico del grupo *multicast*, en Mbps.
- $T(s, N_r)$: árbol *multicast* con origen en s y conjunto de nodos destino N_r .
- $p_T(s, n_i) \subseteq T(s, N_r)$: camino (*path*) que conecta el nodo fuente s y el nodo destino $n_i \in N_r$. Note que $T(s, N_r)$ representa una solución x del Capítulo 2.2
- $d(p_T(s, n_i))$: retardo del camino $p_T(s, n_i)$, dado por:

$$d(p_T(s, n_i)) = \sum_{(i,j) \in p_T(s, n_i)} d_{ij} \quad (3.1)$$

Usando la notación arriba definida, el problema de enrutamiento *multicast* puede ser definido como un MOP que trata de hallar el árbol *multicast* $T(s, N_r)$, que minimiza las siguientes funciones objetivos:

1- Utilización máxima de los enlaces del árbol:

$$\alpha_T = \text{Max}_{(i,j) \in T} \left\{ \frac{\phi + t_{ij}}{z_{ij}} \right\} \quad (3.2)$$

2- Costo del árbol:

$$C_T = \phi \sum_{(i,j) \in T} c_{ij} \quad (3.3)$$

3- Retardo máximo de extremo a extremo:

$$DM_T = \text{Max}_{n_i \in N_r} \{d(p_T(s, n_i))\} \quad (3.4)$$

4- Retardo medio:

$$DA_T = \frac{1}{|N_r|} \sum_{n_i \in N_r} d(p_T(s, n_i)) \quad (3.5)$$

sujeto a restricciones de capacidad en los enlaces:

$$\phi + t_{ij} \leq z_{ij} ; \forall (i, j) \in T(s, N_r) \quad (3.6)$$

esto es: $\alpha_T \leq 1$.

La Ecuación 3.2 da la utilización máxima de los enlaces del árbol. La minimización de esta métrica implica que son preferidos aquellos árboles con la menor utilización máxima de sus enlaces, de forma a balancear la carga sobre la red [SEO02, DON04].

La Ecuación 3.3 da el costo total del árbol *multicast*. Formulado de esta manera, esta métrica puede tener más de un significado. Si el costo representa alguna cantidad monetaria para el proveedor de servicio, entonces sería deseable proveer un servicio *multicast* a un costo mínimo. Por ejemplo, el costo de cada enlace puede tener un significado monetario a pagar por bit de información por unidad de tiempo. En este caso, la función objetivo dada por la Ecuación 3.3 da el costo total a pagar por bit de información por unidad de tiempo para llevar a cabo la transmisión *multicast*. Por otra parte, si el costo de cada enlace tiene un valor unitario, el costo total es el consumo total de ancho de banda del grupo. En cualquiera de los casos, la minimización de esta métrica es siempre deseable. Es claro que formular el problema con enlaces cuyos costos pueden tener valores arbitrarios positivos provee generalidad al problema.

La Ecuación 3.4 define el retardo máximo de extremo a extremo. Debido al creciente número de aplicaciones *multicast* sensibles a esta métrica, es de suma importancia que los árboles *multicast* sean construidos minimizando esta función objetivo. A diferencia de varios trabajos publicados [KOM93a, KOM93b, RAV98, ZHE01, ARA02] en los cuales el retardo máximo es considerado como una restricción, se debe destacar que existen recientes trabajos que formulan este parámetro como una función a minimizar [CRI04a, CRI04b, CRI04c, TAL04a]. En el presente trabajo también se formula explícitamente como una función objetivo a minimizar, donde las ventajas de este enfoque serán resaltadas en los Ejemplo 3.1 y 3.2.

La Ecuación 3.5 define el retardo medio a los nodos destinos. Es importante notar que, si bien las Ecuaciones 3.3 y 3.4 definen funciones objetivos relacionadas, tienen distintos significados. El retardo máximo de extremo a extremo indica cuál es el peor caso de todos los retardos de nodo fuente a destino, y el retardo medio da una apreciación en promedio de los retardos del nodo fuente a todos los destinos.

Por último, la Ecuación 3.6 restringe el tráfico máximo en cada enlace a la capacidad del canal.

3.3 Ejemplos de Problemas Multicast

Ejemplo 3.1. Se propone el siguiente ejemplo de manera a comprender mejor la formulación propuesta [TAL04c]. En el mismo se han utilizado tres de las cuatro métricas mencionadas anteriormente. Considere la red de computadoras mostrada en la Figura 3.1 [BAR00], en donde los números sobre cada enlace (i, j) corresponden a valores de retardo d_{ij} y costo c_{ij} . El retardo está dado en milisegundos. Se genera una solicitud *multicast* en la que el nodo $s=8$ desea transmitir información a los nodos $N_r=\{0, 3, 11, 12\}$.

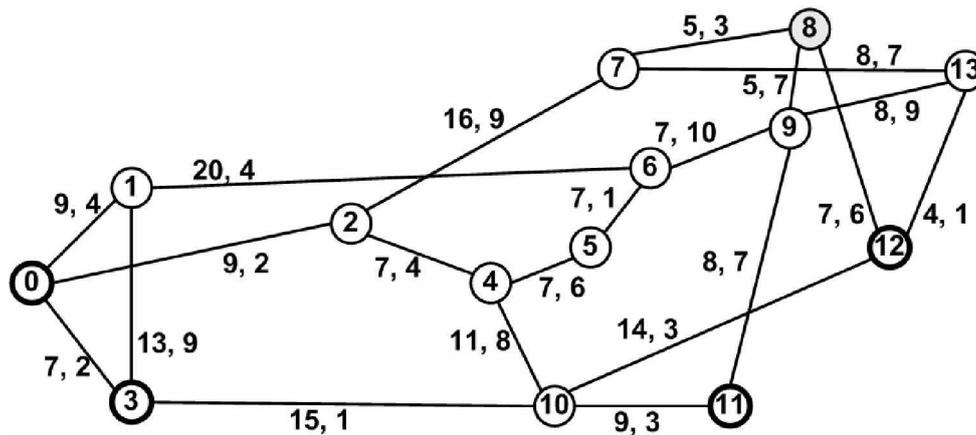


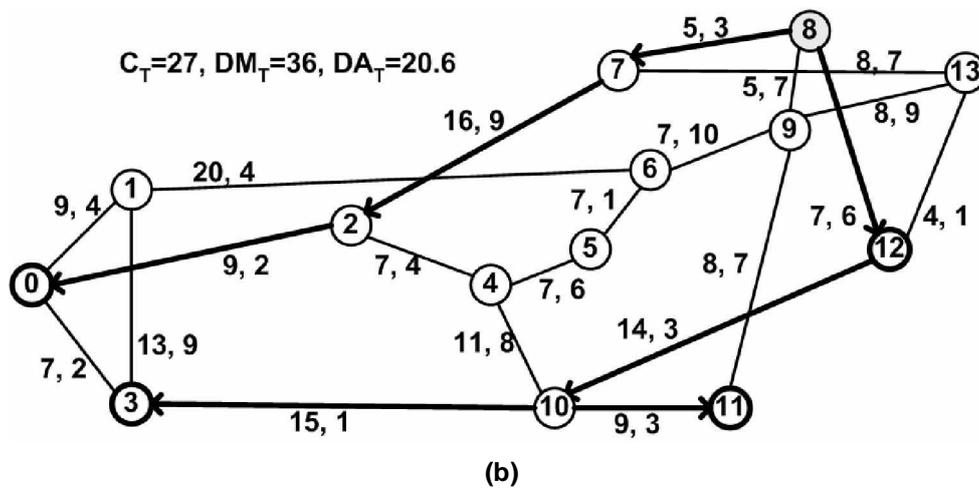
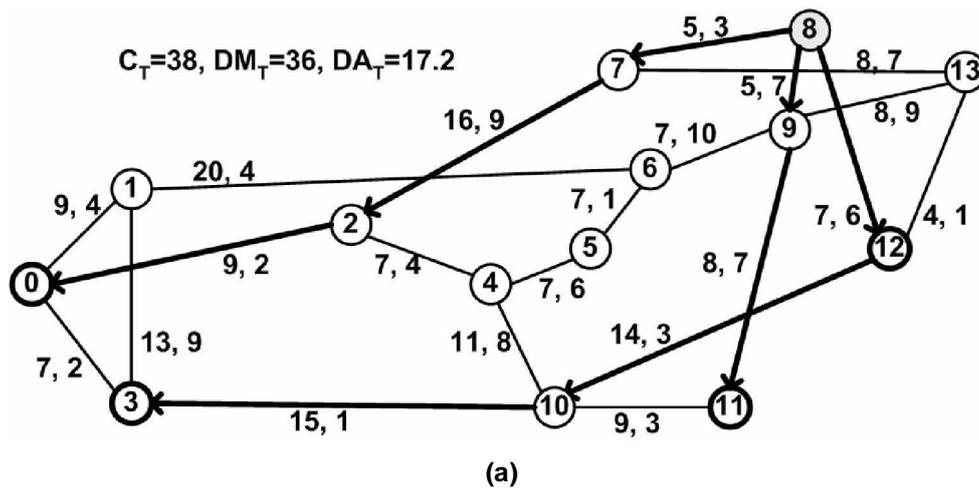
Figura 3.1. Red de la NSF. Cada enlace (i, j) tiene asignado sus correspondientes valores de retardo d_{ij} y costo c_{ij} . Se genera una solicitud *multicast* dada por $s = 8$ y $N_r = \{0, 3, 11, 12\}$.

Una forma tradicional de enrutar la demanda solicitada es a través del árbol del camino más corto, sobre todo para el caso en que la aplicación sea sensible al retardo máximo. Una alternativa es entonces la construcción de un árbol *multicast* que esté compuesto por los caminos más cortos (Shortest Path Tree - SPT), en términos de retardo, a los nodos destinos. El SPT podría encontrar una de las soluciones mostradas en las figuras 3.2 (a) y (b), que poseen el menor retardo máximo. Nótese que la solución (b) posee un valor mucho menor de costo que la solución (a), el SPT podría encontrar cualquiera de las soluciones, no nos asegura que encontrará la solución con menor costo.

Debido a que el SPT solo considera una métrica, el árbol *multicast* puede tener un costo muy elevado, o como en el caso anterior cuando tenemos dos o más soluciones con el mismo retardo máximo, el SPT no garantiza encontrar la solución con menor costo o retardo medio. Por lo tanto, se podrían hallar otras soluciones que, sacrificando el retardo máximo tengan un menor costo o que garanticen el menor costo para un determinado límite de retardo máximo. Para esto se podría usar un algoritmo basado en restricciones como KPP [KOM93a, Kom93b], de forma a optimizar el costo del árbol. KPP y otros

algoritmos basados en restricciones [RAV98, ZHE01, ARA02] requieren un valor de retardo máximo definido *a priori*. Entonces, suponga que el grupo *multicast* desea un árbol con un valor DM_T menor a 38 ms. En dicho caso, KPP hallaría la solución mostrada en la Figura 3.2 (b). Note que el KPP en este caso garantiza encontrar la solución con el menor costo.

El problema de los algoritmos basados en restricciones es la necesidad del valor de retardo máximo dado *a priori*, pues se pueden descartar soluciones de muy bajo costo con valores DM_T apenas superiores al máximo permitido. Si comparamos las soluciones mostradas en las Figuras 3.2 (a) y (e), se puede notar que sacrificando un poco más el retardo máximo en la solución (a), podemos conseguir una solución con un menor costo y un valor similar de retardo medio. Esta podría ser una solución muy interesante que el KPP no encontraría en el caso de considerar la restricción de retardo máximo como inferior a 36.



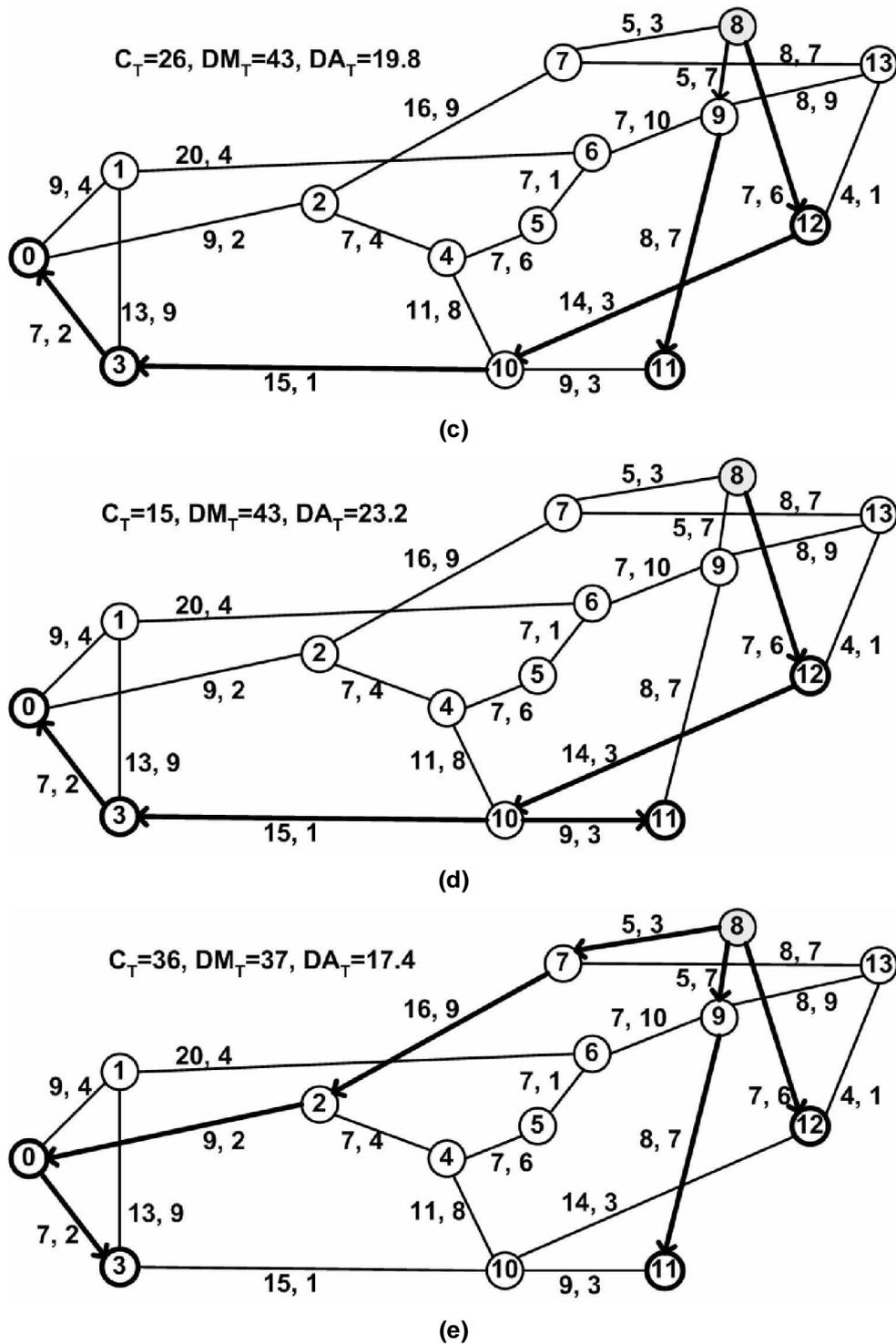


Figura 3.2. Cinco soluciones alternativas para el grupo *multicast* para el Ejemplo 3.1.

Note que las cinco alternativas son soluciones no dominadas. De hecho, por búsqueda exhaustiva, puede comprobarse que estas 5 soluciones forman parte del

conjunto Pareto del problema. Por lo tanto, son soluciones óptimas en un sentido multiobjetivo.

Un conjunto de soluciones óptimas como este puede ser de gran utilidad en la práctica, pues la elección de la mejor opción puede ser tomada *a posteriori* y según las necesidades de cada caso. De la formulación matemática se puede concluir que esta gran variedad de soluciones óptimas solo puede ser obtenida si los objetivos son tratados en forma independiente. Es decir, no forman una función objetivo escalar combinada a través de una combinación lineal, ni tampoco son tratados como simples restricciones. De esta forma, el enfoque multiobjetivo elude el problema de definir los pesos relativos para cada objetivo o hacer restricciones *a priori*, hallando todo un conjunto de soluciones óptimas. Para clarificar los conceptos de dominancia y el cálculo de las métricas propuestas, se presenta el Ejemplo 3.2 que sigue.

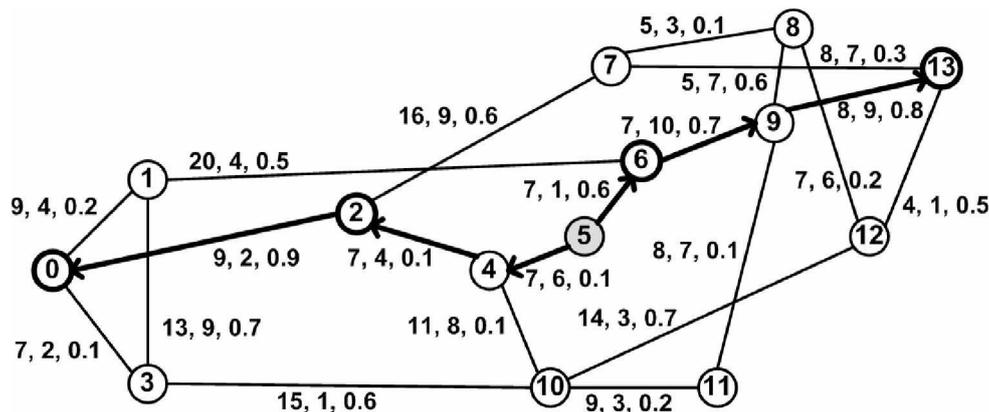


Figura 3.3. Red de la NSF. Cada enlace (i,j) tiene asignado sus correspondientes valores de retardo d_{ij} , costo c_{ij} y tráfico actual t_{ij} . Los Objetivos son $\alpha_T=0.73$, $C_T= 6.4$, $DA_T=16.5$ y $DM_T=23$ para una solicitud *multicast* dada por $s = 5$ y $N_r = \{0, 2, 6, 13\}$.

Ejemplo 3.2. En este ejemplo son utilizadas las cuatro métricas definidas en la Sección anterior para la misma red de computadoras NSF [BAR00], en donde los números sobre cada enlace (i, j) corresponden a valores de retardo d_{ij} , costo c_{ij} y tráfico actual t_{ij} . El retardo está dado en milisegundos y el tráfico actual en Mbps. Suponiendo una demanda *multicast* $\phi = 0.2$ Mbps, para el siguiente grupo *multicast* $s=5$ y $N_r=\{0, 2, 6, 13\}$. La Figura 3.3 muestra la construcción de *un árbol multicast* (T) y la Tabla 3.1 presenta los objetivos calculados para la solución de la Figura 3.3. Para este ejemplo $T \equiv T(s, N_r)$ por simplicidad.

Tabla 3.1. Objetivos Calculados para la solución presentada en la Figura 3.3.

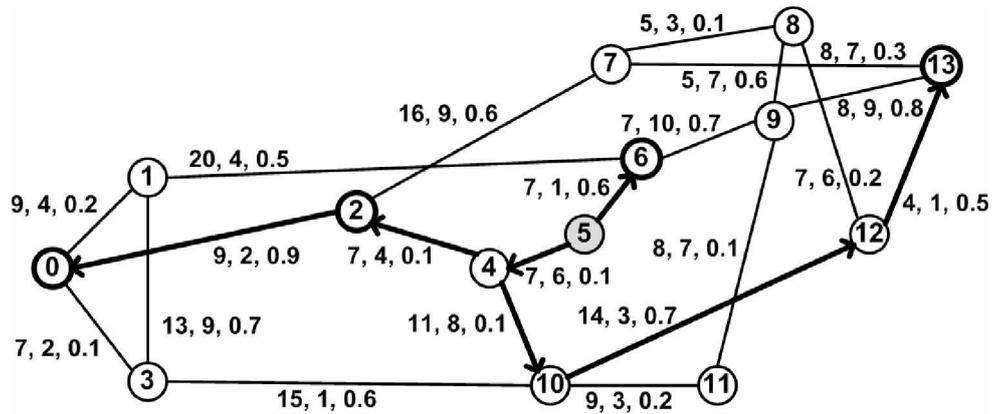
<i>Árbol</i>						
(i,j)	$(5,4)$	$(4,2)$	$(2,0)$	$(5,6)$	$(6,9)$	$(9,13)$
d_{ij}	7	7	9	7	7	8
c_{ij}	6	4	2	1	10	9
t_{ij}	0.1	0.1	0.9	0.6	0.7	0.8
$(\phi+t_{ij})/z_{ij}$	0.2	0.2	0.73	0.53	0.6	0.53
$d(p_T(5,4))$	$d_{5,4} = 7$					
$d(p_T(5,2))$	$d_{5,4} + d_{4,2} = 7+7=14$					
$d(p_T(5,0))$	$d_{5,4} + d_{4,2} + d_{2,0} = 7+7+9=23$					
$d(p_T(5,6))$	$d_{5,6} = 7$					
$d(p_T(5,13))$	$d_{5,6} + d_{6,9} + d_{9,13} = 7+7+8=22$					
<i>Métricas del árbol</i>						
α_T	0.73					
C_T	$0.2*(6+4+2+1+10+9) =$ 6.4					
DA_T	$(7+14+16+7+22)/4 =$ 16.5					
DM_T	23					

Para el mismo ejemplo, la Figura 3.4 presenta en (a), (b) y (c) tres diferentes alternativas de solución, para el mismo grupo *multicast*, de forma a clarificar los conceptos de no-dominancia. Note que cada una de las tres soluciones tiene al menos un objetivo que es mejor que las otras soluciones, por tal motivo, estas soluciones son conocidas como no-comparables. Observe que la solución dada en la Figura 3.4 (a) tiene el menor costo ($C_T=5.0$) entre las tres soluciones, por su parte la solución de la Figura 3.4 (b) tiene una utilización máxima de carga ($\alpha_T=0.6$) menor con solo 0.6 y la última solución dada en la Figura 3.4 (c) posee un retardo medio ($DA_T=19.75$) menor al resto. Estas tres soluciones tienen el mismo retardo máximo de extremo a extremo.

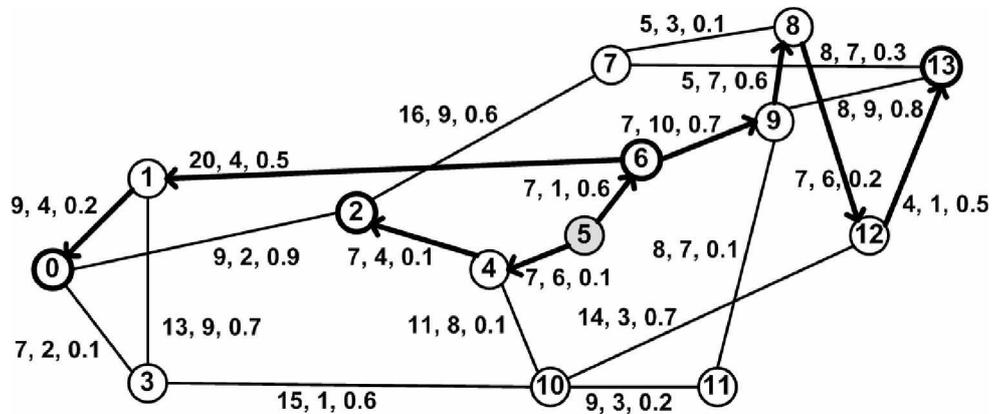
Las tres soluciones presentadas en las Figuras 3.4 pertenecen al conjunto óptimo Pareto. Sin embargo, el conjunto Pareto para este problema en particular esta conformado también por otras soluciones que son presentadas en la Tabla 3.2. Además, en la Tabla 3.3 se da el correspondiente conjunto frente Pareto o Vectores Objetivos, i.e., las métricas de las soluciones que están dadas en la Tabla 3.3.

