



UNIVERSIDAD NACIONAL DEL SUR

TESIS DE DOCTOR EN CIENCIAS DE LA COMPUTACIÓN

*Desarrollo teórico de técnicas meta-heurísticas
para resolver problemas de optimización "TN" (Transit Networks)
en entornos dinámicos*

ANA CAROLINA OLIVERA

BAHIA BLANCA

ARGENTINA

2009

Prefacio

Esta tesis es presentada como parte de los requisitos para optar al grado Académico de Doctor en Ciencias de la Computación, de la Universidad Nacional del Sur y no ha sido presentada previamente para la obtención de otro título en esta Universidad u otras. La misma contiene los resultados obtenidos en investigaciones llevadas a cabo en el Departamento de Ciencias e Ingeniería de la Computación durante el período comprendido entre el 6 de marzo de 2007 y el 7 de septiembre de 2009 bajo la dirección de la Dra. Nélica Beatriz Brignole, profesora del Departamento de Ciencias e Ingeniería de la Computación e investigadora independiente del CONICET y la codirección del Dr. Gustavo Esteban Vazquez, asistente del Departamento de Ciencias e Ingeniería de la Computación e investigador asistente del CONICET.

Deseo expresar en primer término mi más profundo reconocimiento a la Dra. Nélica Beatriz Brignole, por su invaluable colaboración y espíritu de trabajo, por la excelencia de la formación que me brindó y por su constante estímulo, pilares fundamentales en la concreción de esta tesis.

Mi más sincero agradecimiento al Dr. Gustavo Esteban Vazquez, quien me ayudó con paciencia a caminar los primeros pasos y me acompañó con su valiosísima amistad e inmensa sabiduría durante todo el recorrido de este sendero.

Al Consejo Nacional de Investigaciones Científicas y Técnicas y al Departamento de Ciencias e Ingeniería de la Computación, mi agradecimiento por haberme facilitado los medios para realizar este trabajo.

Al Colegio San Vicente de Paul por brindarme la formación ética y moral para realizarme como ser humano, por su apoyo y su constante aliento durante mi tránsito por dicha institución.

A todas las personas que, durante estos años, de alguna u otra manera me acompañaron y ayudaron, me aportaron ideas y con quienes compartí horas de trabajo y desinteresadamente me brindaron su conocimiento en áreas ajenas a mi especialidad.

Y por último, quiero expresar mi mayor gratitud a mis padres, mis hermanos y mi novio quienes con su afecto y constante aliento hicieron posible que concretara esta tesis. A ellos mi mayor agradecimiento por el amor y apoyo brindados.

Ana Carolina Olivera

Bahía Blanca, 7 de septiembre de 2009.-

Departamento de Ciencias e Ingeniería de la Computación

Universidad Nacional del Sur



UNIVERSIDAD NACIONAL DEL SUR
Secretaría General de Posgrado y Educación Continua

La presente tesis ha sido aprobada el/..../..... , mereciendo la calificación de(.....)

Resumen

El objetivo general de esta tesis fue desarrollar técnicas meta-heurísticas híbridas para el tratamiento de problemas *NP-Hard* relacionados con el problema de la red de tránsito (Transit Network Problem: TNP). Estas técnicas están orientadas, en particular, a las dos etapas más importantes del TNP: el diseño y la planificación (Transit Network Design and Scheduling Problem: TNDSP) de la red. El objetivo principal fue proveer una herramienta automatizada para la creación tanto de las rutas de las líneas de la red de tránsito como de la frecuencia de los móviles.

Como resultado colateral de esta tesis se logró además subsanar una deficiencia de los trabajos existentes sobre TNP: su incapacidad para obtener resultados que reflejen la dinámica de las variables que dependen de los tiempos consumidos por el usuario en su espera, acceso y viaje dentro de la red. Esta tesis analiza dichas falencias y presenta como solución la posibilidad de incorporar un elemento dinámico dentro de la resolución del TNDSP. De este modo, se han podido obtener resultados realistas tanto desde el punto de vista del operador del servicio como así también del cliente.

En base a nuestras investigaciones, se obtuvo un algoritmo híbrido evolutivo dinámico multi-objetivo de dos etapas que tiene en cuenta tanto al operador como a los clientes de la red. La primera es una etapa de inicialización de distancias y rutas entre paradas a través de un novedoso procedimiento GRASP (*Greedy Randomized Adaptive Search Procedure*). La segunda es una etapa que implementa un nuevo algoritmo evolutivo que contiene un procedimiento de simulación para el cálculo de la función de aptitud (*fitness*) de los individuos. Finalmente, se estudiaron algunas ciudades hipotéticas estableciendo la mejor configuración de parámetros para las fases del algoritmo híbrido y se analizó un caso real utilizado en la literatura para comparar nuestra metodología con los procedimientos más conocidos sobre el tema, mostrando resultados altamente satisfactorios.

Abstract

The general purpose of this thesis was to develop meta-heuristic hybrid techniques for the treatment of NP-hard problems related to the *Transit Network Problem* (TNP). These techniques are particularly oriented to the most important TNP stages: the design and planning of the network (*Transit Network Design and Scheduling Problem*: TNDSP). The main goal was to provide a tool for the creation of both the routes for the transit network lines and the mobile frequency.

Also, the work performed in this thesis contributes to eliminate a deficiency in the existing work about TNP: the inability to yield results that reflect the dynamics of the variables that depend on the time taken by the users while they are waiting, accessing and tripping within the network. This thesis analyses this dearth and proposes as a solution the possibility of incorporating a dynamic element to the manner of solving the TNDSP. In this way, realistic results could be obtained from the viewpoint of both the service operator and the client as well.

On the basis of our research, a hybrid dynamic two-stage evolutionary algorithm was obtained. In the network, it takes into account both the operator and the clients as well. The first stage is an initialization of the distances and routes between stops through a GRASP (*Greedy Randomized Adaptive Search Procedure*). The second stage implements an evolutionary algorithm that contains a simulation procedure for the calculation of the individuals' fitness.

Finally, some hypothetical cities were studied by establishing the best parameter configuration for the phases of the hybrid algorithm. A real case, which is classic in the literature, was analyzed in order to compare our method against the well-known approaches used to solve these problems. The results of this analysis were highly satisfactory.

Índice

PREFACIO	2
RESUMEN	4
ABSTRACT	5
ÍNDICE	6
LISTA DE PUBLICACIONES	10
CAPÍTULO 1 REDES DE TRÁNSITO	13
1.1 TRÁNSITO Y TRÁFICO	14
1.2 CONCEPTOS GENERALES	14
1.3 EL PROCESO GLOBAL DE PLANEAMIENTO DEL TRÁNSITO	16
1.4 DISEÑO DE LA RED DE TRÁNSITO	17
1.4.1 <i>Entrada</i>	17
1.4.2 <i>Principales restricciones y objetivos</i>	17
1.5 FRECUENCIAS DE LA RED DE TRÁNSITO	19
1.5.1 <i>Entrada</i>	19
1.5.2 <i>Principales restricciones y objetivos</i>	20
1.6 CRONOGRAMA DE LA RED DE TRÁNSITO	20
1.6.1 <i>Entrada</i>	20
1.6.2 <i>Principales restricciones y objetivos</i>	21
1.7 ASIGNACIÓN DE LA FLOTA	21
1.8 PROGRAMACIÓN DE LA TRIPULACIÓN Y OTROS RECURSOS	21
1.9 ORGANIZACIÓN DE LOS CONTENIDOS Y ESTRUCTURA GENERAL DE LA TESIS	22
CAPÍTULO 2 ENFOQUE DEL TNP	23
2.1 $TNDSP = TNDP + TNFSP + TNTP$	23
2.1.1 <i>TNDSP: Terminología</i>	24
2.1.2 <i>Modelos matemáticos para el TNDSP</i>	25
2.1.3 <i>Resolución del problema de optimización para TNDSP</i>	26
2.2 TNDP : DISEÑO DE LA RED DE TRÁNSITO	26
2.2.1 <i>TNDP: Métodos Matemáticos</i>	27
2.2.2 <i>TNDP: Heurísticas</i>	28
2.2.3 <i>TNDP: Métodos de búsqueda local o en el vecindario</i>	29
2.2.4 <i>TNDP: Algoritmos Evolutivos</i>	29
2.2.5 <i>TNDP: Otros métodos</i>	30
2.3 TNFSP: AJUSTE DE FRECUENCIAS	30
2.3.1 <i>TNFSP: Métodos matemáticos</i>	30
2.3.2 <i>TNFSP: Procedimientos heurísticos</i>	31
2.3.3 <i>TNFSP: Otras técnicas</i>	32
2.4 TNTP: CRONOGRAMA DE UNA RED DE TRÁNSITO	32
2.4.1 <i>TNTP: Métodos Matemáticos</i>	33
2.4.2 <i>TNTP: Técnicas heurísticas</i>	34
2.4.3 <i>TNTP: Técnicas de búsqueda local o en el vecindario</i>	34
2.4.4 <i>TNTP: Algoritmos Evolutivos</i>	35
2.4.5 <i>TNTP: Otros métodos</i>	36
2.5 $TNDFSP = TNDP + TNFSP$	36
2.5.1 <i>TNDFSP: Resolución</i>	36
2.6 $TNSP = TNFSP + TNTP$	37
2.6.1 <i>TNSP: Resolución</i>	38
2.7 CARENCIAS DE LOS MÉTODOS ACTUALES PARA RESOLVER EL TNP	38
CAPÍTULO 3 OPTIMIZACIÓN MULTI-OBJETIVO	40
3.1 OPTIMIZACIÓN MONO-OBJETIVO	40
3.2 OPTIMIZACIÓN MULTI-OBJETIVO	41
3.3 ÓPTIMO DE PARETO	43
3.3.1 <i>Condiciones de optimalidad</i>	44

3.4	OPTIMIZACIÓN MULTI-OBJETIVO A TRAVÉS DE MÉTODOS CLÁSICOS	46
3.4.1	<i>Articulación a priori</i>	46
3.4.1.1	Método <i>simplex</i> multi-objetivo	46
3.4.1.2	Programación fraccional	47
3.4.2	<i>Articulación a posteriori</i>	48
3.4.2.1	Combinación lineal de pesos	48
3.4.2.2	El método de las restricciones ϵ	49
3.5	CONCLUSIONES SOBRE LOS MÉTODOS CLÁSICOS Y LA OPTIMIZACIÓN MULTI-OBJETIVO	49
CAPÍTULO 4 TÉCNICAS META-HEURÍSTICAS: ALGORITMOS EVOLUTIVOS		51
4.1	ANÁLISIS DE LOS ALGORITMOS DE OPTIMIZACIÓN CLÁSICOS	52
4.2	ALGORITMOS EVOLUTIVOS	54
4.3	REPRESENTACIÓN DEL INDIVIDUO	59
4.3.1	<i>Representación binaria</i>	59
4.3.2	<i>Representación de recorridos o permutaciones</i>	61
4.4	OPERADORES EVOLUTIVOS	62
4.4.1	<i>Selección</i>	62
4.4.2	<i>Operador de recombinación o cruzamiento</i>	63
4.4.3	<i>Operador de mutación</i>	66
CAPÍTULO 5 ALGORITMOS EVOLUTIVOS PARA PROBLEMAS MULTI-OBJETIVO		68
5.1	ARTICULACIÓN A PRIORI	69
5.1.1	<i>Método de la suma ponderada</i>	69
5.2	ARTICULACIÓN A POSTERIORI	70
5.2.1	<i>Métodos no basados en el concepto de Pareto</i>	70
5.2.1.1	Método de las restricciones ϵ	71
5.2.1.2	Algoritmo genético de vector evaluado (VEGA)	71
5.2.1.3	Algoritmo genético basado en pesos (WPGA)	71
5.2.2	<i>Métodos no elitistas basados en el concepto de Pareto</i>	72
5.2.2.1	Algoritmo de ordenamiento de Pareto	73
5.2.2.2	Algoritmo genético con múltiples objetivos (MOGA)	73
5.2.2.3	Algoritmo evolutivo con fuerzas de Pareto (SPEA)	74
5.2.2.4	Algoritmo genético de ordenamiento no dominado (NSGA)	75
5.2.3	<i>Métodos elitistas basados en el concepto de Pareto</i>	76
5.2.3.1	Algoritmo genético de ordenamiento no dominado elitista	76
5.2.3.2	Algoritmo genético Pareto basado en la distancia (DPGA)	77
5.3	MÉTRICAS PARA COMPARAR CONJUNTOS NO DOMINADOS	78
5.3.1	<i>Escala y Normalización</i>	80
5.3.2	<i>Métricas que evalúan la proximidad a la frontera Pareto óptima</i>	80
5.3.2.1	Razón del Error	80
5.3.2.2	Métrica de cubrimiento de conjunto	81
5.3.2.3	Distancia generacional	81
5.3.3	<i>Métricas que evalúan la diversidad entre los conjuntos no dominados de soluciones</i>	81
5.3.3.1	Espaciado	82
5.3.3.2	Extensión	82
5.3.3.3	Máxima extensión	82
5.3.4	<i>Métricas que evalúan la cercanía y la diversidad</i>	83
5.3.4.1	Hipervolumen	83
5.3.4.2	Superficies de métricas basadas en estadísticas	84
5.3.5	<i>Métricas para evaluar MOEAs</i>	84
5.3.5.1	Ranking de dominancia	84
5.3.5.2	Indicadores de Calidad	85
5.4	PLATAFORMA E INTERFASE INDEPENDIENTE DEL LENGUAJE PARA ALGORITMOS DE BÚSQUEDA	87
5.4.1	<i>Estructura general de PISA</i>	88
5.4.2	<i>IBEA: indicator-based evolutionary algorithm</i>	89
CAPÍTULO 6 SIMULACIÓN		91
6.1	SISTEMA	92
6.1.1	<i>Componentes de un sistema</i>	92
6.1.2	<i>Sistemas discretos y continuos</i>	92
6.2	MODELO DE UN SISTEMA	93
6.2.1	<i>Tipos de Modelos</i>	93

6.2.2	<i>Modelos de Simulación</i>	93
6.3	SIMULACIÓN DE EVENTOS DISCRETOS	94
6.3.1	<i>Conceptos en simulación de eventos discretos</i>	94
6.4	PASOS DE UN ESTUDIO MEDIANTE SIMULACIÓN	95
6.5	SIMULACIÓN EN SISTEMA DE COLAS	98
6.5.1	<i>Características de los sistemas de colas</i>	98
6.5.1.1	Los clientes	98
6.5.1.2	Capacidad del Sistema	99
6.5.1.3	El proceso de arribo	99
6.5.1.4	Comportamiento de la cola y disciplina de cola	100
6.5.1.5	Tiempos de servicio y mecanismo de servicio	101
6.5.2	<i>Notación para los sistemas de colas</i>	101
CAPÍTULO 7 DISEÑO DE UN ALGORITMO HÍBRIDO DINÁMICO MULTI-OBJETIVO PARA REDES DE TRÁNSITO		103
7.1	MODELADO DE LAS ENTIDADES	104
7.2	ASPECTOS GENERALES DEL AHDM	105
7.3	ETAPA GRASP	106
7.3.1	<i>GRASP: Fase constructiva</i>	107
7.3.2	<i>GRASP: búsqueda local</i>	108
7.4	ETAPA EVOLUTIVA DEL AHDM	109
7.4.1	<i>Los individuos: conjunto de líneas</i>	110
7.4.2	<i>Inicialización de la población</i>	111
7.4.3	<i>La función objetivo: Optimizando FO y FU</i>	112
7.4.3.1	Simulación: cálculo de las variables dependientes del tiempo para la evaluación de los individuos	113
7.4.4	<i>Operadores genéticos</i>	115
7.4.4.1	Cruzamiento multipunto multilinea	115
7.4.4.2	Mutación	117
7.4.4.3	Selección	119
7.5	ANÁLISIS DE LA COMPLEJIDAD DE LOS ALGORITMOS DESARROLLADOS	121
CAPÍTULO 8 EXPERIMENTOS		124
8.1	NOTACIÓN URBANA	124
8.2	PRESENTACIÓN CASO DE ESTUDIO 1: UN BARRIO RESIDENCIAL	125
8.3	PRESENTACIÓN CASO DE ESTUDIO 2: UNA CIUDAD DE SUIZA	127
8.3.1	<i>Diagrama de la topología de las paradas</i>	127
8.3.2	<i>Limitaciones y carencias del caso de estudio</i>	128
8.4	ANÁLISIS DE LA CONVERGENCIA DEL AHDM	129
8.4.1	<i>Experimentación: fase 1</i>	130
8.4.2	<i>Experimentación: Fase 2</i>	131
8.5	ELECCIÓN DEL MOEA PARA EL AHDM	132
8.5.1	<i>Análisis utilizando métricas</i>	133
8.6	AHDM APLICADO AL CASO DE MANDL	135
CAPÍTULO 9 ALCANCES		139
9.1	LOGÍSTICA	139
9.2	LOGÍSTICA DEL TRANSPORTE	140
9.2.1	<i>Transporte de Pasajeros</i>	141
9.2.2	<i>Transporte Público Urbano</i>	141
9.2.3	<i>Cruceros</i>	142
9.2.4	<i>Rutas Comerciales Marítimas</i>	142
9.2.5	<i>Distribución de Gas/Electricidad y otros servicios</i>	143
9.2.6	<i>Servicios puerta a puerta</i>	144
9.3	APLICACIÓN DEL ALGORITMO HÍBRIDO	144
CAPÍTULO 10 CONCLUSIONES		146
10.1	EPÍLOGO	148
10.2	PROYECCIÓN	149
REFERENCIAS		151
LISTA DE FIGURAS		161

LISTA DE ALGORITMOS	162
LISTA DE TABLAS	162
ABREVIATURAS	163
NOMENCLATURA	165

Lista de publicaciones

PUBLICACIONES EN REVISTAS INDEXADAS EN ISI Y SCOPUS CON REFERATO Y CESIÓN DE COPYRIGHT

Ana C. Olivera, Mariano Frutos, Jessica A. Carballido, Ignacio Ponzoni, Nélica B. Brignole. "Bus Network Scheduling Problem: GRASP + EAs with PISA * Simulation". 10th International Work-Conference on Artificial Neural Networks (IWANN2009). 10-12/06/2009. Salamanca, España. Organizado por Spanish Chapter of the IEEE Computational Intelligence Society, Universidad de Salamanca, Universitat Politècnica de Catalunya, Universidad de Málaga, Universidad de Granada. *Lecture Notes in Computer Science*. Springer - Verlag. IWANN 2009, Part I, LNCS 5517, pp. 1272–1279. **2009**.

A.C. Olivera and M. Frutos and J.A. Carballido and N.B. Brignole. "Bus Network Optimization with a Time-Dependent Hybrid Algorithm". J.UCS - Journal of Universal Computer Science, Graz University of Technology, and supported by the KNOW-Center and University of Applied Sciences. Vol. 14, Num. 15, pp. 2512-2531. ISSN 0948-695x, OnLine Edition: ISSN 0948-6968 http://www.jucs.org/jucs_14_15/bus_network_optimization_with. **2008**. Trabajo extendido al desarrollado en el 7th. International Conference on Intelligent Systems Engineering and Applications ISDA (2007) -ver más abajo en congresos internacionales.

EN CONGRESOS Y REUNIONES CIENTÍFICAS CON COMITÉ DE REVISIÓN

INTERNACIONALES

Con Publicación en Actas o Proceedings

Ana Carolina Olivera, Mariano Frutos, Jessica Andrea Carballido, Nélica Beatriz Brignole. "GRASP + SPEA2 * Simulation = Allocation and Scheduling of Bus-Networks in Urban Areas". Publicado en el 24th IFIP TC 7 Conference on System Modelling and Optimization. 27-31/07/**2009**. Buenos Aires, Argentina. Organizado por el Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación.

Mariano Frutos, Ana Carolina Olivera, Tohmé Fernando. "A Hybrid Technique for Handling the Flexible Job-Shop Scheduling Problem". Publicado en el 24th IFIP TC 7 Conference on System Modelling and Optimization. 27-31/07/**2009**. Buenos Aires, Argentina. Organizado por el Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación.

Mariano Frutos, Ana Carolina Olivera. "Vehicle Sequencing Problem: Una Técnica Híbrida para El ROADEF'05 Challenge". Publicado en los anales del XIV Congreso Latino Ibero Americano en Investigación de Operaciones (CLAIO 2008) se llevó a cabo en el Centro de Convenciones Cartagena de Indias, Cartagena de Indias, Colombia, del 9-12/09/**2008**. ISBN 978-958-825283-4. Organizado por la Asociación Latino-Iberoamericana de Investigación Operativa. 3 Páginas.

Mariano Frutos, Ana Carolina Olivera. "Job Shop Scheduling Problem: Desarrollo de un Procedimiento Eficiente". Publicado en los anales del XIV Congreso Latino Ibero Americano en Investigación de Operaciones (CLAIO 2008) se llevó a cabo en el Centro de Convenciones Cartagena de Indias, Cartagena de Indias, Colombia, del 9-12/09/**2008**. ISBN 978-958-825283-4. 3 Páginas.

Ana Carolina Olivera, Mariano Frutos, Jessica Andrea Carballido, Nélica Beatriz Brignole. "Bus Network Optimization through Time-Dependent Hybrid Algorithm". Publicado en el International Conference on Intelligent Systems Design and Applications (ISDA'07). Del 22-24/10/2007 en Río de Janeiro, Brasil. volume 0, ISBN 0-7695-2976-3, pp. 857-862, publisher IEEE Computer Society, Los Alamitos, CA, USA. **2007**.

Mariano Frutos, Ana Carolina Olivera e Ricardo Casal. "Algoritmo híbrido para un enfoque alternativo al Job-Shop Scheduling Problem". Publicado en el XIII International Conference on Industrial Engineering and Operations Management (ICIEOM 2007), ISBN CD: 97885-88478-33-6, 9-11/10/2007, en Foz do Iguaçu, Paraná, Brasil, organizado por ABEPRO - Brazilian Association of Production Engineering. 10 páginas.

Mariano Frutos, Ana Carolina Olivera. "Estrategia meta-heurística híbrida aplicada a la programación operacional". Publicado en el II Congresso da Academia Trinacional de Ciências (C3N 2007), ISSN 1982-2758. Del 08-11/10/**2007** en Foz do Iguaçu, Paraná, Brasil. 8 Páginas

Mariano Frutos, Ana Carolina Olivera, Ricardo Casal. “Estudio sobre distintas estrategias utilizadas para establecer secuencias detalladas en entornos productivos”. Publicado en los Anales de XII ICIEOM – “XII International Conference on Industrial Engineering and Operations Management” (ISBN 85-88478-21-8). Organizado por ABEPRO - Associação Brasileira de Engenharia de Produção. Realizado del 09-11/10/2006, Fortaleza - CE, Praia do Futuro, Brasil. 9 Páginas.

Ana Carolina Olivera, Mariano Frutos, Ricardo Casal. “Métodos para determinar Secuencias de Producción en un Ambiente Productivo Complejo”. Publicado en el Congreso Latino-Iberoamericano de Investigación Operativa. Organizado por Universidad de la República, Uruguay. ISBN: 9974-7699-9-X. 27-30/11/2006, Montevideo, Uruguay. 6 Páginas.

NACIONALES

Con Publicación en Actas o Proceedings

Ana Carolina Olivera, Mariano Frutos, Nélidea Beatriz Brignole, Ricardo Casal. “Evaluación del Sistema de Transporte Público Urbano Mediante Simulación de Entornos Dinámicos”. Publicado en el CDROM de anales del VII Congreso Chileno de Investigación Operativa. 21-23/11/2007. Puerto Montt, X° Región, Chile. Organizado por la Asociación Latino-Iberoamericana de Investigación Operativa.

Ricardo Casal, Mariano Frutos, Ana Carolina Olivera. “Combine Meta-Heuristics for Treatment of production Complex Problem”. ISBN 0978-987-98835-1-8. Editores responsables: UTN FRA y UNLaM. 1-2/11/2007. 1er Congreso Argentino de Ingeniería Industrial UTN FRA y UNLaM 2007. Avellaneda, Buenos Aires, Argentina.

Ana C. Olivera, Mariano Frutos, Jessica A. Carballido, Nélidea B.Brignole. “GRASP initialization stage for bus-network hybrid algorithm”. Publicado en los Anales del “Simposio de Investigación Operativa” - SIO 2007 (ISSN: 1850-2865), como parte de las 36° JAIIO, Jornadas Argentinas de la Sociedad Argentina de Informática. Mar del Plata, Buenos Aires, Argentina, entre los días 27-28/08/2007. 8 páginas.

Mariano Frutos, Ricardo Casal, Ana Carolina Olivera. “Algoritmo híbrido estocástico aplicado al diseño de rutas y determinación de frecuencias en el transporte público urbano”. Publicado en los Proceedings (ISBN: 978-84-690-3470-5) del V Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2007). Del 14-16/02/2007. Organizado por Universidad de La Laguna, España. Puerto de La Cruz, Tenerife, España. 7 Páginas.

Ana Carolina Olivera, Mariano Frutos, Ricardo Casal. “Algoritmos Genéticos una Estrategia clave para abordar Problemas de Secuenciamiento en Gran Escala”. Publicado en Anales de JAIIO 35° “Jornadas Argentinas de Informática e Investigación Operativa” (ISSN 1850-2776) “3° Jornadas de Informática Industrial 2006”, pp. 57 a 66 (ISSN 1850-2849), 4-8/11/2006, Ciudad de Mendoza Argentina.

Con publicación de Resúmenes

Mariano Frutos, Ana Carolina Olivera, Ricardo Casal. “Algoritmo Híbrido Aplicado al Job-Shop Scheduling Problem”. Publicado en el CDROM de anales del VII Congreso Chileno de Investigación Operativa (OPTIMA 2007). A realizarse el 21-23/11/2007 en Puerto Montt, X° Región, Chile.

*“When machines learn to feel,
who decides what is human...”¹*

¹ Mitsuhsa, I. y Toshio, S. (Productores), & Mamoru, O. (Director). (2004). *Ghost in the Shell: Innocence* [Película]. Japón: Production I.G..

Capítulo 1 Redes de tránsito

Desde el comienzo de la humanidad, el hombre ha sentido la necesidad de movilizarse con el objetivo de conseguir comodidades que no alcanzaba en su lugar de origen. Desde pequeñas embarcaciones hasta grandes navíos cruzaron océanos en busca de nuevas rutas comerciales, para obtener nuevos productos y aumentar así el comercio. Como resultado de esto, algunos permanecieron en los territorios descubiertos y formaron nuevas comunidades. Mantener conectados a las personas y los lugares a los que estas desean llegar, de manera eficiente dentro del lugar donde viven, constituye el desafío actual. Las personas recorren diariamente grandes distancias por distintos motivos. Todas ellas tienen en común el uso de medios de transporte ya sea particular o público.

La planificación del tránsito público cubre distintas áreas de investigación. Desde el diseño de las redes hasta el tamaño de la tripulación, desde la evaluación de la demanda hasta la asignación del tránsito de los viajes. Los procesos utilizados para obtener una red de tránsito público han sido encarados desde distintos puntos de vista.

Este alto interés por la planificación del tránsito es, en parte, debido a que el desarrollo del transporte público es un tópico crucial en la sociedad moderna. Los continuos congestionamientos urbanos, los problemas de estacionamiento (*parking*) y el incremento de la polución hacen que las personas consideren como opción cambiar al transporte urbano si este tiene una buena calidad de servicio a un costo atractivo. Este problema conocido como **TNP** (Transit Network Problem) es extremadamente complejo. Algunos de sus subproblemas

son considerados por separado dentro de la clasificación NP-*Hard* desde el punto de vista de su complejidad computacional (Ceder e Israeli, 1986; Magnanti y Wong, 1984; Quak, 2003).

En este capítulo se definirán los conceptos fundamentales asociados a TNP, trataremos los principales aspectos de la planificación del tránsito.

1.1 TRÁNSITO Y TRÁFICO

Dos conceptos claves aparecen en la literatura referidos a TNP. Los mismos no deben confundirse para poder comprender el tópico bajo estudio en esta tesis. Estos conceptos son el de **tráfico** y el de **tránsito**.

En inglés se denomina *transit* o tránsito² al transporte público y *traffic* o “tráfico”³ al transporte privado. Sin embargo, en castellano no se hace esa distinción, usándose las palabras tránsito y tráfico indistintamente para referirse a la circulación de vehículos de transporte; en tanto que se le llama transporte pesado al tráfico de mercancías y carga. Para el caso de esta tesis doctoral, haremos uso de la palabra tránsito que se corresponde con la problemática tratada.

1.2 CONCEPTOS GENERALES

Para el tratamiento del problema de las redes de tránsito (**TNP**, Transit Network Problem) comenzaremos definiendo en líneas generales los conceptos que serán utilizados en las secciones siguientes.

Los **operarios** del tránsito son las empresas proveedoras del servicio. Los usuarios o **pasajeros** son los que utilizan el servicio.

² Tránsito (diccionario de la Real Academia Española): Acción de transitar. Actividad de personas y vehículos que pasan por una calle, una carretera, etc. Transitar: Ir o pasar de un punto a otro por vías o parajes públicos.

³ Tráfico (diccionario de la Real Academia Española): Acción de traficar. Circulación de vehículos por calles, caminos, etc. m. Movimiento o tránsito de personas, mercancías, etc., por cualquier otro medio de transporte. Traficar: Comerciar, negociar con el dinero y las mercancías. Andar o errar por varios países, correr mundo.

Un **móvil** es el vehículo que utiliza el operario para transportar al usuario desde su origen a su destino. En el caso particular del problema del transporte público de pasajeros hablaremos indistintamente de móvil, autobús o bus.

Por lo general este móvil deberá recorrer una ruta **o línea**, esta ruta estará compuesta por calles pertenecientes a un **mapa**. El mapa está formado por nodos y arcos. Los arcos poseen sentido. Aquellos nodos del mapa donde es importante que el móvil pase se los denominará **parada**. Por lo general, se definirá más de una ruta en el mapa y se querrá que las rutas estén *interconectadas* por alguna parada, a dicha parada se la denominará **trasbordo**. Al conjunto de rutas se lo denomina *red de tránsito* (TN). La Figura 1.1 resume los conceptos generales presentados.

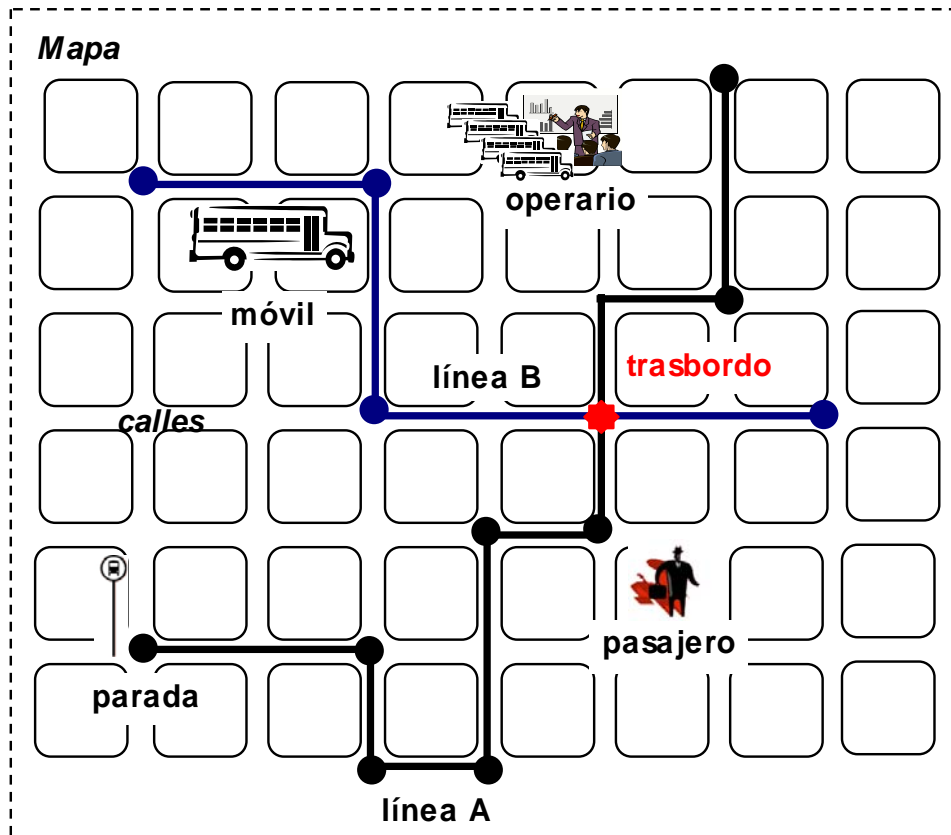


Figura 1.1 Elementos de la red de tránsito

1.3 EL PROCESO GLOBAL DE PLANEAMIENTO DEL TRÁNSITO

En 1986, Ceder y Wilson establecen los lineamientos generales del proceso de planeamiento del tránsito dividiéndolo en una secuencia de cinco pasos:

- 1) Diseño de cada ruta nueva y análisis de las existentes,
- 2) Determinación de las frecuencias de las rutas,
- 3) Determinación de los puntos de transferencia y cronograma de las rutas,
- 4) Asignación de la flota de vehículos,
- 5) Asignación de los empleados y otros recursos para cada línea.

Dentro del transporte público urbano encontramos los tranvías, subtes, metros y buses. Los mismos poseen diferentes características, niveles de flexibilidad y planificación pero comparten los pasos mencionados arriba. La utilidad del recurso transporte esta asociada, por lo general, a como están dispuestos los recorridos de las distintas líneas y la frecuencia con la que realizan dichos recorridos. Los cinco pasos mencionados arriba pueden ser resumidos en la Tabla 1.1.

Tabla 1.1 El Proceso de Planeamiento del Tránsito (Ceder y Wilson, 1986). Los números incorporados en las actividades corresponden a las secciones de este capítulo que las describen

Entradas independientes →	Actividad de Planeamiento	→ Salida
Datos de Demanda Topología	1.4) <i>Diseño de la red de tránsito,</i> ↓	Cambios de Ruta Nuevas Rutas Estrategias de Operación
Disponibilidad de Subsidios Disponibilidad de Móviles Política de Servicio Patrocinio Actual	1.5) <i>Determinación de las frecuencias de las líneas,</i> ↓	Frecuencias del Servicio
Demanda por horarios del día Tiempo del Primer y último viaje Tiempo de recorrido	1.6) <i>Determinación de los puntos de transferencia y cronograma de las líneas,</i> ↓	Tiempo de salida de cada móvil. Tiempo de Arribo de los móviles
Tiempos donde el móvil está vacío Tiempos de Recuperación	1.7) <i>Asignación de la Flota de vehículos,</i> ↓	Planificación de los móviles
Horarios de los empleado Estructura y costos	1.8) <i>Asignación de los empleados y otros recursos para cada línea.</i>	Planificación de los empleados

Idealmente todos estos pasos deben ser tratados simultáneamente con el objetivo de asegurar su interacción y así obtener los mejores resultados posibles. Sin embargo, debido a la complejidad del proceso, este tratamiento ideal es imposible de aplicar en la práctica. Como resultado, varios sub-problemas fueron definidos para resolver el TNP de manera secuencial, lo cual conlleva a la pérdida de garantía de optimalidad.

Para entender como se resuelve la planificación, dividiremos al problema y lo trataremos en secciones diferentes como lo hacen Guihaire y Hao, 2008.

1.4 DISEÑO DE LA RED DE TRÁNSITO

El propósito de este paso, denominado **TNDP** (Transit Network Design Problem), es definir el conjunto de rutas de los vehículos en un área en particular.

1.4.1 Entrada

Los datos necesarios para este paso son dos:

Topología. Contiene indicadores de rendimiento de rutas. La topología del área puede ser definida por calles o similares, posibles lugares para las paradas o algunas veces pueden ser locaciones de paradas ficticias que sirven como terminales o centros extremos.

Matriz origen-destino. (OD, origin-destiny). Contiene datos de demanda. Las matrices OD contienen el conjunto de centros o paradas que satisfacen lo más posible la demanda de la comunidad. Las abscisas se corresponden con los orígenes de los usuarios y las ordenadas con los destinos.

1.4.2 Principales restricciones y objetivos

Dependiendo de la política de la autoridad de tránsito, las restricciones y los objetivos se pueden mezclar, por lo que no existe regla que los diferencie:

Historia. Si existiera una red ya establecida esto puede llegar a jugar un rol en el sentido de que, por alguna razón, sea indeseable interrumpir el servicio en las líneas o rutas ya existentes.

Área de cubrimiento. Este aspecto está relacionado con la definición de nuevas rutas, o cambios de ruta. El área de cubrimiento es medida como el porcentaje de la demanda estimada que puede ser satisfecha a través del tránsito público. Esta razón puede ser calculada de varias maneras (Spasovic *et al.*, 1993), pero usualmente depende de características tales como el tamaño de la ruta, la densidad y la distancia entre paradas. (Murray, 2003; Benn, 1995). Las reglas del pulgar consideran a veces que la gente vive “cerca”, es decir como máximo a 400 o 500 metros del lugar de la parada, como parte de este porcentaje.

Ruta y viaje directos. Desde el punto de vista del usuario, la red debería estar definida de manera de que el viaje que desea realizar sea lo más directo posible. Existen en la literatura distintas definiciones para evaluar este requerimiento. El *número de trasbordos* que debe realizar el usuario es uno de los criterios más utilizados (Desaulniers y Hickman, 2007).

Satisfacción de la demanda. Cuando el origen y el destino de un usuario están muy alejados de las paradas, o cuando el viaje directo no es suficiente, la demanda puede ser considerada como insatisfecha. Computar la satisfacción de la demanda requiere conocer el viaje completo del usuario.

Entre las estrategias de operación se incluyen los siguientes aspectos:

Número de líneas o tamaño total de las rutas. Un objetivo general del operador del servicio es minimizar el tamaño total de la ruta a fin de reducir el número de vehículos y tripulación necesaria para sustentar el sistema de tránsito global. El número de líneas puede ser considerado alternativamente.

Objetivos específicos del operador. Por alguna razón, las agencias de tránsito pueden desear desarrollar una red con una distribución particular. Radial, rectangular o triangular (Van Nes, 2002) son las más utilizadas.

1.5 FRECUENCIAS DE LA RED DE TRÁNSITO

Este paso tiene como objetivo proveer para cada línea su frecuencia dentro de la TN según ciertos *períodos de tiempo*. Abreviaremos al problema del ajuste o asignación de las frecuencias de la red de tránsito como **TNFSP** (Transit Network Frequency Setting Problem). El número de idas y vueltas de las líneas son definidos durante esta etapa. Un **período planificado de servicio** de una línea se corresponde con el tiempo de salida de cada vehículo desde la parada inicial.

1.5.1 Entrada

Rutas o líneas de la red de tránsito. Es la entrada más importante a este paso por lo que ya deben estar definidas.

Demanda. Las matrices OD definidas en el paso 1 (Ceder y Wilson, 1986) son necesarias en esta etapa. Estas matrices proveen datos acordes a *períodos de tiempo* de demanda **uniformes**.

Esos períodos pueden variar de acuerdo a los siguientes criterios:

- Momento del día (mañana/tarde)
- Día de la semana (lunes a viernes/sábados, domingos y feriados)
- Época del año (Estaciones/Vacaciones/otros)

Dado que la demanda es dependiente del tiempo y elástica, la inspección debe ser realizada en períodos de tiempo y regularmente actualizada. Sin embargo, representa una dificultad real para la agencia de tránsito puesto que recolectar estos datos es una tarea compleja y costosa. Este tema está fuertemente relacionado con la política de servicio que definen las autoridades junto con el operario.

Flota de vehículos. Las frecuencias de las líneas también dependen del tamaño de flota disponible y la capacidad de los móviles. En este caso una descripción de los móviles es necesaria, especialmente si la flota es heterogénea. Los tiempos asociados a cada ruta de la red con respecto al tiempo en que un vehículo es utilizado también debe ser provisto.

También es necesario establecer los posibles patrocinios para minimizar los costos de mantenimiento de los móviles y los subsidios entregados por las autoridades.

1.5.2 Principales restricciones y objetivos

Satisfacción de la demanda. Las frecuencias del servicio deben coincidir con la demanda lo más posible a fin de evitar congestionamientos y hacinamiento en las paradas y por ende reducir la espera y las transferencias.

Número de líneas. El número de líneas es un ejemplo de la naturaleza multi-objetivo del problema. Mientras que desde el punto de vista del operador es deseable minimizar el número de recursos, el usuario desea el beneficio de tener una amplia oferta de líneas.

Límite de alcance. El operador puede imponer un mínimo o máximo de tramo o área al que una línea puede llegar según las regulaciones.

Historia. Como explicamos en la sección 1.3.2, puede ser necesario mantener ciertas rutas.

1.6 CRONOGRAMA DE LA RED DE TRÁNSITO

El **cronograma de una red de tránsito** (TNTP, Transit Network Timetabling Problem) se arma teniendo en cuenta para cada línea, una hora de salida del vehículo desde la parada inicial, un tiempo de espera en cada parada y una hora de llegada a cada parada final de la ruta.

1.6.1 Entrada

Rutas o líneas de tránsito. Las rutas de la red de tránsito constituyen la principal entrada de este paso. Los períodos planificados de servicio (tiempo del primer y último viaje y tiempo de recorrido) también son necesarios para computar el cronograma.

Demanda pública. Las frecuencias determinadas en la etapa anterior (TNFSP) definen los tiempos de cobertura de las líneas. Aquí se debe tener en cuenta la demanda de acuerdo a los distintos horarios del día y a nivel de importancia de cada punto de trasbordo, ya que esto es

necesario para asegurar una mejor calidad del servicio y minimizar el tiempo de espera de los usuarios. Este nivel de importancia puede ser deducido con la matriz OD.

1.6.2 Principales restricciones y objetivos

Satisfacción de la demanda. Obtener el cronograma permite computar el tiempo de viaje de los pasajeros. Estos tiempos deben ser minimizados para conseguir la mejor movilidad de los pasajeros. Si el tiempo de viaje para una demanda particular es muy alto puede considerarse como demanda insatisfecha.

Tiempo de salida de cada móvil y Tiempo de arribo de los móviles. Esta etapa debe obtener los tiempos de salida de cada móvil de las cabeceras de las líneas.

Coordinación del trasbordo. El cronograma global de la red debe tener en cuenta cada punto de trasbordo de cada línea de manera tal de minimizar el tiempo entre trasbordos. El número de puntos de trasbordo por línea también puede ser considerado.

Tamaño de la flota. La planificación de los vehículos es creada con el cronograma de la red de tránsito.

Historia. De manera similar a 1.3.2 a veces un cronograma histórico puede ser impuesto además del nuevo.

1.7 ASIGNACIÓN DE LA FLOTA

El propósito de este paso es obtener una secuencia factible de asignación de flota, también llamada servicio del bus, teniendo en cuenta el tiempo donde el móvil está vacío y el tiempo de recuperación del mismo a fin de conocer qué móviles serán asignados a qué rutas. Para más información sobre este punto el lector puede referirse a Bunte *et al.*, 2006.

1.8 PROGRAMACIÓN DE LA TRIPULACIÓN Y OTROS RECURSOS

El propósito principal de esta etapa es asignar los conductores de los móviles y otros recursos a la línea. Cada conductor finaliza su cronograma de un período dado. La fase de

planificación esta usualmente basada en períodos de un día mientras que para los otros recursos son períodos muchos más largos que incluyen otros tipos de restricciones, por ejemplo el número máximo de días de trabajo por semana. Para más información el lector puede consultar en Wren y Rousseau, 1993.

1.9 ORGANIZACIÓN DE LOS CONTENIDOS Y ESTRUCTURA GENERAL DE LA TESIS

Esta tesis doctoral esta organizada en diez capítulos. En el primer capítulo se establecieron los conceptos básicos para el entendimiento de la problemática multi-objetivo abordada y su dificultad de resolución. El Capítulo 2 aborda las principales metodologías utilizadas para resolver el TNP. En el Capítulo 3 se introduce la optimización multi-objetivo y definiciones. En el Capítulo 4 se presenta la teoría de algoritmos evolutivos necesaria para la comprensión de los capítulos posteriores. El Capítulo 5 define los conceptos básicos de algoritmos evolutivos multi-objetivos. El Capítulo 6 da los lineamientos básicos para simulación de eventos discretos y conceptos de teoría de colas. El Algoritmo Híbrido Dinámico Multiobjetivo (AHDM) para TNs es presentado en el Capítulo 7 . El Capítulo 8 describe los casos de estudios abordados con el AHDM y analiza sus resultados. En el Capítulo 9 mostramos las líneas de trabajo futuras y el Capítulo 10 detalla las conclusiones de esta tesis doctoral.

Resumen del capítulo

En este capítulo se introdujo al lector en la problemática de las redes de tránsito y se establecieron los conceptos a su alrededor.

Capítulo 2 Enfoque del TNP

La planificación del tránsito es un problema multi-objetivo, donde los usuarios y los operadores poseen intereses claramente en conflicto. Mientras que el usuario desea cierto nivel de calidad del servicio, el operador desea minimizar sus costos. En la práctica, las autoridades regulan imponiendo líneas que sigan una cierta ruta y frecuencias que aseguren cierto nivel de cobertura espacio-tiempo, mientras que el operador ajusta los tiempo de salida de los móviles para acomodarlos de una manera económicamente conveniente. Por esta razón, se le suele dar muy poca importancia a la sincronización de los trasbordos. Esto resulta muy perjudicial para los usuarios: sincronizar el trasbordo es un elemento crucial de la calidad del servicio y es totalmente dependiente de los horarios de salida de los vehículos.

Teniendo en cuenta lo mencionado anteriormente se realizará una revisión concentrada en las tres primeras etapas. Para establecer puntos en común con las metodologías encontradas en la literatura, se introducirá primero, una serie de conceptos y terminología asociados.

2.1 $TNDSP = TNDP + TNFSP + TNTP$

El **problema de diseño y planificación de la red de tránsito (TNDSP**, Transit Network Design Scheduling Problem) se corresponde con las primeras tres etapas del proceso de programación del tránsito: TNDP (Diseño de la red de tránsito), TNFSP (Ajuste de las Frecuencias de la red de tránsito) y TNTP (Cronograma de la red de Tránsito). Esto consiste en crear, desde cero o desde una red existente, un conjunto de rutas para las líneas, las frecuencias y los cronogramas.

La necesidad de encontrar un balance entre la satisfacción del operario del servicio y de los usuarios hace de este un problema multi-objetivo. A continuación se clasificará y analizará los estudios en este tópico según el sub-problema atacado.

2.1.1 TNDSP: Terminología

En la literatura, varios términos han sido empleados para describir los diferentes pasos del proceso de armado de la red y sus combinaciones. Por ejemplo, el problema de diseño de rutas y frecuencias puede ser nombrado como “Diseño de la red de rutas de tránsito para buses” (Fan y Machemehl, 2004), “Diseño de la red de tránsito” (Baaj y Mahmassani, 1995) o “Programación de las líneas en transporte público” (Borndörfer *et al.*, 2005). Para esta tesis doctoral haremos uso de la terminología propuesta por Guihaire y Hao (2008) que se muestran en la Figura 2.1. De esta manera se organizan de manera sencilla las definiciones; y se relacionan los problemas y sub-problemas.

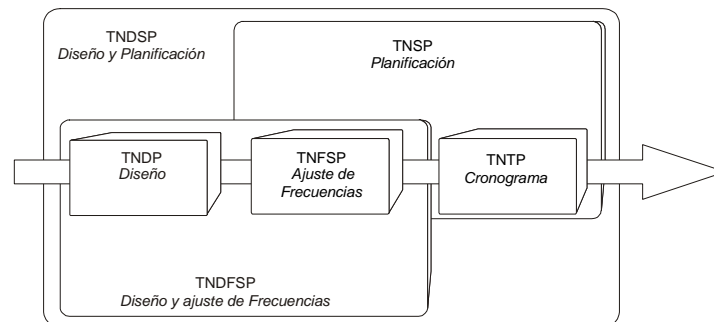


Figura 2.1 Clasificación de subproblemas del TNDSP

A partir de ahora nos referiremos a los siguientes problemas de redes de tránsito, los mismos se abreviarán por sus siglas en inglés: diseño (**TNDP**), ajuste de frecuencias (**TNFSP**) y cronograma (**TNTP**) que se muestran en la Figura 2.1. Como algunos autores abarcan más de uno de los sub-problemas introduciremos además dos problemas combinados. En la Sección 2.14, trataremos el diseño de la red y ajuste de las frecuencias (**TNDFSP=TNDP+TNFP**) y en la sección 2.15, abordaremos la planificación (**TNSP=TNFSP+TNTP**). El árbol de la Figura 2.2 resume las relaciones entre los distintos problemas TNDSP.

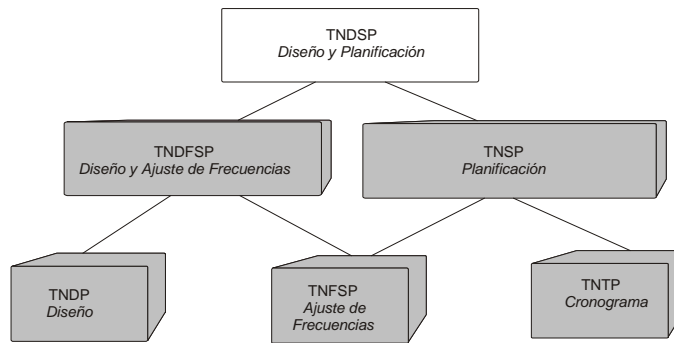


Figura 2.2 Árbol de relaciones entre problemas TN

2.1.2 Modelos matemáticos para el TNDSP

Existen una variedad de propuestas en términos de modelado para abordar el TNDSP. Estas por lo general varían en el sub-problema que atacan. Cada una está caracterizada por sus objetivos específicos y restricciones. El problema del cronograma de la red de tránsito puede ser representado a través de un problema cuadrático de semi-asignación (QSAP, Quadratic Semi Assignment Problem) (Klett y Stemme, 1988; Bookbinder y Désilets, 1992; Daduna y Voss, 1995) y a través de un programa no lineal de enteros mixto (MINLP, Mixed Integer Non-linear Problem), combinando variables binarias con números reales correspondientes a los tiempos de partida y llegada de los móviles, (Chakroborty *et al.*, 1997, 2001, 1995; Deb y Chakroborty, 1998; Jansen *et al.*, 2002). En Ceder *et al.* (2001) y Eranki (2004) el problema de planificación de la red de tránsito es modelado con programación lineal entera mixta. Borndörfer *et al.* (2005) propuso un modelo de flujo basado en el grafo de la ruta de los pasajeros. Un modelo de flujo es también aplicado al TNDSP por Yan y Chen (2002), con redes múltiples espacio-tiempo formulando movimientos de buses y flujo de pasajeros.

Cambios en la oferta de la red de tránsito impactan en la demanda. Modificaciones aplicadas a los componentes de una red de tránsito afectan a los usuarios y la manera en la que viajan. La reacción es un proceso dinámico cuya magnitud fluctúa a través del tiempo, dependiendo entre otros, de la intensidad de los cambios percibidos por cada usuario. Distintos estudios sugieren grandes diferencias entre corto plazo (1 a 2 años), mediano plazo

(5 a 7 años) y largo plazo (12 a 15 años), lo que implica que los efectos en la demanda no pueden ser deducidos en una comparación simple del antes y después. Más aún, debería hacerse un análisis periódico del cambio de la misma.

2.1.3 Resolución del problema de optimización para TNDSP

En términos de estrategias de optimización, una variedad de métodos han sido utilizados. Los métodos de búsqueda exacta son a veces aplicados cuando el problema puede ser formulado con conocimiento del modelo matemático. Estos métodos típicamente incluyen la programación lineal y algunas formas de programación entera y mixta. Note que esto permite poca flexibilidad y a veces se restringe a situaciones basadas en gran número de suposiciones y simplificaciones como así también a limitaciones de cómputo en escalas medias y grandes.

Los trabajos publicados en los últimos años utilizan métodos basados en heurísticas y metaheurísticas (Reeves, 1993). Estos métodos pueden ser clasificados en líneas generales en:

- Heurísticas ad-hoc y específicas las cuales a veces siguen un principio de construcción greedy,
- Búsqueda en vecindarios como *Simulated Annealing* y *Taboo Search*,
- Búsqueda evolutiva como algoritmos genéticos,
- Búsqueda híbrida que combina uno o más métodos.

Diferentes situaciones y modelos han sido considerados en la literatura dentro del dominio del TNDSP. La principal diferencia entre ellos es claramente las variables de decisión que utilizan.

Se explicarán en detalle los principales métodos y modelos empleados para la resolución de cada uno de los subproblemas de TNDSP abordados en la literatura.

2.2 TNDP : DISEÑO DE LA RED DE TRÁNSITO

El problema del diseño de la red de tránsito (TNDP) consiste, como dijimos antes, en determinar un conjunto de rutas, cada una con una parada inicial, una final y paradas

intermedias, un mapa o área topológica y un conjunto de objetivos y restricciones. Desde la perspectiva del usuario, una red de tránsito debe cubrir la mayor área posible de servicio.

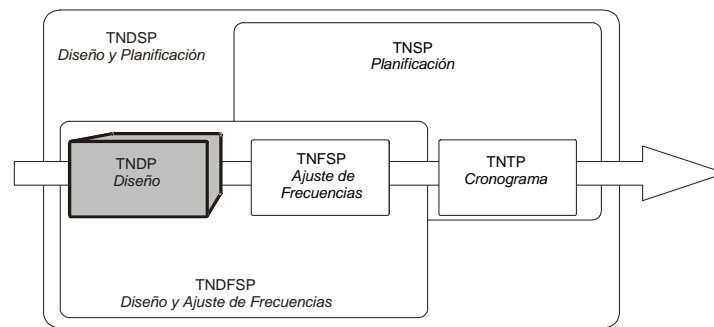


Figura 2.3 Posición del problema de Diseño (TNDP) en el contexto de TNDSP

2.2.1 TNDP: Métodos Matemáticos

Determinar la locación y densidad de las paradas es crucial para el proceso de diseño de la red de tránsito puesto que influye en la accesibilidad, desempeño y nivel de servicio. Los objetivos están asociados a: elevar la cantidad de paradas en la ruta, mejorar la accesibilidad y mantener un tiempo de viaje bajo.

Murray (2003) estudia dos variaciones de este problema. En la primera parte, la localización de las paradas en una red existente es considerada dentro del objetivo de minimizar el número de paradas. Un problema de cubrimiento de locaciones y un problema de maximización del cubrimiento son utilizadas en el modelo. La segunda parte, se dedica a la locación óptima de las paradas para crear o extender la red. Dando un número fijo de paradas a localizar, el objetivo es maximizar el acceso al servicio provisto por las áreas no cubiertas. Un problema híbrido de cubrimiento es propuesto para resolver este otro objetivo. El método tiene como punto a favor que permite expandir tanto el servicio como el acceso.

El problema de la configuración simultánea de las líneas y la asignación de pasajeros es presentado por Guan *et al.* (2003). Guan *et al.* utiliza la ciudad de Hong Kong para testear su procedimiento; el mismo utiliza programación lineal entera binaria uniendo el modelo de planificación de las líneas con el proceso de transferencia de los pasajeros. Dado el tamaño límite de la ruta, el máximo número de transferencias y restricciones de capacidad de los

móviles el objetivo es minimizar una función relacionada con el tamaño total de líneas de tránsito, el número total de líneas de tránsito tomadas y el viaje de los pasajeros. Para mantener el problema tratable bajo resolución matemática Guan *et al.* realiza una computación previa de un conjunto restringido de posibles caminos. Sin embargo, en este paso previo no se tienen en cuenta las frecuencias, el cronograma y las asignaciones de los pasajeros a las líneas no es precisa, lo que cuestiona el proceso completo.

Los métodos basados en procedimientos matemáticos tienen como principal inconveniente el tamaño de las instancias, convirtiéndolo en una limitación para el tratamiento de problemas del mundo real. Por esta razón los procedimientos heurísticos son los más utilizados.

2.2.2 TNDP: Heurísticas

Patz (1925) fue probablemente el primero en atacar el problema de diseño de redes de tránsito a través de heurísticas. Él propuso un procedimiento iterativo para generar las líneas utilizando penalizaciones. Originalmente la red contiene una línea para cada par origen-destino. Para cada línea, es calculada una penalidad basada en el nivel de servicio y el número de pasajeros que necesitan un trasbordo. De esta manera las líneas son iterativamente eliminadas de la red basándose en este valor. Este método tiene como problema principal que no es extensible a redes de gran tamaño.

En Sonntang (1977), el TNDP es atacado con un procedimiento heurístico originalmente creado para los sistemas de subte. Comenzando por una red completa las líneas son iterativamente eliminadas y los pasajeros reasignados al camino más corto según el tiempo de viaje. Este procedimiento es apropiado para redes de tamaño medio con pequeño tiempo promedio de viaje y pequeño número de transferencias.

En 1979 Mandl desarrolla un método para el TNDP comenzando con un conjunto vacío de rutas. Se propone un algoritmo heurístico para definir la red de tránsito dada una

frecuencia constante de móviles para todas las líneas. En una primera parte, rutas en el camino más corto, que contienen un par de centros, son conectadas y luego son seleccionadas el resto de las paradas. De esta manera pueden adaptar el servicio para cubrir todos los sectores. En una segunda parte, la red de tránsito es también iterativamente modificada para minimizar el tiempo de viaje de los usuarios. El método fue aplicado en una red real en Suecia, con 15 nodos y 15570 viajes por día.

2.2.3 TNDP: Métodos de búsqueda local o en el vecindario

Una metaheurística de agregación para el problema de diseño de redes de tránsito es considerada por Zhao *et al.* (2003) y Zhao *et al.* (2004). El objetivo planteado por los autores es el de minimizar el número de trasbordos al mismo tiempo que se optimizan las rutas, mientras se cubre el servicio.

Un algoritmo de Taboo Search y de búsqueda Greedy es propuesto por Zhao y Gan (2003); Zhao y Zheng (2006a). Un algoritmo básico Greedy y búsqueda por ascenso rápido es implementado en Zhao y Ubaka (2004). Estos algoritmos fueron testeados con los datos de la ciudad de Miami en Florida.

2.2.4 TNDP: Algoritmos Evolutivos

Xiong y Scheuder (1993), presentan un método innovador para seleccionar rutas adicionales en una red preexistente. Este método está basado en una mejora al algoritmo genético simple, llamado algoritmo genético acumulado. En principio recolecta todas las soluciones no dominadas y devuelve el conjunto de la última generación. Luego utiliza una red neuronal para la asignación de los viajes de los pasajeros a fin de evaluar la función de *fitness*. Este trabajo está testado con el ejemplo de LeBlank *et al.* (1975).

Chakroborty y Dwivedi (2002) también proponen un algoritmo genético que comienza con un conjunto inicial de rutas construidas heurísticamente. Luego el proceso

evalúa y modifica la ruta de manera iterativa. La solución es un conjunto de rutas. Su *fitness* es evaluado teniendo en cuenta cinco criterios, los mismos de Ceder y Wilson (1986). El testeó se realizó sobre el caso de Mandl (1979).

2.2.5 TNDP: Otros métodos

Yu *et al.* (2005) desarrolla un modelo de colonia de hormigas para el diseño de las rutas que reduce el tiempo de transferencias y maximiza el flujo de pasajeros teniendo en cuenta el tamaño de la ruta y las restricciones. Son considerados los beneficios del operario y del pasajero. El algoritmo fue testeado con los datos de la ciudad de Dalian.

2.3 TNFSP: AJUSTE DE FRECUENCIAS

El problema de ajuste de las frecuencias (TNFSP) consiste en determinar la frecuencia adecuada para cada línea en cada período de tiempo. Un período de tiempo se define según la hora del día, el día de la semana o la estación del año. Es deseable que una frecuencia provea un servicio regular que satisfaga al usuario y minimice los costos del operador.

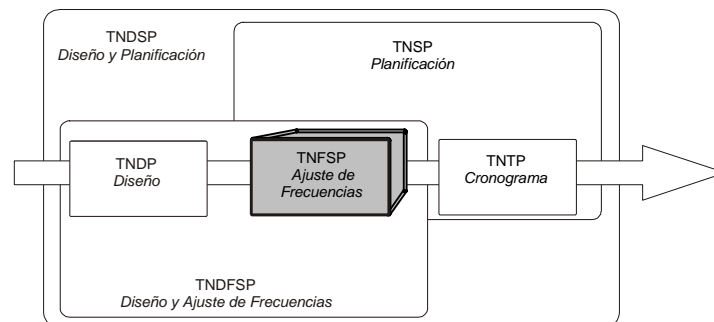


Figura 2.4 TNFSP en el contexto de TNDSP

2.3.1 TNFSP: Métodos matemáticos

Salzborn (1972) determina la frecuencia de los pasajeros dándoles una tasa de arribo para determinar el tamaño de la flota y el tiempo de espera. Más tarde, Salzborn (1980) analiza el problema de un sistema de transporte alimentado entre ciudades, con una ruta de transferencia y determina los requerimientos de tal sistema. El principal inconveniente de esta propuesta es que el modelo está altamente simplificado.

En 1980, Scheele utiliza un modelo no lineal con el objetivo de minimizar el tiempo de viaje de los pasajeros y resolver al mismo tiempo el problema de la frecuencia. El resultado es una localización óptima de móviles a rutas.

Constantin y Florian (1995) presentan un modelo cuyo objetivo es minimizar el tiempo de viaje total y de espera de los pasajeros. Utilizan programación mixta no lineal y no convexa. Para los tests utilizaron un área urbana en Estocolmo, Winnipeg y Portland.

2.3.2 TNFSP: Procedimientos heurísticos

Han y Wilson (1982), obtienen las frecuencias de las líneas teniendo en cuenta el tamaño total de la flota y una capacidad para cada ruta. La heurística tenía dos etapas. En la primera, se minimiza el conjunto de frecuencias para satisfacer la demanda y en la segunda, las frecuencias son incrementadas uniformemente para las líneas según la flota disponible.

En 1986 Ceder y Wilson definen y presentan un modelo conceptual para el proceso de planeamiento completo como una secuencia sistemática de decisiones. El mismo consta en 5 niveles: diseño de las rutas, ajuste de frecuencias, cronograma, programación de los móviles y programación de la tripulación.

Chowdhury y Chien (2001), consideran la coordinación de trasbordo para redes de tránsito intermodales optimizando los recorridos y los tiempos perdidos. El tiempo perdido comprende el tiempo adicional dado a la programación de un viaje con el objeto de compensar los posibles retrasos de los móviles. Estos autores proponen un modelo de programación matemático y luego un procedimiento que optimiza los recorridos primero sin tener en cuenta la coordinación de las líneas y luego optimizando los tiempos perdidos en el contexto de un tránsito intermodal.

Recientemente, Gao *et al.* (2003) propusieron una técnica de programación a dos niveles para tratar el TNFSP. El primer nivel minimiza los tiempos en el vehículo y de

espera de los usuarios. En el segundo nivel utiliza un modelo de asignación equilibrado para describir las rutas alternativas de los usuarios.

2.3.3 TNFSP: Otras técnicas

Park (2005) utiliza algoritmos genéticos y simulación para optimizar la programación de los móviles en redes de área urbana. Dos casos diferentes son considerados. Cuando los buses arriban siguiendo un procedimiento determinístico, un algoritmo genético simple es utilizado en combinación con operadores genéticos específicos para optimizar las rutas. Por otro lado, cuando el arribo de los buses es estocástico, un algoritmo basado en simulación es utilizado para optimizar tanto las rutas como los tiempos perdidos, también utiliza operadores genéticos específicos que incluyen el generador de rutas coordinadas, *crossover* y mutación.

2.4 TNTP: CRONOGRAMA DE UNA RED DE TRÁNSITO

El problema de cronograma de una red de tránsito (TNTP) consiste en ajustar un cronograma para cada línea de una red de tránsito, con tiempos de salida desde su cabecera. Este proceso puede ser dependiente o no de un conjunto preexistente de frecuencias. Para los usuarios, un cronograma es deseable si desean realizar trasbordos. Desde la perspectiva del operador apropiados tiempos de salida pueden ayudar a reducir la flota requerida por la línea. Sin embargo, este punto de vista no ha sido tratado en la literatura o se lo toma de manera muy superficial (por ejemplo, algunos los tratan como costos fijos, o consideran los vehículos que salen de la parada inicial, llegan a la final y no la vuelta).

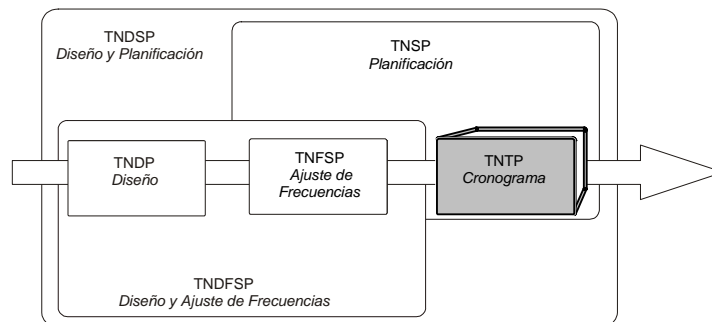


Figura 2.5 TNTP en el contexto de TNDSF

Con el objetivo de reducir la espera de los pasajeros en las zonas de trasbordo, estas transferencias pueden ser coordinadas, especialmente cuando las rutas son largas. En las grandes ciudades las frecuencias tienden a ser altas y perder una conexión sólo extiende el tiempo de espera por unos pocos minutos. Además debido a la congestión coordinar los trasbordos puede resultar en mayor espera y la ausencia de sincronización puede enojar a las personas que utilizan el transporte público.

Existen básicamente dos opciones para abordar el problema de la coordinación de trasbordos: tiempos de trasbordo y optimización del trasbordo (Bookbinder y Désilets, 1992). En el primer caso, los móviles de diferentes líneas se planifican para encontrarse en cierto punto de trasbordo, a través de esperas en la parada, mientras que en el otro caso las líneas son planificadas de manera de minimizar el tiempo total de trasbordo de los pasajeros. Una manera particular de coordinar los trasbordos es normalizar las líneas de manera que vehículos de diferentes rutas arriben a la parada al mismo tiempo a un punto de trasbordo. Por el contrario cuando las líneas tienen frecuencias notablemente diferentes los valores pueden ser fijos o múltiplos de enteros de la ruta más corta (Ting y Schonfeld, 2005).

2.4.1 TNTP: Métodos Matemáticos

De Palma y Lindsey (2001) consideran el problema del cronograma como una conexión simple de tránsito. Cada pasajero tiene un tiempo ideal para abordar y se le asigna un costo por llegar tarde o antes. Un modelo analítico es desarrollado para manejar este proceso de dos pasos: en la primera parte, un problema de locación de la demanda es resuelto asignando a cada individuo una línea. En la segunda parte, un proceso de optimización es llevado a cabo para ajustar los tiempos de partida de manera tal de minimizar los costos de espera del tiempo total de planificación del pasajero. Desafortunadamente, los datos extras necesarios son extremadamente difíciles de obtener para aplicarlo en el mundo real.

Wong y Leung (2004) abordan el problema estático del cronograma y presentan un método para maximizar la sincronización entre líneas y habilitar trasbordos con mínimo tiempo de espera. Este método considera un tiempo de trasbordo conocido para cada estación de intercambio y simplificaciones del problema tales como capacidad de la móvil ilimitada y exacta adherencia a la planificación. La asignación de los usuarios es realizada de la siguiente manera: número de trasbordos y número de paradas en el viaje. Un modelo de programación entera mixta es propuesto con tiempos de ida, tiempos de vuelta y tiempos de salida de cada parada como variables de decisión.

2.4.2 TNTP: Técnicas heurísticas

Klemm y Stemme (1988) abordan el TNTP con la perspectiva de optimización de trasbordos a través de un modelo de programación entero y un método de resolución heurístico. Este último es un proceso constructivo donde las líneas son planificadas una a una teniendo en cuenta la sincronización de los trasbordos.

Teniendo en cuenta los trasbordos cuidadosamente, Fleurent *et al.* (2004) describen los conceptos que son implementados en un software comercial HASTUS (<http://www.giro.ca/en/products/hastus/index.htm>) para generar cronogramas de tránsito sincronizados.

2.4.3 TNTP: Técnicas de búsqueda local o en el vecindario

Daduna y Voss (1995) proponen un método para determinar los tiempos de salida de los móviles en una red *preestablecida* con frecuencias de líneas. El objetivo es sincronizar los tiempos de arribo en las zonas de trasbordo como así también minimizar el tiempo de espera. Sin embargo, este objetivo puede ser refinado y por ejemplo incluir pesos para los diferentes centros de trasbordo o el máximo tiempo de espera en el punto de trasbordo. Utilizan *Simulated Annealing* y *Taboo Search*.

En el 2002, Jansen *et al.*, proponen un método para sincronizar cronograma de buses y minimizar el tiempo de trasbordo de los pasajeros, a través de una red de líneas con cabecera fija. El tiempo de parada y en el móvil se asumen constantes y determinísticos, el problema se reduce ajustando tiempos de salida para la primera salida de cada línea. La ciudad de Copenhagen se encaró con *Taboo Search* a través de este método.

2.4.4 TNTP: Algoritmos Evolutivos

Chakroborty *et al.* (1995), Chakroborty *et al.* (1997) y Deb y Chakroborty (1998) se enfocan en la utilización de algoritmos genéticos para determinar los tiempos de partida de los móviles de una red de tránsito de una manera coordinada, a la vez que se minimiza el tiempo de espera de los clientes. Entre las restricciones de la técnica se encuentran: un tamaño de flota fijo y predeterminado, límite de tiempo de detención en una parada y máximo tiempo de trasbordo. Al principio utilizaron programación no lineal mixta pero rápidamente se dieron cuenta de que no se podía aplicar a una red de tamaño considerable. Por este motivo en 1997 Chakroborty propone un algoritmo genético para el problema. Entre las simplificaciones realizadas por el autor se puede mencionar la capacidad ilimitada de los vehículos y tamaño de flota fija. Por otra parte, Chakroborty *et al.* (2001) definen un problema ligeramente distinto donde la coordinación de trasbordos es combinada con la planificación de los vehículos. El modelo es un problema entero mixto y resuelto por los autores a través de un algoritmo genético binario.

Cevallos y Zhao (2006) presentan una aproximación para el problema de sincronización de trasbordo de los móviles donde consideran como objetivo minimizar el tiempo de espera en trasbordo dado un cronograma bajo la restricción estricta de cabeceras fijas en los recorridos de las líneas. Esta restricción priva la búsqueda de flexibilidad pero complementada con el hecho de que los recursos son dejados a un lado hacen el problema

más tratable que los presentados anteriormente especialmente para problemas de tamaño considerable.

2.4.5 TNTP: Otros métodos

Bookbinder y Désilets (1992) proponen un método híbrido de optimización de los trasbordos que ajusta los tiempos de partida de las líneas en una red dada con cabeceras fijas. Estos autores combinan la flexibilidad de la simulación con la optimización matemática. La simulación les permite tener en cuenta los tiempos estocásticos de viaje de los móviles mientras que el procedimiento matemático optimiza los trasbordos según funciones relacionadas con los tiempos de espera.

2.5 TNDFSP = TNDP + TNFSP

El TNDFSP comprende dos etapas: el diseño de las rutas (TNDP), y el ajuste de las frecuencias (TNFSP). Yu *et al.* (2005) argumenta que el diseño de las rutas y el ajuste de las frecuencias (TNDFSP) no pueden ser abordados simultáneamente. Sin embargo, la interacción que existe entre ambos problemas puede hacer que su resolución simultánea sea beneficiosa para los resultados finales. Más aún, el diseño de la red de tránsito necesita la asignación de viajes de los pasajeros. La Figura 2.6 muestra la relación de los problemas.

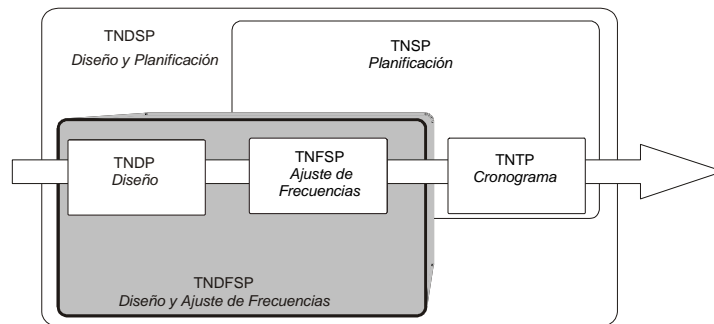


Figura 2.6 Esquema del problema TNDFSP en el contexto del TNDSP

2.5.1 TNDFSP: Resolución

Hasselström (1979, 1981) presenta un modelo de tres pasos aplicable al mundo real para resolver el TNDFSP. En el modelo rutas y frecuencias son determinadas simultáneamente.

Bussieck (1998), enfocado en el transporte férreo, utiliza un modelo matemático para crear rutas y frecuencias. Utilizando algoritmos genéticos, Ngamchai y Lovell (2003) proponen un método para crear rutas y frecuencias minimizando el costo total del tamaño de la flota y el tiempo de espera y en tránsito de los pasajeros. El método fue testeado con los datos de Pattnaik *et al.* (1998). Tom y Mohan (2003) también presentan un algoritmo genético que determina las rutas y las frecuencias. Su esquema de codificación incorpora la frecuencia como variable. Finalmente Fan y Machehmel (2004, 2006a) guían la solución utilizando un algoritmo genético dada una demanda.

Baaj y Mahmassani (1991, 1995) proponen un algoritmo híbrido que combina procedimientos expertos con algorítmicos para resolver el TNDFSP. Shih y Mahmassani (1994) y Shih *et al.* (1998) utilizan un *framework* de cuatro etapas para resolver el problema multi-objetivo. El inconveniente principal de este trabajo es que no es aplicable a problemas del mundo real.

Fusco *et al.* (2002) presenta un procedimiento secuencial de tres pasos combinando algoritmos genéticos con heurísticas simples para generar las rutas de las líneas a un costo mínimo.

2.6 TNSP = TNFSP + TNTP

Dado un conjunto de rutas, el problema de planificación de redes de transporte (TNSP) consiste en obtener la frecuencia (TNFSP) y los tiempos de salida de cada línea (TNTP), teniendo en cuenta un conjunto de restricciones y objetivos. La Figura 2.7 muestra el TNSP dentro del esquema global del problema TNDSP que estamos estudiando.

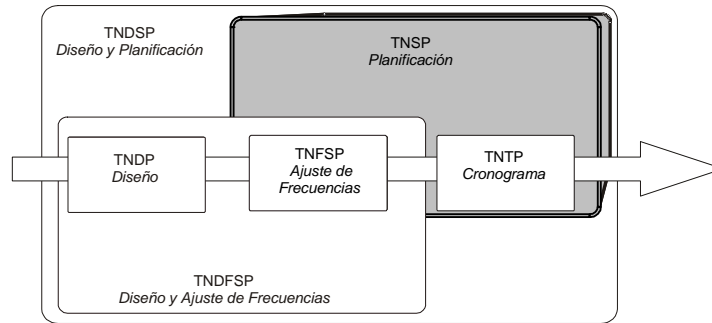


Figura 2.7 Esquema del problema TNDFSP en el contexto del TNDSP

2.6.1 TNSP: Resolución

Koutsopolos *et al.* (1985) desarrollan un modelo matemático y un método de resolución para una simplificación de este problema. Rapp y Gehner (1976) presentan un método para optimizar trasbordos en el contexto de TNSP. Ceder *et al.* (2001) propone una heurística para ajustar las frecuencias y tiempos de salida en orden de maximizar el número de arribo simultaneo a los nodos de trasbordo. Una limitación del método de Ceder *et al.* radica en que son considerados exclusivamente, los arribos simultáneos, cuando a veces puede ser interesante otra perspectiva. En un estudio posterior Ceder (2003b) muestra tres procedimientos basados en la idea anterior para setear frecuencias y tiempos de salida en una red con restricciones de frecuencias y varios objetivos.

2.7 CARENCIAS DE LOS MÉTODOS ACTUALES PARA RESOLVER EL TNP

Como se ha podido observar el problema de diseño y planificación de redes de tránsito (TNDSP) es un tópico activo de investigación. Dado que las regulaciones de tránsito y la demanda del servicio cambian continuamente, los operarios del servicio deben actualizar sus políticas permanentemente. Hemos resumido los modelos y métodos relacionados con el TNDSP. En algunos hemos expresado las carencias principales de los mismos. En resumen podemos decir que existe una falta general de atención en los sub-problemas que trae como consecuencia la sub-optimalidad de los resultados obtenidos por los métodos.

El tamaño de los casos de estudio abordados por varios autores es de pequeña escala y la aplicación de sus métodos en problemas de tamaño real ha sido probado ser imposible. También uno de los tópicos más dejados de lado es el cronograma de los móviles y la sincronización de los trasbordos no es optimizada correctamente debido a la naturaleza estocástica de la demanda.

La mayoría de los estudios están enfocados en el tránsito urbano y no en el interurbano y aunque parezcan similares existen diferencias contextuales que dificultan la transferencia de un método para el tratamiento de tránsito urbano al interurbano.

En varios de los métodos el tiempo de arribo de los buses se toma constante o fijo lo que pierde de contexto el área urbana donde transita, congestionamientos en horas pico, etc. Tener en cuenta rasgo estocástico puede ayudar a asegurar posibilidades de trasbordo y minimización del tiempo viaje de los pasajeros.

La asignación de viaje de los pasajeros es importante en muchas maneras. Cuantos menos trasbordos realice el usuario más satisfecho se encontrará con el servicio. Como vimos, la mayoría de los métodos trata el TNDSP en secuencia, no dando posibilidad a la interacción los problemas dentro de él.

Por lo todo lo expuesto anteriormente en esta tesis doctoral presentaremos un método para la resolución del TNDSP de manera integral, teniendo en cuenta la variabilidad de la demanda y los problemas asociados a los congestionamientos de tránsito. La técnica puede ser utilizada en cualquier ciudad de tamaño real como así también al tránsito interurbano.

Resumen del capítulo

Este capítulo se centró en mostrar los principales enfoques encontrados en la literatura utilizados para la resolución del TNP. Además se clasificó al problema en distintos sub-problemas teniendo en cuenta cómo diferentes autores habían abordado la resolución del mismo

Capítulo 3 Optimización Multi-objetivo

Todos los días en el mundo real se presentan situaciones que poseen objetivos específicos. Cuando un problema de optimización posee únicamente una función objetivo a optimizar estaremos en presencia de una optimización de un único objetivo, denominado *optimización mono-objetivo*. Por otro lado, cuando el problema a optimizar posee más de una función objetivo, la tarea de encontrar una o más soluciones óptimas se conoce como *optimización multi-objetivo*. A continuación definiremos los conceptos fundamentales que nos permitirán analizar la resolución de los problemas multi-objetivo desde un marco teórico.

3.1 OPTIMIZACIÓN MONO-OBJETIVO

Entre las definiciones de *optimización mono-objetivo* que se presentan en la literatura se enunciará a continuación la provista por Coello Coello (2007), la cual nos permitirá inmediatamente definir a la *optimización multi-objetivo*.

Definición 3.1: *Un problema de optimización mono-objetivo está definido como minimizar (o maximizar) $f(x) \in \mathbb{R}$ sujeto a $g_i(x) \leq 0$, $i = \{1, \dots, m\}$ y $h_j(x) = 0$, $j = \{1, \dots, p\}$ con $x \in \Omega$.*

Una solución minimiza (o maximiza) un escalar $f(x)$ con $m + p$ restricciones, donde x es un vector de variables de decisión n -dimensional del universo Ω .

Las inecuaciones $g_i(x) \leq 0$ y las ecuaciones $h_j(x) = 0$ representan las restricciones que debemos satisfacer mientras optimizamos $f(x)$. Ω contiene todas las posibles x que pueden ser usadas para satisfacer $f(x)$ y sus restricciones.

3.2 OPTIMIZACIÓN MULTI-OBJETIVO

Estamos ahora en condiciones de definir formalmente los problemas de *optimización multi-objetivo* (*Multi-Objective Optimization Problem*, **MOOP**).