Cada solución en la Tabla 3.2 es conformada por los enlaces (i,j) y los números en negrita indican los nodos destinos del grupo *multicast*.

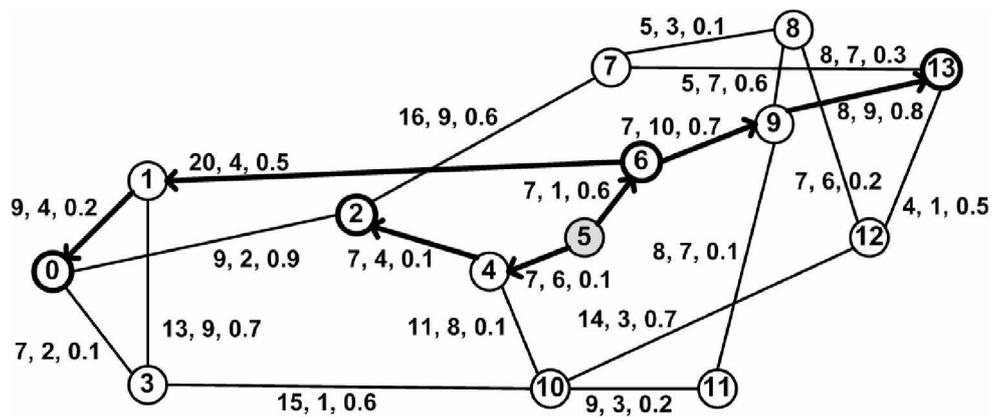
Observe que la solución S_1 dada en las Tablas 3.3 y 3.4 corresponde al árbol de la Figura 3.4 (a), la solución S_2 corresponde al árbol de la Figura 3.4 (b) y la solución S_3 corresponde al árbol de la Figura 3.4 (c).



(a) $\alpha_T=0.73, C_T=5.0, DA_T=20.0, DM_T=36.$



(b) $\alpha_T=0.6, C_T=8.6, DA_T=21.75, DM_T=36.0$



(c) $\alpha_T=0.67, C_T=7.6, DA_T=19.75, DM_T=36.0$

Figura 3.4. Red NSF. (a), (b) y (c) muestran diferentes alternativas de árboles para el mismo grupo *multicast* con $s = 5, N_r = \{0, 2, 6, 13\}$ y $\phi = 0.2$ Mbps. Las tres alternativas son soluciones no-comparables.

Tabla 3.2. Conjunto Pareto Óptimo del Ejemplo 3.2.

<i>Conjunto de enlaces para cada árbol</i>	
S_1	(5,4), (5,6), (4,2), (4,10), (2,0), (10,12), (12,13)
S_2	(5,6), (5,4), (6,1), (6,9), (4,2), (1,0), (9,8), (8,12), (12,13)
S_3	(5,6), (5,4), (6,1), (6,9), (4,2), (1,0), (9,13)
S_4	(5,6), (5,4), (6,9), (4,2), (9,13), (2,0)
S_5	(5,6), (5,4), (6,1), (4,2), (4,10), (1,0), (10,12), (12,13)
S_6	(5,6), (5,4), (6,1), (4,2), (4,10), (1,0), (10,12), (12,13)
S_7	(5,6), (6,1), (1,0), (0,3), (0,2), (3,10), (10,12), (12,13)
S_8	(5,6), (5,4), (6,1), (4,2), (1,0), (2,7), (7,13)
S_9	(5,6), (5,4), (4,2), (4,10), (10,12), (10,3), (12,13), (3,0)
S_{10}	(5,6), (5,4), (4,10), (10,3), (10,12), (3,0), (12,13), (0,2)

Tabla 3.3. Vectores Objetivos (α_T , C_T , DA_T , DM_T) del Ejemplo 3.2.

	α_T	C_T	DA_T	DM_T
S_1	0.73	5	20	36
S_2	0.6	8.6	21.75	36
S_3	0.67	7.6	19.75	36
S_4	0.73	6.4	16.5	23
S_5	0.73	4	26.75	63
S_6	0.6	6.2	23.25	36
S_7	0.73	3.6	41	76
S_8	0.53	7	23.75	38
S_9	0.6	5.2	24.25	4
S_{10}	0.73	4.8	33	49

Dado que el problema del árbol *multicast* en una red de computadoras es un problema NP-completo [KOM93a], el problema formulado en esta sección también lo es. Esencialmente, la intratabilidad del problema se debe a la existencia de un número de árboles que crece exponencialmente con el número de nodos de la red, y las soluciones óptimas solo pueden hallarse por búsqueda exhaustiva [KOM93a]. Evidentemente, dicha técnica no tiene practicidad alguna, por lo que se necesitan métodos alternativos que no exijan la exploración completa del espacio de búsqueda. En general, estos métodos no tienen por objetivo hallar todas las soluciones del conjunto Pareto óptimo, sino simplemente encontrar una aproximación lo suficientemente buena.

Teniendo en cuenta la complejidad del espacio de búsqueda, su tamaño y las recientes propuestas basadas en *Ant Colony Optimization* (ACO) [DOR91], resulta natural el postular la solución del problema de Ingeniería de Tráfico Multicast, utilizando ACO. Cabe destacar que otras propuestas basadas en algoritmos evolutivos multiobjetivo [ZIT99] ya fueron implementadas [CRI04a, CRI04b, CRI04c, TAL04a]. En cambio ACO se

ha implementado por primera vez en este trabajo, debido a que estos algoritmos no han sido todavía suficientemente estudiados en aplicaciones de redes de computadoras. En consecuencia, el presente trabajo es un aporte original y útil tanto en el estudio de ACOs como en la solución del problema de Ingeniería de Tráfico Multicast.

La ventaja principal de los ACOs es su paralelismo inherente, pues son capaces de mantener todo un conjunto de soluciones y por lo tanto, pueden obtener varios miembros del conjunto Pareto óptimo en una única corrida. Además, no requieren de mayor información sobre el espacio de búsqueda; pues basta con especificar los elementos del problema, los objetivos y la manera de calcularlos. Tampoco presentan dificultades relacionadas con la complejidad del espacio de búsqueda, es decir, son bastante generales y robustos en problemas combinatoriales [DOR96].

3.4 Resumen del Capítulo

En el presente Capítulo se ha formulado el problema de enrutamiento *multicast* como un Problema de Optimización Multiobjetivo, donde las funciones objetivos son formuladas explícitamente y deben ser optimizadas simultánea e independientemente. El siguiente Capítulo presenta una introducción a ACO, así como las nuevas propuestas basadas en ACO presentadas en este trabajo.

Capítulo 4

Colonia de Hormigas

4.1. Introducción

En este Capítulo se presenta los conceptos básicos de la Optimización basada en Colonias de Hormigas (*Ant Colony Optimization* - ACO), así como diversos algoritmos basados en ACO para el problema de Ingeniería de Tráfico Multicast.

El presente Capítulo queda organizado de la siguiente manera: en la Sección 4.2 se explica el comportamiento de las hormigas reales en la búsqueda de alimentos. Posteriormente, en la Sección 4.3 se presenta la Optimización basada en Colonia de Hormigas y en la Sección 4.4 el primer algoritmo de optimización estándar de ACO es dado. En la Sección 4.5 se presentan las nuevas propuestas basadas en ACO para la resolución del enrutamiento *multicast* multibojetivo, y finalmente se expone un algoritmo basado en colonia de hormigas para la construcción de una solución o árbol *multicast*.

4.2. Comportamiento de Búsqueda de Alimentos de las Hormigas Reales

Las hormigas son insectos sociales que viven en colonias y que debido a su colaboración mutua, son capaces de mostrar comportamientos complejos y realizar tareas difíciles desde el punto de vista de una hormiga individual. Un aspecto interesante del comportamiento de muchas especies de hormigas es su habilidad para encontrar los caminos más cortos entre su hormiguero y las fuentes de alimento. Este hecho es especialmente interesante si se tiene en cuenta que muchas de las especies de hormigas son casi ciegas, lo que evita el uso de pistas visuales [DOR91].

Mientras que se mueven entre el hormiguero y la fuente de alimento, algunas especies de hormigas depositan una sustancia química denominada *feromona* (una sustancia que puede “olerse”). Si no se encuentra ningún rastro de feromona, las hormigas se mueven de manera básicamente aleatoria, pero cuando existe feromona depositada, tienen mayor tendencia a seguir el rastro. De hecho, los experimentos realizados por biólogos han demostrado que las hormigas prefieren de manera probabilística los caminos marcados

con una concentración superior de feromona [ALO04]. En la práctica, la elección entre distintos caminos toma lugar cuando varios caminos se cruzan. Entonces, las hormigas eligen el camino a seguir con una decisión probabilística sesgada por la cantidad de feromona: cuanto más fuerte es el rastro de feromona, mayor es la probabilidad de elegirlo. Puesto que las hormigas depositan feromona en el camino que siguen, este comportamiento lleva a un proceso de auto-refuerzo que concluye con la formación de rastros señalados por una concentración de feromona elevada. Este comportamiento permite además a las hormigas encontrar los caminos más cortos entre su hormiguero y la fuente del alimento [ALO04]. Nótese que las hormigas solo se comunican de manera indirecta, a través de modificaciones del espacio físico que perciben. Esta forma de comunicación se denomina *Estimergia Artificial* [ALO04].

Como se ilustra en la Figura 4.1 este mecanismo permite a las hormigas encontrar el camino más corto. Inicialmente todas las hormigas siguen el camino directo entre el nido y la fuente de alimento. Si algún obstáculo es puesto en ese camino directo, las hormigas eligen aleatoriamente cualquier camino. Al transcurrir el tiempo y mientras que las hormigas están recorriendo los caminos más prometedores, estos van recibiendo una cantidad superior de feromona. Esto ocurre gracias a que al ser los caminos más cortos, las hormigas que los siguen consiguen encontrar la comida más rápidamente, por lo que comienzan su viaje de retorno antes. Entonces, en el camino más corto habrá un rastro de feromona ligeramente superior y, por lo tanto, las decisiones de las siguientes hormigas estarán dirigidas en mayor medida a dicho camino. Además, este camino irá recibiendo una proporción mayor de feromona gracias a las hormigas que vuelven por él y no por el camino más largo. Este proceso hace que la probabilidad de que una hormiga escoja el camino más corto aumente progresivamente y que al final, el recorrido de la colonia converja al más corto de todos los caminos posibles.

Esta convergencia se complementa con la acción del entorno natural que provoca que la feromona se evapore transcurrida un cierto tiempo. Así, los caminos menos prometedores pierden progresivamente feromona porque son visitados cada vez por menos hormigas [ALO04].

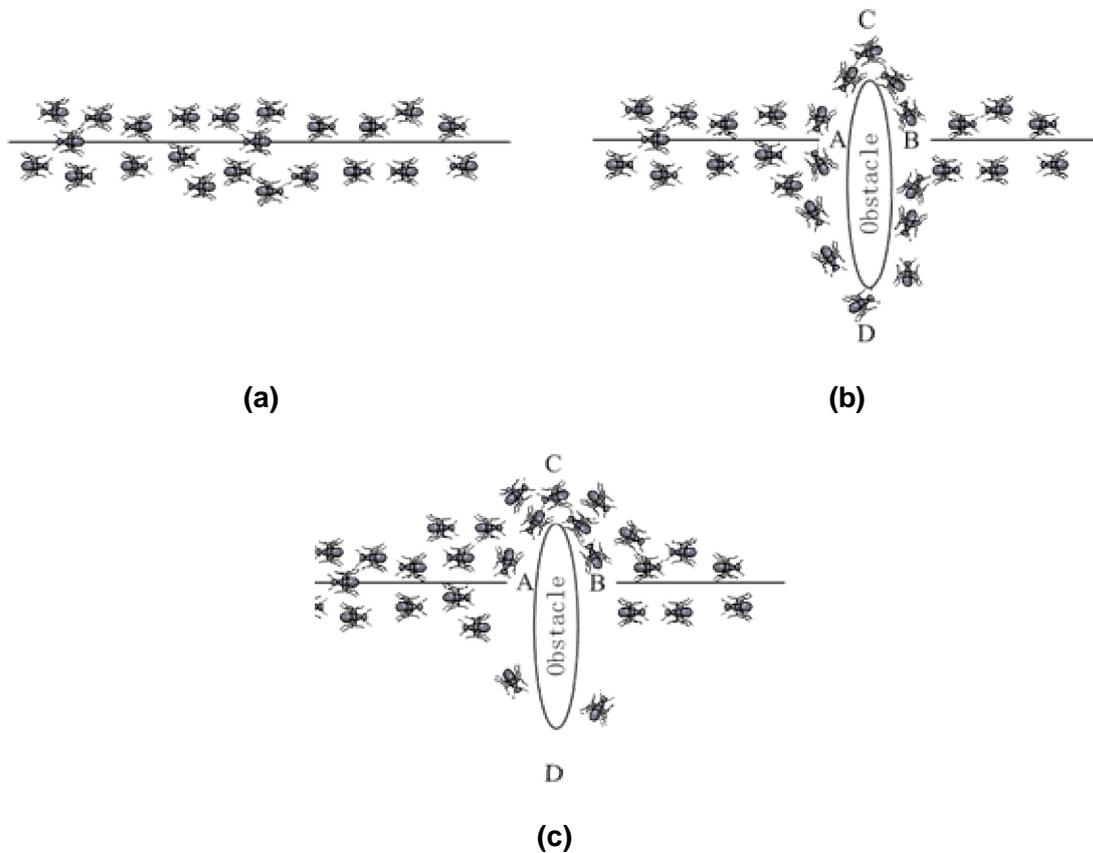


Figura 4.1. Comportamiento de las hormigas en la búsqueda de su alimento entre el nido y su fuente de alimentación. (a) Inicialmente todas las hormigas van por un camino directo, (b) eligen aleatoriamente los caminos alternativos en ausencia de feromonas entre los distintos caminos alternativos, y (c) los altos niveles de concentración de feromonas atraen a las hormigas por el camino más prometedor.

4.3. Optimización por Colonia de Hormigas

Los algoritmos de Optimización por Colonia de Hormigas (*Ant Colony Optimization* - ACO) se inspiran directamente en el comportamiento de las colonias reales de hormigas para solucionar problemas de optimización combinatoria. Se basan en una colonia de hormigas artificiales, esto es, unos agentes computacionales simples que trabajan de manera cooperativa y se comunican mediante rastros de feromona artificiales [DOR99].

Los algoritmos ACO son esencialmente *algoritmos constructivos*: en cada iteración del algoritmo, cada hormiga construye una solución al problema recorriendo un grafo de construcción. Cada arista del grafo, que representa los posibles pasos que la hormiga

puede dar, tiene asociada dos tipos de información que guían el movimiento de la hormiga:

1. *Información heurística*, que mide la preferencia heurística de moverse desde el nodo i hasta el nodo j , o sea, de recorrer el enlace (i,j) . Esta información heurística se denota como η_{ij} y es conocida como visibilidad. Las hormigas no modifican esta información durante la ejecución del algoritmo.
2. *Información de los rastros de feromona artificiales*, que mide la “deseabilidad aprendida” del movimiento de i a j . Imita a la feromona real que depositan las hormigas naturales. Esta información se modifica durante la ejecución del algoritmo dependiendo de las soluciones encontradas por las hormigas. Se denota por τ_{ij} y se la conoce como *Feromona*.

El modo de operación básico de un algoritmo ACO es como sigue: dadas m hormigas (artificiales) de la colonia, estas se mueven concurrentemente y de manera asíncrona, a través de los estados adyacentes del problema (que puede representarse en forma de grafo, con pesos). Este movimiento se realiza siguiendo una regla de transición que está basada en la información local disponible en las componentes (nodos). Esta información local incluye la información heurística y memorística (rastros de feromona) para guiar la búsqueda. Al moverse por el grafo de construcción, las hormigas construyen incrementalmente soluciones. Opcionalmente, las hormigas pueden depositar feromona cada vez que crucen un arco o conexión mientras que construyen la solución, esto es conocido como actualización *en línea, paso a paso*, de los rastros de feromona. Una vez que cada hormiga ha generado una solución, ésta se evalúa y puede depositarse una cantidad de feromona que es función de la calidad de su solución (*actualización en línea de los rastros de feromona*). Esta información guiará la búsqueda de las otras hormigas de la colonia, en el futuro.

Además, el modo de operación genérico de un algoritmo de ACO incluye la evaporación de los rastros de feromona. La evaporación de feromona la lleva a cabo el entorno y se usa como un mecanismo que evita el estancamiento en la búsqueda y permite que las hormigas busquen y exploren nuevas regiones del espacio.

4.4. Enfoque Estándar de ACO

Con el objetivo de proveer un marco de referencia para el diseño de nuevas versiones de algoritmos, la metaheurística ACO fue propuesta posteriormente a las primeras versiones [DOR99]. Aun cuando existen varias versiones de ACO, se presenta a continuación lo que se considera un enfoque estándar [GOM04b].

ACO usa una matriz de feromonas $\tau = \{\tau_{ij}\}$ para almacenar el nivel de feromonas de cada arco de un grafo. Esta matriz de feromonas es inicializada con los valores $\tau_{ij} = \tau_0 \forall (i,j)$, donde $\tau_0 > 0$. También explota información heurística usando $\eta_{ij} = 1/d_{ij}$, donde d_{ij} representa el retardo asociado al enlace (i,j) . Parámetros α y β definen la influencia relativa entre los rastros de feromonas y la visibilidad en el momento de seleccionar un nuevo arco, como se verá en la ecuación (4.1).

En cada iteración del algoritmo, cada una de las θ hormigas de la colonia construye un árbol completo T_k , creándose por lo tanto un conjunto de θ recorridos $T' = \{T_k\}$. Para la construcción de cada árbol se realiza el procedimiento siguiente. En forma general, al inicio es tomado el nodo fuente s del grupo *multicast* como nodo de partida i , luego es seleccionado un nodo vecino j , el enlace formado (i, j) pasa a ser parte de la solución en construcción. Luego, otra vez es seleccionado otro nodo de partida i de entre los nodos ya adheridos a la solución y se selecciona otro nodo j aún no visitado para formar el nuevo enlace, esto se realiza iterativamente hasta llegar a todos los nodos destinos N_r del grupo *multicast*. Mientras es visitado un nodo i , N_i representa el conjunto de nodos todavía no visitados y la probabilidad de elegir un nodo j desde un nodo i es definida como:

$$p_{ij} = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{\forall g \in N_i} \tau_{ig}^\alpha \cdot \eta_{ig}^\beta} & \text{Si } j \in N_i \\ 0 & \text{otro caso} \end{cases} \quad (4.1)$$

Luego se aplica evaporación de feromonas para todo arco $(i, j) \in E$ de acuerdo a $\tau_{ij} = (1 - \rho) \cdot \tau_{ij}$, donde el parámetro de entrada $\rho \in (0,1]$ determina la tasa de evaporación. Finalmente, cada hormiga k deposita feromonas en los arcos componentes del árbol T_k construido por ella, de costo C_{T_k} , con acuerdo a $\tau_{ij} = \tau_{ij} + 1/C_{T_k}$, i.e., que mientras más corto es el recorrido de una hormiga, mayor cantidad de feromonas recibirán los arcos componentes de ese recorrido. En general, los arcos que son utilizados por muchas hormigas y los arcos componentes de recorridos de menor longitud recibirán más

feromonas y por lo tanto serán más frecuentemente elegidos en las iteraciones futuras del algoritmo [GOM04b]. El procedimiento general del enfoque estándar de ACO es presentado en la Figura 4.2, además cabe destacar que este enfoque es mono-objetivo.

```

Inicio ACO estándar
Leer  $\alpha$ ,  $\beta$ ,  $\rho$  y  $G(V,E)$ 
 $\tau_{ij} = \tau_0 \forall (i, j) \in E$  /*inicializa la matriz de feromonas
Hacer {
     $T' = \emptyset$  /*inicializa el conjunto de soluciones
    Desde  $k=1$  hasta  $\theta$  repetir
         $T_k =$  Construir Árbol ( $\tau$ ) /*Construir una solución
         $T' = T' \cup T_k$ 
    Fin Desde
     $\tau =$  Actualizar matriz de feromonas ( $T'$ )
} mientras alguna criterio de parada no sea satisfecho
Fin ACO estándar
    
```

Figura 4.2. Procedimiento general de un algoritmo ACO estándar.

El algoritmo general de ACO inicialmente lee los parámetros α , β , ρ y el grafo $G(V,E)$ e inicializa la matriz de feromonas τ . Mientras algún criterio de parada no sea satisfecho, construye θ árboles y actualiza la matriz de feromonas con estas soluciones. La construcción del árbol *multicast* será explicada en detalles en la Sección 4.6.

4.5. Enfoques Propuestos para Ingeniería de Tráfico Multicast

Como se había mencionado en los Capítulos anteriores, este trabajo propone resolver el problema de enrutamiento *multicast* multiobjetivo utilizando ACO. Cabe mencionar que la resolución de este problema basado en ACO no fue todavía propuesta en la literatura. Por lo tanto, este trabajo propone utilizar tres de los mejores algoritmos *elitistas* basados en ACO. Estos algoritmos demostraron ser efectivos en varios problemas SOPs y hasta MOPs [STÜ00, SCH03, GOM04a], por lo que fueron seleccionados para su adaptación el problema de Ingeniería de Tráfico Multicast.

Estos tres algoritmos son los siguientes:

1. *Max-Min Ant System* – MMAS, de T. Stützle y H. Hoos [STÜ00], que fue propuesto en el año 2000. Este algoritmo fue exitosamente utilizado en varios problemas combinatoriales mono-objetivos como el problema del cajero viajante (*Traveling Salesman Problem* – TSP) entre otros [STÜ00], y es considerado uno de los mejores algoritmos ACO, a la fecha [GOM04a].

2. *Multiobjective Ant Colony System* – MOACS [SCH03], de S. Schaerer y B. Barán, que fue propuesto en el año 2003. Este algoritmo fue exitosamente probado en el problema del cajero viajante bi-criterio [GAR04], así como en el problema de enrutamiento de vehículos con ventajas de tiempo [SCH03] mostrando su potencialidad en un contexto multiobjetivo.
3. *Ómicrom ACO (OA)*, de G. Gómez y B. Barán [GOM04a], que fue propuesto en el año 2004. Este algoritmo fue exitosamente utilizado en el problema del cajero viajante mono-objetivo, y demostró tener buen desempeño, incluso superando al del MMAS en algunas instancias del TSP [GOM04a].

De esta manera, este trabajo justifica la utilización de los tres algoritmos mencionados. Cabe destacar que el MMAS y el OA son algoritmos ACO mono-objetivos, y como el problema a resolver es un problema MOP, estos algoritmos previamente deben ser modificados para poder resolver problemas MOPs. El MMAS y el OA multiobjetivo serán propuestos por primera vez en este trabajo. Además los tres algoritmos optimizarán por primera vez cuatro objetivos (los objetivos fueron presentados en el Capítulo 3). Desde el punto de vista del estado del arte de ACO, es un aporte sumamente significativo.