Definición 3.2: Un problema general de *optimización multi-objetivo* es definido como: *minimizar* (o *maximizar*) $F(x) = (f_1(x), \dots, f_k(x))$ sujeto a $g_i(x) \leq 0$, $i = \{1, \dots, m\}$ y $h_j(x) = 0$, $j = \{1, \dots, p\}$ con $x \in \Omega$. Un problema de optimización multi-objetivo minimiza (o maximiza) los componentes de $F(x)$ donde x es un vector n -dimensional $x = (x_1, \dots, x_n)$.

Una de las diferencias más significativas entre la optimización multi-objetivo y la mono-objetivo es que en la optimización multi-objetivo las funciones constituyen un espacio multidimensional. Este espacio de búsqueda es conocido como el espacio de funciones objetivo Z . Para cada solución x en el espacio de las variables de decisión existe un punto en el espacio de la función objetivo denotada como $F(x) = z = (z_1, \dots, z_k)$. La Figura 3.1 muestra los dos espacios y la correspondencia entre ellos.

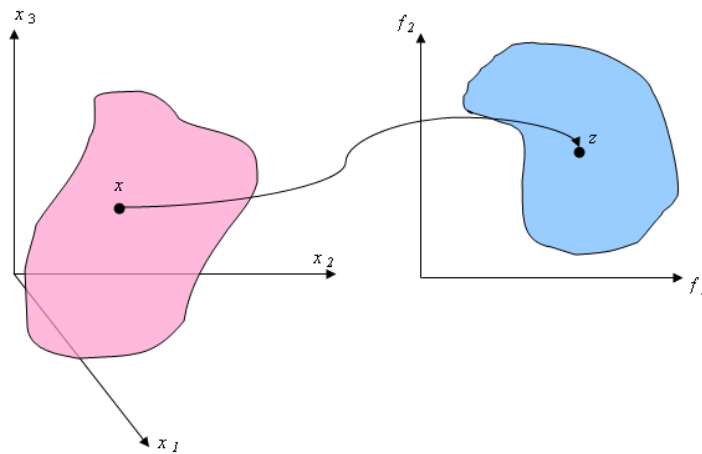
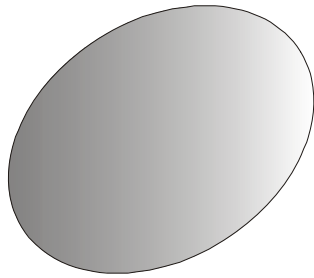


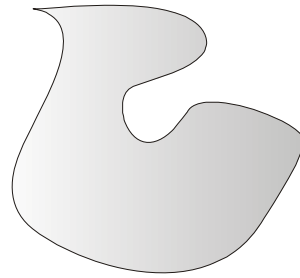
Figura 3.1 Espacio de decisión (en color rosa) vs. Espacio de la función objetivo (en color celeste)

A partir de aquí nos concentraremos en la forma de evaluar la calidad de una solución con respecto a otra. Es importante distinguir la forma del espacio de soluciones. La Figura 3.2

representa los conceptos que yacen tras la idea de convexidad (2.2a) y concavidad (2.2b) de $f(x)$.



a. Conjunto de soluciones convexas



b. Conjunto de soluciones cóncavas

Figura 3.2 Dos espacios de soluciones.

Definición 3.3: Una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ se dice convexa si para cualquier par de soluciones x_1 y $x_2 \in \mathbb{R}^n$,

$$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta)f(x_2) \quad (3.1)$$

donde θ es un escalar en el rango de $0 \leq \theta \leq 1$.

Propiedades: Entre las propiedades asociadas a las funciones convexas podemos nombrar.

1. La aproximación lineal de $f(x)$ en cualquier punto del intervalo $[x_1, x_2]$ siempre estima por debajo el valor actual de la función.
2. La matriz Hessiana de $f(x)$, con $x = (x_1, \dots, x_n)$, es positiva para todos los x . Cabe aclarar que la matriz Hessiana de $f(x)$ es una matriz cuadrada de $n \times n$ cuyos elementos son las segundas derivadas parciales de f .
3. Para toda función convexa, el mínimo local es siempre un mínimo global.

El problema de optimización es complicado de abordar porque generalmente existe un compromiso entre recursos y precisión para seleccionar una solución. En vez de buscar una única solución global, por cuestiones prácticas, a menudo uno debe conformarse con un óptimo local. Al tener varias funciones objetivo, el problema se complica porque es necesario adaptar la noción de óptimo. En muchos casos, los objetivos planteados son contrapuestos y,

por ende, una mejora en uno implica un decremento en otro. Esto implica una toma de decisiones previa o posterior, que caracteriza a un dado método.

3.3 ÓPTIMO DE PARETO

La noción de óptimo para problemas multi-objetivo fue originalmente propuesta por Francis Isidro Edgeworth (1881), y luego generalizada por Vilfredo Pareto (1896). Generalmente se conoce como *óptimo de Pareto*.

Definición 3.4: Una solución $x \in \Omega$ se dice **Pareto óptima**, con respecto a Ω , si y sólo si no existe $x' \in \Omega$ tal que $v = F(x') = (f_1(x'), \dots, f_k(x'))$ domina $u = F(x) = (f_1(x), \dots, f_k(x))$.

Expresado en otras palabras, la definición dice que x' es Pareto óptima si, asumiendo minimización, no existe una solución factible x que al decrecer alguno de los valores de las funciones $f_i(x)$, no cause un incremento simultáneo en el valor de las otras. Es posible en este punto introducir una relación de orden parcial denominada **dominancia** entre vectores solución del problema de optimización multi-objetivo.

Definición 3.5: Un vector solución $u = (u_1, \dots, u_k)$ se dice que **domina** a otro vector solución $v = (v_1, \dots, v_k)$ si y sólo si u es parcialmente menor que v , i.e., $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$. La relación de dominancia se denota como $u \leq v$.

Un conjunto Pareto óptimo para un problema de optimización multi-objetivo está dado por el conjunto de soluciones no dominadas. Esto permite definir el concepto de **frontera de Pareto**.

Definición 3.6: Para un problema de optimización multi-objetivo, un conjunto $F(x)$ y un conjunto Pareto óptimo P^* , la **frontera de Pareto** PF^* (por sus siglas en inglés) se define como:

$$PF^* = \{u = F(x) \mid x \in P^*\} \quad (3.2)$$

La función de evaluación $f : \Omega \rightarrow \Lambda$ transforma las variables de decisión en vectores.

Por último, definiremos las nociones de Pareto optimalidad *fuerte* y *débil*.

Definición 3.7: Una solución x_1 domina fuertemente a otra x_2 ($x_1 \prec x_2$), si la solución x_1 es estrictamente mejor que la solución x_2 para todos los objetivos k . Dentro de un conjunto P de soluciones, el conjunto de soluciones débilmente dominadas P' son aquellas que no son fuertemente dominadas por ningún otro miembro de P .

Encontrar el conjunto de soluciones no dominadas es similar a encontrar el mínimo de un conjunto de número reales. Existen varias formas de identificar el conjunto de soluciones no dominadas. El método fue propuesto por Kung *et al.* (1975). En esta técnica primero se ordenan las soluciones en orden descendente de importancia desde el primer valor de la función objetivo. De esta manera, el conjunto de soluciones es dividido en dos, llamémosles T (*Top*) y B (*Bottom*). Conociendo que la primera mitad de las soluciones es mejor en términos del primer objetivo, tomamos la segunda mitad de soluciones y verificamos la dominación de la otra mitad. Y así sucesivamente con todos los objetivos. Las soluciones de B que no son dominadas por ninguna de T son combinadas con el conjunto T para formar el nuevo conjunto P.

3.3.1 Condiciones de optimalidad

Para concluir con las definiciones asociadas a optimalidad de Pareto introduciremos la condición necesaria para que una solución sea Pareto óptima.

Teorema 2.1: La condición necesaria para que x^* sea Pareto Óptima es que exista un vector $\lambda \geq 0$ y $u \geq 0$ (donde $\lambda \in \mathbb{R}^k$, $u \in \mathbb{R}^j$ y $\lambda, u \neq 0$) tal que las siguientes condiciones sean ciertas:

$$\sum_{h=1}^k \lambda_h \Delta f_h(x^*) - \sum_{i=1}^j u_i \Delta g_i(x^*) = 0 \quad (3.3)$$

$$u_i g_i(x^*) = 0 \text{ para todo } i = 1, \dots, j \quad (3.4)$$

La prueba de este teorema escapa el contenido de esta tesis doctoral, el lector puede encontrarla en Da Cunha y Polak (1967).

La condición suficiente para Pareto optimalidad se puede expresar en el siguiente teorema Kuhn-Tucker (1951).

Teorema 2.2: *Tomemos funciones objetivo convexas y las restricciones para un problema dado con las funciones objetivo y las restricciones continuamente diferenciables en la solución x^* . Una condición suficiente para x^* sea Pareto óptima es que exista un vector $\lambda > 0$ y $u \geq 0$ (donde $\lambda \in \mathbb{R}^k$ y $u \in \mathbb{R}^j$) tal que las siguientes ecuaciones son verdaderas:*

$$\sum_{h=1}^k \lambda_h \Delta f_h(x^*) - \sum_{i=1}^j u_i \Delta g_i(x^*) = 0 \quad (3.5)$$

$$u_i g_i(x^*) = 0 \text{ para todo } i = 1, \dots, j \quad (3.6)$$

Puede el lector buscar la prueba de este teorema en Miettinen (1999).

Definición 3.8: *Si una solución x al problema de optimización multi-objetivo es no inferior, entonces existe un $w_l \geq 0$, $l = 1, \dots, k$ (w_r es estrictamente positivo para algún $r = 1, \dots, k$) y $\lambda_i \geq 0$, $i = 1, \dots, m$ tal que:*

$$x \in \Omega \quad (3.7)$$

$$\lambda_i g_i(x) = 0 \quad i = 1, \dots, m \quad (3.8)$$

$$\sum_{l=1}^k w_l \Delta f_l(x) - \sum_{i=1}^m \lambda_i \Delta g_i(x) = 0 \quad (3.9)$$

Estas condiciones son necesarias para una solución no inferior, y donde todas las $f_l(x)$ son cóncavas, en el caso de que Ω sea un conjunto convexo, ellas son suficientes también. A éstas se las llama **condiciones de no inferioridad** de Kuhn-Tucker.

Dado que hemos definido la teoría que subyace detrás de los problemas multi-objetivo, ahora es necesario concentrarnos en los métodos que se han desarrollado para su tratamiento. En la siguiente sección introducimos los métodos clásicos y las técnicas meta-heurísticas que

más se destacan en la literatura, el lector puede buscar más información sobre dichos métodos en Coello Coello (2007).

3.4 OPTIMIZACIÓN MULTI-OBJETIVO A TRAVÉS DE MÉTODOS CLÁSICOS

Existen numerosas formas enfocar la resolución para problemas con múltiples objetivos. Los métodos o bien generan todas las soluciones no dominadas o bien construyen una solución por compromiso única.

Entre los métodos clásicos más utilizados para optimización multi-objetivo podemos mencionar aquellos que trabajan con problemas de optimización multi-objetivo lineales y los no lineales. Entre los métodos clásicos encontramos dos categorías: los métodos simples y los métodos basados en gradiente (Deb, 1995). En los métodos simples solamente la función objetivo y los valores de las restricciones son utilizados como guía para la estrategia de búsqueda. Por otra parte, los métodos basados en gradiente usan el primer y segundo orden de derivadas de la función objetivo y las restricciones para guiar el proceso de búsqueda.

Por lo general, a estos algoritmos se los clasifica de varias maneras. Una de las más utilizadas es la propuesta por Cohon y Marks (1975),

1. Articulación de preferencia a *priori*, la cual toma decisiones antes de realizar la búsqueda de la solución.
2. Articulación de preferencia a *posteriori*: realizan la búsqueda primero y luego se toman las decisiones sobre cual solución elegir.

3.4.1 Articulación a priori

3.4.1.1 Método *simplex* multi-objetivo

Este método fue desarrollado por Zeleny (1975). El mismo permite encontrar todos los puntos extremos no dominados en problemas multi-objetivo lineales. A partir de estos puntos

obtiene el conjunto de soluciones factibles. Este método crea una tabla *simplex* multi-objetivo similar a la del método *simplex* tradicional

3.4.1.2 Programación fraccional

Existen problemas donde las funciones objetivo están definidas a través de ratios o fracciones. Esto provoca que las funciones se vuelvan no lineales. La forma de resolver estos problemas es a través de programación funcional. La programación funcional multi-objetivo fue propuesta por Tigan (1975) y ampliada por Kornbluth (1979). Aquí sólo explicaremos la programación por metas por ser una de las técnicas más utilizadas.

3.4.1.2.1 Programación por metas

Este método consiste en minimizar el conjunto de desviaciones de las metas preestablecidas (Charnes y Cooper, 1961). Las funciones objetivo y restricciones se definen como:

$$f_i(x) = \sum_{k=1}^p c_{ik}x_k + d_i^- - d_i^+ = b_i, i = 1, \dots, n \quad (3.10)$$

$$g_j(x) = \sum_{k=1}^p a_{jk}x_k = e_j, j = 1, \dots, m \text{ para todo} \quad (3.11)$$

donde d_i^- indica las desviaciones negativas, d_i^+ indica las positivas y b_i son las m metas de las funciones objetivo o valores de las restricciones.

3.4.1.2.2 Programación por compromiso

La programación por compromiso es una metodología mucho más general que permite solucionar problemas multi-objetivo no lineales y se basa en el concepto de distancia a una *solución ideal*. Esta solución ideal indica los valores extremos óptimos que pueden tomar cada una de los objetivos.

Definición 3.9: Sea $FAC = \{x_1, \dots, x_K\}$ el conjunto de soluciones factibles, entonces

$y_i^k = f(x^k)$ indica el valor de la k -ésima solución para el objetivo i -ésimo. La solución ideal

$y^* = (y_1^*, \dots, y_m^*)$ se define como:

$$y^* = \min_{k=1,\dots,K} y_i^k, \quad i=1,\dots,m \quad (3.12)$$

Si tenemos m objetivos, la distancia de una solución cualquiera $y = (y_1, \dots, y_m)$ a la solución ideal $y^* = (y_1^*, \dots, y_m^*)$ puede expresarse de forma generalizada en la siguiente ecuación:

$$d_p = \left(\sum_{i=1}^m \lambda_i^p (y_i^* - y_i)^p \right)^{1/p} \quad (3.13)$$

donde $\lambda_i > 0$, para $i=1,\dots,m$ y $\sum_{i=1}^m \lambda_i = 1$ son los pesos que indican el nivel de contribución de cada objetivo a la distancia total, por último p puede variar entre 1 e infinito.

De esta manera, Zeleny (1973) introdujo el concepto de programación por compromiso como toda solución x que minimiza el valor de d_p para algún vector de pesos $(\lambda_1, \dots, \lambda_m)$ y para algún valor de p . Las soluciones obtenidas a través de esta técnica son no dominadas.

3.4.2 Articulación a posteriori

3.4.2.1 Combinación lineal de pesos

Zadeh (1963) fue el primero en demostrar que la tercera condición Kuhn-Tucker para soluciones no inferiores implicaba que dichas soluciones no inferiores podían ser encontradas resolviendo un problema de optimización escalar. En este problema, la función objetivo es una suma de pesos de los componentes de la función original de vector evaluado. Dicho de otra manera, la solución al problema es en general no inferior a:

$$\min \sum_{i=1}^k w_i f_i(x). \quad (3.14)$$

Sujeta a

$$x \in \Omega \quad (3.15)$$

donde $w_i \geq 0$ para todo i y es estrictamente positivo para al menos uno de los objetivos.

El conjunto de soluciones no inferiores puede ser generado paramétricamente variando los pesos w_i en la función objetivo.

3.4.2.2 El método de las restricciones ε

Este método también proviene de las condiciones de Kuhn-Tucker. La ecuación de condición de no inferioridad (ecuación (3.9)) puede reescribirse de la siguiente manera:

$$w_r \Delta f_r(x) + \sum_{l=1, l \neq r}^k w_l \Delta f_l(x) - \sum_{i=1}^m \lambda_i \Delta g_i(x) = 0. \quad (3.16)$$

Dado que sólo los valores relativos de los pesos poseen significado, el r -ésimo objetivo puede ser seleccionado de tal manera que $w_r = 1$. Luego, la ecuación (3.16) puede ser escrita como:

$$\Delta f_r(x) + \sum_{l=1, l \neq r}^k w_l \Delta f_l(x) - \sum_{i=1}^m \lambda_i \Delta g_i(x) = 0. \quad (3.17)$$

Las soluciones no inferiores pueden ser resueltas como:

$$\min f_r(x) \quad (3.18)$$

sujeto a:

$$f_l(x) \leq \varepsilon_l \text{ para } l = 1, \dots, k \text{ y } l \neq r \quad (3.19)$$

donde ε_l son los valores de las funciones objetivo que deben ser excedidos.

3.5 CONCLUSIONES SOBRE LOS MÉTODOS CLÁSICOS Y LA OPTIMIZACIÓN

MULTI-OBJETIVO

Aunque cada uno de estos métodos posee ventajas y desventajas diferentes, todos ellos comparten ciertas dificultades a la hora de obtener soluciones múltiples Pareto óptimas:

- Sólo una solución Pareto óptima puede ser encontrada con un algoritmo clásico.
- No todas las soluciones Pareto óptimas pueden ser encontradas con algunos de los algoritmos para problemas no-convexos.

- Todos los algoritmos requieren conocimiento del problema estudiado, como pesos o valores.

En conclusión podemos decir que los métodos clásicos por lo general resuelven el problema multi-objetivo reduciéndolo a un único objetivo con la ayuda de algún conocimiento del problema.

Resumen del capítulo

En este capítulo se mostraron los métodos clásicos más reconocidos. Entre ellos están: el método por pesos que sugiere optimizar una suma de pesos de múltiples objetivos, el ϵ -restricciones implica utilizar sólo uno de los objetivos como función a optimizar e indicar el resto como restricciones. Los métodos por pesos de métricas, sugieren minimizar una métrica construida para todos los valores objetivo relacionándolos. La programación por metas sugiere minimizar una suma de pesos de las derivadas de las funciones a partir de metas especificadas por el usuario.

Capítulo 4 Técnicas Meta-Heurísticas:

Algoritmos Evolutivos

Heurística proviene de la palabra griega *heuriskein* que significa descubrir o encontrar. Por heurística entenderemos una estrategia, método o criterio utilizado para hacer más sencilla la resolución de un problema difícil. De esta manera, una meta-heurística (del griego *meta* o más allá) es un método heurístico para resolver problemas computacionales generales combinando procedimientos de caja negra, usualmente heurísticos, con el objetivo de obtener un procedimiento más robusto y eficiente. Una meta-heurística se aplica generalmente a problemas clasificados, al menos, como *NP-Hard* por la teoría de la complejidad computacional.

Los algoritmos evolutivos (*Evolutionary Algorithms*, **EAs**) son meta-heurísticas que tratan de imitar los principios de la naturaleza establecidos en la teoría de la evolución biológica, para realizar búsquedas y procedimientos de optimización (Deb, 2000). El propósito de los EAs consiste en guiar una búsqueda estocástica haciendo progresar a un conjunto de estructuras y seleccionando de modo iterativo las más adecuadas. En este capítulo se tratarán los conceptos fundamentales asociados a los EAs y se estudiarán las alternativas para abordar problemas multi-objetivo desde este paradigma.

4.1 ANÁLISIS DE LOS ALGORITMOS DE OPTIMIZACIÓN CLÁSICOS

De aquí en más a aquellos algoritmos que usan una solución única actualizada en cada iteración y que utilizan una regla de transición determinista se los denominará *métodos clásicos*, los mismos fueron introducidos en el Capítulo 3 .

La mayoría de los algoritmos punto-a-punto clásicos utilizan procedimientos deterministas para aproximarse a la solución óptima. Tales algoritmos comienzan con una solución inicial y luego, a partir de una regla de transición prefijada, el algoritmo sugiere una dirección de búsqueda, teniendo en cuenta la información local con la que cuenta. De esta manera, una búsqueda unidireccional es realizada para encontrar la mejor solución. Esta solución se vuelve la nueva solución y el procedimiento descrito se vuelve a repetir un número de veces. Entre los problemas asociados a los algoritmos de optimización clásicos podemos nombrar:

- La convergencia a una solución óptima depende de la elección de la solución inicial.
- La mayoría de los algoritmos tiende a converger a una solución sub-óptima.
- Un algoritmo clásico eficiente en resolver un problema de optimización puede ser inútil para resolver otros problemas de optimización (mono-propósito).
- Estos algoritmos no son eficientes en manejar problemas que carezcan de un espacio de búsqueda discreto.

Como podemos apreciar en los puntos anteriores, el problema fundamental asociado a los algoritmos de optimización clásicos está ligado a la eficiencia. Por este motivo, es necesario dar una definición dentro de esta tesis de lo que significa el término eficiencia. En este contexto, definimos eficiencia⁴ como la capacidad de producir el máximo resultado (óptimo) con los mínimos recursos (memoria, disco rígido, etc.) y el mínimo tiempo de cómputo. Teniendo en cuenta la similitud de las definiciones de eficiencia, eficacia y

⁴ Del latín *efficiencia*: acción, fuerza, virtud de producir.

efectividad debemos puntualizar las diferencias de estas dos últimas para lograr entender las desventajas de los métodos clásicos. Definiremos aquí a eficacia⁵ como la capacidad de alcanzar las mínimas metas o resultados propuestos. Es decir, llegar a un resultado no necesariamente óptimo. Por último, diremos que efectividad⁶ es la capacidad de responder a las exigencias planteadas.

Cabe destacar que la mayoría de los métodos clásicos son efectivos y eficaces, pero carecen de eficiencia en problemas de tamaño real. Todo algoritmo de optimización clásico está diseñado para resolver un tipo de problema específico (Bazaraa *et al.*, 2006). Dado que un algoritmo puede ser muy bueno para resolver un problema y puede no ser aplicable a otros, es necesario que el usuario conozca una gran cantidad de algoritmos para resolver casos diferentes de optimización. En general, la toma de decisiones sobre la selección del método más adecuado se realiza por analogías entre los problemas típicos asociados a un algoritmo dado y el problema a resolver.

Dada la no linealidad y las interacciones complejas entre las variables de los casos de interés que existen en el mundo real, el espacio de búsqueda usualmente contiene más de una solución óptima, las cuales son en su mayoría, soluciones óptimas locales que poseen un valor en su función objetivo inferior. Mientras se resuelven este tipo de problemas, un método clásico puro es atraído a cualquiera de estas soluciones óptimas locales y no existe forma de escapar de las mismas.

En la mayoría de los problemas prácticos de optimización, las variables de decisión están restringidas a tomar valores solamente discretos. Una práctica común es considerar a dichas variables continuas durante el proceso de optimización. Esto, sin embargo, puede

⁵ Del latín *efficax*, que tiene el poder de producir el efecto deseado.

⁶ Del latín *efficere*, ejecutar, llevar a cabo, obtener como resultado.

provocar que se consuma una gran cantidad de tiempo computando soluciones no factibles. Además, el cálculo post-optimización para n variables discretas involucra evaluar 2^n soluciones adicionales. Por ejemplo, el método *simplex* tiene una complejidad computacional exponencial en términos del tamaño del problema que está resolviendo. Klee y Minty (1972) dieron un ejemplo para el cual el método *simplex* necesita 2^{n-1} iteraciones para encontrar la respuesta, siendo n el número de variables y de restricciones.

4.2 ALGORITMOS EVOLUTIVOS

Desde hace décadas los algoritmos evolutivos (EAs por sus siglas en inglés) vienen siendo extensamente utilizados como herramientas de búsqueda y optimización en una amplia variedad de dominios de problemas, incluyendo ciencia, comercio e ingeniería. La principal razón de su éxito es su poca especificidad, facilidad de uso y perspectiva global (Goldberg, 1989).

Como el nombre lo sugiere, los EAs toman sus características de la naturaleza, son procedimientos de búsqueda y optimización que están motivados por los principios de genética de Mendel⁷ y selección natural de Darwin⁸. Algunas ideas fundamentales de genética son tomadas y utilizadas artificialmente para construir algoritmos de búsqueda que sean robustos y requieran una mínima información del problema.

⁷ Juan Gregorio Mendel: Científico y botánico, 1822 – 1884, Heizendorf, Austria. Autor de las siguientes leyes sobre herencia: Ley de la uniformidad de los híbridos de la primera generación, Ley de la separación de los alelos y Ley de la herencia independiente de características.

⁸ Charles Darwin (1859) *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, John Murray, London; reimpresión moderna Charles Darwin, Julian Huxley (2003). *The Origin of Species*. Signet Classics. ISBN 0-451-52906-5. Publicado en línea en *The complete work of Charles Darwin: On the Origin of Species by Means of Natural Selection, or The Preservation of Favoured Races in the Struggle for Life*. Charles Darwin (1809-1882) expuso en este trabajo su idea de la selección natural como base para la posteriormente denominada teoría de la evolución: por un lado, los organismos se modifican de generación en generación para formar, en una larga escala de tiempo, nuevas especies, pero, por otro, esa misma selección natural garantiza la supervivencia y muerte de los individuos, siendo responsable de la desaparición de las especies y de formas enteras de vida.

El mecanismo de los EAs difiere enormemente de la mayoría de las técnicas clásicas de optimización. De manera general, podemos describir la idea básica de un EA en términos de su contraparte en la naturaleza.

Una estructura o individuo es una solución codificada para algún problema. Típicamente, un *individuo* es representado como un *string* (cadena o cadena de cadenas) correspondiente a un *genotipo* biológico. Este genotipo define un organismo individual cuando es expresado (decodificado) en un *fenotipo*. Un genotipo está compuesto de uno o más *cromosomas*, donde cada cromosoma está compuesto de *genes* separados que toman ciertos valores (*alelos*) de un alfabeto genético. La posición (*locus*) identifica el lugar de un gen dentro del cromosoma. Un conjunto de individuos constituye una *población*. Como en la naturaleza, una población *evoluciona* gracias a operadores evolutivos que tienden a generar soluciones con una mejor función de aptitud (*fitness*). Entre los operadores más conocidos se encuentran la *mutación*, la *recombinación* y la *selección*. De una población, los individuos más aptos son *seleccionados* (**padres**) para brindar su material genético (*recombinación*) a fin de obtener **hijos** que a veces *mutan*.

La **función objetivo** define la condición de optimalidad para los EAs y es un rasgo del dominio del problema, mientras que la función de *fitness* mide en el dominio del algoritmo con qué grado una solución satisface dicha condición y le asigna un valor de peso a la solución.

Para definir formalmente a un algoritmo evolutivo, el algoritmo general debe ser descrito en términos matemáticos, permitiendo una especificación exacta de varias instancias de los EAs. En este contexto, cada EA tiene asociado un conjunto I llamado espacio de soluciones o individuos del EA. Cada individuo que pertenece a I normalmente representa una solución candidata del problema que se desea resolver. Cuando se definen las

transformaciones generacionales de la población, Bäck (1996) denomina a la colección resultante de u individuos como I^u , y a las transformaciones como la relación $T : I^u \rightarrow I^u$, donde $u \in \mathbb{N}$ (Bäck, 1996). Sin embargo, algunas variantes de los EAs obtienen poblaciones cuyo tamaño no es igual que el de sus predecesores. En este contexto, la relación de la transformación de la población se denota $T : I^u \rightarrow I^{u'}$, indicando que las poblaciones sucesivas pueden no contener la misma cantidad de individuos. Teniendo en cuenta lo expuesto se puede definir formalmente un EA según Bäck, 1996 y, Merkle y Glamont, 1997 de la siguiente manera:

Definición 4.1: (Algoritmo Evolutivo) Tomemos I , un conjunto no vacío (el espacio de individuos), $\{u^{(i)}\}_{i \in \mathbb{N}}$ una secuencia en Z^+ (los padres de la población), $\{u'^{(i)}\}_{i \in \mathbb{N}}$ una secuencia de Z^+ (el conjunto de descendientes), $\Phi : I \rightarrow \mathbb{R}$ la función de aptitud o fitness, $\bigcup_{i=1}^{\infty} (I^u)^{(i)} \rightarrow \{true, false\}$ (el criterio de terminación), $\chi \in \{true, false\}$, r una secuencia $\{r^{(i)}\}$ de operadores de recombinación $r^{(i)} : X_r^{(i)} \rightarrow T\left(\Omega_r^{(i)}, T\left(I^{u^{(i)}}, I^{u'^{(i)}}\right)\right)$, m una secuencia $\{m^{(i)}\}$ de operadores de mutación $m^{(i)} : X_m^{(i)} \rightarrow T\left(\Omega_m^{(i)}, T\left(I^{u^{(i)}}, I^{u'^{(i)}}\right)\right)$, s una secuencia $\{s^{(i)}\}$ de operadores de selección $s^{(i)} : X_s^{(i)} \times T(I, \mathbb{R}) \rightarrow T\left(\Omega_s^{(i)}, T\left(I^{u^{(i)} + xu^{(i)}}, I^{u^{(i+1)}}\right)\right)$, $\Theta_r^{(i)} \in X_r^{(i)}$ (los parámetros de recombinación), $\Theta_m^{(i)} \in X_m^{(i)}$ (los parámetros de mutación) y $\theta_s^{(i)} \in X_s^{(i)}$ (los parámetros de selección).

Entonces el algoritmo mostrado en el Algoritmo 4.1 es llamado Algoritmo Evolutivo.

Algorithm Algoritmo Evolutivo

0. $t = 0$;

1. inicializar $P(0) = \{a_1(0), a_2(0), \dots, a_u(0)\} \in I^{u(0)}$;

2. **while** $(\cup(\{P(0), \dots, P(t)\}) \neq \text{true})$ **do**

3. recombinar: $P'(t) = r_{\Theta_r^{(t)}}^{(t)}(P(t))$;

4. mutar: $P''(t) = m_{\Theta_m^{(t)}}^{(t)}(P'(t))$;

5. seleccionar:

6. **if** χ

7. **then** $P(t+1) = s_{(\Theta_s^{(t)}, \phi)}^{(t)}(P''(t))$;

8. **else** $P(t+1) = s_{(\Theta_s^{(t)}, \phi)}^{(t)}(P''(t) \cup P(t))$;

9. **end if**

10. $t = t + 1$;

11. **end while**

12. **end Algorithm**

Algoritmo 4.1 Esquema de un Algoritmo Evolutivo

Y en la Figura 4.1 se puede observar gráficamente la idea general del algoritmo evolutivo.



Figura 4.1 Pasos esenciales de un EA

Es importante en este punto remarcar que con el nombre EAs estamos enmarcando una variedad de algoritmos que difieren unos con otros en ciertos aspectos de su implementación. En la Tabla 4.1 podemos observar las diferencias más destacadas entre las tres formas más acostumbradas de ejemplificar un EA, teniendo en cuenta que u es el número de individuos padre y λ el número de descendientes.

Tabla 4.1 Diferencias clave entre los diferentes algoritmos evolutivos.

Tipo de EA	Representación	Operadores Evolutivos
Programación Evolutiva (EP)	Valores reales	Mutación y selección ($u+\lambda$)
Estrategias Evolutivas (ESs)	Valores reales Parámetros	Mutación, recombinación y selección ($u+\lambda$) o selección (u,λ)
Algoritmos Genéticos (GAs)	Valores reales Es históricamente Binaria	Mutación, recombinación y selección

Dentro de las características comunes que deben cumplir todos los EAs podemos decir que son métodos de búsqueda con las siguientes características:

- Ciega, es decir, no disponen de ningún conocimiento específico del problema, de manera que la búsqueda se basa únicamente en los valores de la función objetivo.
- Codificada, puesto que no trabajan directamente sobre el dominio del problema sino sobre representaciones de sus elementos.
 - Múltiple, esto es, buscan simultáneamente en un conjunto de candidatos.
 - Estocástica, tanto en la selección como en la transformación. Esto proporciona el factor de penetración de la búsqueda.
- De paralelismo implícito, tiene la capacidad de un intercambio estructurado de información en paralelo entre las soluciones posibles.

Como se dijo anteriormente, una posible solución a un problema en un EA es representada a través de un individuo de la población. Por esta razón, cada solución es inherente al problema bajo estudio. Por ejemplo, para un problema de ingeniería una solución puede representar la programación de las tareas a realizar en una fábrica y en problemas de sistemas de control una solución puede ser la variación funcional de los parámetros de control.

4.3 REPRESENTACIÓN DEL INDIVIDUO

Dado que la estructura de una solución varía de un caso a otro, una solución a un problema en particular puede ser representada de varias maneras. En los métodos clásicos de búsqueda y optimización todas las variables de decisión usualmente se representan a través de vectores de números reales y el algoritmo trabaja sobre dicho vector para crear un nuevo vector solución (Deb, 1995; Reklaitis *et al.*, 1983). Los EAs, en especial los GAs, en general utilizan la representación de cadena binaria para sus variables de decisión. Las EEs y la EP utilizan una combinación de variables de decisión con valores reales y un conjunto de parámetros estratégicos como vector de solución.

4.3.1 Representación binaria

La representación clásica, que se usa en el llamado algoritmo genético canónico o simple, consiste en un vector binario (a veces llamado cadena binaria o cadena de bits) de un tamaño fijo l . De esta forma, un espacio de individuos I está dado por $I = \{0,1\}^l$ y los individuos $a \in I$ son denotados como vectores binarios $a = a_1, \dots, a_l \in \{0,1\}^l$ (Goldberg, 1989).

Aunque la codificación binaria de los individuos es la más utilizada en la literatura, en algunos casos dicha codificación no alcanza para representar ciertos aspectos de la solución. Knuth (1973) define una *permutación* de un conjunto finito de elementos como “un arreglo de elementos en una fila”. Teniendo n objetos, existen $n!$ permutaciones de los objetos. Existen varias propiedades de las permutaciones que son relevantes para la manipulación de las representaciones de los individuos en los EAs, tanto desde el punto de vista de la representación en sí como desde una perspectiva analítica. Los problemas donde la solución es una permutación no pueden ser procesados utilizando los mismos operadores de recombinación y mutación que aquellos utilizados por las cadenas de bits.

En los problemas cuyas soluciones se pueden representar mediante permutaciones o combinaciones, pueden existir soluciones equivalentes múltiples (Michalewicz, 1994). Cuando una permutación representa un ciclo, como en el caso del Problema del Viajante⁹ (*Travelling Salesman Problem*, **TSP**) todas las permutaciones son soluciones equivalentes, más aún todas las reversas de las soluciones son soluciones equivalentes al problema. Por lo tanto, esta simetría puede afectar al proceso de recombinación en un EA. Cabe destacar que el problema del viajante es un problema *NP-Hard* (Applegate *et al.*, 2006), y por lo tanto no existen métodos heurísticos eficientes para su resolución. Cabe destacar que también se podrían haber utilizado otras representaciones como el *Vehicle Routing Problem* (VRP) o problema de ruteo de vehículos (Dantzig y Ramser, 1959) de manera satisfactoria. En especial para nuestro enfoque se utilizó el TSP.

Los problemas de permutaciones o combinatorios también abarcan aplicaciones de programación o *scheduling* como el *Job Shop Scheduling Problem* (**JSSP**) (French, 1992). Otros tipos de problemas de programación planifican tiempo de servicio para un conjunto de clientes. Esto envuelve el acceso a recursos críticos, y los clientes deben ser planificados para ser atendidos lo más pronto posible de manera tal que los recursos pueden no estar disponibles para los clientes más tardíos. En estos problemas se trata de optimizar la relación cliente-recurso. El término programación de recursos (*resource scheduling*) es utilizado aquí para referirse a programación de aplicaciones cuyos recursos son consumidos.

Entre los problemas *NP-Hard* de esta categoría se encuentra también el problema de planificación óptima de redes de tránsito (TNDSP), el cual abordamos en el Capítulo 1 y su tratamiento es lo que ha motivado el desarrollo de esta tesis doctoral.

⁹ *Travelling Salesman Problem*: Un viajante debe recorrer un número determinado de ciudades pasando por todas y cada una de ellas una sola vez, encontrando el recorrido óptimo.

Para explicar las características restantes de los AGs, nos enfocaremos en la representación de recorridos o permutaciones mediante cadenas de enteros (definidas originalmente para el TSP) dado que tiene una gran similitud con nuestro enfoque utilizado en el EA híbrido para el TNDSP desarrollado en esta tesis. Luego, en la segunda parte de esta tesis, introduciremos una representación de los individuos original a fin de resolver el problema particular de optimización bajo estudio.

4.3.2 Representación de recorridos o permutaciones

Los problemas de optimización combinatoria o permutaciones son muy variados. No obstante, el problema del viajante de comercio (TSP) es uno de los más tratados en la literatura, no sólo por ser el arquetipo de problema basado en el orden, sino también porque, al ser un problema NP, su resolución sirve como base para el tratamiento de otros problemas combinatorios de permutaciones. Formalmente, el problema del viajante se puede definir como:

Definición 4.2: Dadas m ciudades (o vértices) $\{1, \dots, m\}$ y los costos de viajar de unas a otras c_{ij} ($i, j \in \{1, \dots, m\}$) se desea calcular una ruta (conjunto de aristas o arcos) completa (i.e., que visite todas las ciudades), cerrada (i.e., que comience y termine en la misma ciudad) y conexa que haga mínimo al costo total del recorrido (ver Figura 4.2).

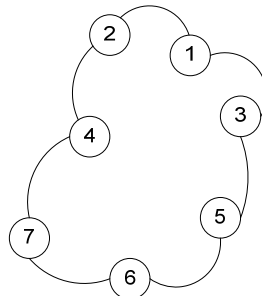


Figura 4.2 Una permutación (solución) para el problema del viajante de comercio (TSP).

El espacio de búsqueda para este problema es el de las permutaciones de las m ciudades o vértices. Cualquier permutación representa un camino completo, cerrado y conexo

(de aquí en adelante, lo abreviaremos como recorrido). Por lo tanto, representa un candidato de la solución.

No existe un modo práctico de codificar el conjunto de las permutaciones de m elementos en cadenas binarias de bits, de modo tal que los operadores evolutivos clásicos sean eficientes en la resolución del TSP (Michalewicz, 1994). Por esta razón, la representación mediante cadenas de enteros combina sencillez y eficiencia.

Luego, sea I el espacio de soluciones, tomemos a_1, \dots, a_m una permutación del conjunto $\{1, \dots, m\}$. Entonces, es posible representar este elemento a través de dos líneas de la siguiente forma:

<i>orden</i>	1	2	3	...	m
<i>permutación</i>	a_1	a_2	a_3	...	a_m

Figura 4.3 Recorrido y orden de una solución del TSP.

De esta manera es posible representar el orden relativo de las ciudades que visita el viajante en el TSP.

4.4 OPERADORES EVOLUTIVOS

En la literatura existen operadores de recombinación (cruzamiento) y mutación para cadenas de enteros. En las próximas secciones se describen dichos operadores en su enfoque tradicional.

4.4.1 Selección

Cuando hablamos de la estructura general de los EAs, especialmente GAs y programación genética, dijimos en la introducción que aplicaban un operador de selección primero para escoger los individuos más aptos y someterlos a las operaciones de recombinación y mutación, con el objeto de obtener individuos mejores. Por lo tanto, la selección no genera ningún individuo nuevo.

Entre los mecanismos más comunes de selección encontramos tres clases:

- **Directos:** estos métodos de selección toman un subconjunto de individuos de la población siguiendo un criterio fijo, del estilo de “los k mejores”, “los k peores”, “a dedo”, etc.
- **Aleatorios simples:** a todos los individuos se les asignan las mismas probabilidades de selección de formar parte del conjunto de padres. Estos padres servirán para crear la nueva población.
- **Estocásticos:** se asignan probabilidades de selección a los elementos de la población en función de su aptitud. Por defecto, la puntuación p_i asociada a un individuo a de la población $P(t) = \{a_1, \dots, a_m\}$ se calcula como la aptitud relativa de dicho individuo. Teniendo

en cuenta las aptitudes de los individuos $f(a_1), \dots, f(a_m)$, se calcula $p_i = \frac{f(a_i)}{f(a_1) + \dots + f(a_m)}$.