4.5.1. Multiobjective Max-Min Ant System

El *Max-Min Ant System* (MMAS) presentado por S. Stützle y H. Hoos en [STÜ00], es un derivado del algoritmo ACO estándar. El MMAS es uno de los mejores algoritmos mono-objetivos basado en ACO en la actualidad, exitosamente probado en varios problemas combinatoriales SOPs como el TSP entre otros [STÜ00]. Para lograr un buen desempeño incorpora tres rasgos claves en comparación al ACO estándar:

1. Solo la mejor solución en cada iteración o durante la ejecución del algoritmo actualiza la matriz de feromonas τ .
2. Para no quedar estancado, los posibles rangos de los componentes de τ están limitados a un intervalo $[\tau_{min}, \tau_{max}]$. Los niveles superiores pueden ser calculado como $\tau_{max} = \Delta\tau / (1-\rho)$ mientras que el nivel inferior $\tau_{min} = \tau_{max} / 2 \cdot \omega$, donde ω es la cantidad de hormigas en cada iteración y $\Delta\tau$ puede estar en función de alguna métrica que se desee optimizar [STÜ00].

3. Inicializa la matriz de feromonas con un τ_{max} , de forma a lograr una alta exploración al inicio del algoritmo. Para más detalles del MMAS refiérase al [STÜ00].

Dado que MMAS es mono-objetivo, los siguientes cambios son realizados para la optimización de problemas MOPs:

1. El nuevo enfoque, que es nombrado en este trabajo como *Multiobjective Max-Min Ant System* – M3AS, halla un conjunto entero de soluciones Pareto óptimas en vez de hallar una simple solución óptima.
2. Para guiar a las hormigas en el espacio de búsqueda, tres visibilidades son propuestas:

$$\eta_{cij} = \frac{1}{c_{ij}} \quad c_{ij} \text{ es el costo asociado al enlace}(i, j) \quad (4.2)$$

$$\eta_{dij} = \frac{1}{d_{ij}} \quad d_{ij} \text{ es el retardo asociado al enlace}(i, j) \quad (4.3)$$

$$\eta_{tij} = \frac{1}{t_{ij}} \quad t_{ij} \text{ es el tráfico actual asociado al enlace}(i, j) \quad (4.4)$$

3. Variables λ^c , λ^d y λ^t definen la influencia relativa entre las visibilidades. Por lo tanto, la probabilidad de elegir un nodo j mientras una hormiga visita un nodo i , está dada por la ecuación (4.5).

$$p_{ij} = \begin{cases} \frac{\tau_{ij}^\alpha \left(\eta_{dij} \right)^{\lambda^d} \cdot \left(\eta_{cij} \right)^{\lambda^c} \cdot \left(\eta_{tij} \right)^{\lambda^t}}{\sum_{\forall g \in N_i} \tau_{ig}^\alpha \left(\eta_{dig} \right)^{\lambda^d} \cdot \left(\eta_{cig} \right)^{\lambda^c} \cdot \left(\eta_{tig} \right)^{\lambda^t}} & \text{Si } j \in N_i \\ 0 & \text{otro caso} \end{cases} \quad (4.5)$$

4. La matriz de feromonas τ es actualizada de acuerdo a $\tau_{ij} = \tau_{ij} + \Delta\tau^k \quad \forall (i,j) \in T_k$ y $\forall T_k \in Y_{know}$ con un nivel superior de feromonas $\tau_{max}^k = \Delta\tau^k / (1-\rho)$ y no menos que un mínimo nivel $\tau_{min}^k = \Delta\tau^k / 2\omega(1-\rho)$.
5. El valor de feromonas $\Delta\tau^k$ expresada en la siguiente ecuación (4.6) que es adicionado a los enlaces, está en función de todas las métricas del árbol *multicast*, donde estas métricas ya fueron definidas en Capítulos anteriores.

$$\Delta\tau^k = \frac{1}{\alpha_{T_k} + C_{T_k} + DA_{T_k} + DM_{T_k}} \quad (4.6)$$

donde las métricas α_{T_k} , C_{T_k} , DM_{T_k} y DA_{T_k} presentadas en el Capítulo 3 (ecuaciones 3.2, 3.3, 3.4 y 3.5 respectivamente) están normalizadas, i.e., cada función objetivo es dividida por un valor máximo definido *a priori*.

El procedimiento general del M3AS es presentado en la Figura 4.3. Inicialmente, M3AS lee los parámetros e inicializa la matriz de feromonas τ . En cada generación, son construidas ω soluciones T . El conjunto de soluciones de Y_{know} es actualizado con las soluciones T no dominadas mientras las soluciones dominadas de Y_{know} son eliminadas.

Para actualizar la matriz τ , una evaporación es primeramente realizada a todos los elementos de la matriz y luego es incrementado en $\Delta\tau^k \forall (i,j) \in T_k$ para $\forall T_k \in Y_{know}$. Terminada la actualización de cada enlace, se verifica si τ_{ij} se encuentra en el rango de feromonas $[\tau_{min}, \tau_{max}]$. Este proceso se repite hasta que algún criterio de parada se cumpla. Finalmente, el algoritmo retorna el frente Pareto conocido Y_{know} como resultado. La construcción del árbol T será presentada en la Sección 4.6.

```

Inicio M3AS
  Leer grupo  $(s, N_r)$ , demanda multicast  $\phi$ , tráfico actual  $t_{ij}$ ,  $\alpha$ ,  $\beta$  y  $\rho$ 
  Inicializar  $\tau_{ij}$  con  $\tau_{max}$  /*  $\tau_{max}$  es el nivel inicial de  $\tau_{ij}$ 
  Hacer{
    Desde  $\lambda^c = 0$  hasta  $\theta - 1$ 
    Desde  $\lambda^d = 0$  hasta  $\theta - 1$ 
       $\lambda^t = \theta - 1 - \lambda^d$  /*Note que  $\omega = \theta \cdot \theta$ 
       $T =$  Construir Árbol  $(\alpha, \beta, \rho, \lambda^c, \lambda^d, \lambda^t, \phi, (s, N_r), t_{ij})$ 
      Si  $(T \not\prec \{T_x / T_x \in Y_{know}\})$  entonces
         $Y_{know} = Y_{know} \cup T - \{T_y \in Y_{know} / T \succ T_y\}, \forall T_y \in Y_{know}$ 
      Fin Si
    Fin Desde
  Fin Desde
   $\tau_{ij} = (1 - \rho) \cdot \tau_{ij} \quad \forall (i,j) \in E$ 
  Si  $(\tau_{ij} < \tau_{min})$  entonces
     $\tau_{ij} = \tau_{min} \quad \forall (i,j) \in E$ 
  Fin Si
  Repetir  $\forall T_k \in Y_{know}$ 
    Repetir  $\forall (i,j) \in T_k$ 
       $\tau_{ij} = \tau_{ij} + \Delta\tau^k$ 
      Si  $(\tau_{ij} > \tau_{max})$  entonces
         $\tau_{ij} = \tau_{max}^k$ 
      Fin Si
    Fin Repetir
  Fin Repetir
} mientras algún criterio de parada no sea satisfecho
Retornar  $Y_{know}$ 
Fin M3AS

```

Figura 4.3. Procedimiento General del *Multiobjective Max-Min Ant System* – M3AS.

4.5.2. Multiobjective Ómicron ACO

El Ómicron ACO (OA) presentado por O. Gómez y B. Barán en [GOM04a] es un algoritmo mono-objetivo inspirado en el MMAS. OA fue desarrollado en la búsqueda de herramientas muy simple de análisis de ACO, que intentaría mantener ciertas características de ACO, y más específicamente de MMAS. De esta manera fue concebido el Ómicron ACO (OA), su nombre proviene del principal parámetro de entrada utilizado, Ómicron (O). OA fue evolucionando hasta convertirse en un algoritmo, al mismo tiempo, conceptualmente simple y competitivo [GOM04a].

En la actualidad, el OA es uno de los mejores algoritmos mono-objetivos basado en ACO, como fuera probado en el problema TSP en la que supero en algunas instancias al MMAS [GOM04a].

Entre las principales analogías entre el OA y el MMAS, se puede destacar que, el OA define un nivel mínimo en el rango de feromonas, es elitista y mono-objetivo.

La principal diferencia entre MMAS y OA se encuentra en la manera en que los algoritmos actualizan la matriz de feromonas, debido que OA mantiene un conjunto de soluciones ordenadas con las que se actualiza la matriz feromonas. Además, OA no realiza evaporación y el incremento de feromonas en los enlaces pertenecientes a las soluciones ordenadas del conjunto es un valor fijo definido *a priori*, i.e. no depende de la calidad de la solución o individuo sino del parámetro de entrada Ómicron (O).

En OA, se define una matriz de feromonas constante τ^0 con $\tau_{ij}^0 = 1, \forall (i,j) \in E$. OA mantiene un conjunto de soluciones $T' = \{T\}$ conformado con " a " individuos que son las mejores soluciones diferentes encontradas hasta el momento. Para OA el mejor individuo en cualquier momento es llamado T^* , mientras que el peor individuo T_{worst} .

En OA la primera población es elegida usando τ^0 . A cada iteración un nuevo, individuo T es generado, reemplazando a $T_{worst} \in T'$ si T es mejor que T_{worst} y diferente de cualquier otro $T_x \in T'$. Luego de l iteraciones, τ es recalculada. Primeramente, $\tau = \tau^0$; luego $\frac{O}{|a|}$ es sumado a cada elemento τ_{ij} por cada vez que un arco (i, j) aparece en cualquiera de los a individuos presentes en T' . El anterior procedimiento es repetido cada l iteraciones hasta que la condición de fin sea satisfecha (para más detalles referirse a [GOM04a]).

Dado que OA es un algoritmo mono-objetivo, los siguientes cambios fueron realizados para resolver problemas MOPs:

1. El nuevo enfoque, que es nombrado en este trabajo *Multiobjective Ómicron ACO* (MOA), halla un conjunto entero de soluciones Pareto óptimas en vez de hallar una simple solución óptima.
2. Para guiar a las hormigas en el espacio de búsqueda, al igual que con M3AS, también tres visibilidades del problema de enrutamiento *multicast* son utilizadas y están dadas por las ecuaciones 4.3, 4.4 y 4.5. Además, variables λ^c , λ^d y λ^t definen la influencia relativa entre las visibilidades. Por lo tanto, la probabilidad de elegir un nodo j mientras una hormiga visita un nodo i , es dado por la ecuación (4.5), misma utilizada en M3AS.
3. MOA utiliza a conjunto de soluciones óptimas conocidas Y^a_{know} , ordenadas de mejor a peor, donde el mejor conjunto de soluciones corresponde al conjunto frente Pareto conocido ($Y^1_{know} = Y_{know}$). En este trabajo en particular solo se utiliza dos conjuntos ($|a|=2$), donde el primer conjunto domina al segundo, i.e. $Y^1_{know} \succ Y^2_{know}$.
4. La actualización de la matriz de feromonas τ solo se realiza con las soluciones pertenecientes a los frentes Y^a_{know} con niveles de feromonas constantes $O/|a|$ donde O (Ómicron) es un parámetro definido *a priori*. Además, el MOA (multiobjetivo) al igual que el OA (mono-objetivo) define un nivel mínimo de feromonas y no utiliza la evaporación de feromonas.

El algoritmo general del MOA es presentado en la siguiente Figura 4.4 y la construcción de un árbol es presentado en la Sección 5.6.

En general, puede observarse que cada una de las w soluciones calculadas es comparada con cada uno de los conjuntos de soluciones Y^a_{know} secuencialmente, hasta hallar al cuál pertenece, para actualizar dicho conjunto Y^a_{know} .

Note que la actualización de la matriz de feromonas τ es llevada en dos pasos:

1. Primeramente se inicializan las feromonas de todos los enlaces (i,j) a un valor τ^0 , i.e. $\tau_{ij} = \tau^0$.
2. Luego al igual que con el caso de OA, se adiciona un valor constante igual a $\frac{O}{|a|}$ en los enlaces pertenecientes a las soluciones de todos los conjuntos Y^a_{know} .

Finalmente cuando algún criterio de parada sea satisfecho MOA retorna el frente Pareto conocido $Y^l_{know} = Y_{know}$.

```

Inicio MOA
Leer grupo  $(s, N_r)$ , demanda multicast  $\phi$ , tráfico actual  $t_{ij}$ ,  $O$ ,  $\alpha$  y  $\beta$ 
 $\tau_{ij} = \tau_0 \quad \forall (i,j) \in E$  /*  $\tau_0$  es el nivel inicial de  $\tau_{ij}$ 
Hacer{
  Repetir  $l$  veces
    Desde  $\lambda^c = 0$  hasta  $\theta - 1$ 
      Desde  $\lambda^d = 0$  hasta  $\theta - 1$ 
         $\lambda^t = \theta - 1 - \lambda^d$  /*Note que  $\omega = \theta$ 
         $T =$  Construir Árbol  $(\alpha, \beta, \rho, \lambda^c, \lambda^d, \lambda^t, \phi, (s, N_r), t_{ij})$ 
        Si  $(T \prec \{T_1 / T_1 \in Y^l_{know}\})$  entonces
           $Y^l_{know} = Y^l_{know} \cup T - \{T_1' \in Y^l_{know} / T \succ T_1'\}, \forall T_1' \in Y^l_{know}$ 
        Sino
          Si  $(T \prec \{T_2 / T_2 \in Y^2_{know}\})$  entonces
             $Y^2_{know} = Y^2_{know} \cup T - \{T_2' \in Y^2_{know} / T \succ T_2'\}, \forall T_2' \in Y^2_{know}$ 
          Fin Si
        Fin Si
      Fin Desde
    Fin Desde
  Fin Repetir
   $\tau_{ij} = \tau_0 \quad \forall (i,j) \in E$ 
  Repetir  $\forall Y^a_{know}$  /*  $a = \{1, 2\}$ 
    Repetir  $\forall T_k \in Y^a_{know}$ 
       $\tau_{ij} = \tau_{ij} + O / |a| \quad \forall (i,j) \in T_k$ 
    Fin Repetir
  Fin Repetir
} mientras algún criterio de parada no sea satisfecho
Retornar  $Y^l_{know}$ 
Fin MOA
  
```

Figura 4.4. Procedimiento general del *Multiobjective Ómicron ACO* – MOA.

4.5.3. Multiobjective Ant Colony System

El *Multiobjective Ant Colony System* – MOACS, fue presentado por M. Schaerer y B. Barán en [SCH03], como un mecanismo de resolver MOPs usando un *Ant Colony System* (ACS) [DOR97]. El MOACS es uno de los ACOs multiobjetivo exitosamente probado en varios problemas MOPs, como en el TSP bi-objetivo y el problema de enrutamiento de vehículos con ventanas de tiempo (VRPWT) [GAR04, SCH03], entre otros.

El MOACS usa una colonia de hormigas y una matriz de feromonas $\tau = \{\tau_{ij}\}$ para la construcción de ω soluciones T en cada generación. En este trabajo, MOACS toma por primera vez la ventaja de tres visibilidades (al igual que el M3AS y el MOA) del problema de enrutamiento *multicast*, las cuáles fueron presentadas en las ecuaciones 4.2, 4.3 y 4.4. Parámetros α y β definen la influencia relativa entre la matriz de feromona y las

visibilidades. Adicionalmente, también utiliza variables λ^c , λ^d y λ^t las cuales también definen la influencia relativa entre las heurísticas. La Figura 4.5 presenta el procedimiento general del MOACS.

```

Inicio MOACS
  Leer grupo  $(s, N_r)$ , demanda multicast  $\phi$ , tráfico actual  $t_{ij}$ ,  $\alpha$ ,  $\beta$  y  $\rho$ 
   $\tau_{ij} = \tau_0 \quad \forall (i,j) \in E$  /*  $\tau_0$  es el nivel inicial de  $\tau_{ij}$ 
  Hacer{
    Desde  $\lambda^c = 0$  hasta  $\theta - 1$ 
    Desde  $\lambda^d = 0$  hasta  $\theta - 1$ 
       $\lambda^t = \theta - 1 - \lambda^d$  /* Note que  $\omega = \theta - \theta$ 
       $T = \text{Construir Árbol}(\alpha, \beta, \rho, \lambda^c, \lambda^d, \lambda^t, \phi, (s, N_r), t_{ij})$ 
      Si  $(T \not\prec \{T_x / T_x \in Y_{know}\})$  entonces
         $Y_{know} = Y_{know} \cup T - \{T_y \in Y_{know} / T \succ T_y\}, \forall T_y \in Y_{know}$ 
      Fin Si
    Fin Desde
  Fin Desde
  Si  $(Y_{know}$  fue modificado) entonces
     $\tau_{ij} = \tau_0 \quad \forall (i,j) \in E$ 
  Sino
    Repetir  $\forall (i,j) T_k \in Y_{know}$ 
       $\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta \tau^k \quad \forall (i,j) \in T_k$ 
    Fin Repetir
  Fin Si
} mientras algún criterio de parada no sea satisfecho
Retornar  $Y_{know}$ 
Fin MOACS
  
```

Figura 4.5. Procedimiento General del *Multiobjective Ant Colony System* – MOACS

Cuando las w soluciones T ya fueron construidas, MOACS actualiza el frente Pareto conocido Y_{know} incluyendo las mejores soluciones no dominadas y eliminando las dominadas.

En general, si el estado de Y_{know} fue modificado, la matriz de feromonas τ es re-iniciado ($\tau_{ij} = \tau_0 \quad \forall (i, j) \in E$) para mejorar la exploración del espacio de búsqueda [SCH03]. En otro caso, la matriz τ es globalmente actualizada usando las soluciones de Y_{know} para una mejor explotación del conocimiento de las mejores soluciones conocidas, donde los niveles de feromonas $\Delta \tau^k$ son dadas en la ecuación (4.6).

Durante la construcción del árbol *multicast*, cuando una hormiga se encuentra en un nodo i y debe seleccionar un nodo j del conjunto vecino de N_i , utiliza una regla pseudo-aleatoria [SCH03], la cuál es presentada en la Figura 4.6, mientras que la probabilidad p_{ij} asignada a los elementos de N_i fue dada en la ecuación (4.5).

```

Inicio Regla Pseudo-aleatoria
Leer parámetro  $q_0$  /* $q, q_0 \in (0,1]$ 
Seleccionar aleatoriamente  $q$  entre  $(0,1]$ 
Si  $q > q_0$  entonces
    Elegir el nodo  $j$  con mayor valor de  $p_{ij}$ 
Sino
    Elegir el nodo  $j$  aleatoriamente usando probabilidades  $p_{ij}$ 
Fin Si
Retornar  $j$ 
Fin Regla Pseudo-aleatoria
    
```

Figura 4.6. Regla Pseudo-aleatoria. Procedimiento de selección de un nodo j del conjunto vecindario N_i .

La Regla Pseudo-aleatoria tiene como parámetro de entrada $q_0 \in (0,1]$, y dependiendo del valor aleatorio tomado por la variable $q \in (0,1]$ se selecciona el nodo j con mayor valor p_{ij} asignado o el nodo j es seleccionado aleatoriamente utilizando las probabilidades sesgadas p_{ij} dadas en la ecuación (4.5).

4.6. Construcción basada en ACO del Árbol Multicast

Cada uno de los tres algoritmos propuestos (M3AS, MOA & MOACS) utiliza un procedimiento para construir una solución (árbol T) por vez. Estos procedimientos son muy semejantes y difieren en algunos aspectos importantes como el método de selección de enlace y la actualización de los niveles de feromonas de los enlaces seleccionados.

El algoritmo “Construir Árbol” es presentado en la Figura 4.7. Este inicia leyendo el grupo *multicast* (s, N_r) , los parámetros de influencia relativa entre visibilidades $(\lambda^c, \lambda^d, \lambda^l)$, parámetros de influencia entre la feromona y el conjunto de visibilidades (α, β) , el factor de evaporación (ρ) , el tráfico actual de la red (t_{ij}) y la respuesta *multicast* (ϕ) .

Seguidamente a la lectura de los datos de entrada, el algoritmo inicializa el árbol a construir T y el conjunto de nodos destinos ya alcanzados D_r que al principio es vacío, i.e. $T = \emptyset$ y $D_r = \emptyset$. Luego un conjunto de nodos de partida representado por N es iniciado con el nodo fuente s . Posteriormente, en el ciclo principal es seleccionado del conjunto de nodos de partida N , un nodo de partida i , hallándose una lista de nodos vecinos factibles a i representado por el conjunto N_i . Los nodos son factibles si *no* se encuentran ya en el árbol T y si cumplen con la restricción de carga (ver ecuación (3.6)). Si N_i es vacío, entonces el nodo i es eliminado de N , por otra parte si N_i no es vacío, las probabilidades p_{ij} son calculadas para cada elemento de N_i utilizando la ecuación (4.5). Luego, se selecciona un nodo j del N_i adhiriendo el enlace (i,j) formado al árbol en construcción T , como también el

nodo j es adherido al conjunto N como un nuevo nodo de partida. Si el nodo j es un *nodo destino* ($j \in N_r$) este nodo es adherido al conjunto D_r , de forma a mantener un registro de destinos alcanzados. La construcción del árbol T sigue con el mismo proceso hasta alcanzar todos los nodos destinos N_r . Finalmente, se eliminan todos los enlaces no utilizados del árbol.

```

Inicio Construir Árbol
  Leer  $\lambda^c, \lambda^d, \lambda^i, \alpha, \beta, \rho, \phi, \tau_{ij}, (s, N_r), t_{ij}$ 
   $T = \emptyset$  /*  $T$  árbol en construcción
   $D_r = \emptyset$  /*  $D_r$  conjunto de nodos destinos alcanzados
   $N = "s"$  /*  $N$  conjunto de nodos de partida
  Hacer{
    Seleccionar nodo  $i$  de  $N$  y construir lista de nodos vecinos  $N_i$ 
    Si ( $N_i = \emptyset$ ) entonces
       $N = N - i$  /* Se elimina el nodo de sin vecino factible
    Sino
      Calcular las probabilidades  $p_{ij}$  de la lista  $N_i$ , según ecuación (4.5)
      Si Construir Árbol es llamado por MOACS entonces
        Seleccionar nodo  $j$  utilizando Regla pseudo-aleatoria
      Sino
        Seleccionar aleatoriamente el nodo  $j$  utilizando  $p_{ij}$ 
      Fin Si
       $T = T \cup (i, j)$ 
       $N = N \cup j$ 
      Si ( $j \in N_r$ ) entonces
         $D_r = D_r \cup j$ 
      Fin Si
      Si Construir Árbol es llamado por MOACS entonces
        Evaporación en línea utilizando la ecuación (4.6)
      Fin Si
    Fin Si
  } mientras los siguientes criterios se cumplan ( $N \neq \emptyset$  o  $D_r \neq N_r$ )
  Podar Árbol  $T$  /* se borran todos los enlaces no utilizados */
  Retornar  $T$ 
Fin Construir Árbol

```

Figura 4.7. Procedimiento General de Construir Árbol.

La selección del nodo j perteneciente al conjunto N_i es realizada aleatoriamente utilizando probabilidades sesgadas p_{ij} (p_{ij} definida en la ecuación (4.5)) para el caso MOA y M3AS. En el caso del MOACS, como fuera mencionado en secciones anteriores, el nodo j es seleccionado utilizando una regla pseudo-aleatoria [SCH03]. Esta regla fue presentada en la Sección 4.4 (ver Figura 4.6).

Otra diferencia importante es que el enlace (i, j) que fuera seleccionado durante la construcción de T sufre una evaporación de feromonas en línea para el caso del MOACS. Esta evaporación es presentada a continuación:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \tau_0 \quad (4.7)$$

El objetivo de la evaporación en línea es intentar que las hormigas no utilicen el mismo camino ya recorrido por hormigas anteriores, de forma a mejorar la exploración de nuevas soluciones y prever una convergencia prematura hacia soluciones sub-óptimas [SCH03].

Es importante aclarar que el algoritmo Construir Árbol puede ser utilizado por cualquier algoritmo mono-objetivo, como el caso del “*Enfoque Estándar de ACO*”. Para su utilización solo será necesario utilizar la ecuación (4.1) en vez de la ecuación (4.5). De esta manera, el algoritmo “Construir Árbol” queda definido como un algoritmo genérico basado en ACO.

4.7. Resumen del Capítulo

En resumen, el presente Capítulo presenta los conceptos del funcionamiento de *Ant Colony System* – ACO. Se da como introducción el “*Enfoque Estándar de ACO*” para luego analizar en forma extensiva los algoritmos de optimización multiobjetivo propuestos en este trabajo para el problema de enrutamiento *multicast*. Estos algoritmos fueron el *Multiobjective Max-Min Ant System* (M3AS), el *Multiobjective Ómicron ACO* (MOA) y el *Multiobjective Ant Colony System* (MOACS). Finalmente, se presenta el procedimiento de construcción de un árbol *multicast* basado en Colonia de Hormigas. Ya presentados los nuevos enfoques para la resolución del problema de enrutamiento *multicast*, en el Capítulo siguiente se procede a presentar las pruebas experimentales con el objetivo de validar las nuevas propuestas.

Capítulo 5

Pruebas Experimentales

5.1 Introducción

En este Capítulo se presentan las pruebas (simulaciones) y los resultados de los algoritmos M3AS, MOA & MOACS en un ambiente *Estático* y *Semi-Dinámico*. A modo de validar los resultados obtenidos de estos algoritmos se los compara con un algoritmo evolutivo “*Multiobjective Multicast Algorithm – MMA*” [CRI04a, CRI04b, CRI04c], recientemente publicado y que fuera diseñado exclusivamente para el problema de enrutamiento *multicast*. El MMA ya fue presentado en la sección 2.5 del Capítulo 2.

Este Capítulo queda organizado de la siguiente manera. En la Sección 5.2 se expone las condiciones del ambiente computacional en el cual fueron llevados a cabo las simulaciones. En la Sección 5.3 se exponen los procedimientos de comparación que fueron utilizados. En las Secciones 5.4 y 5.5 se presentan los resultados de las simulaciones estáticas y semi-dinámicas respectivamente. Finalmente, en la Sección 5.6 se dan las conclusiones basadas en los resultados globales de las simulaciones.

5.2 Ambiente Experimental

Las pruebas o simulaciones correspondientes al presente trabajo fueron realizadas sobre la red de telecomunicaciones de la compañía japonesa *Nipon Telegraph and Telephone, Co. – NTT* (Figura 5.1).

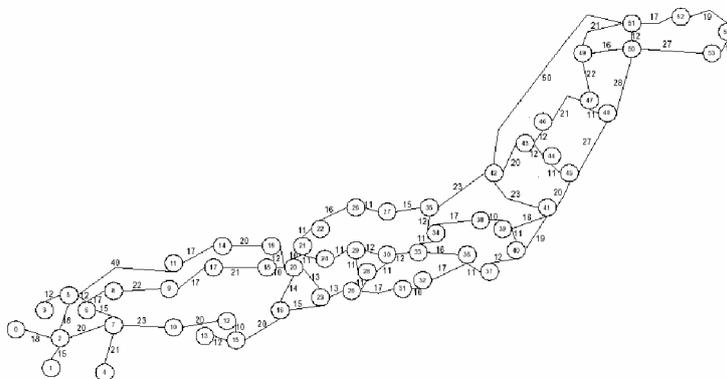


Figura 5.1. Red de la *Nipon Telegraph and Telephone, Co.* (NTT)

La red de la NTT consta de 55 nodos (ruteadores) y 144 enlaces correspondientes. Los números sobre cada enlace de 6 Mbps de capacidad representan los retardos asociados.

En relación a los tipos de pruebas realizadas en el ambiente *multicast* pueden distinguirse tres: el estático, el semi-dinámico y el dinámico. En tipo **estático** el objetivo principal es la verificación de los algoritmos a muy baja carga en la red. En este caso la red está totalmente descargada y los algoritmos deben responder a un pedido *multicast*. En el tipo **semi-dinámico** la red se encuentra a un nivel de carga inicial, i.e. la red tiene tráfico actual, y los algoritmos deben responder a un pedido *multicast* por vez. Tanto para el tipo estático como para el semi-dinámico, los algoritmos deben encontrar el frente Pareto aproximado - Y_{apr} para cada pedido *multicast* entrante. En el caso de las pruebas **dinámicas**, éstas se encuentran conformadas por un conjunto sucesivo de pruebas semi-dinámicas a su vez con políticas de selección de soluciones, debido a que solamente es utilizada una solución por cada pedido *multicast* entrante. Cabe mencionar que en el presente trabajo de tesis se tiene por objetivo principal la realización de pruebas experimentales estáticas como semi-dinámicas, teniendo en cuenta que, en relación a las pruebas dinámicas, el problema de políticas de selección de soluciones ya fue resuelto en otros trabajos relacionados [TAL04b].

En las pruebas realizadas (estáticas y semi-dinámicas) fueron utilizados cuatro grupos *multicast* distintos, los cuales son presentados en la Tabla 5.1.

Tabla 5.1. Grupos Multicast utilizados en todas las pruebas.

Grupo Multicast	<i>Nodo Fuente</i> s	Nodos destinos N_r	$ N_r $
1	{5}	{0, 1, 8, 10, 22, 32, 38, 43, 53}	9
2	{4}	{0, 1, 3, 5, 9, 10, 12, 23, 25, 34, 37, 41, 46, 52}	14
3	{4}	{0, 1, 3, 5, 6, 9, 10, 12, 17, 22, 23, 25, 34, 37, 41, 46, 47, 52, 54}	19
4	{4}	{0, 1, 3, 5, 6, 9, 10, 11, 12, 17, 19, 21, 22, 23, 25, 33, 34, 37, 41, 44, 46, 47, 52, 54}	24

Para cada grupo *multicast* se realizaron varias ejecuciones con distintos tiempos de corrida: 10 segundos, 20 segundos, 30 segundos, y 40 segundos. A su vez, para cada grupo *multicast* y cada tiempo de corrida, fueron hechas cinco ejecuciones distintas, luego de lo cual fueron calculados los promedios para una mejor apreciación del comportamiento de los algoritmos. Los resultados correspondientes, por extenso, constan en el Anexo A y B, respectivamente.

Es importante destacar que para las pruebas semi-dinámicas se generaron tres diferentes cargas iniciales en la red, las cuales son:

1. Baja carga: utilización de los enlaces entre 0 y 0.4.
2. Media carga: entre 0.4 y 0.7 de capacidad de los enlaces de la red, y
3. Alta carga (saturación): entre 0.7 y 1 de capacidad de los enlaces de la red.

En cuanto al ambiente computacional, los cuatros algoritmos fueron implementados sobre una computadora 350 MHz AMDK6 con 128 MB de RAM. El compilador utilizado fue Borland C++ V 5.02.

Como indicado en el capítulo 4, todos los algoritmos ACOs (M3AS, MOA & MOACS) poseen ciertos parámetros que deben ser seleccionados *a priori*. Estos parámetros son: α y β que indican la influencia relativa entre las feromonas y las heurísticas respectivamente, ρ la cuál es un factor de evaporación de los niveles de feromonas, ω la cantidad de hormigas utilizadas en cada iteración, τ_0 el nivel inicial de feromona y el factor q_0 utilizado en la selección pseudo-aleatoria para el del MOACS.

En este trabajo, los parámetros ACO fueron seleccionados después de realizar varias pruebas experimentales con varios grupos *multicast* en varios niveles de carga. En estas pruebas fueron seleccionados los mejores valores de los parámetros que resultaron de las diferentes combinaciones de los mismos (donde se varía un parámetro, por vez, y los otros permanecen constantes). Los mejores valores tomados de estas pruebas arrojaron los mejores frentes Pareto. En Tabla 5.2 se muestran el conjunto de mejores valores de los parámetros ACO de entrada. Estos parámetros fueron utilizados en las diversas pruebas estáticas y semi-dinámicas.

Tabla 5.2. Parámetros de entradas utilizados en los diferentes algoritmos ACO propuestos.

ACOs	Valores de los Parámetros ACO					
	α	β	ρ	τ_0	ω	q_0
M3AS	1	10	0.7	10	10	-
MOA	1	10	-	1	10	-
MOACS	1	4	0.9	1	50	0.8

5.3 Procedimiento de Comparación

La obtención del “frente Pareto real - Y_{true} ” de un problema MOP requiere un tiempo computacional no razonable para hallar todas las soluciones [VAN99]. Por tal motivo, para fines prácticos, en la realización de las pruebas, fue necesario hallar un frente Pareto aproximado - Y_{apr} , donde Y_{apr} es una excelente aproximación a Y_{true} , i.e., $Y_{true} \approx Y_{apr}$.

5.3.1 Obtención del Frente Pareto Aproximado

Para cada grupo *multicast* en las distintas pruebas realizadas, fue calculado su frente Pareto aproximado (Y_{apr}) considerando el siguiente procedimiento de seis pasos:

1. Cada algoritmo (M3AS, MOA, MOACS & MMA) fue ejecutado cinco veces y un promedio fue calculado para fines de comparación entre algoritmos.
2. Para cada algoritmo, cinco conjunto de soluciones no-dominadas: Y_1, Y_2, \dots, Y_5 fueron calculados, uno para cada corrida.
3. Para cada algoritmo, una súper-población Y_{SP} fue obtenida, donde $Y_{SP} = \bigcup_{i=1}^5 Y_i$.
4. Soluciones dominadas fueron borradas de Y_{SP} , obteniendo el frente Pareto calculado Y_{alg} para cada algoritmo, como sigue:

- Y_{M3AS} (Frente Pareto obtenido con las cinco corridas, usando M3AS)
- Y_{MOA} (Frente Pareto obtenido con las cinco corridas, usando MOA)
- Y_{MOACS} (Frente Pareto obtenido con las cinco corridas, usando MOACS)
- Y_{MMA} (Frente Pareto obtenido con las cinco corridas, usando MMA)

5. Un conjunto de soluciones Y' fue obtenido como:

$$Y' = Y_{M3AS} \cup Y_{MOA} \cup Y_{MOACS} \cup Y_{MMA}.$$

6. Las soluciones dominadas fueron eliminadas de Y' , y una aproximación de Y_{true} , llamado Y_{apr} , fue finalmente creada.

En las siguientes Tablas (5.3 al 5.6) se exponen el total de soluciones $|Y_{apr}|$ que fueron halladas para cada grupo *multicast*, para las distintas pruebas.

Tabla 5.3. Pruebas Estáticas. Total de soluciones pertenecientes a Y_{apr} para cada grupo multicast.

	Grupo Multicast 1	Grupo Multicast 2	Grupo Multicast 3	Grupo Multicast 4
$ Y_{apr} $	9	18	24	18

Tabla 5.4. Pruebas Semi-Dinámicas. Total de soluciones pertenecientes a Y_{apr} para cada grupo multicast (carga baja).

	Grupo Multicast 1	Grupo Multicast 2	Grupo Multicast 3	Grupo Multicast 4
$ Y_{apr} $	15	27	30	34

Tabla 5.5. Pruebas Semi-Dinámicas. Total de soluciones pertenecientes a Y_{apr} para cada grupo multicast (carga media).

	Grupo Multicast 1	Grupo Multicast 2	Grupo Multicast 3	Grupo Multicast 4
$ Y_{apr} $	31	66	63	56

Tabla 5.6. Pruebas Semi-Dinámicas. Total de soluciones pertenecientes a Y_{apr} para cada grupo multicast (carga alta).

	Grupo Multicast 1	Grupo Multicast 2	Grupo Multicast 3	Grupo Multicast 4
$ Y_{apr} $	46	162	143	115

5.3.2 Métodos de Comparación

Las Tablas mostradas en las siguientes Secciones como también en los Anexos A y B representan la comparación entre las soluciones halladas (Y_{alg}) con los algoritmos (M3AS, MOA, MOACS & MMA) con respecto al frente Pareto aproximado - Y_{apr} cuyas cantidades fueron presentadas en las Tablas 5.3 al 5.6. Estos frente Pareto calculados (Y_{alg}), por su vez, son comparados entre ellos usando la métrica de *cobertura*, que es una figura de mérito que cuenta el número de soluciones dominadas por los otros algoritmos pertenecientes a un dado conjunto Pareto calculado [VAN99].

Los resultados obtenidos y la comparación relativa entre los algoritmos considerados y correspondientes al tipo de prueba estática (red sin carga), grupo *Multicast 1*, para un tiempo de corrida de 40 segundos, se muestra en la Tabla 5.7.

Tabla 5.7. Grupo Multicast 1 - Tiempo de corrida = 40 seg. – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Dominada	Es-dominado	
Y_{M3AS}	6.4	0.2	6.6	70	0.4	0.2	2
Y_{MOA}	6.4	1.2	7.6	70	0.067	1.2	3
Y_{MOACS}	9	0	9	100	0.47	0	1
Y_{MMA}	5.8	0	5.8	64	0.47	0	4

En la Tabla 5.7, bajo la columna de “Comparación de Solución con Y_{apr} ” se utiliza la siguiente notación:

1. $|Y_{apr} \cap Y_{alg}|$ es el promedio del número de soluciones que pertenecen a la vez al frente Pareto aproximado Y_{apr} y el frente Pareto calculado de un algoritmo Y_{alg} . Note que el subíndice $alg \in \{M3AS, MOA, MOACS \& MMA\}$.

2. $|Y_{apr} \succ Y_{alg}|$ es el promedio del número de soluciones de Y_{alg} que son dominadas por Y_{apr} .
3. $|Y_{alg}|$ es el promedio del número de soluciones pertenecientes al frente Pareto calculado de un algoritmo.
4. $\%Y_{alg}$ es el porcentaje de soluciones halladas por un algoritmo dado, i.e. $100 \cdot (|Y_{apr} \cap Y_{alg}|) / (|Y_{apr}|)$.

Los conceptos correspondientes a la columna “Cobertura” de esta tabla obedecen a la siguiente notación:

5. **Domina** que indica el promedio de soluciones que un algoritmo domina sobre las soluciones encontradas por los otros algoritmos.
6. **Es-dominado** que promedio de soluciones pertenecientes a un algoritmo que son dominadas por los otros algoritmos.

Para casos de desempatar soluciones y establecer un “Ranking” logrado por los algoritmos en base al porcentaje de soluciones encontradas pertenecientes al frente Pareto Y_{apr} , los datos de la columna “**Dominada**” dan el desempate.

7. **Ranking** es la medida de desempeño entre los algoritmos, con escala de desempeño de 1 a 4. El mejor desempeño corresponde a 1 y el peor corresponde a 4.

Observe el bloque de las “Comparaciones de Soluciones con Y_{apr} ”, de la Tabla 5.7 tenemos que son comparadas las soluciones arrojadas por los algoritmos (Y_{M3AS} , Y_{MOA} , Y_{MOACS} & Y_{MMA}) con las soluciones del frente Pareto aproximado Y_{apr} . En esta tabla, tenemos para el caso de Y_{M3AS} :

1. En la fila del Y_{M3AS} se encuentra que **6.6** soluciones en promedio corresponden a $|Y_{M3AS}| = 6.6$ en promedio.
2. La intersección entre la columna $|Y_{apr} \cap Y_{alg}|$ y la fila de Y_{M3AS} indica que **6.4** soluciones de Y_{M3AS} pertenecen también a Y_{apr} .
3. La columna $|Y_{apr} \succ Y_{alg}|$ es la diferencia entre los valores de la columna $|Y_{alg}|$ y $|Y_{apr} \cap Y_{alg}|$, e indica la cantidad de soluciones dominadas por el frente Pareto aproximado. Por ejemplo, **0.2** soluciones de Y_{M3AS} en promedio son dominadas por Y_{alg} .

4. Finalmente, la columna $\%Y_{apr}$ muestra el porcentaje de las soluciones pertenecientes al frente Pareto Y_{apr} que pudo ser hallado por un algoritmo dado. Así, el algoritmo M3AS obtuvo el **70%** del frente Pareto Y_{apr} entre sus soluciones.

En el bloque de “Coberturas”, se comparan las soluciones de un algoritmo con respecto al resto de los algoritmos. Siguiendo el ejemplo para el M3AS, donde se confrontan las soluciones de Y_{M3AS} con las soluciones de $Y_{MOACS} \cup Y_{MOA} \cup Y_{MMA}$ puede verse que:

5. La intersección de la fila de Y_{M3AS} y columna “**Domina**” muestra que **0.4** soluciones del conjunto $Y_{MOACS} \cup Y_{MOA} \cup Y_{MMA}$ el M3AS logró dominar, i.e. $|Y_{M3AS} \succ (Y_{MOACS} \cup Y_{MOA} \cup Y_{MMA})| = 0.4$.
6. Además que, **0.2** soluciones pertenecientes a Y_{M3AS} fueron dominadas por el conjunto $Y_{MOACS} \cup Y_{MOA} \cup Y_{MMA}$, i.e. $|(Y_{MOACS} \cup Y_{MOA} \cup Y_{MMA}) \succ Y_{M3AS}| = 0.2$.

Finalmente, la columna del “**Ranking**” indica que el desempeño logrado por el M3AS corresponde al segundo lugar, por hallar el 70% del frente Pareto. Note que el MOA también halló el 70%; sin embargo en la columna de “**Dominada**” solo obtuvo 0.067. El primer lugar corresponde al MOACS por obtener el 100% de las soluciones y en último puesto queda el MMA por hallar solo el 64%.

5.4 Pruebas Estáticas

En estas pruebas la red NTT se encuentra descargada. El objetivo de estas pruebas, como ya se menciona, es la verificación del funcionamiento de los algoritmos a muy baja carga en la red.

Para una mejor apreciación y claridad de los resultados obtenidos, en las siguientes sub-secciones se exponen solamente las Tablas de Promedio Parciales por Grupo *Multicast*, siendo presentadas, en el Anexo A todas las pruebas realizadas.

Es muy importante destacar que como la red está descargada, los algoritmos solo se optimizan tres de los objetivos: el costo del árbol, el retardo medio y retardo máximo de extremo a extremo, ya que no tiene sentido buscar la optimización del cuarto objetivo que es la máxima utilización de los enlaces.

5.4.1 Resultados para el Grupo Multicast 1 – Prueba Estática

La Tabla 5.8 presenta los Promedios Parciales correspondientes a los distintos tiempos de corrida del grupo *multicast 1*, para las pruebas estáticas.

Tabla 5.8. Grupo Multicast 1 - Promedios Parciales – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	% Y_{apr}	Dominada	Es-dominado	
Y_{M3AS}	8.1	0.05	8.15	90	0.13	0.05	2
Y_{MOA}	7.3	0.4	7.7	81.11	0.0167	0.383	3
Y_{MOACS}	9	0	9	100	0.15	0	1
Y_{MMA}	5.5	0	5.5	61.11	0.133	0	4

Observe, en la columna **Comparación de Soluciones con Y_{apr}** que el MOACS logró el primer lugar al hallar en promedio el 100% de las soluciones del frente Pareto aproximado - Y_{apr} . En promedio, el MOACS calculó 9 soluciones, de las cuáles todas pertenecieron a Y_{apr} , por lo cuál ninguna fue dominada. En la columna de **cobertura**, puede notarse que logro dominar en promedio 0.15 soluciones del resto de los algoritmos y ninguna de sus soluciones fueron dominadas, es decir, genero excelentes soluciones.

Por su parte, el M3AS, logró el segundo lugar al obtener 90% de soluciones de Y_{apr} , en promedio. Note que el M3AS halló 8.15 soluciones, siendo de éstas 8.10 soluciones del frente Pareto aproximado y 0.05 fueron dominadas. Según la columna **cobertura**, logro dominar, en promedio 0.13 soluciones del resto de los algoritmos, y 0.05 de sus soluciones fueron dominadas.

En el tercer lugar se encuentra el MOA. El MOA halló el 81.11% de las soluciones del frente Pareto aproximado. De las 7.7 soluciones que calculo, en promedio, 7.3 de éstas pertenecen al Y_{apr} y 0.4 fueron dominadas. Observando la columna **cobertura**, el MOA dominó 0.0167 soluciones y 0.383 de sus soluciones en promedio fueron dominadas.

Y, finalmente, en estas pruebas el MMA quedó en último lugar en el ranking, con 61.11% de soluciones del frente Pareto aproximado - Y_{apr} . El MMA calculó en promedio 5.5 soluciones, las cuales todas pertenecen al Y_{apr} . Por la columna **cobertura**, el MMA logró dominar en promedio 0.133 soluciones, y como todas sus soluciones pertenecen al frente Pareto aproximado, ninguna fue dominada.

5.4.2 Resultados para el Grupo Multicast 2 – Prueba Estática

El Promedio de las Pruebas Estáticas para el grupo *multicast 2* son mostradas en la Tabla 5.9. Estas corresponden al promedio de todos los tiempos de corrida.

Tabla 5.9. Grupo Multicast 2 - Promedios Parciales – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	12.6	5.15	17.75	70	0.63	1.15	2
Y_{MOA}	11.75	4.45	16.2	65.28	0.383	1.683	4
Y_{MOACS}	15.9	5.8	21.7	88.33	1.35	0.0167	1
Y_{MMA}	11.75	3.65	15.4	65.28	0.867	0.383	3

En estas pruebas, el frente Pareto calculado - Y_{MOACS} , en promedio, obtuvo 21,7 soluciones. Donde 15.9 soluciones de Y_{MOACS} pertenecían al Y_{apr} . Esto corresponde al 88.33% de Y_{apr} . En la **cobertura**, el MOACS dominó 1.35 soluciones y solo 0.0167 de sus soluciones fueron dominadas. Note que estos valores son mejores que los obtenidos que otros algoritmos.

El M3AS por su parte, halló 17.75 soluciones en promedio, siendo 12.6 de ellas pertenecientes a Y_{apr} , correspondiendo esto al 70% del frente Pareto aproximado. En la **cobertura**, dominó en promedio 0.63 soluciones del resto de los algoritmos y 1.15 de sus soluciones fueron dominadas. El M3AS es el segundo en el ranking.

El MOA como el MMA hallaron el 65.28% de las soluciones de Y_{apr} , pero en la cobertura el MMA dominó 0.867 soluciones y el MOA solo 0.383. Por lo cuál, al MMA le corresponde el tercer lugar del ranking, quedando en último lugar el MOA.

El frente Pareto calculado - Y_{MMA} obtuvo en promedio 15.4 soluciones; de las cuáles 11.75 soluciones pertenecen también al Y_{apr} y 3.65 fueron dominadas. Por otra parte, en la cobertura, 0.867 soluciones perteneciente al resto de los algoritmos logró dominar y 0.383 de sus soluciones fueron dominadas.

5.4.3 Resultados para el Grupo Multicast 3 – Prueba Estática

La Tabla 5.10 presenta los Promedios Parciales para el grupo *multicast* 3, los cuáles están dados también en la Tabla A.15 del anexo A.

El MOACS obtuvo el 76.04% del frente Pareto aproximado - Y_{apr} , por lo cuál obtuvo la mejor calificación del ranking. Cabe mencionar que también en la cobertura tuvo mayor dominancia con 1.933 soluciones dominadas y ninguna de sus soluciones fue dominada. El segundo lugar corresponde al M3AS con el 47.92% de Y_{apr} . Por su parte, el MMA logro el tercer lugar al obtener 47.08% de Y_{apr} superando al MOA, el cual quedo último con solo 33.75% de Y_{apr} .