De esta manera, se garantiza $p_1 + \dots + p_m = 1$, tal como corresponde a una distribución de probabilidades. En una generación, la población es reemplazada. Los u padres que aportarán su material genético son seleccionados tirando la bolilla u veces. Entre los métodos estocásticos existentes, el más sencillo de aplicar es el llamado por ruleta sesgada o *roulette wheel*¹⁰. No nos detendremos a explicar más sobre el proceso de selección, puesto que se profundizará cuando tratemos a los algoritmos evolutivos multi-objetivo. Para más información, el lector puede consultar a Bäck *et al.*, 2000.

Una vez que se seleccionaron los u padres de la población actual, se está en condiciones de pasar al siguiente operador evolutivo: la recombinación o cruzamiento.

4.4.2 Operador de recombinación o cruzamiento

Anteriormente dijimos que la manera más natural de codificar soluciones del TSP era enumerando cada una de las ciudades por orden de recorrido. Davis (1985) y Goldberg y

¹⁰ Ruleta sesgada: en este procedimiento la distribución de probabilidades está definida como una ruleta sesgada donde cada porción de la ruleta tiene un tamaño proporcional a la probabilidad de selección del individuo. Un número aleatorio es generado como si fuera una bolilla que se tira a la ruleta en movimiento, se elige el individuo al que corresponde el lugar en la ruleta donde cae la bolilla. Este método posee una alta convergencia a individuos cuya probabilidad de selección es muy alta.

Lingle (1985) definieron algunos de los primeros operadores para problemas de recorrido o permutación. Los operadores de cruzamiento o recombinación más conocidos son:

Cruzamiento por orden (OX). El cruce por orden fue introducido por Davis (1985). Se toman dos recorridos para recombinación o cruzamiento:

<i>orden</i>	1	2	3	4	5	6	7	8	9
p_1	1	2	3	4	5	6	7	8	9
p_2	5	2	9	6	4	1	7	3	8

Figura 4.4 Dos individuos para el TSP.

La primera fila de cada uno de los individuos p_1 y p_2 corresponde al orden del recorrido y la de abajo a las ciudades visitadas. Llamaremos al primer padre p_1 como la cadena de corte y al otro p_2 , como el donador. Se seleccionan dos puntos de corte y se copia la sub-lista de los elementos entre los puntos de corte directamente a la descendencia en la misma posición absoluta. Luego, comenzando en el segundo punto de cruce, se encuentra el siguiente elemento en el padre donador que no aparece en la descendencia. Comenzando desde el segundo punto de cruce, se inserta el elemento del donador en la siguiente posición vacía de la descendencia. Se continúa moviendo posiciones hasta que se llega al final del donador o la descendencia y se regresa al comienzo del individuo. En el ejemplo:

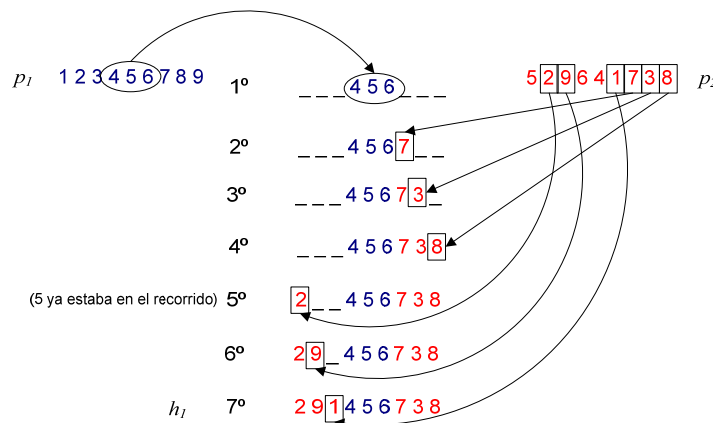


Figura 4.5 Ejemplo de cruzamiento OX entre los padres p_1 y p_2 .

Dado la complicada topología del problema de tránsito, en esta tesis se diseñó, una variante del clásico cruce por orden (OX), la cual será explicada en el Capítulo 7 para el caso del EA híbrido desarrollado.

Cruzamiento por emparejamiento parcial (PMX). Goldberg y Lingle (1985) introdujeron el operador de cruzamiento emparejamiento parcial. Al igual que en el cruzamiento por orden, uno de los progenitores es designado como primer padre p_1 y el otro como donador p_2 . Dos puntos de corte son seleccionados y todos los elementos en el padre p_1 entre los sitios de corte son directamente copiados a la descendencia h_1 , pero la forma en la que los elementos de p_2 son trasladados a la descendencia, difiere del cruzamiento por orden. Lo que se intenta es tomar un tramo de uno de los progenitores y se cruza preservando el orden y la posición de la mayor cantidad de ciudades del otro. Veámoslo para mayor facilidad en el ejemplo:

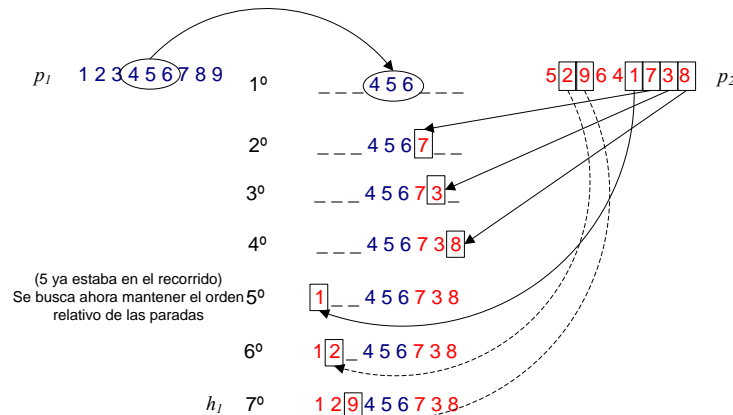


Figura 4.6 Ejemplo de cruzamiento PMX entre los padres p_1 y p_2 .

Los genes de h_1 en rojo corresponden al material genético del padre p_2 . Se puede ver como se trata de que se preserve el orden del recorrido del padre p_2 .

Existen otras formas de cruzamiento pero no nos extenderemos a tratarlas, el lector puede consultarlas en Syswerda (1991) y Davis (1991).

4.4.3 Operador de mutación

Los operadores de mutación pueden ser usados de diversas maneras. Algunos algoritmos evolutivos no aplican recombinación y sólo hacen uso del operador de mutación, mientras que otros combinan el uso de ambos: cruzamiento y mutación. Cuando se aplica una mutación a una permutación o recorrido, se debe obtener un individuo factible del problema que se está resolviendo. El objetivo de la mutación es explorar las vecindades locales de los individuos en busca de mejores soluciones.

La *mutación k-opt.* es la forma más común de mutación para problemas con representaciones del tipo TSP, donde $k \in \mathbb{Z}$ y $k \geq 2$. El operador **k-opt** selecciona k puntos del recorrido e invierte el segmento entre los puntos. Este operador puede aplicarse a todos los pares de nodos $N(N-1)/2$ veces.

Supongamos para nuestro ejemplo que $k = 2$. Dado un recorrido,

<i>orden</i>	1	2	3	4	5	6	7	8	9
a_1	1	2	9	4	5	6	7	3	8

Figura 4.7 un recorrido a_1 .

Dichos puntos de corte son **2** y **6** de la Figura 4.7, el resultado de la mutación se puede observar en la Figura 4.8.

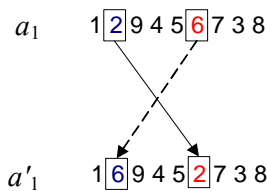


Figura 4.8 mutación de a_1 .

Ahora que hemos establecido la teoría base de nuestro estudio estamos en condiciones de introducir a los algoritmos evolutivos multi-objetivo como alternativa para resolver el tipo problemas de optimización multi-objetivo que fueron descritos en el Capítulo 1 de esta tesis.

Resumen del capítulo

En este capítulo se analizaron los principales aspectos de los algoritmos evolutivos mono-objetivo. Se mostró la limitación de las cadenas binarias para representar permutaciones y se ilustró el uso de cadenas de enteros para resolver el problema del viajante (TSP) y sus afines. Se trataron los operadores evolutivos selección, recombinación y mutación para individuos representados mediante cadenas de enteros.

Capítulo 5 Algoritmos evolutivos para problemas multi-objetivo

En los últimos años se ha generalizado el uso de algoritmos evolutivos para la resolución de problemas de optimización multi-objetivo teniendo en cuenta las falencias de los métodos clásicos para obtener múltiples soluciones alternativas. Los métodos clásicos por lo general resuelven el problema reduciendo el vector de objetivos a uno sólo con la ayuda de algún conocimiento específico del problema modelado. Esto tiene aparejado como consecuencia de que aunque se encuentren soluciones no dominadas, se obtiene únicamente una solución al problema. La persona que toma las decisiones (en inglés *decision maker*, lo cual de ahora en más notaremos DM) debe poseer un alto conocimiento del problema que está modelando antes de desarrollar esa función objetivo.

La potencialidad de los algoritmos evolutivos para el tratamiento de problemas multi-objetivo fue introducida por Rosenberg (1987). El primer algoritmo evolutivo multi-objetivo es acreditado a David Schaffer (1984), que propuso el algoritmo genético con vector de evaluación (*Vector Evaluated Genetic Algorithm*, VEGA). La característica principal de un algoritmo evolutivo multi-objetivo (*Multi-Objective Evolutionary Algorithm*, MOEA) es que el conjunto de objetivos múltiples son optimizados en forma simultánea. Es decir,

Definición 5.1: Sea I un conjunto de individuos. Tomemos $\Phi : I \rightarrow \mathbb{R}^k$, ($k \geq 2$, una función de aptitud o fitness multi-objetivo). Si esta función multi-objetivo es sustituida en la

Definición 4.1 *entonces el procedimiento del Algoritmo 4.1 es llamado Algoritmo Evolutivo Multi-Objetivo.*

De la misma manera que los métodos clásicos se clasificaron en el Capítulo 3 como a priori y a posteriori, los algoritmos evolutivos pueden clasificarse de la misma manera. Un algoritmo evolutivo a priori define con anterioridad la importancia relativa de cada uno de los objetivos, mientras que un algoritmo evolutivo a posteriori realiza la búsqueda con posterioridad al cálculo de las soluciones.

5.1 ARTICULACIÓN A PRIORI

En estas técnicas se deben tomar decisiones de preferencia antes de ejecutar el algoritmo. Estas decisiones involucran por lo general pesos, prioridades o preferencias asignadas a los objetivos, de tal manera que el problema multi-objetivo se vuelva mono-objetivo. Entre las técnicas más conocidas y sencillas están las de la suma ponderada y de la multiplicación, que se detallan en las dos siguientes sub-secciones.

Otros métodos muy renombrados son el “orden lexicográfico” (Fourman, 1985), la “programación por metas” (Charnes y Cooper, 1961; Ijiri, 1965) y la “aproximación min-max con pesos” (Jutler, 1965; Solich 1969) que fue ampliada posteriormente (Osyczka, 1978; Osyczka, 1981). No haremos hincapié en estas técnicas por razones de espacio, puesto que nos hemos concentrado en aquellas que fueron aplicadas para el algoritmo evolutivo híbrido desarrollado.

5.1.1 Método de la suma ponderada

Este método es la primera técnica desarrollada para abordar los problemas multi-objetivo. Consiste en sumar todas las funciones objetivo, utilizando un coeficiente de peso para cada una. El problema pasa de ser multi-objetivo a mono-objetivo de la forma:

$$\text{Minimizar } \sum_{i=1}^n w_i f_i(x) \quad (5.1)$$

donde $w_i \geq 0, i = 1, \dots, n$ son los coeficientes de peso que representan la importancia relativa de cada uno de los objetivos. Normalmente, la suma de los w_i da como resultado un 1.

Los principales problemas de esta técnica radican en la dificultad de determinar los pesos apropiados cuando no hay suficiente información del problema y en su incapacidad de generar soluciones Pareto óptimas para espacios de búsqueda no convexos.

5.2 ARTICULACIÓN A POSTERIORI

En estos casos todos los objetivos se tienen en cuenta. Una vez ejecutado el algoritmo, el resultado es un conjunto de soluciones no dominadas, entre las cuales el DM elegirá aquella solución que le sea de utilidad. La principal ventaja es que no se debe tener un gran conocimiento del problema para aplicar las técnicas, puesto que el análisis de las soluciones se puede realizar totalmente por separado.

En este punto se pueden dividir los algoritmos evolutivos en dos grandes grupos: los basados en el concepto de Pareto y los que no se basan en el mismo. A continuación, explicaremos los métodos más destacados y además aquellos que se utilizaron para el algoritmo evolutivo híbrido desarrollado en el marco de esta tesis.

5.2.1 Métodos no basados en el concepto de Pareto

Entre ellos se destacan, además de los que serán brevemente introducidos a continuación, los métodos denominados: “uso de teoría de juegos”, “uso del género para identificar objetivos”, “estrategia de evolución presa-depredador”, “algoritmo genético con participación distribuida”, “método de aprendizaje por refuerzo distribuido” y “sistema neuro-evolutivo multi-objetivo (MONESSY)”. El lector puede consultar en Deb (2001) o Coello Coello y Veldhuizen (2006) la descripción de todos estos métodos.

5.2.1.1 Método de las restricciones ε

Este método fue introducido por Ritzel *et al.* (1994) como un método para resolver problemas de optimización utilizando GAs. Se basa en la minimización de una sola de las funciones objetivo y se consideran las otras como restricciones para algún nivel permitido ε_i . Variando los niveles de ε_i se puede obtener todo el conjunto de Pareto óptimo.

5.2.1.2 Algoritmo genético de vector evaluado (VEGA)

El método de Schaffer (1984) consiste en utilizar un algoritmo genético simple con una variación en la forma de realizar la selección: para un problema de n objetivos, se crean n sub-poblaciones de tamaño N/n (suponiendo que N es el tamaño de la población total). En cada una de las n sub-poblaciones se realiza la selección de individuos de forma independiente, teniendo en cuenta un objetivo a la vez para obtener una nueva población de tamaño N , en la cual se aplican en la forma usual los operadores genéticos de cruce y mutación.

5.2.1.3 Algoritmo genético basado en pesos (WBGA)

El algoritmo genético basado en pesos fue propuesto por Hajela y Lin (1992). En este procedimiento se utilizan los pesos de cada objetivo dentro del cromosoma. A cada solución x se le asocia un vector de pesos diferente $w = (w_1, \dots, w_n)$ tal que $\sum_{i=1}^n w_i = 1$. La función de *fitness* es de la forma:

$$F(x) = \sum_{i=1}^n w_i f'_i(x) \quad (5.2)$$

donde w_i es el peso para el objetivo i -ésimo asociado a la solución x y $f'_i(x)$ es el valor normalizado de la función objetivo i -ésima para la solución x que se obtiene según la (5.3).

$$f'_i(x) = \frac{f_i(x) - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \quad (5.3)$$

siendo f_i^{\min} y f_i^{\max} los valores mínimo y máximo respectivamente para la función i -ésima.

Dado que es imprescindible mantener diversidad de pesos dentro de la población, Hajela y Lin utilizan una función de participación o *sharing*:

$$sharing(d_{jk}) = \begin{cases} 1 - \left(\frac{d_{jk}}{\sigma_{sharing}} \right)^\alpha, & \text{si } d_{jk} < \sigma_{sharing} \\ 0 & \text{, en otro caso} \end{cases} \quad (5.4)$$

con $\alpha = 1$ y $\sigma_{sharing}$ como el factor de participación que varía entre 0,01 y 0,1. El valor de d_{jk} hace referencia a la distancia entre los pesos de la solución j -ésima x_j y la solución k -ésima x_k .

$$d_{jk} = |w^j - w^k| \quad (5.5)$$

donde w^j y w^k son vectores de pesos asociados a las soluciones x_j y x_k . Para x_j se define el valor de nicho o *niche* como:

$$niche(x_j) = \sum_{k=1}^N d_{jk} \quad (5.6)$$

Aunque es sencillo de implementar, este método tiene como dificultad la de determinar el factor de participación o *sharing*. El *fitness* de una solución x queda, por tanto, de la siguiente manera:

$$F(x) = \frac{F(x)}{niche(x)} \quad (5.7)$$

5.2.2 Métodos no elitistas basados en el concepto de Pareto

Si un MOEA utiliza un operador que favorece a los “mejores” de una población permitiéndoles sobrevivir a la siguiente generación, se dice que estamos ante un algoritmo *elitista*. En caso contrario, el MOEA es *no elitista*. En esta sección nos concentraremos en los MOEAs no elitistas más populares en la literatura, mientras que en la Sección 5.2.3 se abordarán los MOEAs elitistas.

En las próximas sub-secciones se describirán los métodos que de acuerdo a nuestro criterio constituyen los más utilizados en la literatura. Existen además otras técnicas, que no se describen aquí por razones de espacio, entre las que se destacan las denominadas: “estrategia de evolución con archivo Pareto (PAES)”, “algoritmo genético con selección

basada en particiones Pareto (PESA)”, “algoritmo genético desordenado multi-objetivo (MOMGA)”, “micro-algoritmo genético multi-objetivo (Micro-GA)”, “algoritmo evolutivo multi-objetivo elitista con participación co-evolutiva (ERMOCS)”. Para consultar la descripción de estos métodos, el lector puede recurrir a Deb (2001) o Coello Coello y Veldhuizen (2006).

5.2.2.1 Algoritmo de ordenamiento de Pareto

En 1989, Goldberg introduce la asignación de *fitness* basada en la no dominación para resolver los problemas teniendo en cuenta el algoritmo VEGA de Schaffer. La población es ordenada en base a la no dominación a través de la obtención de *frentes de Pareto óptimos*. Se identifican y marcan todos los individuos no dominados, asignando 1 a la primera posición. Estos son ignorados y se identifica el siguiente conjunto de no dominados asignándoles la posición 2 y así sucesivamente hasta agotar la población. Golberg además sugiere el uso de alguna técnica de formación de especies o nichos para mantener la diversidad genética. Es importante puntualizar que una desventaja de este algoritmo es que no existen procedimientos eficientes para comprobar la no dominación de soluciones factibles; como consecuencia la eficiencia disminuye cuando aumenta el tamaño de la población o el número de objetivos. Sin embargo, esta técnica es útil para cubrir el frente de Pareto en una sola ejecución.

5.2.2.2 Algoritmo genético con múltiples objetivos (MOGA)

Fonseca y Fleming (1993) propusieron un algoritmo genético con múltiples objetivos en el cual el *rango* de un cierto individuo corresponde al número de individuos en la población actual que lo dominan. Un rango se puede definir como:

$$rango(x) = 1 + p(x) \quad (5.8)$$

donde $p(x)$ es el número de individuos de la población que dominan a x . Todos los individuos no dominados tienen rango 1; los dominados son penalizados de acuerdo a la densidad de la población de la región correspondiente.

El inconveniente más importante del MOGA es la alta presión selectiva que surge en el uso de esta técnica. Para evitar esto, Fonseca y Fleming introdujeron un método de formación de nichos para distribuir la población por todo el frente de Pareto óptimo. La función de *sharing* se define con la ecuación (5.4) en la cual el exponente es $\alpha = 1$. La distancia d_{jk} entre individuos se calcula como la distancia euclidiana normalizada en el espacio de las funciones objetivo:

$$d_{jk} = \sqrt{\sum_{i=1}^n \left(\frac{f_i(x_k) - f_i(x_j)}{f_i^{max} - f_i^{min}} \right)^2} \quad (5.9)$$

siendo f_i^{min} y f_i^{max} los valores mínimo y máximo respectivamente para la función objetivo i -ésima. El contador de nicho se define sólo entre individuos del igual rango:

$$niche(x_j) = \sum_{k=1}^{\mu(j)} sharing(d_{jk}) \quad (5.10)$$

donde $\mu(j)$ es el número de individuos cuyo rango es igual a x_j .

5.2.2.3 Algoritmo evolutivo con fuerzas de Pareto (SPEA)

El algoritmo evolutivo con fuerzas de Pareto (Strength Pareto Evolutionary Algorithm, SPEA) fue introducido por Zitzler y Thiele en 1999; básicamente, el método utiliza un archivo externo para contener soluciones no dominadas encontradas previamente (conjunto externo). En cada generación los individuos no dominados son copiados en el archivo. Para cada individuo en el conjunto externo se calcula un valor de fuerza (*strength*); este es similar al rango de MOGA y es proporcional al número de soluciones que dominan una solución. En SPEA, el *fitness* de cada miembro actual de la población es calculado según el *strength* de todas las soluciones no dominadas que lo dominan. La dominancia de Pareto es utilizada para asegurar que las soluciones están debidamente distribuidas en la frontera de Pareto.

Existe una versión revisada de SPEA llamada SPEA2 (Zitzler *et al.*, 2002) que posee tres diferencias fundamentales con SPEA:

1. Incorpora una asignación para la función de *fitness* que toma en cuenta para cada individuo el número de individuos que lo dominan y el número de individuos a los que domina.
2. Utiliza una estimación de la densidad de las vecindades para guiar la búsqueda más efectivamente.
3. Utiliza un método de truncado para el archivo contenedor que garantiza la preservación de las soluciones límite.

5.2.2.4 Algoritmo genético de ordenamiento no dominado (NSGA)

Fueron Srinivas y Deb (1994) quienes implementaron la idea de Goldberg con algunas variaciones en su algoritmo genético de ordenamiento no dominado (Non-Dominated Sorting Genetic Algorithm, NSGA). Este algoritmo se basa en tener múltiples niveles de clasificación de los individuos a partir de los *frentes no dominados*.

Antes de realizar la selección se ordena la población en base a la no dominación: todos los individuos no dominados se clasifican en una categoría con un valor de *fitness* ficticio proporcional al tamaño de la población. De esta manera, todos los individuos poseen la misma probabilidad de reproducirse. Con el objeto de mantener la diversidad, se calcula la función de *sharing* para estos individuos a partir del valor de *fitness* ficticio. Se pasa luego al siguiente nivel de individuos no dominados hasta que todos los individuos de la población son clasificados. Se utiliza una selección proporcional estocástica y una función de *sharing* para mantener la dispersión entre los individuos de cada frente no dominado. La distancia de *sharing* entre dos individuos se calcula como la distancia euclídeana d_{jk} normalizada en el espacio de decisión:

$$d_{jk} = \sqrt{\sum_{i=1}^n \left(\frac{x_i^k - x_i^j}{x_i^{max} - x_i^{min}} \right)^2} \quad (5.11)$$

donde n es el tamaño de la población, x_i^{max} y x_i^{min} son los valores máximos y mínimos para el parámetro de decisión i -ésimo. La función de *sharing* se calcula con $\alpha = 2$ teniendo en cuenta sólo los individuos que pertenecen al mismo frente de Pareto.

5.2.3 Métodos elitistas basados en el concepto de Pareto

Los algoritmos evolutivos multi-objetivo con elitismo realizan una articulación de las preferencias a posteriori basándose en el concepto de Pareto.

5.2.3.1 Algoritmo genético de ordenamiento no dominado elitista

Se ha propuesto (Deb *et al.*, 2000) una versión del algoritmo genético de ordenamiento no dominado con elitismo (NSGA-II). Este algoritmo utiliza una estrategia elitista junto con un mecanismo explícito de diversidad. A partir de una población P_t se crea una nueva población de descendientes Q_t . Estas dos poblaciones se mezclan para formar una nueva R_t de tamaño $2N$ (siendo N el tamaño original de P_t). Los individuos de R_t se ordenan por frentes no dominados y se obtiene una población nueva P_{t+1} aplicando a R_t una selección por torneo junto con un mecanismo de nichos, lo que mantiene la diversidad de la población final.

La selección por torneos con nichos compara dos soluciones y devuelve la ganadora, asumiendo que cada solución i tiene dos atributos: un rango de no dominación r_i dentro de la población y una distancia de nicho d_i que es una medida del espacio de búsqueda no ocupado por otra solución de la población.

Teniendo en cuenta lo dicho anteriormente, una solución i se prefiere a una j si:

- i. La solución i tiene mejor rango que j , es decir, si $r_i < r_j$.
- ii. Si las soluciones i y j tienen el mismo rango, pero la solución i tiene mejor distancia de nicho que j .

La distancia se calcula de la siguiente manera: sea Υ_i el conjunto de individuos de la población que pertenecen al mismo frente de Pareto que el individuo i . Luego para cada función objetivo j :

- i. Ordenamos los individuos de Υ_i según el valor de la función objetivo f_j en una lista l_j^i
- ii. f_j^{i+1} es el valor de la función j para el individuo de la lista l_j^i siguiente al individuo i .
- iii. f_j^{i-1} es el valor de la función j para el individuo de la lista l_j^i anterior al individuo i .

Por lo tanto, el valor de la distancia d_i se calcula como:

$$d_i = \sum_{i=1}^n \left(\frac{f_j^{i+1} - f_j^{i-1}}{f_j^{max} - f_j^{min}} \right) \quad (5.12)$$

donde f_j^{max} y f_j^{min} son los valores máximos y mínimos para la función objetivo j -ésima en toda la población.

5.2.3.2 Algoritmo genético Pareto basado en la distancia (DPGA)

Este algoritmo utiliza el teorema de contacto para determinar las distancias relativas de una solución dada al conjunto Pareto; en este algoritmo se mantiene separada una población de individuos elitistas $E = (e_1, \dots, e_K)$ y la ubicación de un individuo en la población viene determinada de forma directa por el valor de la distancia de dicho individuo al conjunto de individuos elitista. La distancia de una solución x a cada individuo $e_k \in E$ se calcula de la forma:

$$d_k(x) = \sqrt{\sum_{i=1}^n \left(\frac{f_i(e_k) - f_i(x)}{f_i(e_k)} \right)^2} \quad (5.13)$$

A partir de la ecuación (5.13) se calcula la distancia mínima d_k del individuo x :

$$d_{min}(x) = \min_{k=1}^K d_k(x)$$

siendo k^* el índice para el cual $d_{k^*}(x) = d_{min}(x)$. En el caso de que x sea una solución no dominada con respecto al conjunto E , ésta es ingresada al conjunto elitista y su *fitness* F se calcula de la siguiente manera:

$$F(x) = F(e_{k^*}) + d_{min}. \quad (5.14)$$

El conjunto E se actualiza eliminando aquellas soluciones dominadas por x . Por otro lado, si x es dominado por cualquier solución en E , el mismo no es aceptado para formar parte del conjunto E y su *fitness* se calcula como:

$$F(x) = \max[0, (F(e_{k^*}) - d_{min})]. \quad (5.15)$$

Luego el *fitness* máximo para todos los miembros de E es:

$$F_{max} = \max_{k=1}^K F(e_k). \quad (5.16)$$

Este método es relativamente simple, aunque requiere definir valores de penalización para las restricciones y otro valor llamado distancia de inicio que se utiliza para escalar las cantidades entre las diferentes soluciones.

5.3 MÉTRICAS PARA COMPARAR CONJUNTOS NO DOMINADOS

En general, es necesario poder establecer con cuál EA y con qué combinación de operadores se obtienen los mejores resultados para un problema dado. Para distinguir la mejor decisión, hay que introducir una medida para evaluar a cada uno de los algoritmos. A continuación presentaremos un conjunto de métricas asociadas a la calidad de los conjuntos no dominados obtenidos a través de los diferentes MOEAs. Para esto se deben tener en cuenta tres criterios (Deb, 2001): precisión, uniformidad y extensión.

Precisión (en inglés, **Accuracy**). La distancia del conjunto no dominado a la frontera de Pareto óptima debe ser mínima.

Uniformidad. Las soluciones deben estar bien distribuidas.

Extensión (en inglés, **Extent**). Las soluciones no dominadas deben cubrir el mayor rango posible para cada valor de la función objetivo.

Varios autores han introducido distintas métricas con el objetivo de evaluar y comparar los MOEAs. Pueden ser clasificadas en:

- i) métricas que sólo evalúan un conjunto de soluciones no dominadas,
- ii) métricas que comparan dos conjuntos de soluciones no dominadas,
- iii) métricas que requieren conocimiento del conjunto Pareto óptimo y
- iv) métricas que miden simultáneamente varias características como las anteriores.

Es importante destacar que existen métricas dedicadas a la calidad de las soluciones dentro de un MOEA y aquellas que comparan la calidad de las soluciones entre MOEAs para un mismo problema. Es importante remarcar que la palabra métrica a veces se puede confundir con medidas de calidad y tiempo. Sin embargo, en esta sección métrica sólo se referirá a la calidad de los resultados obtenidos.

Además, cabe destacar que se utilizará de manera indistinta la palabra métrica e indicador. Tanto Deb (2001) como Zitzler (2000 y 2004) recurren a las palabras métrica e indicador para referirse a los mismos conceptos. Teniendo en cuenta que se tomaron de ambos autores estas definiciones mantendremos la misma terminología, es decir que se emplearán como sinónimos.

Definición 5.2: *Se define que un indicador $I : \Omega \rightarrow \mathbb{R}$ es Pareto compatible, si y sólo si dados dos conjuntos de soluciones no dominadas A y B , $A \leq B \Rightarrow I(A) \geq I(B)$ se lee: A dominado por B , implica que el valor del indicador para el conjunto A es mayor o igual que el valor del indicador para el conjunto B .*

En este contexto, un indicador I Pareto compatible es una función que preserva el orden de (Ω, \leq) hacia (\mathbb{R}, \geq) .

A continuación, introduciremos las métricas más conocidas para evaluar MOEAs. Algunas de ellas serán utilizadas en el Capítulo 8 Sin embargo, a fin de que el indicador sea lo más certero posible, se deben tener en cuenta factores tales como escala y normalización.

5.3.1 Escala y Normalización

En principio, la noción de dominancia de Pareto es completamente independiente de la escala y la normalización. Cuando se trabaja con métricas, estos aspectos son necesarios para permitir que diferentes objetivos contribuyan igualmente a los valores del indicador, en especial para los que presentaremos en la Sección 5.3.5.2.

Un procedimiento de normalización lineal estándar para cada objetivo puede definirse como:

$$z'_i = \frac{z_i - z_i^{(min)}}{z_i^{(max)} - z_i^{(min)}} \quad (5.17)$$

donde $z_i^{(min)}$ y $z_i^{(max)}$ son conocidos o estimados para el i -ésimo objetivo.

5.3.2 Métricas que evalúan la proximidad a la frontera Pareto óptima

Estas métricas evalúan la cercanía de un conjunto Q de N soluciones desde un conjunto conocido Pareto óptimo P^* .

5.3.2.1 Razón del Error

Esta métrica, conocida en inglés como Error Ratio (ER), cuenta el número de soluciones en Q que no son miembros del conjunto Pareto óptimo P^* (Veldhuizen, 1999). Matemáticamente:

$$ER = \frac{\sum_{i=1}^{|Q|} e_i}{|Q|} \quad (5.18)$$

donde $e_i = 1$ si $i \notin P^*$; de lo contrario $e_i = 0$.

Cuando $ER = 0$ todas las soluciones son parte del conjunto P^* . En contraste, $ER = 1$ significa que ninguna solución pertenece a P^* .

5.3.2.2 Métrica de cubrimiento de conjunto

Zitzler (1999) define una métrica que también puede ser utilizada para obtener una idea relativa de la distribución de las soluciones entre dos conjuntos A y B . La métrica de cubrimiento de conjunto $C(A, B)$ calcula la proporción de soluciones B que son dominadas débilmente por las soluciones de A :

$$C(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \leq b\}|}{|B|}. \quad (5.19)$$

Si $C(A, B) = 1$ todos los miembros de B son dominados débilmente por los miembros de A . Por otro lado $C(A, B) = 0$ significa que ningún miembro de B es dominado débilmente por A . Es importante recalcar que este operador de dominancia no es simétrico y es necesario calcular $C(A, B)$ y $C(B, A)$.

5.3.2.3 Distancia generacional

Esta métrica, GD por sus siglas en inglés, encuentra la distancia promedio de las soluciones de Q a P^* :

$$GD = \frac{\left(\sum_{i=1}^{|Q|} d_i^p\right)^{1/p}}{|Q|}. \quad (5.20)$$

Para $p = 2$, el parámetro d_i es la distancia euclidiana (en el espacio de la función objetivo) entre la solución $i \in Q$ y el miembro más cercano de P^* :

$$d_i = \min_{k=1}^{|P^*|} \sqrt{\sum_{m=1}^M \left(f_m^{(i)} - f_m^{*(k)}\right)^2} \quad (5.21)$$

donde $f_m^{*(k)}$ es el m -ésimo valor de la función objetivo del miembro k -ésimo de P^* .

5.3.3 Métricas que evalúan la diversidad entre los conjuntos no dominados de soluciones

Las siguientes métricas miden la diversidad de población dentro de Q de N soluciones que se quieren analizar.

5.3.3.1 Espaciado

Schott (1995) sugiere la métrica S , la cual calcula la distancia relativa medida entre soluciones consecutivas de un conjunto de soluciones no dominadas de la siguiente manera:

$$S = \sqrt{\frac{1}{|Q|} \sum_{i=1}^{|Q|} (d_i - \bar{d})^2} \quad (5.22)$$

donde $d_i = \min_{k \in Q \wedge k \neq i} \sum_{m=1}^M |f_m^i - f_m^k|$ y \bar{d} es el valor medio entre las distancias medido como

$$\bar{d} = \sum_{i=1}^{|Q|} d_i / |Q|.$$

5.3.3.2 Extensión

Deb *et al.* (2000a) proponen la métrica Δ :

$$\Delta = \frac{\sum_{m=1}^M d_m^e + \sum_{i=1}^{|Q|} |d_i - \bar{d}|}{\sum_{m=1}^M d_m^e + |Q| \bar{d}} \quad (5.23)$$

donde d_i puede ser cualquier distancia medida entre las vecindades de las soluciones y \bar{d} es la media de los valores de las distancias. El extremo M es el número de objetivos del problema analizado. El parámetro d_m^e es la distancia entre el extremo de soluciones P^* y Q correspondiente al m -ésimo objetivo.

5.3.3.3 Máxima extensión

Zitzler (1999) define una métrica D para medir el tamaño de la diagonal de la hipercaja¹¹ formada por los valores extremos para la función en el conjunto de soluciones no dominadas:

$$D = \sqrt{\sum_{m=1}^M \left(\max_{i=1}^{|Q|} f_m^i - \min_{i=1}^{|Q|} f_m^i \right)^2}. \quad (5.24)$$

¹¹ Una hipercaja es una representación en 2-dimensiones de una caja n-dimensional. Esta caja es un paralelepípedo rectangular o poliedro de seis lados.

Para un problema con dos objetivos, esta métrica es la distancia euclidiana entre las soluciones extremas en el espacio objetivo como se puede observar en la Figura 5.1. En orden de tener normalizada la métrica, se puede reescribir la ecuación (5.24) de la siguiente manera:

$$D = \sqrt{\sum_{m=1}^M \left(\frac{\max_{i=1}^{|\mathcal{Q}|} f_m^i - \min_{i=1}^{|\mathcal{Q}|} f_m^i}{F_m^{\max} - F_m^{\min}} \right)^2} \quad (5.25)$$

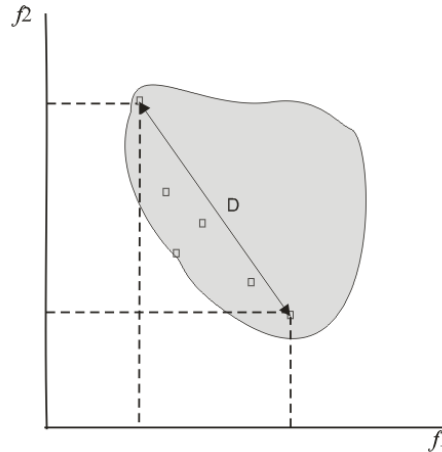


Figura 5.1 La máxima extensión no revela la distribución real de las soluciones

5.3.4 Métricas que evalúan la cercanía y la diversidad

5.3.4.1 Hipervolumen

Esta métrica calcula el volumen en el espacio de las funciones objetivo cubierto por los miembros de \mathcal{Q} (Figura 5.2) para problemas donde todos los objetivos son minimizados (Zitzler and Thiele, 1998).

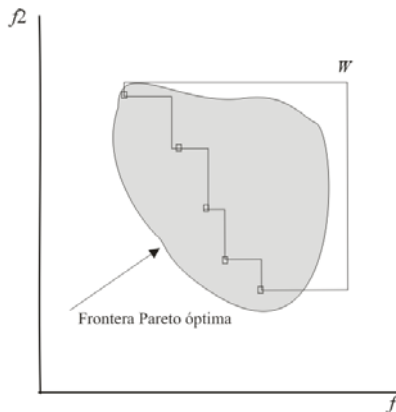


Figura 5.2 El hipervolumen encerrado en las soluciones no dominadas

Matemáticamente, para cada solución $i \in Q$, un hipercubo¹² v_i es construido con el punto de referencia W y la solución i como la diagonal de los extremos. El punto de referencia puede ser encontrado como el vector con peores valores en cada función objetivo. La unión de todos los hipercubos es encontrada y su indicador hipervolumen (I_H) se calcula como:

$$I_H = \text{volumen}\left(\bigcup_{i=1}^{|Q|} v_i\right). \quad (5.26)$$

5.3.4.2 Superficies de métricas basadas en estadísticas

Fonseca y Fleming (1996) sugieren el concepto de superficies o *attainment* en el contexto de la optimización multi-objetivo. Los autores argumentan que, mezclando las soluciones no dominadas obtenidas por una curva, una envoltura puede ser formada marcando todas aquellas soluciones en el espacio de búsqueda que son dominadas por el conjunto de soluciones no dominadas obtenido.

5.3.5 Métricas para evaluar MOEAs

Estas métricas evalúan los conjuntos de aproximación obtenidos por distintos MOEAs con el objetivo de comparar las soluciones obtenidas por cada algoritmo.

5.3.5.1 Ranking de dominancia

Supongamos que se desea comparar la calidad de los conjuntos de aproximación generados para $q \geq 2$ algoritmos de optimización multi-objetivo. Para cada procedimiento $i \in \{1, \dots, q\}$, un número de corrida $r_i \geq 1$ son ejecutadas, generando los conjuntos $A_1^1, A_2^1, \dots, A_{r_i}^1, \dots, A_1^q, \dots, A_{r_q}^q$.

Si tomamos \mathbf{C} como la colección de todos estos conjuntos, entonces típicamente algunos de ellos podrían dominar o ser mejores que otros, mientras que otros pares serían incomparables. Esto se puede resumir en la Tabla 5.1.

¹² Un hipercubo es una figura formada por dos cubos tridimensionales desplazados en un cuarto eje dimensional.

Tabla 5.1 Relaciones entre distintos conjuntos de soluciones

relación	Interpretación en el espacio objetivo	
estrictamente dominado	$A \prec\prec B$	Todo $z^2 \in B$ es estrictamente dominado por al menos un $z^1 \in A$
dominado	$A \prec B$	Todo $z^2 \in B$ es dominado por al menos un $z^1 \in A$
mejor	$A \triangleleft B$	Todo $z^2 \in B$ es débilmente dominado por al menos un $z^1 \in A$ y $A \not\prec B$
Dominado débilmente	$A \leq B$	Todo $z^2 \in B$ es débilmente dominado por al menos un $z^1 \in A$
incomparable	$A \parallel B$	Ni $A \leq B$ ni $B \leq A$
indiferentes	$A \sim B$	$A \leq B$ y $B \leq A$

Existen varias maneras de asignar un ranking para cada uno de los conjuntos de aproximación en base a la dominancia, es decir, contando el número de conjuntos que son dominados (Fonseca y Fleming, 1193) o realizando un ordenamiento de no dominancia de C (Goldberg, 1989). Tomando la primera forma, podemos establecer:

$$rank(C) = 1 + \left| \{C_j \in \mathbf{C} : C_j \triangleleft C_i\} \right|. \quad (5.27)$$

De esta forma, el ranking más pequeño es aquel que contiene la mejor aproximación con respecto a toda la colección C.

El ranking de dominancia puede ser utilizado para comparar dos MOEAs de forma tal de que si el conjunto de aproximación de un algoritmo contiene soluciones extremadamente mejores que el segundo, este último pueda ser descartado sin necesidad de realizar otros estudios de comparación. Sin embargo, cuando los conjuntos son muy similares es necesario trabajar con otros indicadores a fin de obtener mayor información.