Tabla 5.10. Grupo Multicast 3 - Promedios Parciales – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	11.50	2.15	13.65	47.92	1.05	1.583	2
Y_{MOA}	8.10	3.70	11.80	33.75	0.383	3.3	4
Y_{MOACS}	18.25	0.40	18.65	76.04	1.933	0	1
Y_{MMA}	11.30	0.10	11.40	47.08	1.567	0.05	3

5.4.4 Resultados para el Grupo Multicast 4 – Prueba Estática

Los resultados de las pruebas experimentales para el Grupo *Multicast* 4 están presentes en el Anexo A en las Tablas A.16, A.17, A.18 y A.19. Los datos de la Tabla A.10 representan los Promedios Parciales, y son presentados en la Tabla 5.12 para una mejor lectura.

Tabla 5.11. Grupo Multicast 4 - Promedios Parciales – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	4.65	5.20	9.85	25.83	1.53	3.08	2
Y_{MOA}	3.35	6.85	10.20	18.61	0.85	4.516	4
Y_{MOACS}	11.70	3.40	15.10	65	3.95	0.233	1
Y_{MMA}	4.25	0.90	5.15	23.61	1.8	0.3	3

Estas pruebas que se caracterizan por tener el grupo *multicast* más grande. El MOACS nuevamente ha logrado el primer lugar al tener el 65% del frente Pareto aproximado Y_{apr} con una dominancia superior, en la cobertura, con 3.95 soluciones dominadas. El M3AS obtuvo el segundo lugar con 25.83% del frente Pareto Y_{apr} , quedando tercero el MMA y cuarto el MOA. Note que a pesar de tener el MOA mayor cantidad de soluciones su frente Pareto calculado - Y_{MOA} con 10.2 soluciones, solo 3.35 de ellas estaban en Y_{apr} .

5.4.5 Promedios Generales de las Pruebas Estáticas

Para poder tener una visión global sobre las Pruebas Estáticas, se procede a presentar los Promedios Generales. En la Tabla A.21 presenta dichos Promedios Generales y cada elemento de ésta representa el promedio de los elementos correspondientes de las Tablas Parciales de cada Grupo *Multicast*, las cuáles son: Tabla A.5, A.10, A.15 y A.20 del anexo A. Para mayor comodidad la Tabla 5.12 presenta los mismos datos de la Tabla A.21.

Puede observarse que, el MOACS obtuvo mejor ranking, debido que pudo obtener el mayor porcentaje de soluciones pertenecientes a los frentes Pareto, i.e. 82.34% de Y_{apr} .

Con 16.11 soluciones calculadas en Y_{MOACS} , de las cuáles 13.71 soluciones en promedio pertenecen a los frentes Pareto, donde 2.4 de sus soluciones fueron dominadas. En la **cobertura**, también logro dominar más soluciones con un promedio general de 1.845, mientras que solo 0.0625 de sus soluciones fueron dominadas.

Tabla 5.12. Promedios Generales de las Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	9.21	3.14	12.35	58.44	0.84	1.467	2
Y_{MOA}	7.63	3.85	11.48	49.69	0.408	2.47	3
Y_{MOACS}	13.71	2.40	16.11	82.34	1.845	0.0625	1
Y_{MMA}	8.20	1.16	9.36	49.27	1.091	0.183	4

El M3AS por su parte, obtuvo el segundo lugar en el ranking. De las 12.35 soluciones calculadas en promedio, 9.21 de ellas pertenecieron a los frentes Pareto, correspondiendo en promedio a 58.44% de los frentes Pareto, y el resto (3.14 soluciones) fueron dominadas. Por otra parte, en la cobertura 0.84 soluciones en promedio logro dominar, y 1.467 de sus soluciones en promedio fueron dominadas.

En el tercer puesto del ranking se encuentra el MOA, con 49.69% de los frentes Pareto que logro hallar. En promedio calculó 11.48 soluciones, perteneciendo 7.63 soluciones a los frentes Pareto y 3.85 fueron dominadas. En la cobertura, dominó en promedio 0.408 soluciones y 2.47 de sus soluciones fueron dominadas.

El MMA quedó último en el ranking, obteniendo solo 49.27% de las soluciones de los frentes Pareto. En promedio el MMA, halló 9.36 soluciones, de las cuáles 8.20 soluciones pertenecen a los frentes Pareto y 1.16 soluciones en promedio fueron dominadas por los frentes Pareto. En las coberturas 1.091 soluciones del resto de los algoritmos logró dominar siendo el resto (0.183 en promedio) dominado.

En resumen, el MOACS en todas las pruebas estáticas fue mejor, quedando en segundo lugar el M3AS, en tercer lugar el MOA y último el MMA.

5.5 Pruebas Semi-Dinámicas

Las pruebas semi-dinámicas se caracterizan por tener la red inicialmente cargada. Como ya mencionado anteriormente, a modo de verificar el funcionamiento de estos algoritmos, se propone tres niveles de cargas distintas:

- Baja carga – entre 0 y 0.4 de utilización en los enlaces.

- Media Carga – entre 0.4 y 0.7 de utilización en los enlaces.
- Alta Carga – entre 0.7 y 1 de utilización en los enlaces. Nótese que en este tipo de carga existen enlaces totalmente saturados.

En las pruebas dinámicas, a diferencia de las pruebas estáticas, los algoritmos deben optimizar las cuatro funciones objetivos: la utilización máxima de los enlaces, el costo del árbol, el retardo medio y el retardo máximo de extremo a extremo. Análogamente, al tratamiento dado a las pruebas estáticas, en esta sección será expuesto solamente las Tablas de Promedios Parciales y Global.

Luego, serán presentados los resultados de las pruebas con baja carga (sección 5.5.1), con media carga (sección 5.5.2) y con alta carga (sección 5.5.3).

5.5.1 Pruebas con Baja Carga

Para estas pruebas, la red se encuentra a baja carga, donde los enlaces están inicialmente cargados en el rango de 0 y 0.4 de utilización de los mismos.

En las siguientes sub-secciones son presentados los resultados de Promedios Parciales para los distintos tipos de grupos *multicast*, así como el Promedio General a Baja Carga.

5.5.1.1 Resultados para el Grupo Multicast 1 – Baja Carga

Los resultados de las pruebas experimentales para el Grupo *Multicast 1* para todos los tiempos de corridas son presentados en la Tabla 5.13.

Tabla 5.13. Grupo Multicast 1 - Promedios Parciales a Baja Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \supset Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	4.65	2.45	7.10	31	0.33	2.27	2
Y_{MOA}	3.65	4.15	7.80	24.33	0.38	3.42	4
Y_{MOACS}	13.60	1	14.60	90.67	2.48	0	1
Y_{MMA}	4.30	1.25	5.55	28.67	2.82	0.33	3

En estas pruebas, que se caracteriza por tener tamaño grupo *multicast* menor y con baja carga, el MOACS obtuvo el 90.67% de Y_{apr} muy superior al resto de los algoritmos. Además en la **cobertura** notamos que ninguna de sus soluciones fue dominada.

Por su parte, el M3AS logró el segundo lugar al obtener el 31% del frente Pareto aproximado. De las 7.1 soluciones pertenecientes a su frente Pareto calculado - Y_{M3AS} ; de

ellas, 4.65 correspondió al frente Pareto aproximado - Y_{apr} y solo 2.45 fueron dominadas. En la columna **cobertura**, en promedio, solo dominó 0.33 soluciones, pero 2.27 de sus soluciones fueron dominadas.

El tercer lugar corresponde al MMA, que de sus 5.55 soluciones calculadas, 4.3 de ellas pertenecen al frente Pareto aproximado - Y_{apr} que corresponden al 28.67%. En la **cobertura**, 2.82 soluciones logró dominar y 0.33 de sus soluciones solamente fueron dominadas. El cuarto lugar corresponde al MOA, que solo alcanzo el 24.33% del frente Pareto aproximado.

5.5.1.2 Resultados para el Grupo Multicast 2 – Baja Carga

La Tabla 5.14 presenta los promedios parciales de los resultados correspondientes a las pruebas experimentales para el Grupo *Multicast 2*.

Tabla 5.14. Grupo Multicast 2 - Promedios Parciales a Baja Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	8.6	5	13.6	31.78	0.8	4.55	2
Y_{MOA}	6.75	5.95	12.7	25	1.23	2.35	4
Y_{MOACS}	19.65	1.2	20.85	72.78	2.82	0	1
Y_{MMA}	8.5	2.55	11.05	31.48	2.55	0.50	3

En la **Comparación de Soluciones con Y_{apr}** , el 72,78% del frente Pareto aproximado - Y_{apr} fue alcanzado por el MOACS, superior al resto de los algoritmos, logrando de esta manera el primer lugar. Además, note que en la **cobertura** obtuvo 2.82 soluciones dominadas, superior al resto también, con 0 de sus soluciones dominadas.

El segundo mejor algoritmo fue el M3AS con el 31.78% de las soluciones del frente Pareto aproximado. Sin embargo, en la **cobertura** solo pudo dominar en promedio 0.8 soluciones, y 4.55 de sus soluciones fueron dominadas.

Por su parte, el MMA logro el tercer lugar, con el 31.18% del frente Pareto aproximado encontrado, quedando en estas circunstancias con el peor desempeño el MOA.

5.5.1.3 Resultados para el Grupo Multicast 3 – Baja Carga

Los resultados de las pruebas experimentales para el Grupo *Multicast 3* son mostrados en la Tabla 5.15 y presenta los promedios parciales de los resultados.

En estas pruebas el MOACS una vez más obtuvo mayor porcentaje de soluciones, es decir 73.76% de Y_{apr} . En la **cobertura** ninguna de sus soluciones fue dominada, alcanzando en promedio 4.48 soluciones dominar del resto de los algoritmos.

El M3AS fue el segundo mejor algoritmo con el 22.50% del frente Pareto aproximado. En el tercer lugar como se aprecia lo obtuvo el MOA al obtener el 17.33% del Y_{apr} superior al obtenido por el MMA con 16.50%, por lo cuál quedó último.

Tabla 5.15. Grupo Multicast 3 - Promedios Parciales a Baja Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	6.75	7	13.75	22.50	1.30	2.43	2
Y_{MOA}	5.20	7.50	12.70	17.33	1.13	3.77	3
Y_{MOACS}	22.10	4.75	26.85	73.76	4.48	0	1
Y_{MMA}	4.95	1.55	6.50	16.50	0.81	1.53	4

5.5.1.4 Resultados para el Grupo Multicast 4 – Baja Carga

Los resultados de las pruebas experimentales para el Grupo *Multicast* 4 son presentados en la Tabla 5.16 y representa los promedios parciales. Las pruebas para todos los tiempos de corrida son presentadas en las Tablas B1.16, B1.17, B1.18 y B1.19 en el Anexo B - Sección B1.

Tabla 5.16. Grupo Multicast 4 - Promedios Parciales a Baja Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	6.65	9.75	16.40	19.56	1.11	3.83	2
Y_{MOA}	4.65	4.20	8.85	13.68	1.84	2.89	3
Y_{MOACS}	18	5.55	23.55	52.94	2.48	0.94	1
Y_{MMA}	4.55	2.60	7.15	13.38	2.93	0.69	4

En esta última prueba, a baja carga que se caracteriza por tener el grupo *multicast* de mayor cantidad de destinos, el MOACS una vez más obtuvo mayor cantidad de soluciones pertenecientes al frente Pareto aproximado - Y_{apr} , con el 52.94%. Siendo el MMA el de menor porcentaje de soluciones halladas, con solo 13.38% de Y_{apr} . El segundo mejor algoritmo corresponde al M3AS con el 19.56% del frente Pareto aproximado, quedando tercero el MOA que logro encontrar el 13.68% de Y_{apr} . Cabe mencionar que en las coberturas el MMA pudo dominar 2.93 soluciones, en promedio, sobre los otros algoritmos. Además, solo 0.69 de sus soluciones fueron dominadas. Esto es debido a

que el MMA obtuvo en su frente Pareto calculado - Y_{MMA} solo 7.15 soluciones, menor al resto de los algoritmos.

5.5.1.5 Promedios Generales – Baja Carga

Los Promedios Generales a Baja Carga presentados en la Tabla 5.17 expresan una visión superior sobre el desempeño de los diferentes algoritmos a Baja Carga para los distintos grupos *multicast*. El MOACS sin lugar a dudas, fue de mejor desempeño, logrando el 72.51% de los frentes Pareto en promedio, muy superior al resto de los algoritmos. En la **cobertura** el MOACS también tuvo, como corresponde, gran dominancia al obtener 3.06 soluciones cubiertas y solo 0.24 de sus soluciones fueron dominadas.

El resto de los algoritmos lograron desempeños cercanos, quedando como segundo mejor algoritmo el M3AS que obtuvo 26.23% de los frentes Pareto, el MMA logró encontrar el 22.51% y finalmente, el MOA queda con el más bajo desempeño solo logrando 20.09% de soluciones de los frentes Pareto.

Tabla 5.17. Promedios Generales a Baja Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	6.66	6.05	12.71	26.23	0.89	3.27	2
Y_{MOA}	5.06	5.45	10.51	20.09	1.15	3.11	4
Y_{MOACS}	18.34	3.13	21.46	72.51	3.06	0.24	1
Y_{MMA}	5.58	1.99	7.56	22.51	2.28	0.76	3

5.5.2 Pruebas con Media Carga

En estas pruebas, la red se encuentra a media carga, donde los enlaces están inicialmente cargados en un rango entre 0.4 y 0.7 de utilización de enlace.

En las siguientes sub-secciones, son presentados los resultados de Promedios Parciales para los distintos tipos de grupos *multicast* así como el Promedio Generales a Media Carga.

5.5.2.1 Resultados para el Grupo Multicast 1 – Media Carga

Los resultados de las pruebas experimentales para el Grupo *Multicast* 1 están presentes en las Tablas B2.1, B2.2, B2.3 y B2.4 en el Anexo B - Sección B2. La Tabla 5.18 muestra los correspondientes promedios parciales.

Puede observarse que en estas pruebas el MOACS obtuvo el 96.70% del frente Pareto aproximado - Y_{apr} , siendo el mayor alcanzado. Por otra parte, el MMA es el de

menor desempeño con solo 31.17% de Y_{apr} , mientras que el M3AS y el MOA obtuvieron cercanas soluciones, quedando el M3AS en segundo lugar con 47.35% del frente Pareto Y_{apr} y el MOA con 47%.

En la **cobertura**, el MOACS obtuvo mayor dominancia sobre los otros algoritmos, con 5.73 soluciones dominadas. Nótese que ninguna de sus soluciones fue dominada.

Tabla 5.18. Grupo Multicast 1 - Promedios Parciales a Media Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	14.20	8.65	22.85	47.35	1.08	6.00	2
Y_{MOA}	14.10	8.50	22.60	47	1.69	5.43	3
Y_{MOACS}	29	1	30	96.70	5.73	0	1
Y_{MMA}	9.35	1.85	11	31.17	4.05	1.12	4

5.5.2.2 Resultados para el Grupo Multicast 2 – Media Carga

Los promedios parciales de las pruebas para el grupo *multicast 2* a media carga de todos los tiempos de corrida, son presentados en la Tabla 5.19.

Tabla 5.19. Grupo Multicast 2 - Promedios Parciales a Media Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	14.55	9	23.55	22.05	1.41	5.32	2
Y_{MOA}	14.30	8.75	23.05	21.66	1.46	5.35	3
Y_{MOACS}	35.50	6.15	41.65	53.78	4.86	1	1
Y_{MMA}	12.50	2.05	14.55	18.93	4.45	0.51	4

En estas pruebas, el MOACS alcanzó el 53.78% de las soluciones de Y_{apr} y una dominancia 4.86 soluciones. Estos resultados son superiores a los obtenidos por otros algoritmos. Por sus partes, el M3AS ocupa el segundo mejor lugar con 22.05% del frente Pareto aproximado - Y_{apr} , quedando MOA tercero con 21.66% y en último lugar el MMA, con solo 18.93%. Note que, a pesar de que el MMA es el de peor desempeño, pudo lograr una alta dominancia, mejor inclusive al del M3AS y el MOA, esto es porque el MMA generó pocas soluciones dominadas por el Y_{apr} .

5.5.2.3 Resultados para el Grupo Multicast 3 – Media Carga

La Tabla 5.20 presenta los promedios generales a media carga del grupo *multicast 3* correspondientes a todos los tiempos de corridas.

Como puede observarse, el MOACS fue el mejor algoritmo al encontrar el 43.25% del frente Pareto aproximado - Y_{apr} y dominar 6.40 soluciones en la **cobertura**. Estos números son superiores a los logrados por otros algoritmos. El MOA es el mejor segundo algoritmo, con 20.40% del frente Pareto aproximado - Y_{apr} , mientras que el M3AS alcanzó 20.16%, quedando de esta manera tercero. Finalmente, al MMA claramente le corresponde el último lugar por su pobre desempeño con 17.14% del frente Pareto aproximado - Y_{apr} .

Tabla 5.20. Grupo Multicast 3 - Promedios Parciales a Media Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	12.70	11.60	24.30	20.16	2.52	5.67	3
Y_{MOA}	12.85	9.05	21.90	20.40	2.82	6.08	2
Y_{MOACS}	27.25	9.50	36.75	43.25	6.40	2.25	1
Y_{MMA}	10.80	12.45	23.25	17.14	6.15	3.88	4

5.5.2.4 Resultados para el Grupo Multicast 4 – Media Carga

Los resultados de las pruebas experimentales para el Grupo *Multicast 4* están presentes en las Tablas B2.16, B2.17, B2.18 y B2.19 en el Anexo B - Sección B2. La Tabla 5.21 presenta los promedios parciales de dichos resultados.

Tabla 5.21. Grupo Multicast 4 - Promedios Parciales a Media Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	10.90	11.00	21.90	19.46	2.79	6.59	2
Y_{MOA}	8.30	11.70	20.00	14.82	2.79	8.46	3
Y_{MOACS}	16.15	17.45	33.60	28.84	8.92	0.92	1
Y_{MMA}	7.80	7.60	15.40	13.93	6.11	4.65	4

En estas pruebas, una vez más el MOACS demuestra su superioridad alcanzando el 28.84% de Y_{apr} . Además, dominó en promedio 8.92 soluciones con apenas 0.92 de sus soluciones dominadas.

El segundo mejor desempeño le corresponde al M3AS por hallar el 19.46% de Y_{apr} . Por su parte el MOA con 14.82% del frente Pareto aproximado - Y_{apr} se posiciona como tercer mejor algoritmo. Finalmente, el MMA queda último con por haber encontrado solo el 13.93% del frente Pareto aproximado - Y_{apr} .

5.5.2.5 Promedios Generales – Media Carga

Los promedios generales para media carga son presentados en la Tabla 5.22. Como en anteriores sub-secciones el objetivo de los Promedio Generales es obtener un panorama más claro del desempeño de los algoritmos propuesto.

Puede observarse, que el MOACS es claramente el mejor algoritmo debido a que obtuvo el 54.86% del frente Pareto aproximado - Y_{apr} . En la cobertura también fue superior al resto, dominando 6.48 soluciones en promedio y solo 1.04 de sus soluciones fueron dominadas.

El segundo mejor algoritmo es el M3AS con 26.87% del frente Pareto alcanzado, superior al MOA que con sus 25.59% de Y_{apr} queda con el mejor tercer desempeño logrado. Por su parte, el MMA con 20.04% de Y_{apr} encontrado, en promedio, queda con el peor desempeño. A pesar de esto, el MMA pudo cubrir más soluciones que el M3AS y el MOA, dominando 5.19 soluciones dominadas.

Tabla 5.22. Promedios Generales a Media Carga.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	13.09	10.06	23.15	26.87	1.95	5.89	2
Y_{MOA}	12.39	9.50	21.89	25.59	2.19	6.33	3
Y_{MOACS}	26.98	8.53	35.50	54.86	6.48	1.04	1
Y_{MMA}	10.11	5.99	16.10	20.04	5.19	2.54	4

Se puede concluir de estos promedios generales, que las nuevas propuestas (MOACS, M3AS & MOA) obtuvieron resultados alentadores al demostrar ser superiores al MMA.

5.5.3 Pruebas en Saturación

Estas pruebas en saturación, la red se encuentra inicialmente cargados en un rango entre 0.7 y 1 de utilización de enlace, esto implica que existen enlaces a su máxima capacidad de uso. Por lo cuál, no podrán ser utilizados como parte de las soluciones.

5.5.3.1 Resultados para el Grupo Multicast 1 – Saturación

Los resultados de las pruebas experimentales para el Grupo *Multicast* 1 en saturación de todos los tiempos de corrida son presentados en la Tabla 5.23.

Como el ranking indica, el MOACS tuvo el mejor desempeño en estas pruebas, con 35.65% del frente Pareto aproximado - Y_{apr} . Además, note que en la **cobertura** pudo dominar 5.37 soluciones, ampliamente superando al resto (ver columna **Dominada**).

Tabla 5.23. Grupo Multicast 1 - Promedios Parciales en Saturación.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	7.30	8.80	16.10	15.87	2.43	3.82	3
Y_{MOA}	7.40	8.35	15.75	16.09	2.03	5.55	2
Y_{MOACS}	16.40	1.60	18.00	35.65	5.37	1.43	1
Y_{MMA}	5.20	1.80	7.00	11.30	2.43	1.46	4

El MOA por su parte, obtuvo el segundo mejor desempeño con 16.09% de Y_{apr} , y cercanamente el M3AS con 15.87% se ubica en el tercer lugar. El MMA tuvo el más bajo desempeño, con solo 11.30% de Y_{apr} . Aunque el MMA tuvo menor desempeño, en la **cobertura** pudo dominar más soluciones del M3AS y el MOA. Este hecho es debido, a que estos últimos algoritmos tuvieron una gran cantidad de soluciones no pertenecientes al Y_{apr} en comparación al MMA.

5.5.3.2 Resultados para el Grupo Multicast 2 – Saturación

Los resultados de las pruebas experimentales para el Grupo *Multicast 2* están presentes en las Tablas B3.6, B3.7, B3.8 y B3.9 en el Anexo B - Sección B3. La Tabla 5.24 presenta los promedios parciales de dichos resultados, las cuáles se encuentran también en la Tabla B3.10.

Tabla 5.24. Grupo Multicast 2 - Promedios Parciales en Saturación.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	15.6	12.05	27.65	9.63	3.38	3.73	2
Y_{MOA}	13.7	10.55	24.25	8.46	1.89	7.02	3
Y_{MOACS}	75.8	9.2	85	46.79	6.20	1.75	1
Y_{MMA}	11	4.9	15.9	6.79	4.68	3.67	4

En estas pruebas, una vez más, el MOACS tuvo el mejor desempeño con 46.79% de soluciones en Y_{apr} , muy superior al resto de los algoritmos (ver columna $\%Y_{alg}$). Además, en las coberturas también logro 6.20 soluciones dominadas sobre el resto de los algoritmos, también superior.

El segundo mejor desempeño corresponde al M3AS con 9.63% de soluciones en Y_{apr} y el MOA con 8.46% queda como el tercer mejor algoritmo. Por su parte, el MMA tuvo un desempeño bajo en comparación a los ACOs, alcanzando 6.79% del Y_{apr} .

5.5.3.3 Resultados para el Grupo Multicast 3 – Saturación

Los resultados de las pruebas experimentales para el Grupo *Multicast* 3 están presentes en las Tablas B3.11, B3.12, B3.13 y B3.14 en el Anexo B - Sección B3. La Tabla 5.25 presenta los promedios parciales de dichos resultados.

Tabla 5.25. Grupo Multicast 3 - Promedios Parciales en Saturación.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	17.10	13.70	30.80	11.96	2.34	4.72	2
Y_{MOA}	15.65	15.15	30.80	10.94	2.12	7.23	3
Y_{MOACS}	63.60	14.45	78.05	44.48	6.06	2.50	1
Y_{MMA}	14.45	1.80	16.25	10.10	5.33	1.40	4

Como es indicado en el ranking, el MOACS en esta pruebas tuvo el mejor desempeño con 44.40% de Y_{apr} , muy superior al resto como puede observarse en la columna $\%Y_{alg}$.

Por su parte el segundo mejor desempeño, aunque muy inferior al MOACS, corresponde al M3AS con 11.96% de Y_{apr} , quedando como tercer mejor algoritmo el MOA que tuvo 10.94% de Y_{apr} . El peor desempeño fue del MMA que solo alcanzo el 10.10% de Y_{apr} . Note, que en todos los casos anteriores el MMA pudo dominar más soluciones del M3AS y del MOA. Este hecho es debido, como ya fue mencionado anteriormente, que el M3AS y el MOA generaron una gran cantidad de soluciones dominadas por el Y_{apr} , mientras que el frente Pareto calculado del MMA o sea Y_{MMA} tuvo pocas soluciones dominadas por Y_{apr} .

5.5.3.4 Resultados para el Grupo Multicast 4 – Saturación

Los resultados para todos los tiempos de corrida del Grupo *Multicast* 4 están presentes en las Tablas B3.16, B3.17, B3.18 y B3.19 en el Anexo B - Sección B3. En la Tabla 5.26 se muestran los promedios parciales de dichos resultados.

Estas pruebas tienen una connotación especial, porque el nivel de tráfico es el mayor y el grupo *multicast* a ser acomodado en la red también es de mayor tamaño. En este situación, no cabe dudas que una vez más el MOACS fue ampliamente superior al resto de los algoritmos. El MOACS obtuvo 42.74% de Y_{apr} muy superior al resto (ver columna

% Y_{alg}). Note que el segundo mejor algoritmo el M3AS, obtuvo el 10.13% de Y_{apr} muy inferior a los 42.74% del MOACS. Además, en la **cobertura** el MOACS también pudo lograr la mayor cobertura con 12.67 soluciones dominadas.

Tabla 5.26. Grupo Multicast 4 - Promedios Parciales en Saturación.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	% Y_{alg}	Dominada	Es-dominado	
Y_{M3AS}	11.65	18.05	29.70	10.13	4.28	8.23	2
Y_{MOA}	6.90	17.95	24.85	6.00	1.50	11.80	3
Y_{MOACS}	49.15	21.25	70.40	42.74	12.67	7.10	1
Y_{MMA}	3.95	6.50	10.45	3.43	12.52	3.83	4

Por su parte el M3AS, como ya fuera mencionado tuvo el segundo mejor desempeño con 10.13% de Y_{apr} , quedando en tercer lugar el MOA y último el MMA con un muy bajo desempeño de solo 3.43% de Y_{apr} . Sin embargo, el MMA tuvo gran dominancia debido que los frentes Pareto calculados de los ACOs fueron ampliamente superior, como puede observarse en la columna $|Y_{alg}|$ en la Tabla 5.26, por lo que las cantidades de soluciones dominadas por Y_{apr} tuvieron amplio rango de diferencia (ver columna $|Y_{apr} > Y_{alg}|$).

5.5.3.5 Promedios Generales – Saturación

Estos promedios generales darán una visión del desempeño de los algoritmos en el estado que una red se encuentre con altos niveles de tráfico, en la que se desee ubicar un grupo *multicast*.

Tabla 5.27. Promedios Generales en Saturación.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	% Y_{alg}	Dominada	Es-dominado	
Y_{M3AS}	12.91	13.15	26.06	11.90	3.11	5.12	2
Y_{MOA}	10.91	13.00	23.91	10.37	1.89	7.90	3
Y_{MOACS}	51.24	11.63	62.86	42.41	7.57	3.19	1
Y_{MMA}	8.65	3.75	12.40	7.91	6.24	2.59	4

Se puede concluir en base a los resultados expuestos en la Tabla 5.27, que con un alto tráfico en la red, el MOACS es el mejor algoritmo que puede hallar más opciones para acomodar el pedido *multicast*. El MOACS en promedio ha encontrado 42.41% de soluciones del los frentes Pareto, sin dejar de mencionar que en la cobertura tuvo mayor dominancia, con 7.57 soluciones dominadas en promedio. Como se observa en la columna % Y_{alg} el resultado obtenido por el MOACS es muy superior al resto.

Por su parte, el M3AS tuvo el segundo mejor desempeño con 11.90% soluciones de Y_{apr} encontradas, mientras que el MOA en promedio logro un resultado cercano con 10.37% quedando como el tercer mejor algoritmo. Finalmente, el MMA tuvo el más bajo desempeño con solo 7.91% del Y_{apr} .

5.5.4 Promedios Globales de las Pruebas Semi-dinámicas

Así como en las pruebas estáticas, también se presenta los resultados globales para las pruebas semi-dinámicas. Como puede recordarse, en las pruebas semi-dinámicas, a diferencia de las estáticas, la red se encuentra con tráfico de carga inicial al momento que se genera un pedido *multicast*. En estas circunstancias, los resultados globales mostrados en la Tabla 5.28, indican que sin lugar a dudas el MOACS es el algoritmo recomendado para la resolución del problema de enrutamiento *multicast*. El MOACS como puede observarse ha logrado en promedio hallar 56.59% soluciones de Y_{apr} (ver columna $\%Y_{alg}$ - fila Y_{MOACS}). Además, en la **cobertura** también fue superior, al dominar más soluciones, es decir, 5.71 soluciones dominadas.

Tabla 5.28. Promedios Globales de las Pruebas Semi-dinámicas.

	Comparación de Soluciones con Y_{apr}				Cobertura		Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\%Y_{alg}$	Dominada	Es-dominado	
Y_{M3AS}	10.89	9.75	20.64	21.66	1.98	4.76	2
Y_{MOA}	9.45	9.32	18.77	18.68	1.74	5.78	3
Y_{MOACS}	32.18	7.76	39.94	56.59	5.71	1.49	1
Y_{MMA}	8.11	3.91	12.02	16.82	4.57	1.96	4

Por su parte, el siguiente mejor algoritmo es el M3AS que pudo encontrar 21.66% de los frentes Pareto, mientras que el desempeño logrado por el MOA lo ubica en el tercer lugar. Finalmente, puede notarse que el MMA ocupa el último lugar con el peor desempeño, al encontrar solo 16.82% de las soluciones, en promedio, de los frentes Pareto. Note, que en las coberturas el MMA es superior al M3AS y el MOA, esto es debido que la cantidad de soluciones de los frentes Pareto calculado por los ACOs es superior al del MMA (ver columna $|Y_{alg}|$). Por tal motivo, la cantidad de soluciones que fueron dominadas por los frentes Pareto Y_{apr} debe ser mayor también, esto se verifica en la columna $|Y_{apr} > Y_{alg}|$.

5.6 Resumen del Capítulo

En este Capítulo fueron presentados los resultados de las comparaciones entre las soluciones arrojadas por los diversos algoritmos estudiados.

Las pruebas fueron clasificadas en dos grandes grupos: Pruebas Estáticas y Pruebas Semi-Dinámicas. En la primera prueba, la red se encuentra descargada y el objetivo es observar el funcionamiento de los algoritmos a muy baja carga. Los resultados de estas pruebas indican que en estas condiciones, los algoritmos propuestos (M3AS, MOA & MOACS) presentan un excelente desempeño, la cual es superior al MMA.

En estas pruebas, el MOACS fue el mejor algoritmo, quedando como segundo mejor algoritmo el M3AS, tercero el MOA y último el MMA en la escala de ranking.

Por otra parte, en las pruebas Semi-Dinámicas se realizaron pruebas con tres distintos niveles de carga (baja, media y en saturación) para los distintos grupos *multicast* con distintos tiempos de corridas.

Los resultados globales de las pruebas Semi-Dinámicas, indican que los algoritmos propuestos son, sin lugar a dudas, excelentes alternativas para resolución del problema de enrutamiento *multicast* multiobjetivo, debido que los ACOs obtuvieron, en promedio, mejores resultados que el MMA. En estas pruebas (Semi-Dinámicas), que se consideran la más importantes, cabe destacar que claramente el MOACS resultó ser el mejor algoritmo, obteniendo un desempeño muy superior al resto.

Se concluye en éste Capítulo, que las diversas propuestas basadas en ACO son excelentes alternativas de resolución del problema de ingeniería de tráfico *multicast*, donde el más recomendable de este conjunto de algoritmos, es el algoritmo *Multiobjective Ant Colony System* – MOACS.

Capítulo 6

Conclusiones Generales

Como culminación del presente trabajo, en este Capítulo son presentadas las conclusiones finales y son propuestos algunos tópicos como posibles trabajos futuros.

6.1 Conclusiones Finales

El problema de enrutamiento *multicast* ha sido tratado recientemente como un problema multiobjetivo utilizando algoritmos basados en *Algoritmos Evolutivos* - EAs, siendo este enfoque de resolución el más efectivo al compararlo con otras publicaciones que tomaron el mismo problema en un contexto mono-objetivo con restricciones.

Por otra parte, en años recientes ha surgido una novedosa metaheurística conocida como *Ant Colony Optimization* – ACO que fue exitosamente utilizada en varios problemas combinatoriales, y por primera vez, este trabajo propone utilizarla en la resolución del problema de “*enrutamiento multicast multiobjetivo*”, donde se optimizan simultáneamente cuatro funciones objetivos: utilización máxima de los enlaces α_T , costo del árbol C_T , retardo medio DA_T y máximo DM_T . Desde el punto de vista teórico, esto constituye un claro aporte a las Ciencias de Computación, pues se propone una nueva alternativa de solución basada en ACO. Cabe destacar que el problema solo ha sido resuelto en un contexto puramente multiobjetivo con algoritmos basados en EAs.

Para resolver el problema de enrutamiento *multicast*, se propuso re-estructurar tres de los mejores algoritmos basado en ACO: “*Max-Min Ant System* - MMAS”, “*Ómicron ACO* - OA” y el “*Multiobjective Ant Colony Optimization* - MOACS”. Como el MMAS y el OA son originalmente algoritmos mono-objetivos, este trabajo propone una versión multiobjetivo por primera vez, por lo cuál estos fueron nombrados “*Multiobjective Max-Min Ant System* – M3AS” y “*Multiobjective Ómicron ACO* - MOA” respectivamente. Además, el MOACS es utilizado por primera vez en la optimización de cuatro objetivos simultáneamente, para lo cuál fueron necesarios otros ajustes importantes en su estructura funcional. Desde el punto de vista de ACO, esto constituye un aporte significativo al proponer por primera vez una versión multiobjetivo de varios algoritmos hasta hoy mono-objetivos.

Los algoritmos propuestos fueron evaluados sobre varios grupos *multicast* de distintos tamaños en la red NTT tomadas de trabajos anteriores. Además, las pruebas se dividieron en Pruebas Estáticas y Semi-Dinámicas, en las cuales los algoritmos deben hallar un conjunto entero de soluciones Pareto óptimas. En las Pruebas Estáticas, la red se encuentra vacía y el objetivo es verificar el funcionamiento de los algoritmos a muy baja carga en la red. En las pruebas Semi-Dinámicas la red se encuentra con tráfico inicial. En estas pruebas se tomaron tres niveles de tráfico inicial: baja carga (0 - 0.4 de utilización de enlaces), media carga (0.4 - 0.7 de utilización de enlaces) y saturación (0.7 - 1 de utilización de enlaces).

Para validar las nuevas alternativas, éstas fueron comparadas con el “*Multiobjective Multicast Algorithm – MMA*”, el cuál es un algoritmo genético basado en un algoritmo evolutivo de segunda generación “*Strength Pareto Evolutionary Algorithm – SPEA*” diseñado exclusivamente para el problema de enrutamiento *multicast* en el contexto puramente multiobjetivo. Los resultados de las Pruebas Estáticas y Semi-Dinámicas son presentadas en los Anexos A y B respectivamente.

Se concluye finalmente en este trabajo, teniendo en cuenta los resultados de las extensas pruebas que fueron llevadas a cabo, que los diversos algoritmos propuestos basados en ACO (MOACS, M3AS & MOA) constituyen una nueva alternativa para la resolución del problema de enrutamiento *multicast*. Donde el MOACS es el algoritmo que se recomienda para el problema de enrutamiento *multicast*, por obtener el mejor desempeño en todas las pruebas realizadas.

6.2 Trabajos Futuros

De forma a continuar con el trabajo iniciado en esta tesis, los siguientes tópicos son propuestos como trabajos futuros:

1. Mejorar las nuevas propuestas con heurísticas de iniciación de población y de prevención de convergencia prematura.
2. Evaluación comparativa de M3AS, MOA, MOACS con otros algoritmos multiobjetivo ACOs y diversos MOEAs.
3. Aplicación de las propuestas al problema de enrutamiento *multicast* sobre redes ópticas.

4. Considerando que la optimización con más de cuatro funciones objetivos no es trivial en los enfoques ACOs, sería muy interesante optimizar mayor cantidad de funciones objetivos como: jitter, tamaño de paquetes encolados, números de saltos, etc.
5. Estudio e implementación de las nuevas propuestas en esquemas con múltiples árboles y múltiples flujos.

En resumen, existe una gran cantidad de temas abiertos que deben ser imperiosamente resueltos, de forma a seguir avanzando con el desarrollo de la Ingeniería de Tráfico Multicast.

Anexo A

Tablas de Pruebas Estáticas

Este anexo presenta los resultados de las Pruebas Estáticas llevadas a cabo sobre la red NTT. Las Pruebas Estáticas se caracterizan por estar la red descargada, es decir, no existe tráfico de datos. En estas condiciones, los algoritmos respondieron a un solo pedido *multicast* y en la optimización solo se tuvo en cuenta el costo del árbol, el retardo medio y retardo máximo de extremo a extremo.

Cada Tabla presenta el promedio sobre cinco pruebas distintas realizadas para cada grupo *multicast* y cada tiempo de corrida. En cada Tabla puede distinguirse dos bloque de pruebas llamados “**Comparaciones de Soluciones con Y_{apr}** ” y “**Cobertura**”.

En las “**Comparación de Soluciones con Y_{apr}** ” cada frente Pareto calculado por los algoritmos (Y_{M3AS} , Y_{MOA} , Y_{MOACS} & Y_{MMA}) es comparado con el frente Pareto aproximado - Y_{apr} , donde la notación siguiente es utilizada:

1. $|Y_{alg}|$ es el promedio del número de soluciones del frente Pareto calculado por cada algoritmo. Ejemplo para el M3AS, $|Y_{alg}| = |Y_{M3AS}|$.
2. $|Y_{apr} \cap Y_{alg}|$ es el promedio del número de soluciones que pertenecen a la vez al frente Pareto calculado por cada algoritmo Y_{alg} y también en el frente Pareto aproximado Y_{apr} .
3. $|Y_{apr} \succ Y_{alg}|$ es el promedio del número de soluciones del frente Pareto calculado por cada algoritmo Y_{alg} que son dominados por el frente Pareto aproximado Y_{apr} .
4. $\%Y_{apr}$ es el promedio del porcentaje de soluciones halladas del frente Pareto aproximado Y_{apr} , i.e. $100 \cdot (|Y_{apr} \cap Y_{alg}|) / (|Y_{apr}|)$.

En el bloque de “**Cobertura**” se compara el frente Pareto calculado por un algoritmo con el frente Pareto calculado de otro algoritmo.

Sea:

i el subíndice correspondiente a la fila, donde $i \in \{M3AS, MOA, MOACS, MMA\}$

j el subíndice correspondiente a la columna, donde $j \in \{M3AS, MOA, MOACS, MMA\}$

x_{ij} es un elemento de la tabla que representa la cantidad de soluciones que Y_i domina al de Y_j , i.e. $|Y_i \succ Y_j|$.

Teniendo en cuenta las notaciones anteriores, se define que:

Si $x_{ij} > x_{ji}$ o sea $|Y_i > Y_j| > |Y_j > Y_i|$, entonces el algoritmo i es mejor que el algoritmo j , en otro caso, si $x_{ij} < x_{ji}$ o sea $|Y_i > Y_j| < |Y_j > Y_i|$, entonces el algoritmo j es mejor que el algoritmo i .

A.1 Grupo Multicast 1 – Tablas A.1 a A.5.

Tabla A.1. Grupo Multicast 1 - Tiempo de corrida = 40 seg. – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.4	0.2	6.6	70		1.2	0	0
Y_{MOA}	6.4	1.2	7.6	70	0.2		0	0
Y_{MOACS}	9	0	9	100	0.2	1.2		0
Y_{MMA}	5.8	0	5.8	64	0.2	1.2	0	

Tabla A.2. Grupo Multicast 1 - Tiempo de corrida = 80 seg. – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	8.4	0	8.4	90		0	0	0
Y_{MOA}	7.8	0	7.8	80	0		0	0
Y_{MOACS}	9	0	9	100	0	0		0
Y_{MMA}	5.2	0	5.2	57	0	0	0	

Tabla A.3. Grupo Multicast 1 - Tiempo de corrida = 120 seg. – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	9	0	9	100		0.2	0	0
Y_{MOA}	7.6	0.2	7.8	84	0		0	0
Y_{MOACS}	9	0	9	100	0	0.2		0
Y_{MMA}	5.2	0	5.2	57	0	0	0	

Tabla A.4. Grupo Multicast 1 - Tiempo de corrida = 160 seg. – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	8.6	0	8.6	95		0.2	0	0
Y_{MOA}	7.4	0.2	7.6	82	0		0	0
Y_{MOACS}	9	0	9	100	0	0.2		0
Y_{MMA}	5.8	0	5.8	64	0	0.2	0	

Tabla A.5. Grupo Multicast 1 - Promedios Parciales – Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	8.8	0	8.8	88.75		0.1	0	0
Y_{MOA}	7.4	0.1	7.6	79	0		0	0
Y_{MOACS}	9	0	9	100	0	0.1		0
Y_{MMA}	5.8	0	5.8	60.5	0	0.067	0	

A.2 Grupo Multicast 2 – Tablas A.6 a A.10.

Tabla A.6. Grupo Multicast 2 - Tiempo de corrida = 40 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	9.8	5.4	15.2	54		2.2	0	0
Y_{MOA}	10	4.6	14.6	55	1.6		0	0
Y_{MOACS}	15.6	5.6	21.2	86	2.8	3.2		0
Y_{MMA}	9	3.2	12.2	50	1.6	2.2	0	

Tabla A.7. Grupo Multicast 2 - Tiempo de corrida = 80 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12.4	5.2	17.6	68		2	0	0.4
Y_{MOA}	10.8	5	15.8	60	0.8		0	0.2
Y_{MOACS}	16	5.6	21.6	88	1.8	2.6		0.4
Y_{MMA}	11.6	3.8	15.4	64	1.4	1.8	0	

Tabla A.8. Grupo Multicast 2 - Tiempo de corrida = 120 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	13.8	4.8	18.6	76		1.6	0	0.6
Y_{MOA}	12.8	4.2	17	71	0.6		0	0.4
Y_{MOACS}	16	6	22	89	0.8	1.6		0.6
Y_{MMA}	12.8	3.6	16.4	71	0.4	1.6	0.2	

Tabla A.9. Grupo Multicast 2 - Tiempo de corrida = 160 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	14.4	5.2	19.6	79		0.2	0	0.6
Y_{MOA}	13.4	4	17.4	74	0.4		0	0.6
Y_{MOACS}	16	6	22	89	1	0.6		0.8
Y_{MMA}	13.6	4	17.6	75	0.6	0.6	0	

Tabla A.10. Grupo Multicast 2 - Promedios Parciales – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	14.4	5.2	19.6	69.25		1.1	0	0.3
Y_{MOA}	11.7	4.5	17.2	65	0.4		0	0.3
Y_{MOACS}	16	6	22	88	1	0.6		0.4
Y_{MMA}	11.3	4	17.6	65	0.6	0.6	0	

A.3 Grupo Multicast 3 –Tablas A.11 a A.15.

Tabla A.11. Grupo Multicast 3 - Tiempo de corrida = 40 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.4	5.4	11.8	26		3.6	0	0
Y_{MOA}	5	4.8	9.8	20	2.6		0	0
Y_{MOACS}	16.4	0.4	16.8	68	5.2	4.6		0
Y_{MMA}	11	0	11	45	4.6	4.4	0	

Tabla A.12. Grupo Multicast 3 - Tiempo de corrida = 80 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12.8	0.8	13.2	52		3.8	0	0
Y_{MOA}	7.8	4.4	12.2	32	0.4		0	0
Y_{MOACS}	17.8	0.6	18.2	73	0.6	4.4		0
Y_{MMA}	11	0	11	45	0.2	3.8	0	

Tabla A.13. Grupo Multicast 3 - Tiempo de corrida = 120 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12.4	1.4	13.8	52		3.8	0	0
Y_{MOA}	9	4	13	37	1		0	0
Y_{MOACS}	19.4	0.2	19.6	80	1.4	4		0
Y_{MMA}	11.8	0	11.8	49	0.8	3.6	0	

Tabla A.14. Grupo Multicast 3 - Tiempo de corrida = 160 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	14.4	1	15.4	60		1.2	0	0.2
Y_{MOA}	10.6	1.6	12.2	44	0.6		0	0
Y_{MOACS}	19.4	0.4	19.8	80	1	1.6		0.4
Y_{MMA}	11.4	0.4	11.8	47	0.6	0.8	0	

Tabla A.15. Grupo Multicast 3 - Promedios Parciales – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	14.4	1	15.4	47.5		1.2	0	0.05
Y_{MOA}	8.2	2.8	12.6	33.25	0.8		0	0
Y_{MOACS}	19.4	0.4	19.8	75.25	1	2.8		0.1
Y_{MMA}	11.13	0.1	11.27	46.5	0.6	0.8	0	

A.4 Grupo Multicast 4 – Tablas A.16 a A.20.