5.3.5.2 Indicadores de Calidad

Los indicadores de calidad están orientados a evaluar la calidad de las soluciones de los conjuntos de aproximación en contraste con un conjunto R de referencia.

5.3.5.2.1 El indicador hipervolumen I_H

Teniendo en cuenta un conjunto de referencia R, podemos definir el indicador I_H^- de la siguiente manera: dado un conjunto de aproximación A, el resultado del indicador se define como:

$$I_H^- = I_H(R) - I_H(A) \quad (5.28)$$

donde el valor más pequeño corresponde a una calidad alta en contraste con el hipervolumen original I_H (sección 4.3.3.1).

5.3.5.2.2 Los indicadores unarios epsilon I_ε^1 e $I_{\varepsilon+}^1$

La familia del **indicador epsilon** fue introducida por Zitzler *et al.* (2003) y comprende una versión multiplicativa y una aditiva, ambas con opciones unarias y binarias.

En el caso particular del **indicador binario multiplicativo epsilon** $I_\varepsilon^1(A, B)$, el mismo devuelve el mínimo factor ε donde cada punto en B puede ser multiplicado de manera tal que el conjunto de aproximación resultante es débilmente dominado por A .

$$I_\varepsilon(A, B) = \inf_{\varepsilon \in \mathbb{R}} \{ \forall z^2 \in B, \exists z^1 \in A : z^1 \leq z^2 \}. \quad (5.29)$$

Para ambos indicadores, si el valor menor es 1 en el caso de I_ε^1 y 0 para $I_{\varepsilon+}^1$ implica que A domina estrictamente al conjunto R .

5.3.5.2.3 Los indicadores I_{R2}^1 e I_{R3}^1

Los indicadores R propuestos por Hansen y Jaskiewicz (1998) pueden ser usados para comparar conjuntos de aproximación en base a un conjunto de funciones de utilidad. La función de utilidad u es definida como una función del conjunto Z de vectores objetivo n -dimensionales a un conjunto de números reales $u : Z \rightarrow \mathbb{R}$.

Supongamos que las preferencias de los DMs están dadas en términos de una función de utilidad u_λ y un correspondiente conjunto Λ de parámetros; por ejemplo, u_λ podría representar una suma de pesos de los valores de la función objetivo, donde $\lambda = (\lambda_1, \dots, \lambda_n) \in \Lambda$. Hansen y Jaskiewicz (1998) proponen varias maneras de transformar esta familia de funciones de utilidad en indicadores de calidad. En particular los indicadores binarios I_{R2} e I_{R3} son definidos como:

$$I_{R2}(A, B) = \frac{\sum_{\lambda \in \Lambda} u^*(\lambda, A) - u^*(\lambda, B)}{|\Lambda|} \quad (5.30)$$

$$I_{R3}(A, B) = \frac{\sum_{\lambda \in \Lambda} [u^*(\lambda, B) - u^*(\lambda, A)] / u^*(\lambda, B)}{|\Lambda|} \quad (5.31)$$

donde u^* es el máximo valor alcanzado por la función de utilidad u_λ con un vector de peso λ en un conjunto de aproximación A, es decir, $u^*(\lambda, A) = \max_{z \in A} u_\lambda(z)$.

5.3.5.2.4 Combinando indicadores de calidad Pareto compatibles

Existen en la literatura algoritmos que son evaluados simultáneamente con diferentes indicadores. Esto sirve para proveer de información adicional a los DMs, sólo si los indicadores son Pareto compatibles. Una combinación de indicadores Pareto compatibles da más información en conjunto que un único indicador. En particular, si dos indicadores Pareto compatibles se contradicen, esto implica que los conjuntos son incomparables.

5.4 PLATAFORMA E INTERFASE INDEPENDIENTE DEL LENGUAJE PARA

ALGORITMOS DE BÚSQUEDA

Tomando como punto de partida la gran variedad de MOEAs en la literatura y el número de métricas para evaluarlas, Bleuler *et al.* (2003) desarrollaron una plataforma e interfaz independiente del lenguaje para algoritmos de búsqueda (***Platform and Programming Language Independent Interface for Search Algorithms, PISA***). Esto permitió ahorrar una gran cantidad de tiempo de programación del algoritmo híbrido que será estudiado en el Capítulo 7 como así también realizar un estudio comparativo entre los MOEAs en base a las métricas implementadas por PISA. Bleuler *et al.* proponen esta plataforma en base a la separación entre la implementación de la sección específica del problema y la etapa independiente del mismo. Esta plataforma ahorra tiempo de codificación y además provee una batería de indicadores que permiten la evaluación de varios MOEAs.

5.4.1 Estructura general de PISA

La arquitectura de PISA se caracteriza por separar la selección de los individuos y el cálculo del *fitness*, comunicarse vía archivos de texto que permite la independencia tanto del lenguaje de programación como del sistema operativo donde se implementa el algoritmo y minimizar la comunicación entre las secciones de PISA.

PISA es una interfaz para algoritmos de búsqueda dividida en dos módulos. El módulo llamado *variator* contiene todas las partes específicas del problema de optimización a resolver. Otro módulo, denominado *selector* es independiente del problema a resolver y contiene el proceso de selección. Estos módulos son programas separados que se comunican a través de archivos de texto. Además del *variator* y el *selector*, PISA contiene una librería de evaluación del desempeño (Knowles *et al.*, 2006) que contiene un conjunto de herramientas estadísticas para evaluar y comparar diferentes métodos de optimización. Para esta tesis doctoral se hizo uso de los selectores NSGA-II, SPEA2 e IBEA implementados en PISA. Este último método de selección, IBEA (indicator-based evolutionary algorithm) es introducido por Zitzler y Künzli (2004). En la Figura 5.3 se puede observar la estructura general de PISA adaptada al trabajo realizado en esta tesis:

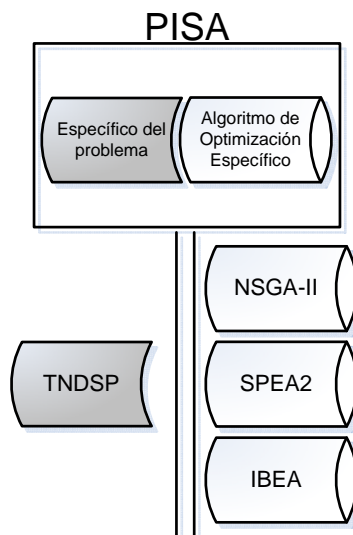


Figura 5.3 Visión global de PISA para el caso particular del TNDSP.

A continuación explicaremos IBEA el cual será utilizado como inspiración para una de las implementaciones de nuestro algoritmo híbrido mostrado en el Capítulo 7 .

5.4.2 IBEA: indicator-based evolutionary algorithm

Teniendo en cuenta la definición para los indicadores I_ϵ que se estableció en la sección 4.3.5.2.2, la población P representa una porción del espacio de decisión. La asignación del *fitness* trata de realizar un ranking de los miembros de la población de acuerdo a su valor de aptitud en el objetivo a optimizar. Sea $F(x_1)$ el valor de *fitness* de x_1 calculado como:

$$F(x_1) = \sum_{x_2 \in P \setminus \{x_1\}} -e^{-I(\{x_2\}, \{x_1\})/k} . \quad (5.32)$$

En esta ecuación utilizaremos una propiedad para el indicador de preservación de dominancia:

$$I(\{x_2\}, \{x_1\}) < I(\{x_2\}, \{x_1\}) \text{ Si } x_1 \succ x_2 . \quad (5.33)$$

De esta manera, la influencia de un valor pequeño del indicador contribuye mucho más a todo el *fitness* que los valores grandes. El parámetro k es un factor de escala que depende de I y del problema a resolver; $k > 0$.

Para la asignación se utiliza el indicador I_ϵ aditivo, aunque también se pueden utilizar otros indicadores como I_{HD} :

$$I_{HD}(A, B) = \begin{cases} I_H(B) - I_H(A) & \text{si } \forall x_2 \in B \exists x_1 \in A : x_1 \succ x_2 \\ I_H(A+B) - I_H(A) & \text{sino} \end{cases} . \quad (5.34)$$

I_{HD} mide el volumen del espacio que es dominado por B pero no por A respecto de un punto de referencia Z .

Resumen del capítulo

En este capítulo se trataron los principales MOEAs presentados en la literatura. Además se introdujeron indicadores de análisis que permiten comparar distintos conjuntos de soluciones de un mismo MOEA o bien de distintos MOEAs. Dada la necesidad de probar varios MOEAs

para el problema del tránsito se introduce la plataforma PISA como herramienta de implementación y análisis.

Capítulo 6 Simulación

Existe una variedad de sistemas en la naturaleza que son modelados para tratar de predecir distintos comportamientos y resultados. Con este propósito, se ha debido recurrir a distintas formas de modelar la realidad en diferentes disciplinas y con el tiempo se han vuelto indispensables. La simplicidad y el tipo de modelo están condicionados por los requerimientos del problema a resolver. Estos modelos abarcan desde maquetas a escala para edificios hasta la modelización de procesos químicos complejos. La **simulación** es la imitación del comportamiento del mundo real o de un sistema en el tiempo, representado a través de un modelo. Con el surgimiento de las ciencias de la computación, el avance de la tecnología y las computadoras digitales, los modelos lógicos y matemáticos de los sistemas se volcaron a la computadora. Algunas veces, los modelos son lo suficientemente simples que pueden ser resueltos a través de métodos matemáticos estandarizados con suficiente precisión. Estas soluciones pueden ser obtenidas a través de cálculo, teoría de probabilidades, métodos algebraicos u otras técnicas matemáticas. Sin embargo, cuando el sistema del mundo real es lo suficientemente complejo, resolverlo por métodos matemáticos estandarizados se vuelve virtualmente imposible. En estos casos, la simulación computarizada puede ser utilizada para imitar el comportamiento del sistema en el tiempo recolectando datos del mundo real que se está observando, los cuales son utilizados para estimar las medidas de desempeño del sistema. En este capítulo abordaremos los principales conceptos relacionados con simulación.

6.1 SISTEMA

Para modelar y simular un sistema es necesario entender la definición del mismo y sus límites.

Definición 6.1: *Un sistema es un grupo de objetos relacionados que interactúan entre sí con un propósito.*

Entre los ejemplos podemos mencionar el sistema respiratorio, el circulatorio y hasta sistemas tan diversos como el sistema de fabricación de automóviles.

Por lo general, los sistemas son afectados por cambios que ocurren fuera del mismo, estos cambios se dice que son el *entorno* de dicho sistema (Gordon, 1979). Por este motivo, es importante decidir los límites entre el sistema y su entorno. Esta decisión puede depender del propósito del estudio del sistema.

6.1.1 Componentes de un sistema

Antes de analizar un sistema es necesario definir los objetos que lo componen. Diremos que una **entidad** es un *objeto* de interés para el sistema, dichos objetos tendrán **atributos** o propiedades. Una **actividad** representa un *período de tiempo* de tamaño específico. El **estado** de un sistema se define como la colección de variables necesarias para describir el sistema en cualquier momento del tiempo y es relativo a los objetos en estudio. Un **evento** es una ocurrencia *instantánea* que cambia el estado del sistema. El término **endógeno** lo utilizaremos cuando este evento o actividad ocurre dentro del sistema y **exógeno** en caso contrario.

6.1.2 Sistemas discretos y continuos

Los sistemas pueden ser divididos en discretos o continuos. Un sistema **discreto** es aquel en el que los estados de las variables cambian en puntos discretos del tiempo. Un sistema **continuo** es aquel donde el estado de las variables cambia de manera continua.

6.2 MODELO DE UN SISTEMA

Con el objeto de estudiar un sistema, a veces es necesario entender las entidades que lo componen y sus relaciones a fin de predecir su comportamiento bajo ciertas circunstancias. Es cierto que muchas veces es posible experimentar directamente con el sistema real, pero otras veces por varios motivos esto no se puede realizar. Por ejemplo, si se trata de un sistema que aún no existe. Incluso a veces es imposible experimentar con el sistema, aunque exista. Por este motivo, es que se suele estudiar un sistema a través de su modelo.

Definición 6.2: *Un modelo de un sistema es una representación del mismo con el propósito de estudiarlo.*

En la mayoría de los casos el modelo sólo tiene en cuenta aquellos aspectos que desean estudiarse. Podemos decir entonces que el modelo es una simplificación del sistema.

6.2.1 Tipos de Modelos

Los modelos de un sistema pueden clasificarse como matemáticos o físicos. Un modelo matemático utiliza notación simbólica y ecuaciones matemáticas para representar un sistema. En particular, un modelo de **simulación** es un tipo de modelo matemático de un sistema.

6.2.2 Modelos de Simulación

Los modelos de simulación pueden ser clasificados en estáticos y dinámicos, deterministas o estocásticos y discretos o continuos (Pidd, 1998). Un modelo de simulación *estático* es a veces llamado simulación de Monte Carlo y representa al sistema en un punto particular del tiempo (Banks *et al.*, 2001). Los modelos de simulación *dinámicos* representan a sistemas que cambian en el tiempo.

Por otro lado, los modelos de simulación que no contienen variables aleatorias son denominados *deterministas*. Los modelos deterministas tienen conocimiento del conjunto de entradas que resulta en un único conjunto de salidas. Un modelo de simulación *estocástico*

tiene una o más variables aleatorias como entrada. Las variables en una simulación pueden cambiar en forma *continua* o *discreta*. La naturaleza del cambio dependerá de la situación, el propósito del modelo y el tipo de equipo de computación disponible.

Dado que el problema tratado en esta tesis puede ser en parte modelado a través de eventos discretos, introduciremos a continuación algunas definiciones relacionadas con este tópico.

6.3 SIMULACIÓN DE EVENTOS DISCRETOS

En la simulación de eventos discretos el estado de las variables cambia únicamente en momentos discretos del tiempo. Los modelos de simulación son analizados por lo general de una manera numérica más que analítica. Los métodos analíticos emplean razonamiento deductivo para resolver el modelo, mientras que los métodos numéricos utilizan procedimientos computacionales para resolver el modelo matemático.

Para el desarrollo de esta tesis haremos uso de los métodos numéricos, los modelos son “corridos” más que resueltos. De esta manera, se genera una historia artificial del sistema basada en supuestos y observaciones que son recolectadas y analizadas con el objetivo de estimar el comportamiento real del sistema.

6.3.1 Conceptos en simulación de eventos discretos

A continuación se exponen los conceptos básicos que forman un *framework* o marco para el desarrollo de modelos de simulación de eventos discretos (Banks *et al.*, 2001).

Un **sistema de eventos discretos** es una colección de entidades que interactúan juntas en el tiempo con el objetivo de alcanzar una o más metas. Un **modelo en eventos discretos** es una representación abstracta del sistema que usualmente contiene relaciones estructurales, lógicas o matemáticas que describen al sistema en términos de estados, procesos, eventos, actividades, demoras, entidades y sus atributos. El **estado** del sistema es la colección de

variables que contienen toda la información necesaria para describir al sistema en cualquier momento del tiempo.

Una **entidad** es cualquier objeto o componente del sistema que requiere explícitamente su representación en el modelo. Los **atributos** son las propiedades de una entidad. Una **lista** es una colección (permanente o temporaria) de entidades asociadas y organizadas de alguna manera lógica.

Un **evento** es una ocurrencia *instantánea* que cambia el estado del sistema. Una **notificación de evento** es el registro de la ocurrencia actual o futura, junto con los datos asociados para ejecutar ese evento; requiere como mínimo el tipo de evento y el momento en que ocurre.

En contraste, una **actividad** es una acción que transcurre durante un intervalo de tiempo de un tamaño específico que tiene un comienzo conocido definido en términos de una distribución estadística. Una duración en el tiempo de tamaño indefinido, cuya terminación no puede ser conocida hasta su fin se denominará **demora**. Se llama **reloj** a la variable que representa el tiempo simulado.

6.4 PASOS DE UN ESTUDIO MEDIANTE SIMULACIÓN

Varios autores han propuesto una serie de pasos más o menos similares para el abordaje de una simulación (Gordon, 1979). En general, podemos resumirlos en la Figura 6.1.

Formulación del problema. Todo estudio debe comenzar con un planteo escrito del problema. Este planteo es proporcionado por los que desean tomar una decisión sobre el problema.

Fijar objetivos. Los objetivos indican las preguntas que debe responder la simulación.

Construcción del modelo conceptual. La construcción de un modelo es la habilidad de abstraer las características esenciales del sistema en estudio para enriquecer y elaborar un modelo hasta que se obtengan resultados aproximados.

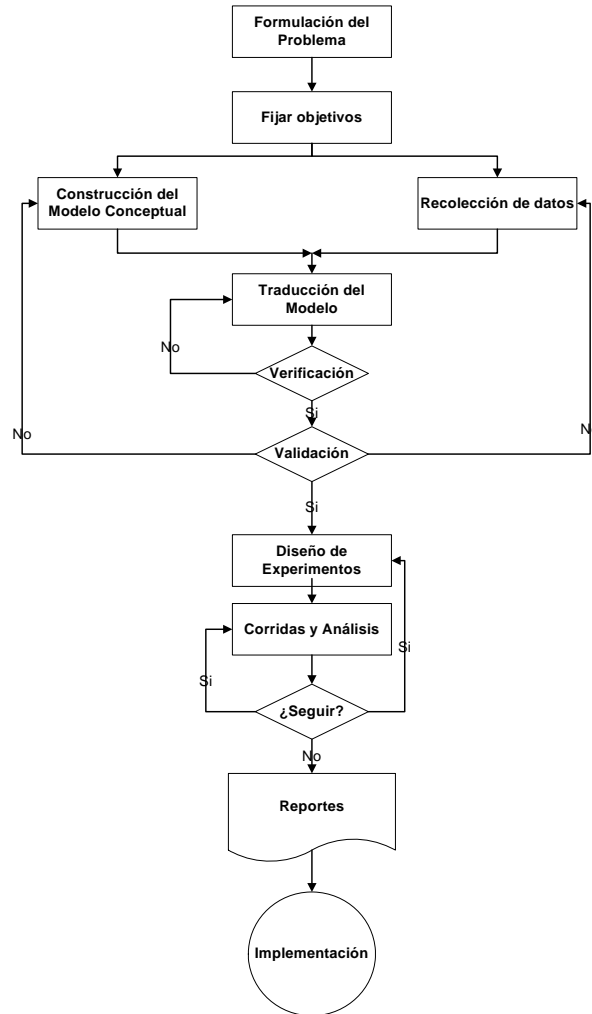


Figura 6.1 Pasos de un estudio utilizando simulación

Recolección de datos. Existe según Shannon (1975) una interrelación directa entre la construcción del modelo y la colección de datos de entrada de una simulación. A medida que la complejidad del modelo cambia, también lo hacen los datos que necesita. Los objetivos del estudio marcan en gran escala el tipo de datos que se recolectan.

Traducción del modelo. Este paso tiene como objetivo volcar el modelo que representa el sistema del mundo real en un formato reconocible para la computadora. Existen en el mercado una gran cantidad de software dedicado a simular aunque a veces, dependiendo del

problema, puede ser necesario programar la simulación de otra manera. Estos lenguajes de programación son poderosos y flexibles por lo que el tiempo de desarrollo del modelo es altamente reducido.

Verificación. La verificación permite a la computadora estar preparada para la simulación del modelo. Permite saber si el programa está ejecutándose correctamente.

Validación. La validación determina si el modelo es una representación apropiada del sistema real. Usualmente ayuda a calibrar el modelo.

Diseño de experimentos. Las alternativas a ser simuladas deben ser determinadas. A veces la decisión concerniente a cuáles alternativas simular puede estar en función de las corridas. Por lo general, los principales aspectos de este paso son determinar el tiempo de inicialización, el tamaño de una corrida y el número de réplicas.

Corridas y análisis. Para estimar medidas de desempeño para del sistema, la simulación debe ejecutarse repetidamente y se debe realizar un análisis de los resultados.

Seguir corriendo. Basándose en el análisis de las corridas que ya han sido completadas el analista puede determinar si corridas adicionales son necesarias y cuáles experimentos deben diseñarse.

Documentación y reportes. Existen dos tipos de documentación: del programa y del progreso. La documentación del programa es necesaria por numerosas razones. Si el programa es utilizado por un analista y luego por otro, es necesario que esté documentado para entender cómo opera. Los reportes deben ser devueltos de manera clara y concisa. Por otro lado, los reportes de progreso son aquellos que muestran la cronología del trabajo y las decisiones tomadas en el tiempo. Estos ayudan a proveer la historia del proyecto de simulación.

Implementación. El éxito de la implementación depende en gran parte de cómo se realizaron los pasos anteriores.

6.5 SIMULACIÓN EN SISTEMA DE COLAS

La simulación es a veces utilizada para analizar los modelos de colas. En un típico sistema de colas encontramos a los clientes deseando acceder por un recurso por lo que se van colocando en una cola de espera para ser atendidos. Estos clientes eventualmente serán atendidos y dejarán el sistema.

Los modelos de colas pueden ser resueltos matemáticamente o analizados a través de la simulación. Las típicas medidas que se pueden obtener en un sistema de colas son tiempo de servicio y de espera. Para sistemas relativamente sencillos las medidas pueden ser calculadas matemáticamente, pero en modelos realistas de sistemas complejos, como es el caso del abordado en esta tesis, la simulación es requerida.

6.5.1 Características de los sistemas de colas

Los elementos claves de un sistema de colas son los clientes y los servidores. El término cliente puede referirse a personas, máquinas, mecánicos, pacientes, aviones, etc.; es decir, cualquier cosa que arribe y necesite un servicio. El término servidor puede referirse a la recepción de personas, un operario que repara máquinas, personal médico, máquinas de almacenamiento automático, la CPU de una computadora, o un lavadero automático; es decir, cualquier recurso (persona o máquina) que brinda un servicio.

6.5.1.1 Los clientes

La población de potenciales clientes se conoce como población y puede ser finita o infinita. En sistemas con gran cantidad de clientes se suele suponer que se trata de población infinita. La principal diferencia entre estos tipos de población de clientes tiene que ver con la forma en que se define la tasa de arribo.

En un modelo de población infinita la tasa de arribo o el número promedio de arribos por unidad de tiempo no es afectado por el número de clientes que pasan de la población de

potenciales clientes a la cola del sistema. Por otra parte, para modelos de población finita, la tasa de arribo a la cola del sistema depende del número de clientes que fueron atendidos y los que están esperando.

6.5.1.2 Capacidad del Sistema

En muchos sistemas de colas existe un límite en la cantidad de clientes que pueden estar esperando en la cola del sistema. Por este motivo cuando el sistema tiene una capacidad limitada, se debe hacer una distinción entre la tasa de arribo (el número de clientes que llegan por unidad de tiempo) y la tasa de arribo efectivo (el número de arribos que entraron y se quedaron en el sistema por unidad de tiempo).

6.5.1.3 El proceso de arribo

El proceso de arribo para una población infinita es usualmente caracterizado en términos de los tiempos entre arribos de los clientes. Estos arribos pueden ocurrir de manera planificada o aleatoria (los clientes que llegan a la parada de buses). En el caso de que los arribos sean aleatorios, el tiempo entre arribos se define a través de una distribución de probabilidad. Además, los clientes pueden llegar uno cada vez o en conjunto. Uno de los modelos más reconocidos para arribos es el proceso de Poisson.

Si A_n representa el intervalo de tiempo entre el cliente $n-1$ y el cliente n (A_1 es actual tiempo de arribo para el primer cliente), entonces para un proceso de arribo de Poisson A_n esta exponencialmente distribuido con media de $1/\lambda$ unidades de tiempo.

Otra clase de arribo son los arribos planificados, tales como aquellos de los pacientes a un consultorio médico. En este caso los tiempos entre arribos pueden ser constantes o con un pequeño número aleatorio que suma o resta a esa constante. La Tabla 6.1 muestra la notación estándar de teoría de colas que se utiliza en esta tesis.

Tabla 6.1 Notación para sistemas de colas

Notación	Significado
$Prob_n$	Probabilidad de tener n clientes en el sistema
$Prob_n(t)$	Probabilidad de tener n clientes al momento t
λ	Tiempo de arribo
λ_e	Tiempo de arribo efectivo
μ	Tiempo de servicio en un servidor
ρ	Utilización del servidor
A_n	Intervalo entre arribos del cliente $n-1$ y n
S_n	Tiempo de servicio del n -esimo cliente
W_n	Tiempo en el sistema del n -esimo cliente
W_n^Q	Tiempo total de espera en la cola de n
$L(t)$	Número de clientes en el sistema en el momento t
$L_Q(t)$	Número de clientes en la cola en el momento t
L	Tiempo promedio de clientes en el sistema
L_Q	Tiempo promedio de clientes en la cola
w	Tiempo promedio de un cliente en el sistema
w_Q	Tiempo promedio de un cliente en la cola

Para modelos de población finita, el proceso de arribos es caracterizado de una manera diferente. Se define como pendiente cuando un cliente esta fuera de la cola del sistema y como miembro de la población de potenciales clientes.

6.5.1.4 Comportamiento de la cola y disciplina de cola

El comportamiento de la cola esta relacionado con la actitud de los clientes mientras esperan en la cola a ser atendidos. Es común que un cliente que arriba al sistema se vaya sin formar parte de la cola cuando la cola es muy larga, cuando ha arribado a la cola y ve, luego de un tiempo que se mueve muy lento o bien cambie de cola porque la cola actual es muy lenta.

La disciplina de cola se refiere al orden lógico en el que los clientes serán atendidos cuando el recurso se libere. Las disciplinas más comúnmente usadas son primero en entrar, primero en salir (FIFO); último en entrar, primero en salir (LIFO); aleatorio (SIRO); atender primero al procedimiento más corto (SPT); prioridad (PR).

En el caso particular del problema que motiva esta tesis doctoral, el arribo de los clientes a la parada de buses y su ingreso al móvil se toma comúnmente como FIFO.

6.5.1.5 Tiempos de servicio y mecanismo de servicio

Los tiempos de servicio de los arribos sucesivos suelen denotarse como $S_1, S_2,$ etc. Los mismos pueden ser constantes o aleatorios. Para el caso de aleatorio $\{S_1, S_2, \dots\}$ es usualmente caracterizado como una secuencia de variables independientes con idéntica distribución probabilística. Distribuciones como Weibull, Gamma, lognormal, y normal son utilizadas para modelar tiempos de servicio.

Por último, los **sistemas de colas** consisten en un número de centros de servicio y colas. Cada centro de servicio consiste en un número de servidores c , trabajando en paralelo. Los servidores en paralelo pueden ser de servidor simple ($c=1$) o múltiple ($c>1$), o ilimitados ($c=\infty$).

Los modelos de colas son aplicables a gran cantidad de escenarios y poseen un atractivo particular cuando se desea analizar los tiempos de espera y de servicio del sistema y las congestiones y situaciones que pueden llevar a un problema futuro para el sistema.

6.5.2 Notación para los sistemas de colas

Debido a la variedad de sistemas de colas, Kendall (1953) propuso un sistema de notación para los sistemas de servidor paralelo que fue rápidamente adoptado. Una versión abreviada del formato sería $A / B / c / N / K$ donde A representa la distribución probabilística del tiempo entre arribos y B la distribución probabilística del tiempo de servicio. Comúnmente A y B se reemplazan por M (exponencial o Markov), D (constante o determinista), E_k (Erlang de orden k), PH (tipo fase), H (hiper-exponencial), G (arbitraria o general) y GI (general independiente). C representa el número de servidores paralelos, N representa la capacidad del sistema y K representa el tamaño de una población de posibles clientes.

Por ejemplo, $M / M / 1 / \infty / \infty$ indica que se trata de un sistema de capacidad de cola ilimitada y población potencial infinita con una sola cola y el tiempo de arribos y de servicio son en ambos casos distribuciones exponenciales.

La simulación es una herramienta poderosa de análisis que permite observar los estados del sistema bajo estudio. En el Capítulo 7 , estableceremos el modelo de simulación del problema de diseño de redes de tránsito y la utilización de la teoría de colas en la simulación del sistema.

Resumen del capítulo

En este capítulo desarrollamos los conceptos fundamentales que rigen los modelos de simulación como así también los aspectos relacionados con teoría de colas. Como observamos la simulación puede ser utilizada para generar historia de un sistema complejo. Estos conceptos serán abordados nuevamente cuando se introduzca el algoritmo utilizado para resolver el problema planteado en el Capítulo 1 y el método de simulación utilizado.

Capítulo 7 Diseño de un algoritmo híbrido dinámico multi-objetivo para redes de tránsito

Existe evidencia experimental (Davis, 1991; Michalewicz, 1994) que asegura que el uso de algoritmos evolutivos (EAs) combinados con alguna heurística adicional que introduzca conocimiento del problema puede mejorar la calidad de los resultados obtenidos. Estos sistemas a veces son denominados *algoritmos evolutivos híbridos*.

Varios investigadores han reconocido el potencial de hibridizar los EAs. Davis (1991) subraya el empleo de tres principios para realizar la hibridización:

- 1) Utilizar una codificación propia del individuo.
- 2) Hibridizar donde sea posible.
- 3) Adaptar los operadores genéticos incorporándoles características propias del dominio del problema.

En este capítulo se propone un algoritmo evolutivo híbrido para resolver el **problema del diseño y planificación de las redes de tránsito** (TNDSP). El mismo captura las características propias del TNDSP y las incorpora a la resolución, con el objetivo de obtener resultados realistas desde el punto de vista del dominio del problema teniendo en cuenta los tres principios descritos por Davis (1991).

7.1 MODELADO DE LAS ENTIDADES

Como dijimos en el primer capítulo, el TNDSP es un problema multi-objetivo donde los intereses de los usuarios y del operador del servicio son contrapuestos. El operador del servicio esta interesado en obtener la mayor ganancia utilizando los mínimos recursos y el usuario esta interesado en llegar lo más pronto posible a su destino realizando la menor cantidad de trasbordos a un costo mínimo.

Por estos motivos se proponen como objetivos: maximizar la rentabilidad del operario y al mismo tiempo minimizar los tiempos y el costo de viaje del cliente en la TN. El modelo propuesto en esta tesis doctoral es una variante al presentado por Gruttner *et al.* (2002).

Las ecuaciones (7.1), (7.2) y (7.3) están asociadas al operador del servicio:

$$FO = \sum_{L=1}^M FO_L = \sum_{L=1}^M (IO_L - CO_L) \quad (7.1)$$

$$IO_L = AF_L T_L \quad (7.2)$$

$$CO_L = D_L K_L \quad (7.3)$$

donde L es una línea de la red de tránsito, AF_L es la afluencia total de viajes para L, T_L es el precio de un viaje en L, D_L es la distancia recorrida por L y K_L es el costo de operación por kilómetro de L. La ecuación (7.2) es la ganancia del operador del servicio mientras que la ecuación (7.3) representa el costo operativo de la línea L. Finalmente, la ecuación (7.1) representa la ganancia neta del operador.

Para definir las ecuaciones asociadas al usuario es necesario introducir la definición de tres variables temporales. El tiempo de **acceso** (t_{ij}^A) el cual se define como el tiempo entre que el autobús arriba a la parada y el usuario efectivamente ingresa al móvil. El tiempo de **espera** (t_{ij}^W) es el tiempo que pasa desde el momento en el que el usuario llega a la parada y el autobús que debe tomar arriba a la misma. Por último, el tiempo de **viaje** (t_{ij}^J) es el tiempo que el usuario pasa efectivamente viajando dentro del móvil.

La ecuación (7.4) representa el costo de viaje del usuario del servicio (FU):

$$FU = \sum_{L=1}^M FU_L = \sum_{L=1}^M \left[\sum_{i=1}^N \sum_{j=1}^N (\delta t_{ijL}^A + t_{ijL}^J + \eta t_{ijL}^W) VST_{ijL} V_{ijL} \right] \quad (7.4)$$

donde L es una línea de la TN, VST_{ijL} es el valor subjetivo correspondiente al tiempo de viaje entre cada par origen-destino que emplea la línea L . V_{ijL} es el número de viajes entre cada par (i,j) que emplea la línea L . t_{ijL}^A es el tiempo de acceso para L . t_{ijL}^J es el tiempo de viaje para la línea L y t_{ijL}^W es el tiempo de espera para la línea L . Por último, δ y η representan pesos para el tiempo de acceso y el tiempo de espera con respecto al tiempo de viaje. En resumen, la ecuación (7.4) representa el costo de transporte asociado a los clientes de la TN; de esta ecuación se desprende la importancia del tiempo como factor determinante del costo del usuario.

Para un tratamiento real del TNDSP se desarrolló como principal aporte de esta tesis un algoritmo híbrido dinámico multi-objetivo (AHDM). El mismo captura las características estocásticas del problema haciendo uso de la simulación en el marco de la función de *fitness* de un algoritmo evolutivo multi-objetivo que constituye el núcleo de este nuevo método.

7.2 ASPECTOS GENERALES DEL AHDM

En líneas generales, el AHDM comprende dos etapas: la primera consiste en inicializar las distancias junto con las rutas entre las paradas a través de un GRASP (*Greedy Randomized Adaptive Search Procedure*); la segunda etapa es un algoritmo evolutivo que resuelve el TNDSP y utiliza un procedimiento de simulación para obtener los valores de las variables dependientes del tiempo necesarias para el cálculo de la función de aptitud. El esquema general del algoritmo se muestra en la Figura 7.1.

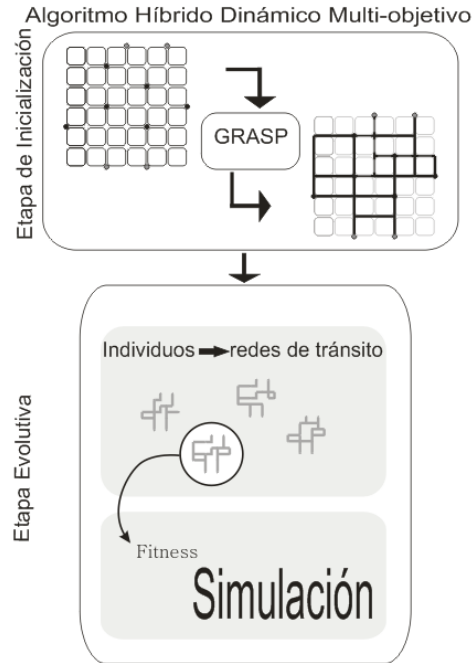


Figura 7.1 Esquema general del algoritmo híbrido dinámico multi-objetivo.

Dado que el modelo abordado tiene como uno de sus objetivos maximizar la ganancia, un aspecto crucial es obtener las distancias entre las paradas a fin de poder calcular la función objetivo asociada al operador (FO). Existen varias formas de calcular estas distancias y rutas; para esta tesis doctoral hemos elegido una meta-heurística que se ha vuelto popular en los últimos años denominada GRASP (*Greedy Randomized Adaptive Search Procedure*).

7.3 ETAPA GRASP

GRASP es un método de búsqueda local que se utiliza para problemas de optimización combinatoria (Pitsoulis y Resende, 2001). En su versión más simple es un procedimiento que consiste de dos fases: una fase constructiva y un procedimiento de búsqueda local. La solución es construida paso a paso agregando un nuevo elemento en cada momento. El siguiente elemento que formará parte de la solución es elegido de manera aleatoria de un conjunto de candidatos.

7.3.1 GRASP: Fase constructiva

GRASP constituye la etapa de inicialización: toma dos paradas i (origen) y j (destino) y devuelve la ruta y la distancia entre ellas. Dada S , una ruta factible parcial entre dos paradas y k el último nodo visitado en la ruta parcial, el algoritmo selecciona los posibles arcos que tienen a k como origen. Luego se calcula la probabilidad de cada arco para ser elegido que contiene a k como origen. Para cada nodo en el conjunto de nodos destino, asociados a los arcos e que parten de k , la probabilidad de selección tiene en cuenta la distancia a la parada destino j (ver Algoritmo 7.1).

Con el objetivo de converger a una solución factible se introduce una variable α dentro de una función f , la cual depende de un parámetro β y la iteración actual. La función f converge a cero a medida que el número de iteración crece:

$$\alpha = f(\beta, \text{iter}); \text{ iter} \in [1, \text{iterMax}], \beta \in \mathfrak{R}^{[0,1]}. \quad (7.5)$$

$$f(\beta, \text{iter}) = \frac{\beta}{\text{iter}}, \beta \in \mathfrak{R}^{[0,1]} \quad (7.6)$$

La probabilidad $P(k_j)$ de elegir un nodo k_j es definida teniendo en cuenta α y la inversa de la distancia, d_{kj}^{-1} . Se comienza generando una distorsión en la probabilidad de elección. Luego se elige el próximo nodo en la ruta a la parada destino j teniendo en cuenta la probabilidad generada. Sea n el número de nodos finales asociados a los arcos de e (ver Algoritmo 7.1); la distorsión considera m nodos cuya distancia es menor o igual que la del nodo anterior. La probabilidad de elección se calcula aplicando la ecuación (7.8). Luego la probabilidad del nodo con distancia mayor que su antecesor es calculada con la ecuación (7.7).

$$P(k_j) = \frac{\frac{(1-\alpha)}{m} d_{k_j}^{-1}}{\frac{1}{m} \sum_{j=(n-m)}^n d_{k_j}^{-1} + \alpha \left[\left(\frac{1}{(n-m)} \sum_{i=1}^{n-m} d_{k_i}^{-1} \right) - \left(\frac{1}{m} \sum_{j=(n-m)}^n d_{k_j}^{-1} \right) \right]} \quad (7.7)$$

$$P(k_i) = \frac{\frac{\alpha}{(n-m)} d_{k_i}^{-1}}{\frac{1}{m} \sum_{j=(n-m)}^n d_{k_j}^{-1} + \alpha \left[\left(\frac{1}{(n-m)} \sum_{i=1}^{n-m} d_{k_i}^{-1} \right) - \left(\frac{1}{m} \sum_{j=(n-m)}^n d_{k_j}^{-1} \right) \right]} \quad (7.8)$$

$i = 1, \dots, (n-m) \quad j = (n-m), \dots, n \quad k = 1, \dots, n$

$d_{k_1} \geq \dots \geq d_{k_i} \geq \dots \geq d_{k_{(n-m)}} \quad d_{k_{(n-m)}} = \dots = d_{k_j} = \dots = d_{k_n}$

Algorithm GRASP

Input: i, j : paradas, iterMax: entero, beta: real;
Output: bestRoute Ruta, mejorDist: real;
Var: alpha, distancia: real;
 e : conjunto de arcos;
 k : nodo;
 n : entero;
 r : Ruta;

1. mejorDist:= infinito;
2. **for** $n := 1$ **to** iterMax
3. alpha := (beta / n)
4. $e :=$ arcosQueSalen(i)
5. CrearProbabilidades(e , alpha)
6. $r(i, j)$.Actual := SeleccionarAleatorio(e)
7. **Repeat**
8. $r(i, j)$.Distancia := $r(i, j)$.Distancia
 + $r(i, j)$.Actual.arcoDistancia
9. $r(i, j)$.ListAgregar(Ruta(i, j).Actual)
10. $k := r(i, j)$.Actual.NodoFinal
11. $e :=$ arcosQueSalen(k)
12. CrearProbabilidades(e , alpha)
13. $r(i, j)$.Actual := SeleccionarAleatorio(e)
14. **until** $k = j$
15. $r(i, j) :=$ búsquedaLocal($r(i, j)$)
16. **if** $r(i, j)$.Distancia < mejorDist **then**
17. mejorRuta := $r(i, j)$
18. mejorDist := $r(i, j)$.Distancia
19. **end if**
20. **end for**
21. **return** mejorRuta, mejortDist
22. **end Algorithm**

Algoritmo 7.1 Esquema de GRASP

7.3.2 GRASP: búsqueda local

La búsqueda local es necesaria para retornar soluciones cercanas al óptimo. Por este motivo las vecindades de cada nodo perteneciente a la ruta devuelta por la fase constructiva son exploradas con el objetivo de encontrar una mejor solución (ver Algoritmo 7.2).