Tabla A.16. Grupo Multicast 4 - Tiempo de corrida = 40 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	1.8	7.6	9.4	10		1.8	0	0.2
Y_{MOA}	2	7	9	11	4.4		0.2	0
Y_{MOACS}	8.4	4.8	13.2	46	7.2	6.2		0.2
Y_{MMA}	4	0.4	4.4	22	4.4	2.6	1.6	

Tabla A.17. Grupo Multicast 4 - Tiempo de corrida = 80 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	4.4	4.8	9.2	24		6.4	0	0.2
Y_{MOA}	3.6	7.8	11.4	20	2.2		0.2	0.2
Y_{MOACS}	12.2	2.6	14.8	67	4.2	7.8		0.4
Y_{MMA}	4.2	0.6	4.8	23	2.8	2.4	0.2	

Tabla A.18. Grupo Multicast 4 - Tiempo de corrida = 120 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	5.2	5.6	10.8	29		3.8	0	0.2
Y_{MOA}	4	5.4	9.4	22	2.4		0.2	0
Y_{MOACS}	12.2	3.6	15.8	68	4.6	5.4		1
Y_{MMA}	4.4	1.4	5.8	24	2.2	2.6	0	

Tabla A.19. Grupo Multicast 4 - Tiempo de corrida = 160 seg. – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	7.2	2.8	10	40		5.6	0	0.2
Y_{MOA}	3.8	7.2	11	21	0		0.2	0.2
Y_{MOACS}	14	2.6	16.6	78	2.4	7.2		0.8
Y_{MMA}	4.4	1.2	5.6	24	0.2	2.4	0.2	

Tabla A.20. Grupo Multicast 4 - Promedios Parciales – Pruebas Estáticas.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	7.2	2.8	10	25.75		5.6	0	0.2
Y_{MOA}	3.27	7.1	10	18.5	0		0.2	0.07
Y_{MOACS}	14	2.6	16.6	64.75	2.4	7.2		0.9
Y_{MMA}	4.2	1.2	5.6	23.25	0.2	2.4	0.1	

A.5 Promedios Generales de las Pruebas Estáticas

Para poder tener una visión general sobre los resultados arrojados por las Pruebas Estáticas, se presentan los Promedios Generales. La Tabla A.21 presenta dichos promedios y cada elemento de ésta representa el promedio de los elementos correspondientes de las Tablas Parciales de cada Grupo *Multicast* (Tablas A.5, A.10, A.15 y A.20). Además, note que la Tabla A.21 presenta una columna del ranking, que indica el desempeño de los algoritmos en función del % Y_{apr} . El mejor ranking es 1 y el peor 4.

Tabla A.21. Promedios Generales de las Pruebas Estáticas.

	Comparación de Soluciones con Y_{apr}				Cobertura				Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{apr} $	% Y_{apr}	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}	
Y_{M3AS}	9.21	3.14	12.35	58.44		2.35	0	0.16	2
Y_{MOA}	7.63	3.85	11.48	49.69	1.08		0.05	0.10	3
Y_{MOACS}	13.71	2.40	16.11	82.34	2.08	3.18		0.29	1
Y_{MMA}	8.20	1.16	9.36	49.27	1.25	1.89	0.14		4

En resumen, se observa que el MOACS fue el mejor algoritmo al obtener mayor cantidad de soluciones calculadas (16.11) en su frente Pareto calculado, donde 13.71 soluciones fueron de los frentes Pareto aproximados y solo 2.4 fueron dominadas. Esto corresponde al 82.34% de los frentes Pareto aproximados que fue alcanzado hallar. Por otra parte, en las coberturas logró dominar más soluciones que otros algoritmos, i.e.:

$$|Y_{MOACS} \succ Y_{M3AS}| = 2.08 > |Y_{M3AS} \succ Y_{MOACS}| = 0,$$

$$|Y_{MOACS} \succ Y_{MOA}| = 3.18 > |Y_{MOA} \succ Y_{MOACS}| = 0.05,$$

$$|Y_{MOACS} \succ Y_{MMA}| = 0.29 > |Y_{MMA} \succ Y_{MOACS}| = 0.14.$$

Brevemente mencionamos, que en segundo lugar se encuentra el M3AS alcanzando 58.44% de los frentes Pareto aproximados, logrando dominar más soluciones de MOA en la cobertura, esto es:

$$|Y_{M3AS} \succ Y_{MOA}| = 2.35 > |Y_{MOA} \succ Y_{M3AS}| = 1.08.$$

El MOA por su parte, se encuentra tercero en el ranking con 49.69% de los frentes Pareto aproximados, no logrando ganar a otro algoritmo en las coberturas.

Finalmente, el MMA obtuvo el cuarto lugar del ranking al obtener solo el 49.27% de los frentes Pareto aproximados, sin embargo, halló algunas soluciones de muy buena calidad por que logro dominar más soluciones pertenecientes al M3AS y MOA.

Anexo B

Tablas de Pruebas Semi-Dinámicas

El Anexo B presenta las pruebas realizadas sobre la red NTT, de 6Mbps de capacidad en los enlaces, con diferentes niveles de tráfico de carga inicial, para los distintos grupos *multicast* y tiempos de corrida. Este anexo queda dividido en tres partes: Pruebas a Baja Carga, Pruebas a Media Carga y Pruebas en Saturación.

Las *Pruebas a Baja Carga* se caracterizan por tener un tráfico de carga entre 0 y 0.4 de utilización en los enlaces. La respuesta *multicast* a ser acomodada en la red es de $\phi = 1$ Mbps.

Las *Pruebas a Media Carga* tienen por su parte, un tráfico inicial de carga entre 0.4 y 0.7 de utilización en los enlaces de la red. Con un pedido *multicast* $\phi = 1$ Mbps de tráfico a ser ubicado.

En las *Pruebas en Saturación* se encuentran los enlaces de la red entre 0.7 y 1 de utilización. Note que existen enlaces totalmente saturados, por lo cuál no podrán ser utilizados para la transmisión de nuevos datos. La respuesta *multicast* generada es de un tráfico igual a $\phi = 0.2$ Mbps.

Cada Tabla (exceptuando aquellas de Promedio Parciales y Globales) presenta el promedio de cinco corridas distintas para el mismo grupo *multicast* y el mismo tiempo de ejecución. Además, en cada Tabla puede distinguirse dos bloques, la “**Comparación de Soluciones con Y_{apr}** ” y la “**Cobertura**”.

En las “**Comparación de Soluciones con Y_{apr}** ” cada frente Pareto calculado por los algoritmos (Y_{M3AS} , Y_{MOA} , Y_{MOACS} & Y_{MMA}) es comparado con el frente Pareto aproximado Y_{apr} , donde la notación siguiente es utilizada:

1. $|Y_{alg}|$ es el promedio del número de soluciones del frente Pareto calculado por cada algoritmo. Ejemplo para el M3AS, $|Y_{alg}| = |Y_{M3AS}|$.
2. $|Y_{apr} \cap Y_{alg}|$ es el promedio del número de soluciones que pertenecen a la vez al frente Pareto calculado por cada algoritmo Y_{alg} y también en el frente Pareto aproximado Y_{apr} .
3. $|Y_{apr} \succ Y_{alg}|$ es el promedio del número de soluciones del frente Pareto calculado por cada algoritmo Y_{alg} que son dominados por el frente Pareto aproximado Y_{apr} .
4. $\%Y_{apr}$ es el promedio del porcentaje de soluciones halladas del frente Pareto aproximado Y_{apr} , i.e. $100 \cdot (|Y_{apr} \cap Y_{alg}|) / (|Y_{apr}|)$.

En el bloque de “Cobertura” se compara el frente Pareto calculado por un algoritmo con el frente Pareto calculado de otro algoritmo.

Sea:

i el subíndice correspondiente a la fila, donde $i \in \{M3AS, MOA, MOACS, MMA\}$

j el subíndice correspondiente a la columna, donde $j \in \{M3AS, MOA, MOACS, MMA\}$

x_{ij} es un elemento de la tabla que representa la cantidad de soluciones que Y_i domina al de Y_j , i.e. $|Y_i \succ Y_j|$.

Teniendo en cuenta las notaciones anteriores, se define que:

Si $x_{ij} > x_{ji}$ o sea $|Y_i \succ Y_j| > |Y_j \succ Y_i|$, entonces el algoritmo i es mejor que el algoritmo j , en otro caso, si $x_{ij} < x_{ji}$ o sea $|Y_i \succ Y_j| < |Y_j \succ Y_i|$, entonces el algoritmo j es mejor que el algoritmo i .

B1. Pruebas a Baja Carga.

B1.1. Grupo Multicast 1 – Tablas B1.1 a B1.5.

Tabla B1.1. Grupo Multicast 1 - Tiempo de corrida = 40 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	3.6	2.6	6.2	24		1.2	0	0
Y_{MOA}	2	4.4	6.4	13.3	1.6		0	0
Y_{MOACS}	13	1	14	86.6	2.6	4.2		1
Y_{MMA}	8.4	1	9.4	56	2.6	4.2	0	

Tabla B1.2. Grupo Multicast 1 - Tiempo de corrida = 80 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	4.4	2.2	6.6	29.3		0.6	0	0
Y_{MOA}	3.6	4	7.6	24	0.8		0	0
Y_{MOACS}	13.6	1	14.6	90.7	2.2	3.6		1
Y_{MMA}	4.6	1.6	6.2	30.7	2.4	3.8	0	

Tabla B1.3. Grupo Multicast 1 - Tiempo de corrida = 120 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	5	2.4	7.4	33.3		1.2	0	0
Y_{MOA}	4.4	3.8	8.2	29.3	1		0	0
Y_{MOACS}	13.8	1	14.8	92	2.4	3.2		1
Y_{MMA}	3	1.4	4.4	20	7	5	0	

Tabla B1.4. Grupo Multicast 1 - Tiempo de corrida = 160 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	5.6	2.6	8.2	37.3		1	0	0
Y_{MOA}	4.6	4.4	9	30.67	1.2		0	0
Y_{MOACS}	14	1	15	90.3	2.4	5.2		1
Y_{MMA}	1.2	1	2.2	8	1	7.8	0	

Tabla B1.5. Grupo Multicast 1 - Promedios Parciales a Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	4.53	2.45	7.1	33.3		1	0	0
Y_{MOA}	4.2	4.2	7.4	24.43	1.2		0	0
Y_{MOACS}	13.7	1	14.7	89.2	2.4	4.05		1
Y_{MMA}	4.73	1.25	5.55	30.7	2.5	5.27	0	

B1.2. Grupo Multicast 2 – Tablas B1.6 a B1.10.

Tabla B1.6. Grupo Multicast 2 - Tiempo de corrida = 40 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	5.2	5.6	10.8	19.2		2.6	0	0.2
Y_{MOA}	4.6	5.8	10.4	17	3.2		0	0.2
Y_{MOACS}	18.4	1.6	20	68	5	5		0.6
Y_{MMA}	11.4	2.4	13.8	42	4.6	1	0	

Tabla B1.7. Grupo Multicast 2 - Tiempo de corrida = 80 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	8.2	5.2	13.4	30.4		2.4	0	0.2
Y_{MOA}	6.4	6	12.4	23.7	2.8		0	0.4
Y_{MOACS}	19.8	1.2	20.1	73.3	4.4	4.4		0.4
Y_{MMA}	10	2.6	12.6	37	5.6	0.2	0	

Tabla B1.8. Grupo Multicast 2 - Tiempo de corrida = 120 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	9.8	5	14.8	36.3		2	0	0.4
Y_{MOA}	7	6.2	13.2	25.9	3		0	0.4
Y_{MOACS}	20.2	1	21.2	74.8	4	3		0.4
Y_{MMA}	7.8	2.6	10.4	28.8	8.2	1.2	0	

Tabla B1.9. Grupo Multicast 2 - Tiempo de corrida = 160 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	11.2	4.2	15.4	41.5		1.4	0	0.4
Y_{MOA}	9	5.8	14.8	33.3	2.8		0	2
Y_{MOACS}	20.2	1	21.2	74.8	3.2	3		0.4
Y_{MMA}	4.8	2.6	7.6	17.8	7.8	2	0	

Tabla B1.10. Grupo Multicast 2 - Promedios Parciales a Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	8.6	5	13.6	31.85		2.1	0	0.3
Y_{MOA}	6.75	5.95	12.7	24.975	2.95		0	0.75
Y_{MOACS}	19.65	1.2	20.625	72.725	4.15	3.85		0.45
Y_{MMA}	8.5	2.55	11.1	31.4	6.55	1.1	0	

B1.3. Grupo Multicast 3 – Tablas B1.11 a B1.15.

Tabla B1.11. Grupo Multicast 3 - Tiempo de corrida = 40 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6	5.2	11.2	20		2.4	0	0.4
Y_{MOA}	5.2	6.4	11.6	17	2.8		0	0.4
Y_{MOACS}	19.8	5.8	25.6	66	5.2	6.2		0.6
Y_{MMA}	4.6	2	6.6	15	2	2	0	

Tabla B1.12. Grupo Multicast 3 - Tiempo de corrida = 80 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.2	6	12.2	20.6		2.2	0	0.6
Y_{MOA}	4.8	7.6	11.4	16	3.2		0	0.6
Y_{MOACS}	21	6	27	70	2.6	7.2		6
Y_{MMA}	4.8	1.6	5.4	16	0.9	1	0	

Tabla B1.13. Grupo Multicast 3 - Tiempo de corrida = 120 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	8.6	7.8	16.4	28.6		3.2	0	1
Y_{MOA}	6	7.4	13.4	20	2		0	0.7
Y_{MOACS}	23.4	3.8	27.2	78	3	7		3
Y_{MMA}	6	1.4	7.4	20	1	2	0	

Tabla B1.14. Grupo Multicast 3 - Tiempo de corrida = 160 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.2	9	15.2	20.6		3.8	0	2
Y_{MOA}	4.8	8.6	12.4	16	2.8		0	1
Y_{MOACS}	24.2	3.4	27.6	80.7	3	8		2
Y_{MMA}	4.4	1.2	5.6	14.6	0.6	0.2	0	

Tabla B1.15. Grupo Multicast 3 - Promedios Parciales a Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.75	7	13.75	22.45		2.9	0	1
Y_{MOA}	5.2	7.5	12.2	17.25	2.7		0	0.675
Y_{MOACS}	22.1	4.75	26.85	73.675	3.45	7.1		2.9
Y_{MMA}	4.95	1.55	6.25	16.4	1.125	1.3	0	

B1.4. Grupo Multicast 4 – Tablas B1.16 a B1.20.

Tabla B1.16. Grupo Multicast 4 - Tiempo de corrida = 40 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.6	8.6	15.2	20		2.6	0	0.4
Y_{MOA}	5	2.4	7.4	15.2	4.2		0	1
Y_{MOACS}	15.6	5.6	21.2	47.3	6	3		1
Y_{MMA}	5	3.4	8.4	15.2	3	2.6	3	

Tabla B1.17. Grupo Multicast 4 - Tiempo de corrida = 80 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.8	9.6	16.4	20.6		2.4	0	0.3
Y_{MOA}	4.2	2.4	6.6	12.7	3.6		0	0.3
Y_{MOACS}	17.2	6.4	23.6	52	3	2.9		0.6
Y_{MMA}	5.6	2.6	8.2	17	5	2.1	4	

Tabla B1.18. Grupo Multicast 4 - Tiempo de corrida = 120 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	7.6	10	17.6	23		3.4	0	0.6
Y_{MOA}	4.2	5.8	10	12.7	5		0	1
Y_{MOACS}	19	5.4	24.4	57.6	2	3		0.2
Y_{MMA}	4.2	2.6	6.8	12.7	3	2	2.3	

Tabla B1.19. Grupo Multicast 4 - Tiempo de corrida = 160 seg. – Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	5.6	10.8	16.4	17		2.6	0	1
Y_{MOA}	5.2	6.2	11.4	15.7	6		0	1
Y_{MOACS}	20.2	4.8	25	61	4	3.1		0.9
Y_{MMA}	3.4	1.8	5.2	10	1.2	5	2	

Tabla B1.20. Grupo Multicast 4 - Promedios Parciales a Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.65	9.75	16.4	20.15		2.75	0	0.575
Y_{MOA}	4.65	4.2	8.85	14.075	4.7		0	0.825
Y_{MOACS}	18	5.55	23.55	54.475	3.75	3		0.675
Y_{MMA}	4.55	2.6	7.15	13.725	3.05	2.925	2.825	

B1.5. Promedios Generales a Baja Carga

La Tabla B1.21 presenta los Promedios Generales a Baja Carga, donde cada elemento de ésta representa el promedio de los elementos correspondientes de las Tablas Parciales de cada Grupo *Multicast* (Tablas B1.5, B1.10, B1.15 y B1.20).

Tabla B1.21. Promedios Generales a Baja Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.66	6.05	12.71	26.23		2.19	0	0.47
Y_{MOA}	5.06	5.45	10.51	20.09	2.88		0	0.56
Y_{MOACS}	18.34	3.13	21.46	72.51	3.44	4.50		1.26
Y_{MMA}	5.58	1.99	7.56	22.51	3.49	2.63	0.71	

B2. Pruebas a Media Carga.

B2.1. Grupo Multicast 1 – Tablas B2.1 a B2.5.

Tabla B2.1. Grupo Multicast 1 - Tiempo de corrida = 40 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	10.4	9.4	19.8	34.7		4.8	0	0.6
Y_{MOA}	12.4	8.2	20.6	41.3	4.4		0	0.3
Y_{MOACS}	29	1	30	96.7	9.2	7.6		1
Y_{MMA}	10.8	1.4	12.2	36	9.2	7	0	

Tabla B2.2. Grupo Multicast 1 - Tiempo de corrida = 80 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	13.8	8.6	22.4	46		2.8	0	0.8
Y_{MOA}	13.4	8.4	21.8	44.7	4.6		0	0.2
Y_{MOACS}	29	1	30	96.7	7.6	7.4		2
Y_{MMA}	10.8	1.8	12.6	36	6.4	6	0	

Tabla B2.3. Grupo Multicast 1 - Tiempo de corrida = 120 seg. – Media Carga

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	15.6	8	23.6	52		1.4	0	0.6
Y_{MOA}	14.6	8.6	23.2	48.7	4.8		0	0.5
Y_{MOACS}	29	1	30	96.7	7.6	7.6		3
Y_{MMA}	8.6	2.2	10.8	28.7	4.4	7.6	0	

Tabla B2.4. Grupo Multicast 1 - Tiempo de corrida = 160 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	17	8.6	25.6	56.7		1.2	0	0.7
Y_{MOA}	16	8.8	24.8	53.3	4.8		0	0.7
Y_{MOACS}	29	1	30	96.7	5	7.8		3
Y_{MMA}	7.2	2	8.4	24	4	4	0	

Tabla B2.5. Grupo Multicast 1 - Promedios Parciales a Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	14.2	8.65	22.85	47.35		2.55	0	0.675
Y_{MOA}	14.1	8.5	22.6	47	4.65		0	0.425
Y_{MOACS}	29	1	30	96.7	7.35	7.6		2.25
Y_{MMA}	9.35	1.85	11	31.175	6	6.15	0	

B2.2. Grupo Multicast 2 – Tablas B2.6 a B2.10.

Tabla B2.6. Grupo Multicast 2 - Tiempo de corrida = 40 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	15.8	8	24.6	24		2.8	0	0.6
Y_{MOA}	15.8	9	24.8	24	3.8		0.6	1
Y_{MOACS}	28.4	6.4	34.8	43	6.8	5		0.2
Y_{MMA}	16.4	3.2	19.4	24.8	6.8	8.2	5.6	

Tabla B2.7. Grupo Multicast 2 - Tiempo de corrida = 80 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	18.2	9.8	28	27.5		5	0	0.5
Y_{MOA}	18.8	8.2	27	28.4	3.2		1	0.3
Y_{MOACS}	36.2	6.2	42.4	54.8	8.6	7.2		0.4
Y_{MMA}	11.4	2.6	14	17.3	7	5.8	3	

Tabla B2.8. Grupo Multicast 2 - Tiempo de corrida = 120 seg. – Media Carga

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	11.4	9	19.4	17.3		3.4	0	0.6
Y_{MOA}	10.4	8.4	18.8	15.7	3		0.2	0.2
Y_{MOACS}	37.4	6.6	44	56.7	7.8	6		0.6
Y_{MMA}	10.2	1.4	11.6	15.5	3.6	3.6	0.6	

Tabla B2.9. Grupo Multicast 2 - Tiempo de corrida = 160 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12.8	9.2	22	19.4		3.6	0	0.4
Y_{MOA}	12.2	9.4	21.6	18.5	3.2		0.4	0.6
Y_{MOACS}	40	5.4	45.4	60	5	10		0.7
Y_{MMA}	12	1	13	18	5	3.6	0.6	

Tabla B2.10. Grupo Multicast 2 - Promedios Parciales a Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	14.55	9	23.5	22.05		3.7	0	0.525
Y_{MOA}	14.3	8.75	23.05	21.65	3.3		0.55	0.525
Y_{MOACS}	35.5	6.15	41.65	53.625	7.05	7.05		0.475
Y_{MMA}	12.5	2.05	14.5	18.9	5.6	5.3	2.45	

B2.3. Grupo Multicast 3 – Tablas B2.11 a B2.15.

Tabla B2.11. Grupo Multicast 3 - Tiempo de corrida = 40 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	11.4	10.2	21.6	18		4.4	0	3
Y_{MOA}	11.8	7.8	19.6	18.7	3		0	5
Y_{MOACS}	20.8	10.6	39.4	33	9.2	7.6		2
Y_{MMA}	10.8	13.2	24	17	9	7	5	

Tabla B2.12. Grupo Multicast 3 - Tiempo de corrida = 80 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12	11.6	23.6	19		4.8	0	2
Y_{MOA}	12.6	8.4	21	20	3.2		0	6
Y_{MOACS}	26	9.2	35.2	41.3	7.6	7.4		2
Y_{MMA}	10.8	12.2	23	17	6.4	7	6	

Tabla B2.13. Grupo Multicast 3 - Tiempo de corrida = 120 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	13.2	12.4	25.6	21		6	0	0.6
Y_{MOA}	13.2	9.8	22.2	21	2.8		0	4
Y_{MOACS}	30.2	9.4	39.6	48	7	7.6		4
Y_{MMA}	10.8	12.2	23	17	4.4	5	12	

Tabla B2.14. Grupo Multicast 3 - Tiempo de corrida = 160 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	14.2	12.2	26.4	22.5		4.4	0	5
Y_{MOA}	13.8	10.2	24	21.9	3.8		0	6
Y_{MOACS}	32	8.8	40.8	50.8	7.6	7.8		7
Y_{MMA}	10.8	12.2	23	17	4	4	4	

Tabla B2.15. Grupo Multicast 3 - Promedios Parciales a Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12.75	11.6	24.3	20.125		4.9	0	2.65
Y_{MOA}	12.85	9.05	21.7	20.4	3.2		0	5.25
Y_{MOACS}	27.25	9.5	38.75	43.275	7.85	7.6		3.75
Y_{MMA}	10.8	12.45	23.25	17	5.95	5.75	6.75	

B2.4. Grupo Multicast 4 – Tablas B2.16 a B2.20.

Tabla B2.16. Grupo Multicast 4 - Tiempo de corrida = 40 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	10.4	8.6	19	18.5		3.6	0	5
Y_{MOA}	6.2	11	16.2	11	5.2		0	3
Y_{MOACS}	16.2	12.4	28.6	28.9	9.2	10.8		8
Y_{MMA}	13.2	8.8	22	23.6	9	10	2	

Tabla B2.17. Grupo Multicast 4 - Tiempo de corrida = 80 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	10.4	12.8	23.2	18.6		6.4	0	4
Y_{MOA}	8.4	10.2	18.6	15	4.6		0	3
Y_{MOACS}	17.6	16	33.6	31.4	7.6	10.2		7
Y_{MMA}	7.2	7.2	14.4	12.9	6.2	8	2	

Tabla B2.18. Grupo Multicast 4 - Tiempo de corrida = 120 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	11.2	11.4	22.6	20		3.8	0	6
Y_{MOA}	7.6	12.2	19.8	13.6	4.8		0	2
Y_{MOACS}	16.4	19.4	35.8	29.3	7	12.2		5
Y_{MMA}	5.4	7.2	12.6	9.6	4.4	9	4	

Tabla B2.19. Grupo Multicast 4 - Tiempo de corrida = 160 seg. – Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	11.6	11.2	22.8	20.7		3.8	0	2
Y_{MOA}	11	13.4	24.4	19.6	6.6		0	3
Y_{MOACS}	14.4	22	36.4	25.7	7.6	13.4		7
Y_{MMA}	5.4	7.2	12.6	9.6	4	10	4	

Tabla B2.20. Grupo Multicast 4 - Promedios Parciales a Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	10.9	11	21.9	19.45		4.4	0	4.25
Y_{MOA}	8.3	11.7	19.75	14.8	5.3		0	2.75
Y_{MOACS}	16.15	17.45	33.6	28.825	7.85	11.65		6.75
Y_{MMA}	7.8	7.6	15.4	13.925	5.9	9.25	3	

B2.5. Promedios Generales a Media Carga

La Tabla B2.21 presenta los Promedios Generales a Media Carga, donde cada elemento de ésta representa el promedio de los elementos correspondientes de las Tablas Parciales de cada Grupo *Multicast* (Tablas B2.5, B2.10, B2.15 y B2.20).

Tabla B2.21. Promedios Generales a Media Carga.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	13.09	10.06	23.15	26.87		3.93	0	1.92
Y_{MOA}	12.39	9.50	21.89	25.59	4.14		0.14	2.28
Y_{MOACS}	26.98	8.53	35.50	54.86	7.58	8.44		3.42
Y_{MMA}	10.11	5.99	16.10	20.04	5.96	6.63	2.99	

B3. Pruebas con Alta Carga - Saturación.

B3.1. Grupo Multicast 1 – Tablas B3.1 a B3.5.

Tabla B3.1. Grupo Multicast 1 - Tiempo de corrida = 40 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	8.8	9.8	18.6	19		5.2	0	0.6
Y_{MOA}	10	8.8	18.8	21.7	1.6		3	1
Y_{MOACS}	15.8	2.2	18	34.3	5.8	8.6		2
Y_{MMA}	8.2	1.6	9.8	14.6	6	3	1	

Tabla B3.2. Grupo Multicast 1 - Tiempo de corrida = 80 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.8	9.6	16.4	12.2		6.6	0	0.9
Y_{MOA}	7.6	7.4	15	13.6	0.8		2	1
Y_{MOACS}	16.6	1.4	18	29.6	7.4	7		1.2
Y_{MMA}	6.4	1.4	7.8	14.6	2	2	0.2	

Tabla B3.3. Grupo Multicast 1 - Tiempo de corrida = 120 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	6.4	8.4	14.8	14.6		6	0	2
Y_{MOA}	5.2	8.4	13.6	9.3	1.6		3	2
Y_{MOACS}	16.6	1.4	18	29.5	7.2	8		1.3
Y_{MMA}	3.6	2	5.6	6.4	3	2	2	

Tabla B3.4. Grupo Multicast 1 - Tiempo de corrida = 160 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	7.2	7.4	14.6	12.8		5.8	0	2
Y_{MOA}	6.8	8.8	15.6	12.2	1.4		4	3
Y_{MOACS}	16.6	1.4	18	29.5	7	8.4		0.5
Y_{MMA}	2.6	2.2	4.8	4.6	2	4	1.9	

Tabla B3.5. Grupo Multicast 1 - Promedios Parciales en Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	7.3	8.8	16.1	14.65		5.9	0	1.375
Y_{MOA}	7.4	8.35	15.75	14.2	1.35		3	1.75
Y_{MOACS}	16.4	1.6	18	30.725	6.85	8		1.25
Y_{MMA}	5.2	1.8	7	10.05	3.25	2.75	1.275	

B3.2. Grupo Multicast 2 – Tablas B3.6 a B3.10.

Tabla B3.6. Grupo Multicast 2 - Tiempo de corrida = 40 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	16	12.6	28.6	9.8		5.2	0	5
Y_{MOA}	15.2	9.2	24.4	9.4	0.6		0.6	6
Y_{MOACS}	65.6	13.2	78.8	40.5	7	8.6		3.4
Y_{MMA}	14	6.2	20.2	8.6	2	6.8	5	

Tabla B3.7. Grupo Multicast 2 - Tiempo de corrida = 80 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	18.8	13	31.8	11.6		6.6	0	4
Y_{MOA}	17.2	10.6	27.8	10.6	0.6		0.8	3
Y_{MOACS}	74.6	8.8	83.4	46	8	7		3.4
Y_{MMA}	13	5.2	18.2	8	2	6.2	4	

Tabla B3.8. Grupo Multicast 2 - Tiempo de corrida = 120 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12.4	11	23.4	7.7		6	0	5
Y_{MOA}	10.4	11.2	21.6	6.4	1.4		0.8	4
Y_{MOACS}	80.6	7.6	88.2	49.8	7.4	8		3.4
Y_{MMA}	10.2	5.2	15.4	6.3	2.4	7	5	

Tabla B3.9. Grupo Multicast 2 - Tiempo de corrida = 160 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	15.2	11.6	26.8	9.4		5.8	0	3
Y_{MOA}	12	11.2	23.2	7.4	1.6		0.8	3
Y_{MOACS}	82.4	7.2	89.6	50.9	9	8.4		0.8
Y_{MMA}	6.8	3	9.8	4.2	3.2	8.6	4	

Tabla B3.10. Grupo Multicast 2 - Promedios Parciales en Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	15.6	12.05	27.65	9.625		5.9	0	4.25
Y_{MOA}	13.7	10.55	24.25	8.45	0.93		0.75	4
Y_{MOACS}	75.8	9.2	85	46.8	7.85	8		2.75
Y_{MMA}	11	4.9	15.9	6.775	2.4	7.15	4.5	

B3.3. Grupo Multicast 3 –Tablas B3.11 a B3.15.

Tabla B3.11. Grupo Multicast 3 - Tiempo de corrida = 40 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	11.8	11.2	23	8.3		5	0	2
Y_{MOA}	11.6	12.8	24.4	8	12.4		0	1.9
Y_{MOACS}	48.6	20.4	69	34	5.8	9		1.5
Y_{MMA}	20.2	2	22.2	14	2	3.6	11	

Tabla B3.12. Grupo Multicast 3 - Tiempo de corrida = 80 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	24.2	12.6	36.8	16.9		4.6	0	0.9
Y_{MOA}	12.8	14.2	27	9	2.6		0	0.8
Y_{MOACS}	60.6	15.8	76.4	42.4	7.4	10.2		1
Y_{MMA}	18	1.4	19.4	12.6	2	5	5	

Tabla B3.13. Grupo Multicast 3 - Tiempo de corrida = 120 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	15.6	14.8	30.4	10.9		5.4	0	1.6
Y_{MOA}	23.6	16.4	40	16.5	3.2		0	1
Y_{MOACS}	70.2	11.8	82	49	7.2	10.8		1
Y_{MMA}	11.6	1.6	13.2	8	2.4	8.8	6	

Tabla B3.14. Grupo Multicast 3 - Tiempo de corrida = 160 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	16.8	16.2	33	11.7		6.6	0	2
Y_{MOA}	14.6	17.2	31.8	10	1.4		0	2.1
Y_{MOACS}	75	9.8	84.8	52.5	7	10.8		1
Y_{MMA}	8	2.2	10.2	5.6	3.2	7	8	

Tabla B3.15. Grupo Multicast 3 - Promedios Parciales – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	17.1	13.7	30.8	11.95		5.4	0	1.625
Y_{MOA}	15.65	15.15	30.8	10.875	4.9		0	1.45
Y_{MOACS}	63.6	14.45	78.05	44.475	6.85	10.2		1.125
Y_{MMA}	14.45	1.8	16.25	10.05	2.4	6.1	7.	

B3.4. Grupo Multicast 4 – Tablas B3.16 a B3.20.

Tabla B3.16. Grupo Multicast 4 - Tiempo de corrida = 40 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	10.6	15.8	26.4	9.2		7.2	0	7
Y_{MOA}	8.4	13.8	22.2	7.3	1.4		0.2	5
Y_{MOACS}	37.4	24.2	61.6	32.5	12.8	13.6		7
Y_{MMA}	1.6	8.4	10	1.2	7.8	6.8	13	

Tabla B3.17. Grupo Multicast 4 - Tiempo de corrida = 80 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	11.2	16.6	27.8	9.7		8	0	3
Y_{MOA}	6.4	17.8	24.2	5.6	2		0	2
Y_{MOACS}	47.4	22	69.4	41	13	17.6		5
Y_{MMA}	3.4	7.4	10.8	3	8	8	16	

Tabla B3.18. Grupo Multicast 4 - Tiempo de corrida = 120 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12	19.6	31.6	10.4		11.6	0	1
Y_{MOA}	5.8	20	25.8	5	1.8		0	1
Y_{MOACS}	53.6	20.6	74.2	46.6	16.2	19.8		4
Y_{MMA}	4.4	5.8	10.2	3.8	11.2	8.2	35	

Tabla B3.19. Grupo Multicast 4 - Tiempo de corrida = 160 seg. – Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12.8	20.2	33	11		11.6	0	2
Y_{MOA}	7	20.2	27.2	6	1.6		0	3
Y_{MOACS}	58.2	18.2	76.4	50.7	17	20		6
Y_{MMA}	6.4	4.4	10.8	5.7	6	9.2	21	

Tabla B3.20. Grupo Multicast 4 - Promedios Parciales en Saturación.

Comparación de Soluciones con Y_{apr}					Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} > Y_{alg} $	$ Y_{alg} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	11.65	18.05	29.7	10.075		9.6	0	3.25
Y_{MOA}	6.9	17.95	24.85	5.975	1.7		0.05	2.75
Y_{MOACS}	49.15	21.25	70.4	40.2	14.75	17.75		5.5
Y_{MMA}	3.95	6.5	10.45	3.425	8.25	8.05	21.25	

B3.5. Promedios Generales en Saturación

La Tabla B3.21 expone los Promedios Generales en estado de Saturación de la red, donde cada elemento de ésta indica el promedio de los elementos correspondientes de las Tablas Parciales de cada Grupo *Multicast* (Tablas B3.5, B3.10, B3.15 y B3.20).

Tabla B3.21. Promedios Generales en Saturación.

	Comparación de Soluciones con Y_{apr}				Cobertura			
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}
Y_{M3AS}	12.91	13.15	26.06	11.90		6.7	0	2.63
Y_{MOA}	10.91	13	23.91	10.37	2.22		0.95	2.49
Y_{MOACS}	51.24	11.63	62.86	42.41	9.08	10.99		2.66
Y_{MMA}	8.65	3.75	12.40	7.91	4.08	6.01	8.63	

B4. Promedios Globales de las Pruebas Semi-Dinámicas

El objetivo de los Promedios Globales presentados en esta sección, es definir una visión compacta de los extensos resultados que fueron arrojados en estas pruebas, de forma a dar una clasificación del desempeño de los algoritmos.

La Tabla B4 presenta los resultados promediados de todos los distintos resultados para los distintos grupos *multicast*, en los variados tiempos de corrida y todos los escenarios de carga inicial de la red. Además, una columna que define el **Ranking** obtenido en función del porcentaje de soluciones aportadas al frente Pareto aproximado es dada.

Tabla B4. Promedios Globales de las Pruebas Semi-Dinámicas.

	Comparación de Soluciones con Y_{apr}				Cobertura				Ranking
	$ Y_{apr} \cap Y_{alg} $	$ Y_{apr} \succ Y_{alg} $	$ Y_{apr} $	$\% Y_{apr}$	Y_{M3AS}	Y_{MOA}	Y_{MOACS}	Y_{MMA}	
Y_{M3AS}	10.89	9.75	20.64	21.66		4.57	0	1.67	2
Y_{MOA}	9.45	9.32	18.77	18.68	3.08		0.36	1.78	3
Y_{MOACS}	32.18	7.76	39.94	56.59	6.70	7.98		2.44	1
Y_{MMA}	8.11	3.91	12.02	16.82	4.51	5.09	4.11		4

Sin lugar a dudas, el MOACS es el mejor algoritmo de este conjunto. El MOACS como primero en el **ranking** logro aportar 32.18 soluciones a los frentes Pareto aproximado en promedio, con 7.76 de sus soluciones dominadas, teniendo 39.94 soluciones en promedio su frente Pareto calculado - Y_{MOACS} . Los resultados anteriores indican que el 56.59% de los frentes Pareto pudo encontrar el MOACS, la cuál es amplia en comparación al de otros algoritmos (ver columna $\% Y_{apr}$ de la Tabla B4).

En las coberturas, se puede observar que el MOACS dominó más soluciones sobre el M3AS y el MOA. i.e.:

$$|Y_{MOACS} \succ Y_{M3AS}| = 6.70 > |Y_{M3AS} \succ Y_{MOACS}| = 0,$$

$$|Y_{MOACS} \succ Y_{MOA}| = 7.98 > |Y_{MOA} \succ Y_{MOACS}| = 0.36.$$

Por su parte el M3AS logro el segundo lugar al lograr hallar 21.66% de las soluciones de los frentes Pareto. Además, el M3AS sus soluciones dominaron más soluciones del MOA:

$$|Y_{M3AS} \succ Y_{MOA}| = 4.27 > |Y_{MOA} \succ Y_{M3AS}| = 3.08.$$

El tercer lugar corresponde al MOA, el cuál alcanzo el 18.68% de las soluciones de los distintos frentes Pareto. Sin embargo, no pudo dominar más soluciones que los otros algoritmos.

Finalmente el MMA, quedo último en la ranking con solo 16.82% de soluciones alcanzadas en los distintos frentes Pareto. A pesar de quedar último, el MMA pudo dominar más soluciones sobre los otros algoritmos, como puede verse a continuación:

$$|Y_{MMA} \succ Y_{M3AS}| = 4.51 > |Y_{MMA} \succ Y_{M3AS}| = 1.67,$$

$$|Y_{MMA} \succ Y_{MOA}| = 5.09 > |Y_{MMA} \succ Y_{MOA}| = 1.78,$$

$$|Y_{MMA} \succ Y_{MOACS}| = 4.11 > |Y_{MMA} \succ Y_{MOACS}| = 2.44.$$

Ésta dominancia es debida a que el MMA pudo encontrar alguna muy buena solución en alguna zona particular del frente Pareto, a costa resto del frente Pareto, que sí fue explorado por los ACOs, las cuales por tal motivo, aportaron muchas más soluciones de compromiso, acorde a los resultados de Promedios Globales.

Bibliografía

- [ALO04] S. Alonso, O. Cordón, I. Fernández de Viana y F. Herrera. "La Metaheurística de Optimización Basada en Colonias de Hormigas: Modelos y Nuevos Enfoques". G. Joya, M.A. Atencia, A. Ochoa, S. Allende (Eds.), Optimización Inteligente: Técnicas de Inteligencia Computacional para Optimización, 2004, Servicio de Publicaciones de la Universidad de Málaga, páginas 261-313.
- [ARA02] P. T. de Araujo, y G. M. Barbosa, "Multicast Routing with Quality of Service and Traffic Engineering Requirements in The Internet, Based On Genetic Algorithm", *Proceedings of the 7th Brazilian Symposium on Neural Networks (SBRN'02)*, Brasil, 2002.
- [BAR00] B. Barán, y R. Sosa, "A New Approach for Antnet Routing, " *IEEE International Conference on Computer Communication and Networks (ICCCN'2000)*, USA, 2000.
- [COH78] J. Cohon, *Multiobjective programming and planning*, Academic Press, 1978.
- [CHU02] C. Chu, J. Gu, X. Hou, and Q. Gu. "A heuristic ant algorithm for solving QoS multicast routing problem" *Evolutionary Computation, 2002. CEC'02. Volume 2*, pp 1630-1635.
- [CRI04a] J. Crichigno, y B. Barán, "Multiobjective Multicast Routing Algorithm", *11th International Conference on Telecommunications*, Ceara, Brasil, 2004.
- [CRI04b] J. Crichigno, y B. Barán, "A Multicast Routing Algorithm Using Multiobjective Optimization", *11th International Conference on Telecommunications*, Ceara, Brasil, 2004.
- [CRI04c] J. Crichigno, y B. Barán, "Multiobjective Multicast Routing Algorithm for Traffic Engineering", *IEEE International Conference on Computer and Communications ICC'2004*, Chicago, US, 2004.

- [CRI04d] J. Crichigno, F. Talavera, J. Prieto y B. Barán, "Enrutamiento multicast utilizando optimización multiobjetivo", *CACIC'2004*, Buenos Aires, Argentina, 2004. pp. 147-158.
- [DAS02] S. Das, S. Gosavi, S. Vaze, y W. H. Hsu. "An Ant Colony Approach for the Steiner Tree Problem". Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002), New York, NY, 2002.
- [DEB99] K. Deb, "Evolutionary algorithms for multi-criterion optimization in engineering design", *Proceedings of the Evolutionary Algorithms in Engineering and Computer Science (EUROGEN'99)*, 1999.
- [DIJ59] E. Dijkstra, "A note on two problems in connexion with graphs", *Numerische Mathematik*, Vol. 1, pág. 269-271, 1959.
- [DON02] Y. Donoso. "Extensión de los Métodos Hop-by-Hop, CR-LDP y RSVP-TE para Multicast IP sobre MPLS", *XXVIII Conferencia Latinoamericana de Informática*, Uruguay, 2002.
- [DON04] Y. Donoso, R. Fabregat, y J. Marzo, "Multi-objective Optimization Algorithm for Multicast Routing with Traffic Engineering", *IEEE 3rd International Conference on Networking (ICN'2004)*, Guadalupe – Caribe Francés, 2004.
- [DOR91] M. Dorigo, V. Maniezzo y A. Coloni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 26: 1, páginas 29-41, 1996.
- [DOR97] M. Dorigo y L.M. Gambardella. "Ant Colony System: A cooperative learning approach to the traveling salesman problem". *IEEE Transactions on Evolutionary Computation*, 1:1, páginas 53-56, 1997.

- [DOR99] M. Dorigo y G. Di Caro. "The Ant Colony Optimization Meta-Heuristic". En David Corne, Marco Dorigo, y Fred Glover, editores, *New Ideas in Optimization*, páginas 11–32. McGraw-Hill, London, 1999.
- [DOR01] M. Dorigo y T. Stützle. "An Experimental Study of the Simple Ant Colony Optimization Algorithm". En 2001 WSES International Conference on Evolutionary Computation (EC'01). WSES-Press International, 2001.
- [GAR04] C. García-Martínez, O. Cordón y F. Herrera, "An Empirical Analysis of Multiple Objective Ant Colony Optimization Algorithms for the Bi-criteria TSP". En M. Dorigo, M. Birattari, C. Blum, L. M. Gambardella, F. Mondada, y T. Stützle, editores, *Proceedings of ANTS 2004 -Fourth International Workshop on Ant Colony Optimization and Swarm Intelligence*. Tomo 3172 de LNCS. Springer-Verlag, Bruselas, septiembre 2004.
- [GOM04a] O. Gómez y B. Barán. "Òmicron ACO". En *Proceedings de la Conferencia Latinoamericana en Informática (CLEI'04)*. Septiembre 2004.
- [GOM04b] O. Gómez. "Razones del Éxito de Optimización por Colonia de Hormigas en el Problema del Cajero Viajante". Tesis de Maestría. Universidad Nacional de Asunción. Septiembre 2004.
- [HOR97] J. Horn, "The nature of niching: genetic algorithms and the evolution of optimal, cooperative population", *Ph.D Thesis*, University of Illinois at Urbain Champaign, USA, 1997.
- [HWA00] R. Hwang, W. Do, y S. Yang, "Multicast Routing Based on Genetic Algorithms", *Journal of Information Science and Engineering*, Vol. 16, pág. 885-901, 2000.
- [KOM93a] V. Kompella, "Multicast Routing Algorithm for Multimedia Traffic", *Ph.D. Thesis*, University of California, San Diego, USA, 1993.

- [KOM93b] V. Kompella, J. Pasquale, y G. Polyzos, "Multicast routing in multimedia communication", *IEEE/ACM Transactions on Networking*, Vol. 1 N° 3, pág. 286-291, 1993.
- [KOU81] L. Kou, G. Markowsky, y L. Berman, "A Fast Algorithm for Steiner Trees", *Acta Informatica*, Vol. 15, N° 2, pág. 141-145, 1981.
- [LAN02] R. Landa. "Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Multiobjetivo". Tesis de Maestría, Centro de Investigación y Estudios Avanzados, Instituto Politécnica Nacional. México, 2002.
- [LIU03] Y. Liu, and J. Wu. "The degree-constrained multicasting algorithm using ant algorithm" IEEE 10th International Conference on Telecommunications, 2003.
- [OOM02] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul, y F. Ansari, "Overview of IP Multicast in a Multi-Protocol Label", *RFC 3353*, 2002.
- [PAR98] G. Parks, y I. Miller, "Selective breeding in a multiobjective genetic algorithm", *5th International Conference on Parallel Problem Solving from Nature (PPSN-V)*, Berlín, Alemania, pág. 250-259, 1998.
- [RAV98] C. P. Ravikumar, y R. Bajpai, "Source-based delay bounded multicasting in multimedia networks", *Computer Communications*, Vol. 21, pág. 126-132, 1998.
- [SEO02] Y. Seok, Y. Lee, Y. Choi, y C. Kim, "Explicit Multicast Routing Algorithm for Constrained Traffic Engineering", *IEEE Proceedings of 7th International Symposium on Computer and Communications (ISCC'02)*, Italia, 2002.
- [SCH03] M. Schaerer y B. Barán. "A Multiobjective Ant Colony System For Vehicle Routing Problem With Time Windows", *IATED International Conference on Applied Informatics*, Innsbruck, Austria, 2003.

- [STA00] W. Stallings, *Comunicaciones y Redes de Computadores*, Prentice Hall, 2000.
- [STA01] W. Stallings. "MPLS", *The Internet Protocol Journal*, Vol. 4, N° 3, 2001.
- [STE86] R. Steuer, *Multiple criteria optimization: theory, computation and application*, New York, 1986.
- [STÜ00] T. Stützle y H. Hoos. "MAX-MIN Ant System". *Future Generation Computer System*. 16(8): páginas 889-914, Junio 2000.
- [STÜ02] T. Stützle y M. Dorigo. "A Short Convergente Proof for a Class of Ant Colony Optimization Algorithms". *IEEE Transactions on Evolutionary Computation*, 6:358-365, agosto 2002.
- [TAL04a] F. Talavera, J. Prieto, J. Crichigno y B. Barán. "Comparación de algoritmos evolutivos multi-objetivos en un ambiente multicast," *CACIC'2004*, Buenos Aires, Argentina, 2004. pp. 1611-1622.
- [TAL04b] F. Talavera, y B. Barán. "Policies for Dynamical Multi-Objective Environment of Multicast Traffic Engineering". *12th International Conference on Telecommunications*. Cape Town, Sudáfrica, 2005.
- [TAL04c] F. Talavera. "Comparación de Algoritmos Multi-objetivos en un Ambiente Multicast". Tesis de Ingeniería, Universidad Católica Nuestra Señora de la Asunción, Febrero, 2005.
- [TAN03] A. Tanenbaum, *Computer Networks*, Prentice Hall, 2003.
- [VAN99] D. Veldhuizen, "Multiobjective Evolutionary Algorithms: Classifications, Analysis, and New Innovations", *Ph.D Thesis*, Graduated School of Engineering of the Air Force Institute of Technology, Air University, 1999.
- [WAN01] Z. Wang, *Internet QoS, Architectures and Mechanism for Quality of Service*, Morgan Kaufmann, 2001.

- [ZHE01] W. Zhengying, S. Bingxin, y Z. Erdun, "Bandwidth-delay-constraint least-cost multicast routing based on heuristic genetic algorithm", *Computer Communications*, Vol. 24, pág. 685-692, 2001.
- [ZIT99] E. Zitzler, y L. Thiele, "Multiobjective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach", *IEEE Trans. Evolutionary Computation*, Vol. 3, N° 4, pág. 257-27, 1999.