Tomemos i como origen y j como destino, las paradas de las cuales deseamos conocer una ruta que las conecte. Utilizaremos la variable auxiliar k para denotar cada nodo del camino entre i y j ; luego construiremos un conjunto de arcos, llamémoslo e , que salen del nodo k . El conjunto e es justamente la vecindad del nodo k . Luego se buscan todos los nodos que están conectados a k con j , formalmente, $r_{k,j} = (k, \dots, j)$. Si existe un nodo l en la sub-ruta

$r_{k,l} = (k, \dots, h, \dots, l)$ tal que l pertenece a e y $h \neq \emptyset$, entonces reemplazamos la sub-ruta $r_{k,l} = (k, \dots, h, \dots, l)$ por (k,l) y el nuevo camino de k a j queda definido como $r_{k,j} = (k, l, \dots, j)$.

Algorithm búsqueda local

Input: $r(i, j)$: Ruta;
Output: $r_{i,j} = (i, \dots, j)$: Ruta;
Var: k, l : nodo;

1. $r_{i,j} := r(i, j)$
2. $k := i$
3. **while** not $(k = j)$ **do**
4. $e := \text{arcosQueSalen}(k)$
5. **if** existe l tal que $(k, l) \in e$ y $r_{k,j} = (k, \dots, h, \dots, l)$ contiene una sub-ruta $r_{k,l} = (k, \dots, h, \dots, l)$ **then**
 reemplazar $r_{k,l} = (k, \dots, h, \dots, l)$ con (k, l)
6. **end if**
7. $k := \text{sig}(k, r_{k,j})$
8. **end while**
9. **return** $r_{i,j} = (i, \dots, j)$
10. **end Algorithm**

Algoritmo 7.2 Procedimiento de búsqueda local

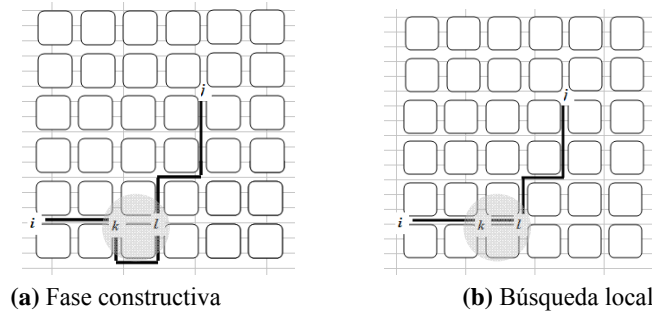


Figura 7.2 Gráfica del procedimiento de búsqueda local

Este procedimiento es repetido para todas las secuencias de nodos hasta que $k = j$. La Figura 7.2 (b) muestra el resultado de la búsqueda local en la ruta obtenida en la fase constructiva Figura 7.2 (a).

7.4 ETAPA EVOLUTIVA DEL AHDM

Varios autores han mostrado el éxito de los algoritmos evolutivos obteniendo soluciones para problemas NP-Hard (Baaj and Mahmassani, 1991).

La etapa evolutiva del AHDM comienza con el conjunto de parámetros genéticos, las distancias entre las paradas obtenidas a través de GRASP y una población inicial generada aleatoriamente. Los parámetros necesarios para esta etapa son: el mapa, las paradas iniciales $\{B_S\}$, intermedias $\{B_I\}$ y finales $\{B_F\}$, las rutas y distancias generadas por GRASP, el número de líneas (M , cuyo valor es mayor que el número de paradas iniciales y finales), el

costo de viaje (T_L), la matriz de demanda OD descrita en el Capítulo 1 , el costo unitario por kilómetro para cada línea (K_L), la capacidad máxima de los móviles (C_{MAX}) y los parámetros propios de la etapa evolutiva tales como probabilidad de mutación, de cruzamiento, número de individuos por generación (N) y número de generaciones (g).

7.4.1 Los individuos: conjunto de líneas

Cada **individuo** de la población representa una red de tránsito a través de un conjunto de líneas. Cada ruta o **línea** es modelada por una lista ordenada de enteros o recorrido comenzando por una parada inicial, una serie de paradas intermedias y una parada final. En tal sentido debemos tener presentes los conceptos de nodo, parada y trasbordo introducidos en el Capítulo 1 Sección 1.2. Los individuos sólo están formados por paradas utilizando una simplificación del mapa urbano como se puede ver en la Figura 7.3.

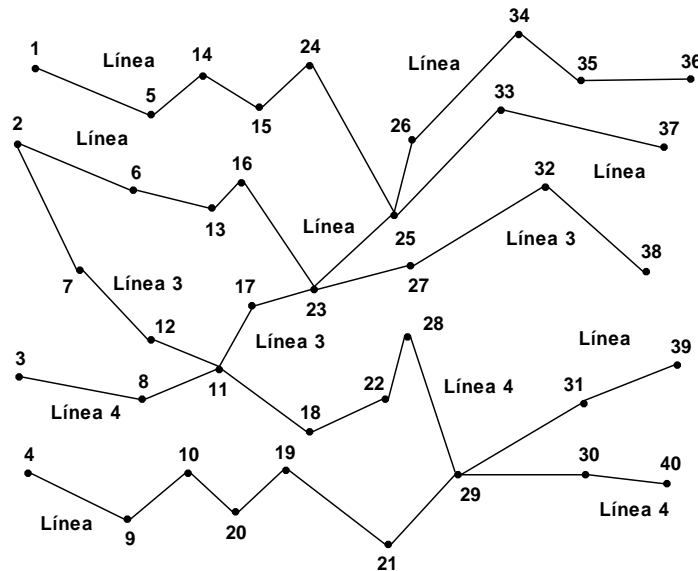


Figura 7.3 Ejemplo de un individuo con 40 paradas y 5 líneas.

Es importante destacar que no todas las asociaciones de líneas como cadenas de enteros constituyen una solución factible. Las restricciones asociadas con la factibilidad de los individuos pueden ser expresadas de acuerdo a la teoría de conjuntos. Sean L_1, L_2, \dots, L_M las

líneas pertenecientes a un individuo x , este individuo será una solución factible del TNDSP si y sólo si cumple las siguientes restricciones:

1. Una solución debe contener la misma cantidad de rutas como tamaño de líneas (M), el mismo es definido como parámetro de la etapa evolutiva:

$$x = \{L_1, \dots, L_M\}, \quad (7.9)$$

2. La red de tránsito debe ser conexa,

$$\forall L_i, \forall L_j, i \neq j \Rightarrow \exists \text{ un camino entre } i \text{ y } j \quad (7.10)$$

donde i y j son líneas de x .

3. Todas las paradas deben estar en al menos una línea

$$\{B_S\} \cup \{B_I\} \cup \{B_F\} = \{L_1\} \cup \{L_2\} \cup \dots \cup \{L_M\}. \quad (7.11)$$

7.4.2 Inicialización de la población

Como dijimos anteriormente, la población inicial del algoritmo se genera de forma aleatoria. A cada parada del conjunto $\{B_S\}$ se le da una probabilidad de 1 dividido el número de paradas iniciales. Para el caso de las paradas intermedias se realiza un procedimiento similar: a cada parada se le da una probabilidad de 1 dividido el número de paradas intermedias y de la misma manera se hace con las paradas finales. Luego para cada línea de la red representada por un individuo se toma un número al azar y se obtiene la parada inicial que le corresponde así la línea tendrá como parada inicial la elegida de manera aleatoria. Luego se genera un número aleatorio entre 1 y el número de paradas intermedias, llamémoslo a , y se toman sucesivamente a paradas intermedias de manera aleatoria, las cuales se añaden a la línea. Si se selecciona más de una vez la misma parada se descarta y se genera otra hasta cubrir a paradas intermedias. Por último se elige una parada final y se la asigna como parada final de la línea. Esto se realiza para todas las líneas que tiene un individuo.

Cuando el individuo está completo se evalúan las ecuaciones (7.9), (7.10) y (7.11). La ecuación (7.9) siempre se cumplirá, teniendo en cuenta la forma en que son construidos los

individuos. En el caso de que una línea no esté conectada con ninguna otra línea (7.10), se elige aleatoriamente una parada intermedia de una de las líneas y se inserta a la línea desconectada. Para la ecuación (7.11), se debe tener en cuenta que el número de paradas iniciales y finales no debe superar al número de líneas por individuo. En el caso de que una parada inicial o final no fuera elegida para encabezar o terminar ninguna línea existe entonces una parada repetida como inicial/final -según la definición de M-; dicha parada se reemplaza por la faltante. Si se trata de una parada intermedia que no fue agregada en ninguna línea se elige aleatoriamente una línea y se la inserta.

7.4.3 La función objetivo: Optimizando FO y FU

Teniendo en cuenta que FO y FU son dos objetivos claros del problema, no existe una solución única puesto que ambos objetivos son igualmente importantes y contrapuestos como fue dicho en el Capítulo 1 . Por este motivo, se propuso para el abordaje del modelo de la Sección 7.1 un algoritmo híbrido multi-objetivo basado en el concepto de dominancia de Pareto. En este contexto, el objetivo FO que era naturalmente maximizado, es redefinido para ser minimizado, y la ecuación (7.12) muestra el nuevo objetivo.

$$\begin{cases} 1/\left(1 + \sum_{L=1}^M FO_L\right), & \text{si } FO_L > 0 \\ 1, & \text{en caso contrario} \end{cases} \quad (7.12)$$

Luego, la optimización multi-objetivo para el TNDSP puede ser formalizada encontrando una variable de decisión x que minimice la siguiente función objetivo:

$$f(x) = \begin{bmatrix} FO(x) \\ FU(x) \end{bmatrix} \quad (7.13)$$

$$FO(x) = 1/\left(1 + \sum_{L=1}^M FO_L\right), \quad (7.14)$$

$$FU(x) = \sum_{L=1}^M FU_L .$$

El conjunto solución para el TNDSP consiste en todos los vectores de decisión cuyos vectores objetivo no pueden ser mejorados en una dimensión sin degradar al mismo tiempo la otra. A continuación explicaremos cómo se calculan cada una de estas variables a partir de un proceso de simulación al que es sometido cada individuo en el momento de la evaluación.

7.4.3.1 Simulación: cálculo de las variables dependientes del tiempo para la evaluación de los individuos

La forma en que la simulación es llevada a cabo depende en gran medida del modelo construido para resolver el problema. Para el tratamiento formal de TNDSP fue necesario aplicar técnicas relacionadas con teoría de colas y acceso a recursos, orientadas a la planificación de eventos. Es importante tener en cuenta que es posible el uso de herramientas de *Software* ya desarrolladas siempre que permitan obtener los valores necesarios para el cálculo de la función de aptitud.

7.4.3.1.1 Estructura de la red simulada

La estructura básica de una red de tránsito esta compuesta por diferentes líneas, la flota del operador, los usuarios y los puntos de trasbordo. Un punto de trasbordo es, como dijimos en el Capítulo 1 , compartido por más de una ruta lo que permite a los usuarios llegar a cualquier punto de la red de tránsito.

7.4.3.1.2 Entidades

Durante la simulación, cada entidad tiene asociada la información relevante para la simulación y aquella que será utilizada para el cálculo del *fitness* de la etapa evolutiva. Para cada **usuario** estaremos interesados en obtener el tiempo de espera (t_{ijL}^W), el tiempo de viaje (t_{ijL}^J), y el tiempo de acceso (t_{ijL}^A). Además, cada entidad mantiene información de la parada destino, de las paradas trasbordo en caso de que no llegue directamente a destino a través de la línea a la que puede acceder y el móvil en el cual se encuentra (en el caso de estar viajando). Cada entidad **parada** mantiene dos listas: una lista de clientes que están esperando

a tomar un móvil y una lista de usuarios o elementos que ya arribaron a la parada (V_{ijL}) y accedieron a algún móvil. Entre los atributos de la entidad **línea** podemos incluir la lista de paradas que la línea recorre, su flota de buses y la afluencia total de viajes (AF_L). Finalmente cada **móvil** tiene información sobre el punto en el que se encuentra, la parada siguiente a la que debe llegar, la línea a la que pertenece, la capacidad actual (C_A), la capacidad máxima del móvil (C_M) y el momento en el que arribará a la parada siguiente.

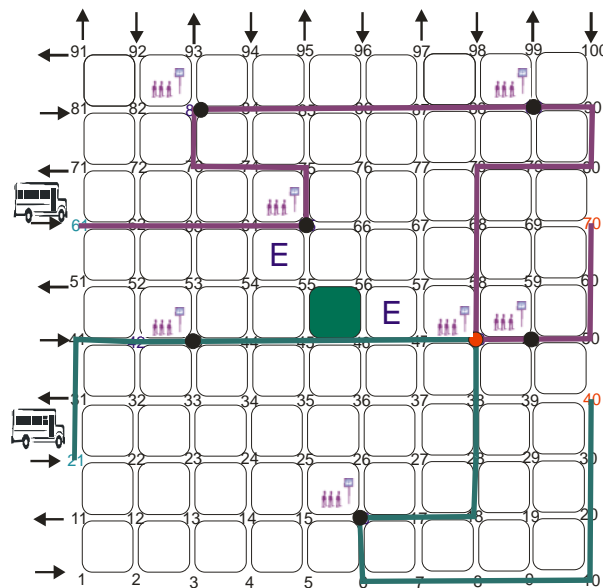


Figura 7.4 Estructura de una red con dos líneas, varias paradas, usuarios y buses.

En la Figura 7.4 podemos observar a los clientes, la cola de espera en cada una de las paradas, los móviles, los recorridos de las líneas y un punto de trasbordo de una red de tránsito pequeña.

7.4.3.1.3 Implementación del planificador de eventos

La construcción del modelo computacional para la simulación constituye un desafío debido a la complejidad del problema abordado. El **reloj** de la simulación avanza mediante un esquema de planificación de eventos. Antes de comenzar la simulación para un día de trabajo los arribos de todos los clientes a las respectivas paradas son generados y se colocan en la cola de potenciales clientes. El evento planificado que hace avanzar el reloj de simulación es el *arribo de un móvil a una parada*.

La simulación comienza generando los arribos de cada móvil a su parada de línea inicial teniendo en cuenta una frecuencia para los buses ω definida antes de comenzar la simulación. El reloj de simulación avanza hasta el primer arribo del primer móvil de la flota. Asimismo, el arribo de los móviles a la parada siguiente de cada línea es planificado, teniendo en cuenta el reloj actual y la distancia recorrida que fue obtenida a través de GRASP. Los siguientes atributos son actualizados: los de las líneas (AF_L , V_{ijL}), los del usuario (t_{ijL}^W , t_{ijL}^J , t_{ijL}^A) y los del móvil (C_A , t_{Now}). El reloj avanza al próximo arribo continuado repetitivamente hasta que finalmente la simulación termina.

7.4.4 Operadores genéticos

A continuación se introducen un operador de cruzamiento y dos operadores de mutación desarrollados especialmente para su utilización en los individuos descritos en la Sección 7.4.1, además del método de selección.

7.4.4.1 Cruzamiento multipunto multilínea

En esta tesis se desarrolló una variante del operador de cruzamiento de dos puntos OX que denominamos cruzamiento multipunto multilínea. Es importante mencionar que el operador de un punto no es satisfactorio para este TNDSP, puesto que en los casos de estudio analizados en el Capítulo 8 los hijos tenían un *fitness* inferior al de sus padres. Por esta razón sólo se analiza en esta sección la variante de dos puntos.

El método de cruzamiento trabaja de la siguiente manera: tomemos dos individuos, redes de tránsito en el TNDSP y dos arcos para cada línea o ruta de cada padre (cruzamiento de dos puntos). Para el primer hijo (Figura 7.6a), el algoritmo toma la primera parte de las sub-rutas del primer padre (Línea 1: 1-5-14; Línea 2: 2-6-13-16-23; Línea 3: 2-7-12-11; Línea 4: 3-8-11-18-22; Línea 5: 4-9-10, ver Figura 7.5a) y las mismas son conectadas con las sub-rutas medias del segundo padre (Línea 1: 15-24; Línea 2: 16-25-33; Línea 3: 12-17-23;

Línea 4: 22-28; Línea 5: 20-19-21-29, ver Figura 7.5b). La última conexión, es realizada con las sub-rutas finales del primer padre (Línea 1: 25-26-34-35-36; Línea 2: 37; Línea 3: 27-32-38; Línea 4: 30-40; Línea 5: 31-39, ver Figura 7.5a). De esta manera se obtiene el primer hijo (Figura 7.6a).

De manera similar ocurre con el segundo hijo (Figura 7.6b). Las sub-rutas primeras del segundo padre son tomadas para formar la primera parte de las sub-rutas del segundo hijo (Línea 1: 1-5-14-16; Línea 2: 2-7-6-13; Línea 3: 3; Línea 4: 3-8-11-18-23; Línea 5: 4-8-9-10, ver Figura 7.5b). Luego, estas son conectadas con las sub-rutas medias del primer padre (Línea 1: 15-24; Línea 2: 25-33; Línea 3: 17-23; Línea 4: 28-29; Línea 5: 20-19-21-29, ver Figura 7.5a). Finalmente, éstas son conectadas con las últimas partes de las sub-rutas del segundo padre (Línea 1: 26-34-35-37; Línea 2: 37; Línea 3: 27-33-32-38; Línea 4: 30-40; Línea 5: 31-39, ver Figura 7.5b). En el caso de paradas repetidas en las rutas se resuelve de la misma manera descrita para el cruzamiento OX en el Capítulo 4 .

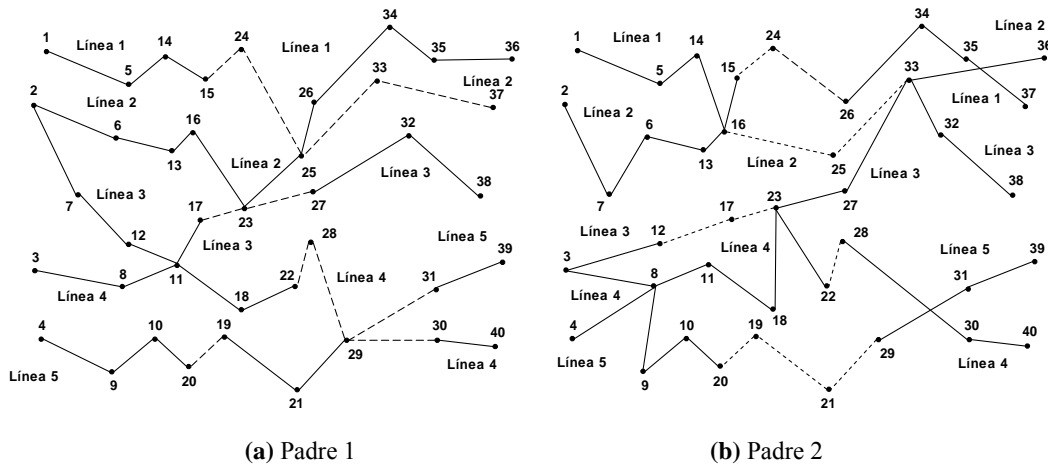


Figura 7.5 Padres seleccionados para cruzamiento

Utilizando las restricciones definidas en las ecuaciones (7.9), (7.10) y (7.11), el algoritmo detecta los individuos no factibles. En orden de mantener individuos factibles en la población, cuando el cruzamiento genera hijos no factibles, estos son descartados (se aplica una penalidad mortal) y nuevos cortes son realizados en los padres. Sin embargo, existe una

posibilidad en la que se repara el hijo obtenido si el mismo es no factible. Cuando una parada inicial o final queda desconectada, se busca la parada inicial o final que aparece repetida en más de una línea y esta es intercambiada por la parada desconectada. Del conjunto de individuos no factibles (un 1% de la población) que se generan, el 90% de los casos corresponde a esta situación, la cual es tratada sin volver a realizar los cortes.

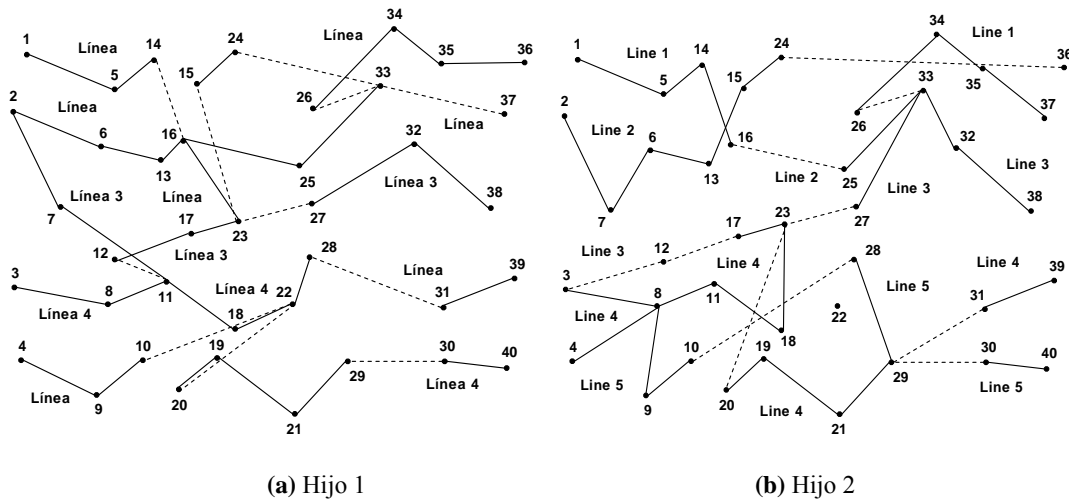


Figura 7.6 Hijos resultantes del crossover

7.4.4.2 Mutación

Para el caso de la mutación se desarrollaron en el marco de esta tesis doctoral dos operadores de mutación: mutación por **arco** y mutación por **parada**. Las mismas tienen por objetivo lograr la diversidad genética necesaria para conseguir una población de individuos que permita obtener buenas soluciones para un análisis posterior de los no dominados. La Figura 7.7 muestra un hijo al que se le va a aplicar el operador de mutación (Línea 1: 1-5-14-16-15-24-26-34-35-37; Línea 2: 2-7-6-13-16-25-33-36; Línea 3: 3-7-12-17-23-27-32-38; Línea 4: 3-8-11-18-22-28-30-40; Línea 5: 4-8-9-10-20-19-21-29-31-39).

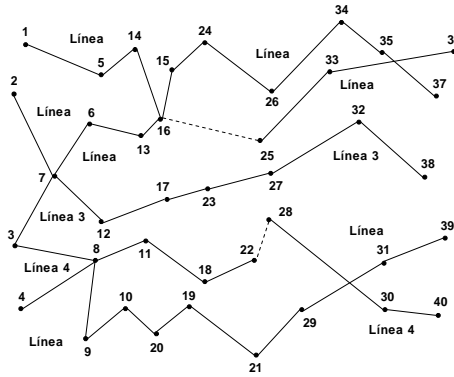
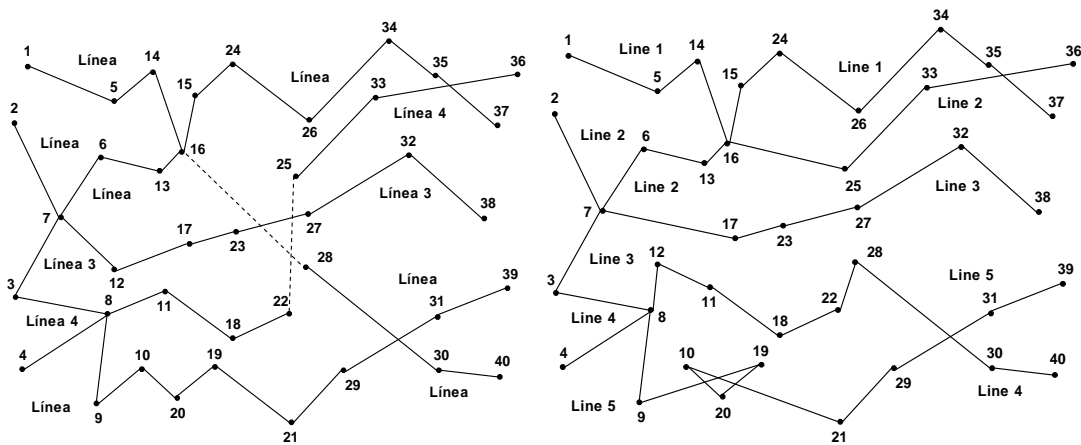


Figura 7.7 Red de tránsito a mutar.

La mutación por **arco** selecciona de manera aleatoria dos arcos para dos diferentes líneas e invierte su conexión (ver Figura 7.8a; Línea 1: 1-5-14-16-15-24-26-34-35-37; Línea 2: 2-7-6-13-16-28-30-40; Línea 3: 3-7-12-17-23-27-32-38; Línea 4: 3-8-11-18-22-25-33-36; Línea 5: 4-8-9-10-20-19-21-29-31-39).

Para el caso de mutación por **parada** existen dos alternativas: insertar aleatoriamente la parada seleccionada en otra línea o introducirla en una nueva posición en la misma línea (ver Figura 7.8b; Línea 1: 1-5-14-16-15-24-26-34-35-37; Línea 2: 2-7-6-13-16-25-33-36; Línea 3: 3-7-17-23-27-32-38; Línea 4: 3-8-12-11-18-22-28-30-40; Línea 5: 4-8-9-19-20-10-21-29-31-39).



(a) Individuo resultante de la mutación por arco (b) Individuo resultante de la mutación por parada

Figura 7.8 Diferentes operadores de mutación.

En ambos casos, si el individuo resultante es no factible, se descarta y el individuo original es mutado nuevamente hasta obtener otro factible. En este punto es importante tener en cuenta que siempre se podrán obtener individuos factibles; basta con seleccionar un individuo factible, tomar dos paradas sin trasbordos, e intercambiar las mismas. De manera similar si se toman dos arcos cuyos nodos no son trasbordos y se intercambian, el individuo resultante será factible si el original lo era. Es importante remarcar que al momento de identificar un individuo para ser mutado se elige de manera aleatoria la mutación a aplicar.

7.4.4.3 Selección

Para seleccionar el conjunto de individuos que forman parte de las sucesivas generaciones se tuvieron en cuenta las etapas de selección de los tres MOEAs que fueron descritos en el Capítulo 5 esta tesis doctoral. De esta forma, algoritmos evolutivos inspirados en NSGA-II (Deb *et al.*, 2000), SPEA2 (Zitzler *et al.*, 2002) e IBEA (Zitzler and Künzli, 2004) fueron implementados en el marco de esta tesis doctoral sobre la base de la plataforma PISA (Bleuler *et al.*, 2003). Esto último permitió el uso de las herramientas de estadística provistas por PISA para la evaluación de las alternativas de implementación que serán analizadas en este capítulo.

En la Figura 7.9 se observa la estructura del AHDM con la plataforma PISA utilizada en la etapa evolutiva. De esta forma se implementó el *variator* de PISA (ver Sección 5.4) y los selectores fueron tomados de PISA junto con las herramientas de análisis de calidad provistos por la misma. Realizando una visión disgregada del AHDM el método completo basado en NSGA-II se muestra en el Algoritmo 7.3, para el caso de SPEA2 en el Algoritmo 7.4 y para IBEA en el Algoritmo 7.5

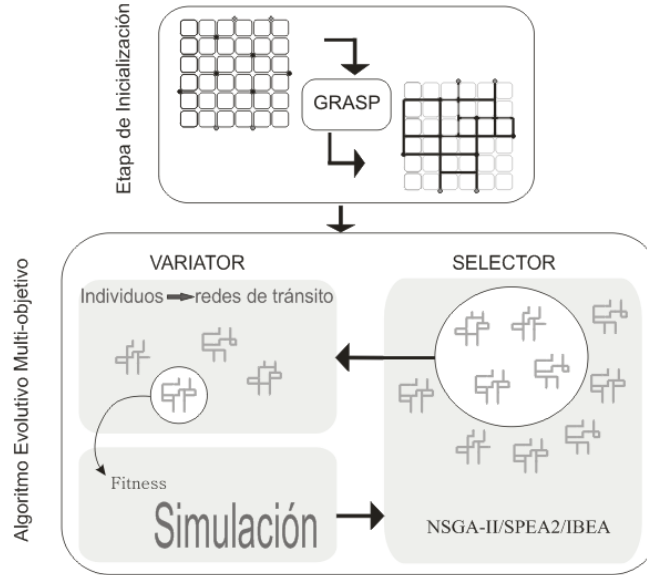


Figura 7.9 Estructura general de AHDM utilizando la herramienta PISA.

Algorithm AHDM = GRASP + NSGA – II * Simulación

Input: M: Mapa; B_S, B_F, B_I: conjuntos de paradas;
 [AF_L, T_L, D_L, K_L]: conjuntos of reales; [VST_{ijL}]: conjunto of reales;
 δ, η : real; g: entero; (g es el número de generaciones)
 N: entero; (N es el número de individuos en la población)

OutPut: P^g

Var distanciaEntreParadas: Lista de distancias; rutaEntreParadas: Lista de rutas;

0. Inicializar las distancias entre paradas y rutas entre todas las paradas(**GRASP**: B_S, B_F, B_I)
1. Generación de la población Inicial P₀
2. **for each** x ∈ P₀ **do** Simular x [AF_L, T_L, D_L, K_L, t^A_{ijL}, t^I_{ijL}, t^W_{ijL}] **end for**
3. Evaluar FO y FU para cada individuo x ∈ P₀
4. Asignar un valor de ranking basado en Pareto
5. Padres := Seleccionar Padres de la Población
6. Q₀ := Cruzar(Padres)
7. Mutar(Q₀)
8. **for** i := 0 **to** g-1 **do**
9. **for each** x ∈ Q_i **do**
10. Simular x [AF_L, T_L, D_L, K_L, t^A_{ijL}, t^I_{ijL}, t^W_{ijL}]
11. **end for**
12. Evaluar FO y FU para cada individuo x ∈ Q_i
13. R_i := P_i ∪ Q_i
14. **for each** individuo x en R_i **do**
15. Asignar un valor de ranking basado en Pareto
16. Generar conjuntos de vectores no dominados
17. **end for**
17. Seleccionar los puntos elitistas teniendo en cuenta el ranking y eliminarlos
18. Crear la siguiente generación P_{i+1}
19. Padres := Seleccionar Padres de la Población
20. Q₀ := Cruzar(Padres)
21. Mutar(Q₀)
22. **end for**
23. **end Algorithm**

Algoritmo 7.3 Pseudo-código del AHDM para el evolutivo inspirado en NSGA-II

Algorithm AHDM = GRASP + SPEA2 * Simulation

Input: M: Mapa; B_S, B_F, B_I: conjuntos de paradas;
 [AF_L, T_L, D_L, K_L]: conjuntos of reales; [VST_{ijL}]: conjunto of reales;
 δ, η : real; g: entero; (g es el número de generaciones)
 N: entero; (N es el número de individuos en la población)

OutPut: P^g

24. Inicializar las distancias entre paradas y rutas entre todas las paradas(**GRASP**: B_S, B_F, B_I)
25. Generación de la población Inicial P₀
0. Crear un conjunto externo vacío N'
1. **for each** x ∈ P₀ **do** Simular x [AF_L, T_L, D_L, K_L, t^A_{ijL}, t^I_{ijL}, t^W_{ijL}] **end for**
2. Evaluar FO y FU para cada individuo x ∈ P₀

```

3. Copiar todos los individuos no dominados en  $P_0$  y  $N'$  a  $N'$ 
4. Utilizar el método de truncado para remover los elementos de  $N'$  cuando la capacidad del
   archivo es superada
5. if capacidad( $N'$ )>limiteCapacidad then
6.     utilizar los elementos dominados en  $P_0$  para rellenar  $N'$ 
7. end if
8. Padres := Seleccionar Padres de  $P_0$ 
9.  $Q_0$ := Cruzar(Padres)
10. Mutar( $Q_0$ )
11. for  $i := 0$  to  $g-1$  do
12.     for each  $x \in Q_i$  do Simular  $x$  [ $AF_L, T_L, D_L, K_L, t_{ijL}^A, t_{ijL}^J, t_{ijL}^W$ ] end for
13.     Evaluar FO y FU para cada individuo  $x \in Q_i$ 
14.     Copiar todos los individuos no dominados en  $Q_i$  y  $N'$  a  $N'$ 
15.     Utilizar el método de truncado para remover los elementos de  $N'$  cuando la capacidad del
   archivo es superada
16.     if capacidad( $N'$ )>limiteCapacidad then
17.         utilizar los individuos dominados en  $Q_i$  para completar  $N'$ 
18.     end if
19. Padres:= Seleccionar Padres ( $Q_i+N'$ )
20.  $Q_i$ := Cruzar (Padres)
21. Mutar( $Q_i$ )
22. end for
23. end Algorithm

```

Algoritmo 7.4 Pseudo-código del AHDM para la implementación de SPEA2

Algorithm AHDM = GRASP + IBEA * Simulation

Input: M: Mapa; B_S, B_F, B_I : conjuntos de paradas;

$[AF_L, T_L, D_L, K_L]$: conjuntos of reales; $[VST_{ijL}]$: conjunto of reales;

δ, η : real; g : entero; (g es el número de generaciones)

N : entero; (N es el número de individuos en la población)

T : entero; (el tamaño máximo de P)

OutPut: P_g

0. Inicializar las distancias entre paradas y rutas entre todas las paradas(**GRASP**: B_S, B_F, B_I)

26. Generación de la población Inicial P_0

1. **for each** $x \in P_0$ **do** Simular x [$AF_L, T_L, D_L, K_L, t_{ijL}^A, t_{ijL}^J, t_{ijL}^W$] **end for**

2. **for** $i := 0$ to $g-1$ **do**

3. Evaluar FO y FU para cada individuo $x \in P$

4. **for each** $x \in P_0$ **do** Simular x [$AF_L, T_L, D_L, K_L, t_{ijL}^A, t_{ijL}^J, t_{ijL}^W$] **end for**

5. Calcular el *fitness* para cada $x \in P$ como $F(x) = \sum_{x' \in P_0 \setminus x} -e^{-f((x'),(x))/k}$

6. Iterar siguiendo los siguientes paso hasta que el tamaño de la población P no exceda un tamaño T predefinido

7. Elegir un $x' \in P$ con el menor valor de *fitness*

8. remover a x' de la población

9. actualizar el valor de *fitness* para los individuos restantes

24. Padres:= Seleccionar Padres (Q_i+N')

25. Q_i := Cruzar (Padres)

10. Mutar(Q_i)

11. **end for**

12. **end Algorithm**

Algoritmo 7.5 Pseudo-código del AHDM para la implementación de IBEA

7.5 ANÁLISIS DE LA COMPLEJIDAD DE LOS ALGORITMOS DESARROLLADOS

La complejidad computacional de los algoritmos desarrollados está compuesta por la complejidad de las etapas que los componen. Así un análisis paso a paso de cada etapa debe ser realizado para llevar a cabo un estudio de su orden de complejidad.

El algoritmo de inicialización (GRASP) es ejecutado únicamente al principio del AHDM con un tiempo de ejecución que es de orden $O(N^3)$ (Feo, 1995; Resende, 1999).

En la etapa evolutiva debemos tener en cuenta el MOEA en el que está inspirada y el tiempo de ejecución del procedimiento de simulación de los individuos.

Para el caso de NSGA-II, ha sido demostrado en investigaciones previas que se necesita un tiempo del orden de $O(g*N^2)$ teniendo en cuenta una población de tamaño de $2*N$ para el criterio de dominación. El cálculo de las distancias (Deb, 2002) que requiere el NSGA-II necesita no menos de $O(g*N*\log N)$ por lo que el procedimiento NSGA-II completo es del orden de $O(g*N^2)$.

La parte más costosa del algoritmo basado en SPEA2 (Zitzler, 2002) es el procedimiento de corte utilizado para preservar la diversidad, el cual posee un tiempo de ejecución de orden $O((N+N')^2*\log(N+N'))$, donde N es el tamaño de la población y N' es el tamaño del archivo donde guarda a los individuos mejores.

En el caso de IBEA (Zitzler y Künzli, 2004) tiene un orden de complejidad computacional de orden de $O(N^2)$.

Finalmente, la simulación realizada en el cálculo de la función de fitness de cada individuo toma un tiempo de orden de $O(N*M)$, considerando que esta etapa se realiza sobre los N individuos con M líneas y ω frecuencias predefinidas. Sin embargo, ω es un número constante por lo cual se descarta del análisis de orden.

Resumen del capítulo

En este capítulo se presentaron las principales características del algoritmo híbrido dinámico multi-objetivo para el tratamiento del TNDSP de una manera realista. El algoritmo posee dos etapas: una de inicialización y otra etapa evolutiva que incluye un procedimiento de simulación para obtener los valores de las variables contenidas en la función de aptitud.

Además se desarrollaron en el marco de esta tesis un nuevo operador de cruzamiento para individuos con cromosomas de múltiples cadenas –las líneas de la red de tránsito- y dos novedosos operadores de mutación (por arco y por parada). Además se trabajó con tres de los métodos de selección más utilizados en la literatura - NSGA-II, SPEA2 e IBEA - para obtener los individuos de las generaciones subsiguientes.

Capítulo 8 Experimentos

Con el objetivo de evaluar la calidad del AHDM y establecer su desempeño en la resolución del TNDSP y además, comparar y analizar las diferentes implementaciones incluidas dentro de la etapa evolutiva, se llevaron a cabo distintos experimentos. Los casos de estudio poseen características interesantes desde el punto de vista de su topología y distribución urbana. Uno de ellos es un modelo real - de alto impacto en la literatura referida a TNP - el cual se trata de una ciudad de Suiza (Mandl, 1979).

8.1 NOTACIÓN URBANA

Teniendo en cuenta la variedad de casos de estudio que se pueden armar y las características propias de las ciudades es necesario introducir una notación común que servirá al lector para entender con facilidad cada uno de los ejemplos e identificar rápidamente las características propias de cada modelo.

Dentro de un mapa urbano las tres posibles clasificaciones de las paradas se notarán con colores especiales: iniciales (B_S - verde), intermedias (B_I - rojo) y finales (B_F - azul). Los espacios verdes se notarán con verde oscuro mientras que los establecimientos educacionales con la letra E (con color azul). Por último, las direcciones de las calles se mostrarán con flechas.

Para el caso particular de la matriz OD (origen-destino) se introduce otra tabla que contiene los datos de las distribuciones que modelan el arribo de los clientes. Por lo tanto la matriz OD siempre se mostrará junto con otra tabla indicando tiempo y frecuencia horaria.

A continuación introduciremos los casos de estudio que se utilizan en este capítulo para realizar las corridas y el posterior análisis del AHDM.

8.2 PRESENTACIÓN CASO DE ESTUDIO 1: UN BARRIO RESIDENCIAL

La primera ciudad hipotética creada para estudiar el AHDM posee 100 nodos, 2 paradas iniciales (B_S), 6 intermedias (B_I) y 2 finales (B_F). Se trata de un barrio dentro de lo que podría ser una ciudad más completa. La Figura 8.1 muestra la disposición de los nodos dentro del mapa, una identificación de los mismos de forma numerada y un esquema de las direcciones de las calles. El barrio residencial posee una plaza, una escuela primaria y una secundaria. La Tabla 8.1 muestra la distancia entre los nodos del mapa.

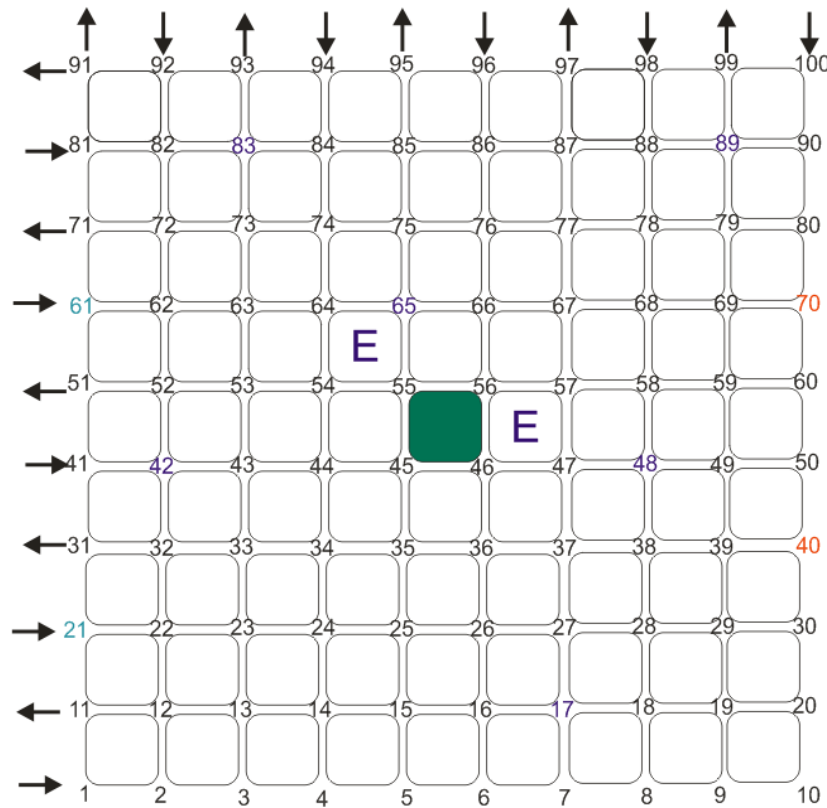


Figura 8.1 Ubicación de los nodos en el barrio residencial

Por último, la matriz de origen-destino está armada según datos generados para una frecuencia horaria en particular, de 6:00 a.m. hasta 10:00 a.m. La Tabla 8.2 contiene los datos de la matriz OD. Teniendo en cuenta que se desea realizar un análisis temporal de los viajes

de los usuarios del servicio, se utilizó una frecuencia de arribo de los clientes a las paradas siguiendo distintas distribuciones de probabilidad exponenciales (Banks et al., 2000); los valores del parámetro α de la distribución se muestran en la Tabla 8.3. Es importante remarcar que para este caso de estudio particular la Tabla 8.2 no es simétrica puesto que no comprende toda una jornada sino únicamente la mañana. Esta es una variante a las matrices OD utilizadas hasta el momento en la literatura puesto que las mismas sólo consideran distribuciones uniformes.

Tabla 8.1 Distancias del mapa de cada nodo desde el nodo 1

Nodo	Ubicación en el mapa		Nodo	Ubicación en el mapa		Nodo	Ubicación en el mapa		Nodo	Ubicación en el mapa	
1	0	0	26	500	200	51	0	500	76	500	700
2	100	0	27	600	200	52	100	500	77	600	700
3	200	0	28	700	200	53	200	500	78	700	700
4	300	0	29	800	200	54	300	500	79	800	700
5	400	0	30	900	200	55	400	500	80	900	700
6	500	0	31	0	300	56	500	500	81	0	800
7	600	0	32	100	300	57	600	500	82	100	800
8	700	0	33	200	300	58	700	500	83	200	800
9	800	0	34	300	300	59	800	500	84	300	800
10	900	0	35	400	300	60	900	500	85	400	800
11	0	100	36	500	300	61	0	600	86	500	800
12	100	100	37	600	300	62	100	600	87	600	800
13	200	100	38	700	300	63	200	600	88	700	800
14	300	100	39	800	300	64	300	600	89	800	800
15	400	100	40	900	300	65	400	600	90	900	800
16	500	100	41	0	400	66	500	600	91	0	900
17	600	100	42	100	400	67	600	600	92	100	900
18	700	100	43	200	400	68	700	600	93	200	900
19	800	100	44	300	400	69	800	600	94	300	900
20	900	100	45	400	400	70	900	600	95	400	900
21	0	200	46	500	400	71	0	700	96	500	900
22	100	200	47	600	400	72	100	700	97	600	900
23	200	200	48	700	400	73	200	700	98	700	900
24	300	200	49	800	400	74	300	700	99	800	900
25	400	200	50	900	400	75	400	700	100	900	900

Tabla 8.2 Matriz OD para la Figura 8.1 (viajes de la mañana)

Matriz origen-destino de 6 a.m. a 10 a.m.						
Parada	17	42	48	65	83	89
17	200	200	80	200	80	
42	0	0	400	200	80	200
48	100	800	0	80	200	80
65	0	0	0	0	80	120
83	800	0	0	100	0	400
89	100	400	100	0	0	0

Tabla 8.3 Matriz OD: Valores del parámetro α de la distribución de probabilidad exponencial que modela los arribos (viajes de la mañana)

Parámetro α para los arribos de 6 a.m. a 10 a.m.						
Parada	17	42	48	65	83	89
17	0.000000	0.833333	0.833333	0.333333	0.833333	0.333333
42	0.000000	0.000000	1.666667	0.833333	0.333333	0.833333
48	0.416667	3.333333	0.000000	0.333333	0.833333	0.333333
65	0.000000	0.000000	0.000000	0.000000	0.333333	0.500000
83	3.333333	0.000000	0.000000	0.416667	0.000000	1.666667
89	0.416667	1.666667	0.416667	0.000000	0.000000	0.000000

8.3 PRESENTACIÓN CASO DE ESTUDIO 2: UNA CIUDAD DE SUIZA

Mandl (1979) trata la problemática de una ciudad Suiza, la cual consiste en un total de 15 paradas y una demanda de 15570 viajes diarios. Este modelo es uno de los casos de estudio más frecuentemente citados en la literatura sobre redes de tránsito. La disponibilidad de estos trabajos previos permitió evaluar el desempeño del AHDM frente a los métodos publicados anteriormente.

8.3.1 Diagrama de la topología de las paradas

A continuación la Figura 8.2 muestra el mapa de la ciudad Suiza. El número en el arco que une a las diferentes paradas corresponde con el tiempo en minutos de viaje del nodo i al nodo j . La Matriz OD para este caso se muestra en la Tabla 8.4, la cual es simétrica.

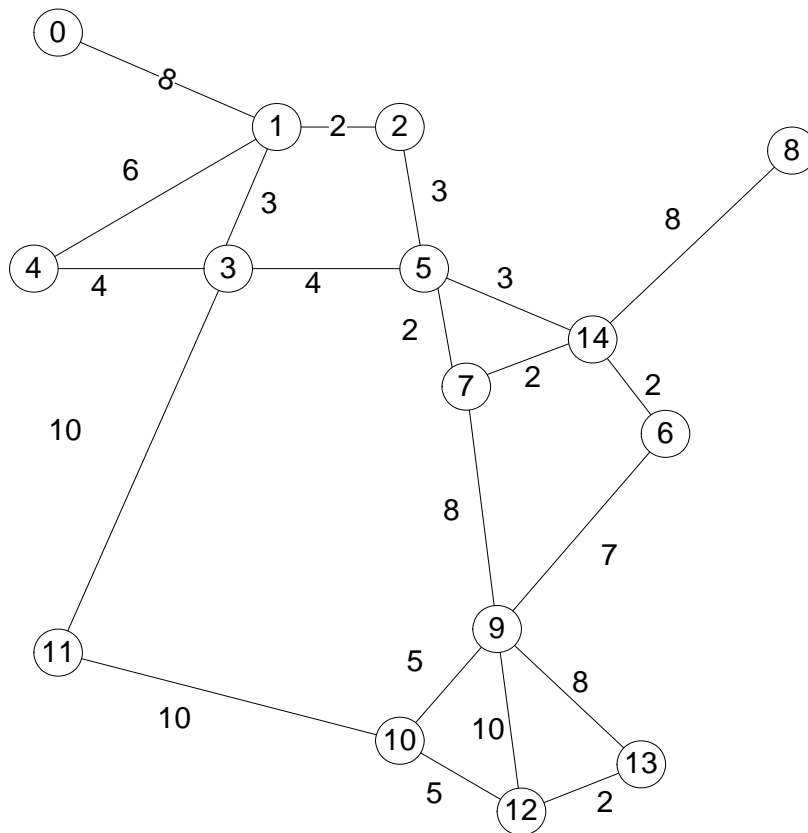


Figura 8.2 Mapa de una ciudad Suiza (Mandl, 1980)

Tabla 8.4 Matriz OD de la ciudad de Mandl (1979)

Parada	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	400	200	60	80	150	75	75	30	160	30	25	35	0	0
1	400	0	50	120	20	180	90	90	15	130	20	10	10	5	0
2	200	50	0	40	60	180	90	90	15	45	20	10	10	5	0
3	60	120	40	0	50	100	50	50	15	240	40	25	10	5	0
4	80	20	60	50	0	50	25	25	10	120	20	15	5	0	0
5	150	180	180	100	50	0	100	100	30	880	60	15	15	10	0
6	75	90	90	50	25	100	0	50	15	440	35	10	10	5	0
7	75	90	90	50	25	100	50	0	15	440	35	10	10	5	0
8	30	15	15	15	10	30	15	15	0	140	20	5	0	0	0
9	160	130	45	240	120	880	440	440	140	0	600	250	500	200	0
10	30	20	20	40	20	60	35	35	20	600	0	75	95	15	0
11	25	10	10	25	15	15	10	10	5	250	75	0	70	0	0
12	35	10	10	10	5	15	10	10	0	500	95	70	0	45	0
13	0	5	5	5	0	10	5	5	0	200	15	0	45	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.3.2 Limitaciones y carencias del caso de estudio

Aunque este caso es uno de los más citados en la literatura carece de detalle y deja ciertos aspectos del problema sin resolución o ambiguos. Los mismos se pueden resumir como sigue:

- No brinda las orientaciones de las calles. Pareciera que todas las calles son doble mano.
- El tiempo de viaje entre una parada y otra es igual tanto para la ida como para la vuelta.
- El tiempo de viaje es fijo entre paradas sin tener en cuenta congestionamientos, horas picos, poco tránsito, pocos pasajeros, etc.
- Existen paradas desconectadas del mapa que sólo se pueden acceder pasando por otra parada.
- Carece de una discriminación horaria de la demanda. La matriz OD sólo muestra el promedio de viajes entre una parada y otra. No tiene en cuenta horas pico o fluctuaciones provenientes de actividades humanas bien marcadas como por ejemplo ir a trabajar, ir al colegio, horarios nocturnos, etc. Por este motivo se introduce la Tabla 8.5 para generar los arribos siguiendo una distribución exponencial con los α provistos.

En las secciones siguientes se presentan los resultados obtenidos por el algoritmo híbrido desarrollado. Los experimentos se realizaron en varias etapas con el objeto de obtener resultados relacionados con los diferentes MOEAs implementados para la etapa evolutiva. Además se estudió el número de generaciones necesarias para converger a soluciones buenas desde el punto de vista de las entidades modeladas por las funciones objetivo.

Tabla 8.5 Matriz OD con los parámetros α de las distribuciones de probabilidad para los arribos de los clientes (jornada completa)

Par.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0	0.8333	0.4167	0.125	0.16666	0.3125	0.1563	0.15625	0.063	0.333	0.063	0.05	0.073	0	0
1	0.8333	0	0.1042	0.25	0.04167	0.375	0.1875	0.1875	0.031	0.271	0.042	0.02	0.021	0.0104	0
2	0.4167	0.1042	0	0.083	0.125	0.375	0.1875	0.1875	0.031	0.094	0.042	0.02	0.021	0.0104	0
3	0.125	0.25	0.0833	0	0.10417	0.20833	0.1042	0.104167	0.031	0.5	0.083	0.05	0.021	0.0104	0
4	0.1667	0.0417	0.125	0.104	0	0.10417	0.0521	0.052083	0.021	0.25	0.042	0.03	0.01	0	0
5	0.3125	0.375	0.375	0.208	0.10417	0	0.2083	0.208333	0.063	1.183	0.125	0.03	0.031	0.2083	0
6	0.1563	0.1875	0.1875	0.104	0.05208	0.20833	0	0.104167	0.031	0.917	0.073	0.21	0.208	0.0104	0
7	0.1563	0.1875	0.1875	0.104	0.05208	0.20833	0.1042	0	0.031	0.917	0.073	0.21	0.208	0.0104	0
8	0.0625	0.0313	0.0313	0.031	0.02083	0.0625	0.0313	0.03125	0	0.292	0.042	0.01	0	0	0
9	0.3333	0.2708	0.0938	0.5	0.25	1.18333	0.9167	0.916667	0.292	0	1.25	0.52	1.042	0.4167	0
10	0.0625	0.0417	0.0417	0.083	0.04167	0.125	0.0729	0.072917	0.042	1.25	0	0.16	0.198	0.0313	0
11	0.0521	0.0208	0.0208	0.052	0.03125	0.03125	0.2083	0.208333	0.01	0.521	0.156	0	0.146	0	0
12	0.0729	0.0208	0.0208	0.021	0.01042	0.03125	0.2083	0.208333	0	1.042	0.198	0.15	0	0.0938	0
13	0	0.0104	0.0104	0.01	0	0.20833	0.0104	0.010417	0	0.417	0.031	0	0.094	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

8.4 ANÁLISIS DE LA CONVERGENCIA DEL AHDM

Inicialmente, para analizar la convergencia del método propuesto, se llevó a cabo un experimento en dos etapas sobre el caso de estudio del barrio residencial presentado en la sección 8.2. En la primera fase, todos los parámetros se mantuvieron fijos excepto por el número de generaciones (g) que varió entre 50, 100, 150 y 200. El resto de los parámetros tomaron los siguientes valores: para GRASP: $iterMax = 1$ y $\beta = 0.4$; para la etapa evolutiva: $\alpha=100$, $CrossoverProb = 0.9$, $MutationProb = 0.2$. Se utilizó el NSGA-II para la etapa evolutiva y para el procedimiento de simulación la frecuencia de los móviles se estableció en 10 minutos.

Para la segunda etapa, el tamaño de las generaciones fue establecido teniendo en cuenta los resultados de la primera fase. El objetivo fue testear el algoritmo híbrido en diferentes escenarios variando las frecuencias de los buses en 10, 20 y 40 minutos como muestra la Tabla 8.6 y determinar la frecuencia más adecuada.

Tabla 8.6 Frecuencia de los buses para cada escenario

	Simulación
Escenarios	Frecuencia de los Buses
1	10
2	20
3	40

En este punto es importante remarcar que el parámetro $iterMax$ es igual a 1 porque una iteración simple de GRASP es suficiente para obtener resultados favorables con un

tiempo de cómputo pequeño para el caso del barrio residencial. Sin embargo, para casos de mayor envergadura, varias iteraciones del método son necesarias.

8.4.1 Experimentación: fase 1

Para todos los casos de la primera fase, el algoritmo se testeó a través de diez corridas independientes para cada cantidad de generaciones. Se almacenaron los individuos de la última corrida. Luego para cada población final se analizó su *fitness* a fin de construir una frontera de individuos no dominados. Las fronteras obtenidas para cada conjunto de individuos se muestran en la Figura 8.3 según los objetivos definidos en el Capítulo 7 .

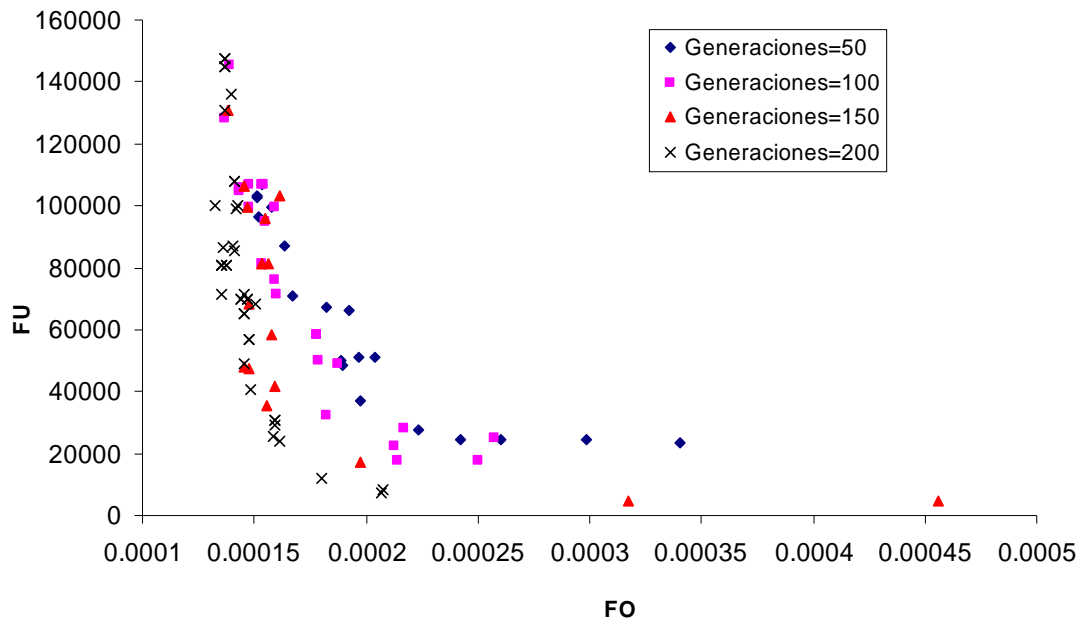


Figura 8.3 Individuos no dominados obtenidos con los diferentes tamaños de generaciones

Como se puede observar en el gráfico y como era de esperar, los individuos obtenidos tras 200 generaciones son mejores que los encontrados con tan sólo 50 generaciones. Sin embargo, entre 150 y 200 generaciones no existe una mejora significativa. Más aún, para la función FO se percibe una convergencia. De esta manera se estableció que no eran necesarias más de 200 generaciones.

8.4.2 Experimentación: Fase 2

Durante la segunda fase, veinte corridas con 200 generaciones fueron realizadas para cada escenario. Es importante remarcar que un promedio de 20 soluciones no dominadas se obtienen en cada ejecución de la etapa evolutiva del AHDM.

Para decidir que solución era mejor a otra se utilizó el criterio de las entidades (operarios vs. usuarios). Con el objeto de ilustrar algunos ejemplos antes de realizar el análisis, fueron seleccionadas cuatro soluciones del conjunto de individuos no dominados, dos que favorecen al usuario y otras dos que benefician al operario. La Tabla 8.7 ilustra en detalle las líneas L_0 y L_1 para cada una de las soluciones seleccionadas.

Finalmente, la Tabla 8.8 muestra las estadísticas calculadas a partir de las soluciones cercanas a la frontera de Pareto para los escenarios de frecuencias de los buses definidas en la Tabla 8.6. La comparación está basada en las variables que caracterizan a la calidad de la red de tránsito según lo expuesto en el Capítulo 1 de esta tesis. El porcentaje de trasbordos es calculado a partir del total de viajes que se realizaron (primeras tres columnas de la Tabla 8.8). El valor asociado a la demanda insatisfecha se muestra en la fila 4 de la Tabla 8.8 y representa la cantidad de usuarios que no llegaron a destino durante la simulación. El tiempo de espera de los usuarios (fila 6 de la Tabla 8.8) es la suma del tiempo que esperaron en la parada hasta que finalmente el móvil arribó a la misma. El costo de operación (fila 7 de la Tabla 8.8) es el promedio de valores del objetivo FO para las soluciones no dominadas.

Tabla 8.7 Ejemplos de cuatro soluciones no dominadas para los escenarios de la Tabla 8.6

	Escenarios	1	2	3
Rutas de las redes de tránsito	1	Línea 0: 21-42-83-48-40	Línea 0: 21-83-16-42-65-48-40	Línea 0: 21-83-16-48-65-42-89-40
		Línea 1: 61-65-16-89-48-70	Línea 1: 61-89-16-70	Línea 1: 61-65-70
	2	Línea 0: 21-83-65-42-16-48-89-40	Línea 0: 21-42-48-16-40	Línea 0: 21-83-42-16-48-65-89-40
		Línea 1: 61-48-70	Línea 1: 61-65-83-89-48-70	Línea 1: 61-16-70
	3	Línea 0: 21-65-83-42-16-48-40	Línea 0: 21-65-83-42-16-48-40	Línea 0: 21-83-16-48-42-65-89-40
		Línea 1: 61-89-16-70	Línea 1: 61-89-16-70	Línea 1: 61-83-70
	4	Línea 0: 21-83-42-83-48-40	Línea 0: 21-16-48-40	Línea 0: 21-83-65-42-16-48-89-40
		Línea 1: 61-65-16-89-48-70	Línea 1: 61-83-65-42-89-48-70	Línea 1: 61-83-70

Tabla 8.8 Resultados estadísticos

Escenarios		1	2	3
Resultados Estadísticos	<i>% demanda con 0-trasbordos promedio</i>	68.680	70.505	46.800
	<i>% demanda con 1-trasbordos promedio</i>	31.300	29.245	13.200
	<i>% demanda con +1-trasbordos promedio</i>	0	0	0
	<i>% demanda insatisfecha promedio</i>	0	0	40
	<i>Promedio de los tamaños de las rutas</i>	Línea 0: 5533	Línea 0: 3200	Línea 0: 8900
		Línea 1: 3566	Línea 1: 3800	Línea 1: 2500
	<i>tiempo de espera promedio de los usuarios de soluciones no dominadas</i>	2 min 30 seg	12 min 28 seg	25 min 54 seg
	<i>Costo operativo promedio de las soluciones no dominadas</i>	5932.03 u. m.	3983.24 u.m	3936.67 u.m.

Como se puede observar, los mejores resultados se obtienen en el escenario 2 (frecuencia de 20 minutos). Es interesante además notar que el escenario 3 (40 minutos) puede ser descartado del análisis ya que los resultados mostraron que el 40% de los pasajeros no llegaban a destino al concluir la simulación. De esta manera, uno de los objetivos principales del TNDSP no era alcanzado en ese escenario. Por otra parte, en el escenario 2 las líneas están balanceadas en cuanto al tamaño de las rutas y los usuarios necesitan menos trasbordos para llegar a su destino. Más aún, se puede ver en la fila 1 de la Tabla 8.8 que existen más usuarios que no necesitan trasbordo. Por último y como era de esperarse, el escenario 2 (frecuencia de 20 minutos), posee un tiempo de espera más grande que el del escenario 1 (10 minutos). A pesar de esto, el balance entre los intereses del operador y del usuario es mejor en el escenario 2.

8.5 ELECCIÓN DEL MOEA PARA EL AHDM

Habiendo definido una cantidad adecuada de generaciones para el algoritmo evolutivo y la frecuencia de los buses para la etapa de la simulación, se trabajó con la herramienta PISA (Bleuler *et al.*, 2003) descrita en el Capítulo 4 y se comparó el desempeño global de la etapa evolutiva intercambiando entre tres selectores: IBEA, NSGA-II y SPEA2.

Para este experimento $\alpha=100$, $\mu=50$ y $\lambda=50$ son los parámetros de PISA que corresponden con el número de individuos (100), el número de padres (50) y de descendientes de la población (50) de una generación. Estos valores fueron elegidos teniendo

en cuenta resultados las corridas preliminares de la Sección 8.4. En el caso del selector IBEA se eligió el indicador *epsilon* aditivo y el resto de los valores de los parámetros se mantuvieron con su valor predefinido en PISA.

8.5.1 Análisis utilizando métricas

Las métricas aplicadas en la evaluación de los MOEAs incluyen el ranking de dominancias, el Indicador Hipervolumen Γ_H , la versión multiplicativa del Indicador *epsilon* unario (I^I_E) y el Indicador $R2$ (I^I_{R2}).

El propósito de estos indicadores de calidad es caracterizar las diferencias entre los conjuntos de aproximación cuando existe una similitud significativa en el ranking de dominancia (Knowles *et al.*, 2005). Para los parámetros de los indicadores se mantuvieron los valores originales provistos por PISA. De esta manera, los objetivos fueron normalizados en el intervalo [1..2].

Los selectores a evaluar se tomaron de a pares a fin de aplicar el ranking por suma de Mann-Witney (Conover, 1999). Ninguno de los resultados fue estadísticamente significativo con un nivel de confianza de $\alpha = 0.05$. Esto demostró que ninguno de los MOEAs generó una mejor aproximación con respecto a otro.

Luego se aplicaron los indicadores unarios utilizando los conjuntos de aproximación normalizados y el conjunto de referencia generado por PISA. La distribución de los conjuntos de aproximación resultantes puede ser visualizada gráficamente en la Figura 8.4.

Un test no paramétrico fue aplicado a los conjuntos de aproximación transformados con el objeto de obtener conclusiones válidas sobre la calidad de los métodos de optimización. Para nuestro análisis se utilizó el test de Fisher (Zitzler y Künzli, 2004) con nivel de confianza de $\alpha = 0.05$. La Tabla 8.9 muestra que no existe diferencia significativa entre los resultados obtenidos para los diferentes algoritmos e indicadores aplicados al TNDSP.

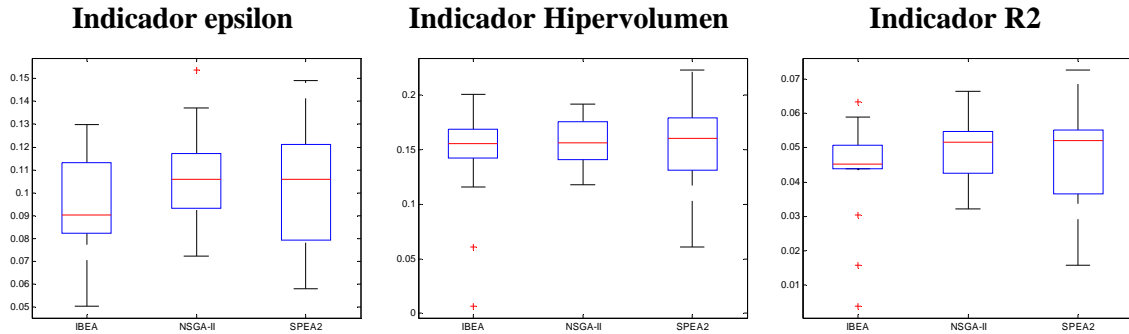


Figura 8.4 Diagramas de caja (boxplots) para cada indicador unario

Tabla 8.9 Test de Fisher aplicado a los indicadores de calidad

Indicador Epsilon Multiplicativo

	IBEA	NSGA-II	SPEA2
IBEA	-	0.059980	0.089980
NSGA-II	0.937920	-	0.525180
SPEA2	0.912100	0.469180	-

Indicador Hipervolumen

	IBEA	NSGA-II	SPEA2
IBEA	-	0.147820	0.245460
NSGA-II	0.851750	-	0.627420
SPEA2	0.758040	0.368880	-

Indicador R2

	IBEA	NSGA-II	SPEA2
IBEA	-	0.097320	0.168560
NSGA-II	0.902720	-	0.574120
SPEA2	0.833560	0.423320	-

Teniendo en cuenta que fueron aplicados varios test sin ningún resultado que se inclinara por uno u otro de los MOEAs, la última opción fue evaluar la superficie generada por los MOEAs en cada corrida.

La Figura 8.5 (a-b-c) muestra que no existen diferencias considerables entre los 4 mejores frentes de Pareto generados por los diferentes MOEAs. Sin embargo, la gráfica para SPEA2 posee claramente el conjunto de soluciones más convenientes y el espacio de soluciones obtenido en cada corrida es más disperso que en el resto de los algoritmos. Debido a esto y dado que los DMs serán los encargados de dar la última opinión sobre una solución u otra, el método que provee un conjunto de soluciones amplio es el más conveniente para resolver el TNDSP. Por este motivo, SPEA2 se prefiere con respecto a los otros MOEAs.

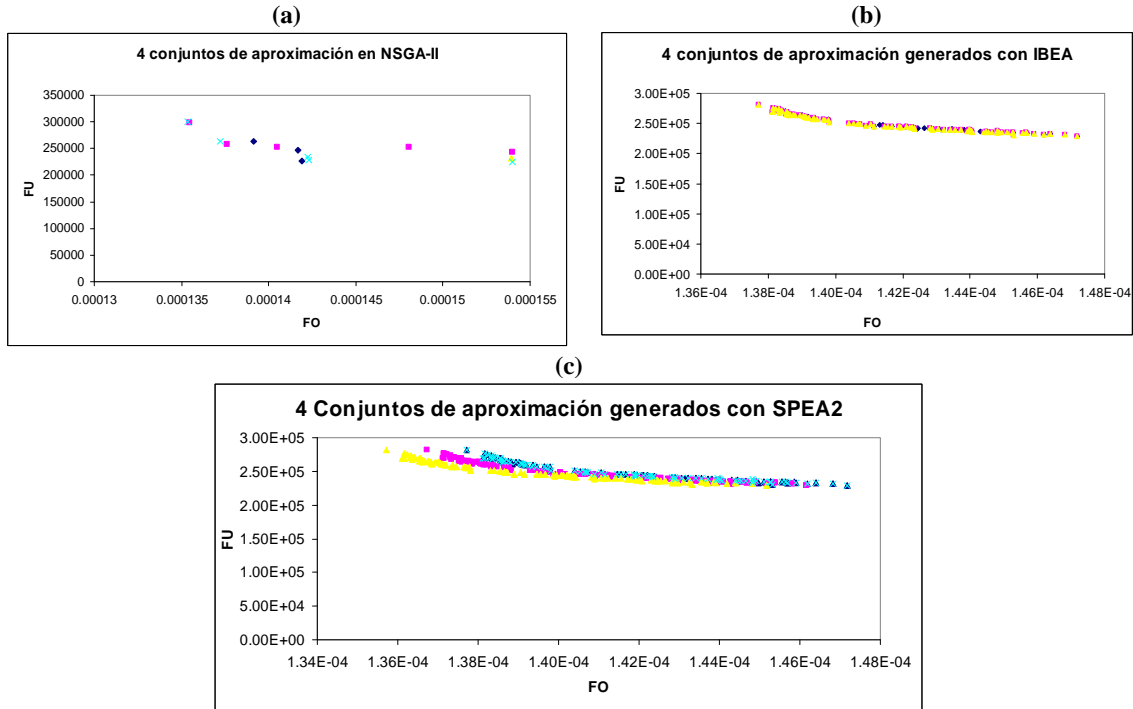


Figura 8.5 Gráfica de 4 conjuntos de aproximación generados con IBEA, NSGA-II y SPEA2

8.6 AHDM APLICADO AL CASO DE MANDL

En esta sección el caso de estudio de Mandl es presentado teniendo en cuenta los datos obtenidos en la Sección 8.5. Los experimentos para este caso en particular se realizaron sobre el mapa original provisto por Mandl presentado en la Sección 8.3 y las modificaciones planteadas en dicha sección para un tratamiento más realista del caso de estudio. El objetivo de este experimento fue utilizar el AHDM para un caso real y compararlo con otras técnicas existentes.

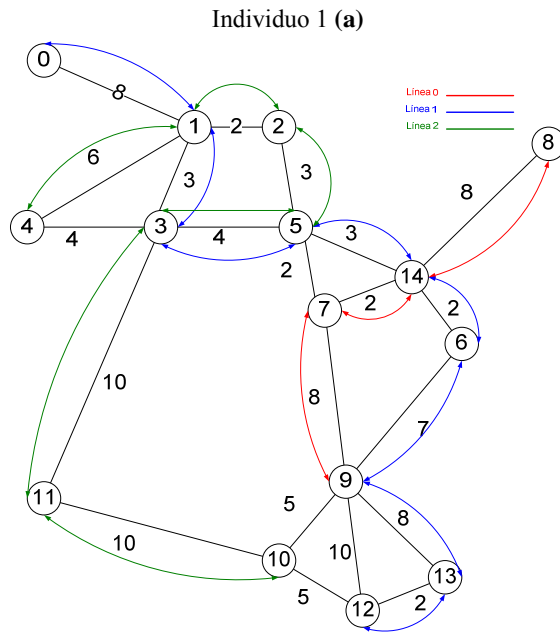
Tabla 8.10 Resultados estadísticos de los diferentes algoritmos para resolución del problema de Mandl

Resultados Estadísticos	AHDM	Baaj & Mahmassani	Mandl
% 0-trasbordos	83.10	80.99	69.94
% 1-trasbordo	16.90	19.01	29.93
% 2-trasbordos	0	0	0.13
% demanda insatisfecha	0	0	0

Los parámetros fijados fueron: para la etapa evolutiva $\alpha = 100$, $CrossoverProb = 0.9$, $MutationProb = 0.2$; $Generations = 200$ y para la simulación la frecuencia de los móviles se estableció cada 20 minutos en base a corridas previas. Se realizaron veinte corridas con estos parámetros.

La Tabla 8.10 muestra una comparación entre el método propuesto por Mandl (1979), el propuesto por Baaj y Mahmassani (1991) y el AHDM. La comparación está basada en las variables que caracterizan la calidad de una configuración de red de tránsito público (Desaulniers, 2007). Como se puede observar los mejores resultados se obtuvieron con el algoritmo presentado en esta tesis: se aumentó el porcentaje de viajes que no requieren trasbordos, se redujo el porcentaje de viajes con uno o más trasbordos y al mismo tiempo se cumplió con la demanda total de los usuarios. Más aún, estas ventajas se lograron aplicando el AHDM sobre el modelo de Mandl que incorpora los aspectos dinámicos mencionados en la Sección 8.3, lo cual lo convierte en un problema de mayor complejidad.

En las Figuras 8.6 (a-b-c) se muestran tres posibles soluciones factibles obtenidas en el frente no dominado. El tiempo de espera promedio de los usuarios en los 3 casos es de 6:54min, 7:08min y 5:44min respectivamente.



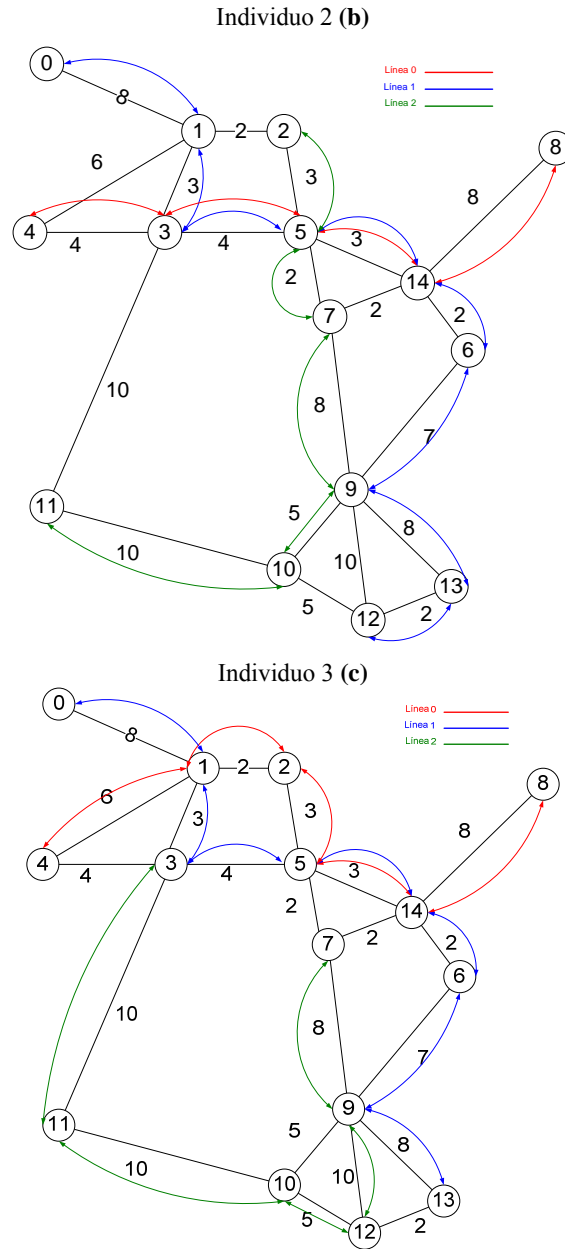


Figura 8.6. Tres soluciones no dominadas para el problema de Mandl

En la Figura 8.7 se muestra la población final obtenida luego de 200 generaciones para el problema de Mandl. Se observa claramente la buena distribución de los 100 individuos sobre los dos objetivos perseguidos.

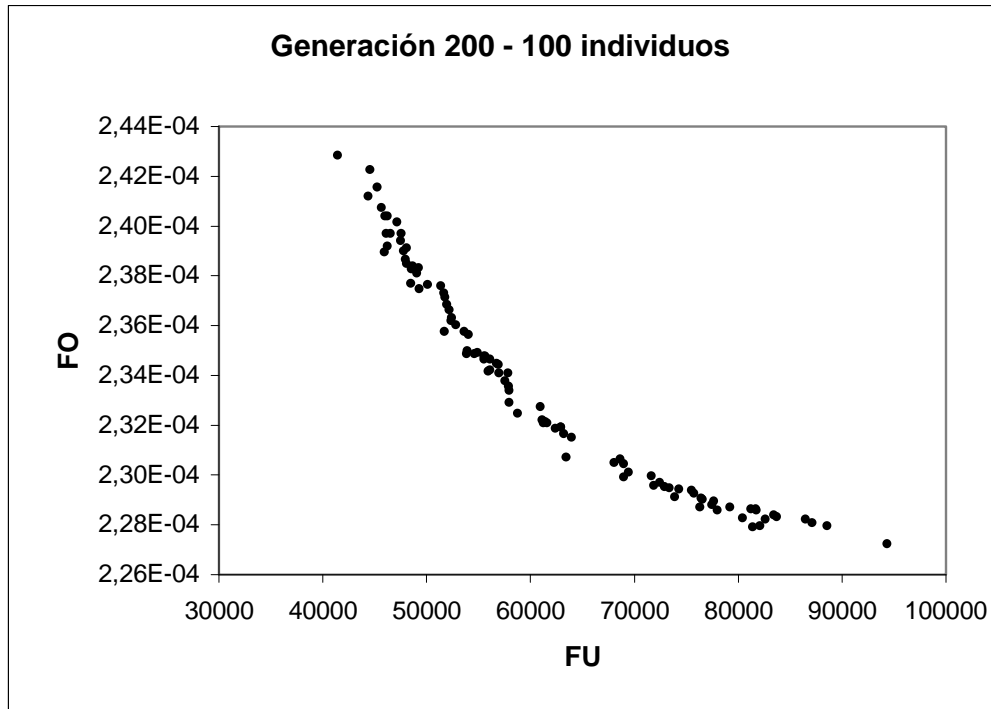


Figura 8.7 Población final obtenida luego de 200 generaciones para el problema de Mandl.

Resumen del capítulo

En este capítulo mostramos los casos de estudio elegidos para analizar el algoritmo híbrido propuesto en esta tesis doctoral y desarrollamos una serie de experimentos con el fin de demostrar la aplicabilidad del mismo al TNDSP. Se analizaron distintos MOEAs para la etapa evolutiva y también se comparó su desempeño con otros métodos presentados en la literatura, en base a un caso real.

Capítulo 9 Alcances

El algoritmo híbrido dinámico multi-objetivo desarrollado en el marco de esta tesis tiene múltiples aplicaciones pertenecientes a una clase de problemas. La salida del algoritmo es el conjunto de recorridos que optimizan los objetivos predefinidos. Entre los problemas en los que se puede utilizar este algoritmo se destacan aquellos donde existe una necesidad de definir la mejor localización de un conjunto de nodos con el propósito de cumplir ciertos objetivos económicos o sociales.

9.1 LOGÍSTICA

La logística (del inglés *logistics*, a su vez del francés *logistique* y *loger*), es definida por la Real Academia Española como el conjunto de medios y métodos necesarios para llevar a cabo la organización de una empresa, o de un servicio, especialmente de distribuciones. En el ámbito empresarial existen múltiples definiciones del término logística. Este término ha evolucionado desde la logística militar hasta el concepto contemporáneo del arte y la técnica que se ocupa de la organización de los **flujos** de mercancías, energía e información.

En negocios, la logística puede tener un enfoque bien interno, y/o bien externo que cubre el flujo desde el origen hasta la entrega al usuario final. Existen dos etapas básicas de logística:

1. Una optimiza un flujo de material constante a través de una red de enlaces de transporte y de centros del almacenaje.
2. La otra coordina una secuencia de recursos para realizar un determinado proyecto.

Todo ello al mínimo coste global para la empresa.

En particular la **logística de distribución** incluye la gestión de los flujos físicos, de información y administrativos siguientes:

- la previsión de la actividad de los centros logísticos
- el almacenamiento
- la preparación de los pedidos o la ejecución de cross docking (tránsito)
- a veces la realización de pequeñas actividades de transformación del producto (kitting, etiquetado)
- el transporte de distribución hasta el cliente
- traslado de mercancías de un lugar a otro con los recursos y equipos indispensables para su acción.

El marco de esta tesis doctoral nos enfocaremos en la previsión de la actividad de los centros logísticos y el transporte de distribución hasta el cliente.

9.2 LOGÍSTICA DEL TRANSPORTE

Se denomina **transporte** (del latín *trans*, "al otro lado", y *portare*, "llevar") al traslado de **personas** o **bienes** de un lugar a otro. Dentro de esta acepción se incluyen numerosos conceptos, de los que los más importantes son infraestructuras, vehículos y operaciones. Los transportes pueden también distinguirse según la posesión y el uso de la red.

La **planificación del tránsito** se define como un proyecto que estudia demandas presentes y futuras de movilidad de personas y material. Estos proyectos están precedidos por estudios de movimientos y necesariamente involucran a los diferentes medios de transporte. Está estrechamente relacionado con el campo de la ingeniería de tráfico (transporte). Podemos definir en líneas generales a la planificación como "*el desarrollo de planes y programas de acción dentro del contexto de un proceso continuo mediante el cual se alcanza un objetivo predeterminado*". La planificación clásica se refiere a la formulación, evaluación

y elección de alternativas para poder cumplir un objetivo determinado, por ejemplo la construcción de un tranvía, establecimiento de líneas de autobuses urbanos, etc.

Son ejemplos de transporte el **transporte público**, sobre el que se entiende que los vehículos son utilizables por cualquier persona previo pago de una cantidad de dinero también el **transporte privado**, aquel que es adquirido por personas particulares y cuyo uso queda restringido a sus dueños.

9.2.1 Transporte de Pasajeros

El **transporte de pasajeros** comprende los medios de transporte en que los pasajeros no son los propietarios de los mismos, siendo servidos por terceros. Los servicios de transporte de pasajeros pueden ser suministrados tanto por empresas públicas como privadas. Existe actualmente una gran variedad de transporte de pasajeros.

9.2.2 Transporte Público Urbano

Los transportes públicos en una ciudad procuran el desplazamiento de personas de un punto a otro en el área de esa ciudad. La gran mayoría de las áreas urbanas de medio y gran tamaño poseen algún tipo de **transporte público urbano**. Su suministro adecuado, en varios países, es, generalmente, de responsabilidad municipal, aunque el municipio pueda conceder licencias, a veces acompañadas de subsidios, a las compañías particulares.

El **transporte público urbano** es parte esencial de una ciudad. Disminuye la contaminación, ya que se usan menos coches para el transporte de personas, además de permitir el desplazamiento de personas que, no teniendo coche necesitan recorrer largas distancias. Tampoco debemos olvidar que hay personas que, teniendo coche, a veces no lo usan por los embotellamientos o las dificultades de estacionamiento y prefieren (al menos en algunas ocasiones) el transporte público. Este problema en particular es el que motivo el desarrollo de esta tesis.

9.2.3 Cruceros

El barco de pasajeros es la nave específicamente diseñada para el transporte, generalmente colectivo, de pasajeros. Según la duración del viaje el barco puede tener muchas instalaciones, llegando a los grandes barcos de crucero. Los cruceros son viajes realizados en barcos de pasajeros a cualquier parte del mundo como viaje de placer, siendo una forma de turismo única, diferente y especial de conocer países que son imposibles de explorar a través de otro medio de transporte.

El turismo en crucero se ha convertido en una actividad económica de suma relevancia. Esta industria mantiene cientos de miles de empleos directos e indirectos en diferentes países. Desde el diseño de los barcos, su construcción en astilleros, su tripulación, el aprovisionamiento de los barcos, las empresas de mantenimiento, las escalas, el personal en los puertos de escala, las empresas comercializadores de los cruceros. En particular para un viaje en crucero específico es preciso determinar previamente a la salida del barco el lugar de las escalas y el tiempo en que se llegará a cada una de ellas como así también el orden en el cual se recorren y su estadía en las mismas de forma tal que beneficie a la empresa y a la vez brinde el mayor confort a sus pasajeros.

9.2.4 Rutas Comerciales Marítimas

Las rutas comerciales generan los enlaces geográficos entre centros de producción de mercancías y los mercados de consumo. Las rutas comerciales, su exploración y explotación, son las verdaderas causas de los grandes descubrimientos y viajes de la historia, desde Zhang Qian, Erik el Rojo, Leif Eriksson, Zheng He y Marco Polo, hasta los viajes a la Luna, pasando por viajeros como Vasco de Gama y Cristóbal Colón, entre otros. Han sido también grandes impulsos para la ciencia y la tecnología de la navegación y el transporte terrestre y aéreo. Gracias a las necesidades de nuevas rutas comerciales, se crearon objetos como las

brújulas, los sextantes, los relojes y los modernos sistemas de navegación GPS, así como la cartografía.

Se denomina ruta marítima a la empleada por los buques mercantes en su tráfico comercial. Inicialmente estas rutas estuvieron limitadas a la navegación costera. Con el advenimiento de la navegación astronómica las rutas marítimas se internaron más y más en alta mar hasta circunnavegar el globo. Las rutas marítimas son el resultado de contemplar variados factores entre otros:

- Economía de combustible.
- Factores climatológicos y oceanográficos (corrientes marítimas, estadísticas de vientos predominantes, presencia de hielos flotantes etc.).
- Factores Legales (líneas de máxima carga).
- Cercanía a puertos intermedios para el caso de tener que efectuar recaladas forzosas para recibir asistencia o auxilio de tierra, etc.

Estas rutas utilizan vías acuáticas naturales como lagos, mares y ríos. Sin embargo, esas vías en general no se pueden utilizar de manera continua sin un mejoramiento considerable y sucesivos mantenimientos. Como por ejemplo los ríos canalizados, diseño de esclusas, embalses, diques y dragados.

9.2.5 Distribución de Gas/Electricidad y otros servicios

Los oleoductos son parte de plantas gigantescas desde las que se transporta petróleo de los campos a las refinerías o de éstas al mercado. Los oleoductos se pueden clasificar en tres categorías: de alimentación que son los que transportan el crudo desde el lugar de bombeo, aquellos que bombean el crudo en sí y las líneas de abastecimiento de productos refinados que transportan gasolina, queroseno y combustibles destilados.

Un proyecto de distribución de gas natural incluye la construcción de redes primarias de acero, estaciones de regulación, que reducen la presión para atender los sectores

residencial y comercial, además de las necesarias para atender el sector industrial, de acuerdo con sus requerimientos y legislación vigentes.

Dicho de otra manera este problema consiste en el armado de la red de suministro de gas natural de tal manera que el costo del operador sea mínimo y la calidad del gas cumpla los estándares mientras que el costo del servicio se reduce. En general podemos definir las redes de distribución de servicios de tal manera que el mismo esquema es aplicable a otros servicios como electricidad, GNC (gas natural comprimido para uso vehicular), entre otros. De la misma manera es posible trabajar sobre redes ya construidas con el objetivo de ampliar la zona de servicio de la red. Esta variante plantea un desafío adicional en lo que respecta a mantener la estructura original de la red y agregar los nuevos centros de servicio a la misma.

9.2.6 Servicios puerta a puerta

En los últimos años han cobrado importancia las empresas de entrega de paquetes y mensajería Express. Este servicio se encarga del traslado de mercadería u objetos desde un lugar en particular a destino en un tiempo límite establecido. Este tipo de problema requiere un alto conocimiento de la red de distribución, de los posibles destinos y orígenes de la mercadería a trasladar de tal manera que minimice el costo del transportista a la vez que lo hace con el del cliente y cumple con el plazo establecido. (Ejemplo UPS: <http://www.ups.com/content/ar/es/index.jsx>)

9.3 APLICACIÓN DEL ALGORITMO HÍBRIDO

Para todos los problemas descritos anteriormente el algoritmo híbrido puede ser utilizado como medio para obtener soluciones que contemplen las características particulares de cada uno. La flexibilidad inherente de los algoritmos evolutivos sumado a la capacidad de capturar la variabilidad del entorno provista por la simulación hace a las soluciones alternativas

brindadas por el mismo de especial interés tanto para el que realiza el análisis a nivel empresarial como para las autoridades que regulan y establecen las normativas de cada caso. Sumado a esto se pueden realizar pronósticos de acuerdo a cambios posibles de demanda y contexto, lo cual no sería posible con la utilización de otra meta-heurística. El hecho de encontrar soluciones adecuadas a las demandas específicas de todos los problemas relacionados con la logística del transporte incluyen la selección de un nivel y una calidad de servicio que satisfagan la demanda dentro de los límites de financiamiento posible. El nivel de servicio relaciona a la capacidad y la frecuencia del servicio. La calidad de dicho servicio incluye su seguridad, confiabilidad, privacidad, rapidez y efectos tanto en la comunidad como en el medio ambiente. Cada una asociadas con un costo.

Las metas de un buen sistema de transporte deben incluir la capacidad de proporcionar a un costo operativo razonable un cupo suficiente, una disponibilidad, un tiempo mínimo, confiabilidad en cualquier estación del año, comodidad razonable, conveniencia y efectos adversos mínimos en las áreas que abarca.

Resumen del capítulo

En este capítulo hemos tratado los principales alcances del algoritmo híbrido implementado como marco de esta tesis doctoral. Con esta herramienta es posible abordar distintos aspectos logísticos como por ejemplo transporte de mercadería o personas. En particular, esta tesis se ha enfocado en detalle en el problema de diseño y programación de redes de tránsito.

Capítulo 10 Conclusiones

El objetivo general de esta tesis ha sido desarrollar técnicas meta-heurísticas híbridas para el tratamiento de problemas *NP-Hard* relacionados con las redes de tránsito (*Transit Network Problem*, TNP). Estas técnicas están orientadas en particular a las dos etapas más importantes del TNP: el diseño y la planificación (TNDSP). El objetivo fue proveer una herramienta para la obtención tanto las rutas de las líneas de la red de tránsito como la frecuencia de los móviles.

La tesis se centra en subsanar la incapacidad de trabajos previos para obtener resultados que reflejen la dinámica de las variables que dependen de los tiempos consumidos por el usuario en su espera, acceso y viaje dentro de la red. Esta tesis analiza dichas falencias y presenta como solución la incorporación de un esquema de simulación dinámica dentro de la resolución del TNDSP para obtener resultados realistas tanto desde el punto de vista del operador del servicio como así también del cliente.

Más aún, dado que los operadores del servicio y los clientes de la red tienen objetivos contrapuestos, las técnicas más conocidas se vuelcan a beneficiar a una de las entidades de la solución o bien sólo optimizan un aspecto del problema. Nuestro estudio se concentró principalmente en resolver el problema desde el punto de vista de ambas entidades a partir de un enfoque multi-objetivo.

En particular se analizó la posibilidad de utilizar algoritmos evolutivos multi-objetivo para su tratamiento. Sin embargo, esto no era suficiente para capturar la realidad inmersa en el problema. Por esta razón, se consideró a la simulación como herramienta de soporte para

obtener valores realistas de los tiempos de demora, viaje y espera de los clientes. Como resultado se desarrolló un algoritmo híbrido dinámico multiobjetivo (AHDM) que provee un conjunto de soluciones factibles al TNDSP. Este algoritmo permite a los *decision maker* (DM) analizar múltiples soluciones que contemplan los objetivos contrapuestos del TNP. Asimismo, también se desarrollaron nuevos operadores de recombinación de los individuos.

La autoridad reguladora del servicio, por lo general el estado es el encargado de: establecer la tarifa, precios, el sistema de cobro del pasaje; brindar soportes de validación y control (GPS); determinar las paradas mínimas o principales que deben tener las líneas; desarrollar las políticas en infraestructura; estar en contacto directo con el usuario a través de sociedades de fomento, consejo deliberante etc; establecer las políticas de estacionamientos y circulación de vehículos; por último, también debe controlar y supervisar la prestación del servicio (horarios, cumplimiento de rutas).

Además, dado que no siempre se cuenta con las distancias entre las paradas de los móviles se incorporó un algoritmo de inicialización basado en *Greedy Randomized Adaptive Search Procedure* (GRASP) que obtiene la ruta y la distancia entre las paradas de interés de un mapa bajo estudio.

Finalmente, se trabajó con algunos casos de estudio para establecer la mejor configuración de parámetros de las fases del algoritmo híbrido. Se analizó una ciudad real utilizada en la literatura relacionada para comparar nuestra propuesta con los procedimientos más conocidos sobre el tema. Los resultados obtenidos han sido altamente satisfactorios.

En este capítulo se resumen los principales resultados obtenidos con el AHDM. Además, se presentan recomendaciones y lineamientos generales para futuras investigaciones en el tema, así como también sugerencias sobre extensiones naturales del trabajo desarrollado en esta tesis.

10.1 EPÍLOGO

Dado que nuestro algoritmo es aplicable a cualquier mapa y configuración de líneas, se introdujo como etapa inicial el cálculo de las distancias a las paradas teniendo en cuenta el sentido de las calles y la topografía del mapa. Se desarrolló un procedimiento basado en GRASP que obtiene las distancias y las rutas entre las paradas destacadas de una red de tránsito.

Como hemos dicho, el TNDSP tiene en todo momento dos entidades cuyos intereses están en contraposición: operadores y usuarios. Por un lado, los operadores del servicio desean obtener una red que les permita cubrir con pocos kilómetros el mayor tramo posible de paradas y aumentar la frecuencia al máximo a fin de minimizar el costo operativo. Por su parte, el usuario desea una red flexible que le permita llegar a través de varias líneas a su destino, hacer trasbordos en lugares estratégicos y que exista un servicio para transportarse hacia/desde su trabajo.

Como resultado de nuestras investigaciones se obtuvo un algoritmo híbrido evolutivo dinámico multi-objetivo de dos etapas que tiene en cuenta tanto al operador como a los clientes de la red. La primera es una etapa de inicialización de distancias y rutas entre paradas. La segunda implementa un algoritmo evolutivo que contiene un procedimiento de simulación para el cálculo del *fitness* de los individuos.

Dada la variedad de métodos de selección de los algoritmos evolutivos multi-objetivo fue necesario implementar aquellos más conocidos en la literatura y determinar cuál era el más apropiado para el TNDSP. Por ende, se utilizó una plataforma de desarrollo denominada PISA que permite no sólo la implementación ágil de los algoritmos más utilizados, sino también el análisis de conjuntos de aproximación para establecer naturalmente puntos de comparación entre los métodos.

De los tres selectores implementados (NSGA-II, SPEA2 e IBEA), SPEA2 reunía las características más favorables para resolver el TNDSP. Se utilizaron distintos indicadores de calidad para evaluar los conjuntos de aproximación de cada implementación propuesta. En todos los casos, los indicadores demostraron que no existían diferencias. Por lo tanto, se analizó como último recurso la dispersión de las soluciones sobre las fronteras no dominadas. SPEA2 obtenía un conjunto más disperso de individuos lo que significa para este problema que el DM tendrá más alternativas para elegir. En particular, dada la necesidad de analizar una red de tránsito desde el punto de vista de los operadores, los usuarios y las reglamentaciones vigentes en cada lugar es necesario contar con soluciones en extensión más que en profundidad.

Se analizó entonces su desempeño con respecto a otros métodos utilizados para abordar el TNDSP. El caso de estudio elegido fue el propuesto por Mandl (1980), el cual estudia una ciudad de Suiza. Dada la simplicidad del caso de estudio se agregaron características dinámicas al problema para modelar con realismo los arribos de los usuarios. Sin embargo, este enriquecimiento del modelo no fue en detrimento de nuestro algoritmo; los resultados obtenidos por el AHDM son comparables con los encontrados en la literatura sobre el tema (Baaj y Mahmassani, 1991; Mandl, 1980), e incluso mejora aspectos asociados al tiempo de espera, acceso y viaje de los usuarios del servicio.

10.2 PROYECCIÓN

En primer lugar, como extensión natural a las investigaciones realizadas en esta tesis, se proyecta la posibilidad de incorporar más detalle al modelo utilizado para la simulación. Cuanto más fiel sea el modelo de simulación para representar la dinámica de la red de tránsito, mejores resultados podrán obtenerse en la realidad. Para este propósito se deben agregar características tales como la tolerancia a la espera, la posibilidad de avería en los móviles, los cambios de recorrido imprevistos por cortes de tránsito, etc. Se proyecta la expansión del

simulador desarrollado y/o la interoperabilidad de nuestra técnica con otros simuladores que ya se han implementado para este problema y se encuentran disponibles.

Además, se desea incorporar en futuras extensiones del trabajo un objetivo que tenga en cuenta consideraciones relacionadas con el medio ambiente. Dicho objetivo está orientado a la disminución de la polución en el micro centro de las ciudades. Esta polución está íntimamente relacionada con la afluencia de autos y buses dentro de un área que contiene edificios públicos y plazas principales.

Con respecto a la etapa de inicialización de las distancias y rutas entre paradas, se ha comenzado a trabajar con el algoritmo Floyd-Warshall (1962). El mismo, ampliamente utilizado en la literatura, identifica el camino más corto (óptimo) entre dos nodos cualesquiera de un grafo dirigido con un tiempo de ejecución de orden de $O(N^3)$.

Por último, la urbanización no planificada en Bahía Blanca ha llevado al empobrecimiento de las redes de tránsito, la duplicación de recorridos y la falta de paradas en lugares con edificaciones recientes; estas falencias terminan por precarizar a todo el sistema de transporte público. Esta tesis no sólo se propone resolver el TNDSP sino también dar soluciones factibles a la falta de infraestructura del tránsito local en la que deseamos aplicar el algoritmo en busca de resultados aceptables y satisfactorios sin olvidar que toda respuesta viable requiere de un proceso de mediación en el que se tengan presente todas las partes afectadas.

Referencias

-A-

Applegate, D.L., Bixby, R.E., Chvátal V. and W.J. Cook., 2006. The Traveling Salesman Problem: A Computational Study. Princeton University Press. ISBN 978-0-691-12993-8.

-B-

Baaj, M. H. and Mahmassani, H. S., 1991. Hybrid route generation heuristic algorithm for the design of transit networks. *Transportation Research C* 3, 31-50.

Bäck, T., 1996. *Evolutionary Algorithms in Theory and Practice*. Oxford University Press, New York.

Banks, J., Carson II, J. S., Nelson B. L., and Nicol D. M., 2001. *Discrete-Event System Simulation*. Prentice-Hall International Series in Industrial and System Engineering.

Bazaraa, M. S., Jarvis, J. J. and Sherali, H. D., 2006. *Linear Programming And Network Flows*. 3rd Edition. 704 Pages. John Wiley & Sons Inc.

Benn, H. P., 1995. Bus route evaluation standards. Tech. rep., Transportation Research Board, Washington.

Bleuler, S., Laumanns, M., Thiele, L. and Zitzler, E. 2003. PISA - A Platform and Programming Language Independent Interface for Search Algorithms, in: *Proceeding of Evolutionary Multi-Criterion Optimization*, pp. 494-508.

Bookbinder, J. H., Désilets, A., 1992. Transfer optimization in a transit network. *Transportation Science* 26, 106-118.

Borndörfer, R., Grötschel, M., Pfetsch, M. E., 2005. A path-based model for Línea planning in public transport. Tech. Rep. Report 05-18, ZIB.

Bunte, S., Kliewer, N., Suhl, L., 2006. An overview on vehicle scheduling models in public transport. In *Proceedings of the 10th International Conference on Computer-Aided Scheduling of Public transport*, Leeds UK, 2006. Springer Verlag.

Bussieck, M. R., 1998. *Optimal Líneas in public rail transport*. Ph.D. thesis, TU Braunschweig.

-C-

Carrese, S., Gori, S., 2002. An urban bus network design procedure. *Applied Optimization* 64, 177-196.

Castelli, L., Pesenti, R., Ukovich, W., 2004. Scheduling multimodal transportation systems. *European Journal of Operational Research* 155, 603-615.

Ceder, A. and Israeli Y., 1997. Creation of Objective functions for transit network design. *Proceedings of IFAC Transportation Systems Chania Greece* 684-689.

Ceder, A. and Wilson, N. H. M., 1986. Bus Network Design, *Transportation Research*, Vol. 20 (4), 331-344.

Ceder, A., Golany, B., Tal, O., 2001. Creating bus timetables with maximal synchronization. *Transportation Research A* 35, 913-928.

Ceder, A., 2003a. Designing public transport network and routes. Pergamon Imprint, Elsevier Science Ltd, 59-91.

- Ceder, A., 2003b. Public transport timetabling and vehicle scheduling. Pergamon Imprint, Elsevier Science Ltd, 31-57.
- Charnes, A. and Cooper, W. W., 1961. Management Models and Industrial Applications of Linear Programming, volume 1. John Wiley, New York.
- Chakroborty, P., 2003. Genetic algorithms for optimal urban transit network design. Journal of Computer Aided Civil and Infrastructure Engineering 18, 184-200.
- Chakroborty, P., Deb, K., Porwal, H., 1997. A genetic algorithm based procedure for optimal transit systems scheduling. In Proceedings of the 5th International Conference on Computers in Urban Planning and Urban Management. Mumbai, India, 330-341.
- Chakroborty, P., Deb, K., Sharma, R. K., 2001. Optimal fleet size distribution and scheduling of urban transit systems using genetic algorithms. Transportation Planning and Technology 24 (3), 209-226.
- Chakroborty, P., Deb, K., Subrahmanyam, P. S., 1995. Optimal scheduling of urban transit systems using genetic algorithms. Journal of Transportation Engineering 121 (6), 544-553.
- Chakroborty, P., Dwivedi, T., 2002. Optimal route network design for transit systems using genetic algorithms. Engineering Optimization 34 (1), 83-100.
- Charnes, A. and Cooper, W. W., 1961. Management Models and Industrial Applications of Linear Programming, volume 1. John Wiley, New York.
- Chowdhury, S., Chien, S., 2001. Optimization of transfer coordination for intermodal transit networks. In: 80th Annual Meeting, Transportation Research Board. Washington, DC, paper No 01-0205.
- Coello, C. A., Lamont, G. B., and Veldhuizen, D. A., 2006. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag New York, Inc.
- Cohon, J. L. and Marks, D. H.. A Review and Evaluation of Multiobjective Programming Techniques. Water Resources Research, 11(2):208–220, apr 1975.
- Conover, W., 1999. *Practical Nonparametric Statistics*. John Wiley & Sons. New York, NY.
- Constantin, I., Florian, M., 1995. Optimizing frequencies in a transit network: a nonLinear bi-level programming approach. International Transactions in Operational Research 2 (2), 149-164.

-D-

- Da Cunha, N. O., Polak, E., 1967. Constrained minimization under vector valued-criterial in finite dimensional spaces. J. Mathematical Analysis & Applications, vol. 19 (1), 103-124.
- Daduna, J. R., Voss, S., 1995. Practical experiences in schedule synchronization. Lecture Notes in Economics and Mathematical Systems 430, 39-55.
- Dantzig, G.B., Ramser, J.H. 1959. *The Truck Dispatching Problem*. Management Science 6 (1): 80–91.
- Davis, L., 1985. Applying adaptive algorithms to epistatic domains, in: Proceedings of the International Joint Conference on Artificial Intelligence, pp.162–164.
- Davis, L., 1991. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold Library.
- De Palma, A., Lindsey, P., 2001. Optimal timetables for public transportation. Transportation Research B 35 (8), 789-813.

- Deb, K. and Agrawal, R. B., 1995. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, 9:115–148.
- Deb, K., 1995. *Optimization for engineering design: Algorithms and examples New Delhi*. Prentice-Hall.
- Deb, K., Agarwal, S., Pratap, A., Meyarivan T., 2000. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France. Springer. Lecture Notes in Computer Science No. 1917
- Deb, K., 2001. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK. ISBN 0-471-87339-X.
- Desaulniers, G. and Hickman M., 2007. Public transit. *Handbooks in Operations Research and Management Science, Transportation*, Vol. 14, G. Laporte and C. Barnhart (eds), Elsevier, Amsterdam, 69-127.
- Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

-E-

- Edgeworth, F. Y. 1881. *Mathematical Psychics*. P. Keagan, London, England.
- Eranki, A., 2004. *A model to create bus timetables to attain maximum synchronization considering waiting times at transfer stops*. Master's thesis, University of South Florida.
- Fan, W., Machemehl, R., 2004. Optimal transit route network design problem: Algorithms, implementations, and numerical results. Tech. Rep. SWUTC/04/167244-1, Center for Transportation Research, University of Texas.

-F-

- Fleurent, C., Lessard, R., Seguin, L., 2004. Transit timetable synchronization: Evaluation and optimization. In *Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT)*. San Diego, California.
- Floyd, R.W., 1962. *Algorithm 97: Shortest Path*. *Communications of the ACM* 5 (6): 345.
- Fourman, M. P. 1985. Compaction of Symbolic Layout using Genetic Algorithms. In J. J. Grefenstette, editor, *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 141–153. Lawrence Erlbaum, Hillsdale, New Jersey.
- Fonseca, C. M. and Fleming, P. J. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416-423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kaufmann Publishers.
- French, S., 1982. *Sequencing and Scheduling: An Introduction to the Mathematics of the Job-Shop*. Wiley.
- Frutos M., Olivera A. C., 2008a. Job Shop Scheduling Problem: Desarrollo de un Procedimiento Eficiente. Publicado en los anales del XIV Congreso Latino Ibero Americano en Investigación de Operaciones (CLAIO 2008) se llevó a cabo en el Centro de Convenciones Cartagena de Indias, Cartagena de Indias, Colombia, del 9-12/09/2008. ISBN 978-958-825283-4. 3 Páginas.
- Frutos M., Olivera A. C., 2008b. Vehicle Sequencing Problem: Una Técnica Híbrida para El ROADEF'05 Challenge. Publicado en los anales del XIV Congreso Latino Ibero Americano en Investigación de

Operaciones (CLAIO 2008) se llevó a cabo en el Centro de Convenciones Cartagena de Indias, Cartagena de Indias, Colombia, del 9-12/09/2008. ISBN 978-958-825283-4. Organizado por la Asociación Latino-Iberoamericana de Investigación Operativa. 3 Páginas.

Frutos M., Olivera A. C., Casal R., 2006. Estudio sobre distintas estrategias utilizadas para establecer secuencias detalladas en entornos productivos”. Publicado en los Anales de XII ICIEOM – “XII International Conference on Industrial Engineering and Operations Management”. Organizado por ABEPRO - Associação Brasileira de Engenharia de Produção. Realizado del 09-11/10/2006, Fortaleza - CE, Praia do Futuro, Brasil. 9 Páginas.

Frutos M., Olivera A. C., Casal R., 2007a. Algoritmo híbrido para un enfoque alternativo al Job-Shop Scheduling Problem. Publicado en el XIII International Conference on Industrial Engineering and Operations Management (ICIEOM 2007), en Foz do Iguaçu, Paraná, Brasil, organizado por ABEPRO - Brazilian Association of Production Engineering. 10 páginas.

Frutos M., Olivera A. C., Casal R., 2007b. Algoritmo Híbrido Aplicado al Job-Shop Scheduling Problem. Publicado en el CDROM de anales del VII Congreso Chileno de Investigación Operativa (OPTIMA 2007). A realizarse el 21-23/11/2007 en Puerto Montt, X° Región, Chile.

Frutos M., Casal R., Olivera A. C., 2007c. Algoritmo híbrido estocástico aplicado al diseño de rutas y determinación de frecuencias en el transporte público urbano. Publicado en los Proceedings del V Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2007). Del 14-16/02/2007. Organizado por Universidad de La Laguna, España. Puerto de La Cruz, Tenerife, España. 7 Páginas.

Frutos M., Olivera A. C., Tohmé F., 2009. A Hybrid Technique for Handling the Flexible Job-Shop Scheduling Problem. Publicado en el 24th IFIP TC 7 Conference on System Modelling and Optimization. 27-31/07/2009. Buenos Aires, Argentina. Organizado por el Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación.

-G-

Gao, Z., Sun, H., Shan, L., 2003. A continuous equilibrium network design model and algorithm for transit systems. *Transportation Research B* 38, 235-250.

Goldberg, D. E., 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts.

Goldberg, D. E. and Lingle, R. Jr., 1985. Alleles/Loci and the Traveling Salesman Problem, *Proceedings of the 1st International Conference on Genetic Algorithms*, p.154-159.

Gordon, G., 1979. The design of the GPSS language. In N. R. Adam & A. Dogramaci (eds). *Current Issues in Computer Simulation*. Academic Press. New York.

Guan, J. F., Yang, H., Wirasinghe, S. C., 2003. Simultaneous optimization of transit Linea configuration and passenger Linea assignment. *Transportation Research B* 40(10), 885-902.

Guihaire, V. and Hao J., 2008. Transit network design and scheduling: a global review. *Transportation Research Part A: Policy and Practice*. In Press, Draft.

Gruttner, E., Pinninghoff, M. A., Tudela, A., Díaz, H., 2002. Recorridos Óptimos de Líneas de Transporte Público Usando Algoritmos Genéticos, *Jornadas Chilenas de Computación*, Chile.

-H-

- Hajela, P. and Lin, C. Y. , 1992. Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4:99–107.
- Hasselström, D., 1979. A method for optimization of urban bus route networks. Tech. report, Volvo Bus Corporation. Göteborg.
- Hasselström, D., 1981. *Public transportation planning - a mathematical programming approach*. Ph.D. thesis, University of Göteborg, Sweden.
- HASTUS (<http://www.giro.ca/en/products/hastus/index.htm>)
- Horn, J. and Nafpliotis, N. 1993. Multiobjective Optimization using the Niche Pareto Genetic Algorithm. Technical Report IlliGAI Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA.
- Horn, J., Nafpliotis, N. and Goldberg, D. E., 1994. A Niche Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, IEEE Service Center.

-I-

- Ijiri, Y., 1965. *Management Goals and Accounting for Control*. North-Holland, Amsterdam.

-J-

- Jansen, L. N., Pedersen, M. B., Nielsen, O. A., 2002. Minimizing passenger transfer times in public transport timetables. In *Proceedings of the 7th Conference of the Hong Kong Society for Transportation Studies : Transportation in the information age*. Hong Kong, 229-239.
- Jutler, H., 1967. *Liniejnaja model z nieskolkimi celevymi funkcijami* (Linear Model with Several Objective Functions). *Ekonomika i matematiceckije Metody*, 3:397–406. (In Polish).

-K-

- Klee, V., Minty, G. J. 1972. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities*, III, pages 159-175. Academic Press, New York.
- Klemt, W. D., Stemme, W., 1988. Schedule synchronization for public transit networks. *Computer-Aided Transit Scheduling, Proceedings of the 4th International Workshop on Computer-Aided Scheduling of Public Transport*. Springer Verlag, Hamburg, Germany, 327-335.
- Knowles, J., Thiele, L., Zitzler, E., 2005. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. *TIK Computer Engineering and Networks Laboratory*.
- Köppen, M. and Rudlof, S., 1998. Multiobjective Optimization by Nussy Algorithm. In R. Roy, T. Furuhashi, and P. Chawdhry, editors, *Advances in Soft Computing*, pages 357–368, London. Springer.
- Kornbluth J. S. H. and Steuer R. E., 1981. Multiple objective Linear fractional programming. *Management Science* 27(9):1024–1039.
- Koutsopoulos, H. N., Odoni, A., Wilson, N. H. M., 1985. Determination of headways as function of time varying characteristics on a transit network. North-Holland, Amsterdam, 391-414.

- Kuhn, H. W., Tucker, A. W., 1951. NonLinear programming. Proceedings of 2nd Berkeley Symposium. Berkeley: University of California Press. 481-492.
- Kung, H. T., Luccio, F., Preparata, F. P., 1975. On finding the maxima of a set of vectors. Journal of the ACM. volume 22, 469-476.

-L-

- Lampkin, W., Saalmans, P. D., 1967. The design of routes, service frequencies, and schedules for a municipal bus undertaking: A case study. Operational Research Quarterly 18 (4), 375-397.
- LeBlanc, L., Morlok, E., Pierskalla, W., 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. Transportation research 9 (5), 309-318.

-M-

- Magnanti, T. L., Wong, R. T., 1984. Network design and transportation planning: models and algorithms. Transportation Science 18 (1), 1-55.
- Mandl, C. E., 1979. Evaluation and optimization of urban public transportation networks. European Journal of Operational Research 5, 396-404.
- Mandl, C. E., 1979. *Applied Network Optimization*. London, Academic Press.
- Merkle, L. D. and Lamont, G. B.1 1997. A Random Function Based Framework for Evolutionary Algorithms. In T. Bäck, editor, Proceedings of the Seventh International Conference on Genetic Algorithms, pages 105–112. Morgan Kaufmann Publishers, San Mateo, California.
- Michalewicz, Z., 1994. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, 2nd (extended) edition.
- Murray, A. T., 2003. A coverage model for improving public transit system accessibility and expanding access. Annals of Operations Research 123, 143-156.

-N-

- Ngamchai, S., Lovell, D., 2003. Optimal time transfer in bus transit route network design using a genetic algorithm. Journal of Transportation Engineering 129 (5), 510-521.

-O-

- Osyczka, A. and Kundu, S., 1995 A Genetic Algorithm-Based Multicriteria Optimization Method. In Proceedings of First World Congress of Structural and Multidisciplinary Optimization, pp 909-914, Goslar, Germany. Elsevier Science.
- Olivera A. C., Frutos M., Carballido J. A., Ponzoni I., Brignole N. B., 2009a. Bus Network Scheduling Problem: GRASP + EAs with PISA * Simulation. 10th International Work-Conference on Artificial Neural Networks (IWANN2009). 10-12/06/2009. Salamanca, España. Organizado por Spanish Chapter of the IEEE Computational Intelligence Society, Universidad de Salamanca, Universitat Politècnica de Catalunya, Universidad de Málaga, Universidad de Granada. *Lecture Notes in Computer Science*. Springer - Verlag. IWANN 2009, Part I, LNCS 5517, pp. 1272–1279.

- Olivera A. C., Frutos M., Carballido J. A., Brignole N. B., 2008. Bus Network Optimization with a Time-Dependent Hybrid Algorithm. J.UCS - Journal of Universal Computer Science, Graz University of Technology, and supported by the KNOW-Center and University of Applied Sciences. Vol. 14, Num. 15, pp 2512-2531. ISSN 0948-695x, OnLine Edition: ISSN 0948-6968 http://www.jucs.org/jucs_14_15/bus_network_optimization_with. 2008.
- Olivera A. C., Frutos M., Carballido J. A., Brignole N. B., 2009b. GRASP + SPEA2 * Simulation = Allocation and Scheduling of Bus-Networks in Urban Areas. Publicado en el 24th IFIP TC 7 Conference on System Modelling and Optimization. 27-31/07/2009. Buenos Aires, Argentina. Organizado por el Ministerio de Ciencia, Tecnología e Innovación Productiva de la Nación.
- Olivera A. C., Frutos M., Carballido J. A., Brignole N. B., 2007a. Bus Network Optimization through Time-Dependent Hybrid Algorithm. Publicado en el International Conference on Intelligent Systems Design and Applications (ISDA'07). Del 22-24/10/2007 en Río de Janeiro, Brasil. volume 0, ISBN 0-7695-2976-3, pp. 857-862, publisher IEEE Computer Society, Los Alamitos, CA, USA.
- Olivera A. C., Frutos M., Brignole N. B., Casal R., 2007b. Evaluación del Sistema de Transporte Público Urbano Mediante Simulación de Entornos Dinámicos. Publicado en el CDROM de anales del VII Congreso Chileno de Investigación Operativa. 21-23/11/2007. Puerto Montt, X° Región, Chile. Organizado por la Asociación Latino-Iberoamericana de Investigación Operativa.
- Olivera A. C., Frutos M., Carballido J. A., Brignole N. B., 2007c. GRASP initialization stage for bus-network hybrid algorithm. Publicado en los Anales del "Simposio de Investigación Operativa" - SIO 2007 (ISSN: 1850-2865), como parte de las 36° JAIHO, Jornadas Argentinas de la Sociedad Argentina de Informática. Mar del Plata, Buenos Aires, Argentina, entre los días 27-28/08/2007. 8 páginas.
- Olivera A. C., Frutos M., Casal R. 2006a. Algoritmos Genéticos una Estrategia clave para abordar Problemas de Secuenciamiento en Gran Escala. Publicado en Anales de JAIHO 35° "Jornadas Argentinas de Informática e Investigación Operativa" (ISSN 1850-2776) "3° Jornadas de Informática Industrial 2006", pp. 57 a 66 (ISSN 1850-2849), 4-8/11/2006, Ciudad de Mendoza Argentina.
- Olivera A. C., Frutos M., Casal R. 2006b. Métodos para determinar Secuencias de Producción en un Ambiente Productivo Complejo". Publicado en el Congreso Latino-Iberoamericano de Investigación Operativa. Organizado por Universidad de la República, Uruguay. ISBN: 9974-7699-9-X. 27-30/11/2006, Montevideo, Uruguay. 6 Páginas.

-P-

- Pareto, V., 1896. Cours D'Economie Politique, volume I & II. F. Rouge, Lausanne.
- Park, S. J., 2005. *Bus network scheduling with genetic algorithms and simulation*. Master's thesis, University of Maryland.
- Pattnaik, S. B., Mohan, S., Tom, V. M., 1998. Urban bus transit route network design using genetic algorithm. *Journal of Transportation Engineering* 124 (4), 368-375.
- Patz, A., 1925. Die richtige Auswahl von Verkehrslinien bei großen Straßenbahnnetzen. *Verkehrstechnik* 50/51.
- Perlis, A. J., 1998. *Commun ACM*. Vol 5, Num 6, issue date 1962, ACM, New York, NY, USA.
- Pidd, M. 1998. *Computer Simulation in Management Science*. John Wiley & Sons.
- Pitsoulis, L. S., Resende, M.G.C. 2001. Greedy Randomized Adaptive Search Procedures. In P.M. Pardalos and M.G.C. Resende editors, *Handbook of Applied Optimization*. Oxford University Press.

-Q-

Quak, C. B., 2003. *Bus Line planning*. Master's thesis, Delft University of technology, The Netherlands.

-R-

Rapp, M. H., Gehner, C. D., 1976. Transfer optimization in an interactive graphic system for transit planning. *Transportation Research Record* 619, 27-33.

Reeves, C. R., 1993. *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Press, Oxford.

Reklaitis, G.V., Ravindran, A., Ragsdell, K.M., 1983. *Engineering Optimization*. New Jersey: Wiley Interscience.

Ritzel, B. J., Eheart, W. J. and Ranjithan, S., 1994. Using genetic algorithms to solve a multiple objective groundwater pollution containment problem. *Water Resources Research*, 30(5):1589-1603.

Rosenberg, R. S., 1967. *Simulation of genetic populations with biochemical properties*. PhD thesis, University of Michigan, Ann Arbor, Michigan, USA.

-S-

Salzborn, F. J. M., 1972. Optimum bus scheduling. *Transportation Science* 6 (2), 137-148.

Salzborn, F. J. M., 1980. Scheduling bus systems with interchanges. *Transportation Science* 14 (3), 211-220.

Scheele, S., 1980. A supply model for public transit services. *Transportation Research B* 14, 133-146.

Schaffer, J. D., 1984. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, Nashville, Tennessee.

Shannon, R. E., 1975. *Systems Simulation: The Art and Science*. Prentice-Hall, New York.

Shih, M., Mahmassani, H. S., 1994. A design methodology for bus transit networks with coordinated operations. Tech. Rep. SWUTC/94/60016-1, Center for Transportation Research, University of Texas, Austin.

Shih, M., Mahmassani, H. S., Baaj, M., 1998. A planning and design model for transit route networks with coordinated operations. *Transportation Research Record* 1623, 16-23.

Solich, R. I. 1969. *Zadanie programowania liniowego z wieloma funkcjami celu* (Linear Programming Problem with Several Objective Functions). *Przegląd Statystyczny*, 16:24-30. (In Polish).

Sonntag, H., 1977. *Linienplanung im öffentlichen Personennahverkehr*. Ph.D. thesis, Technische Universität Berlin.

Spasovic L. N., Boile M. P., Bladikas A. K., 1994. A Methodological Framework for Optimizing Bus Transit Service Coverage. 73rd Annual Meeting of the Transportation Research Board, January 9-13, Washington, DC.

Syswerda, G., 1991. Schedule Optimization using Genetic Algorithms in *Handbook of Genetic Algorithms* Van Nostrand Reinhold Library, Lawrence Davis (ed).

Syswerda, G. and Palmucci, J., 1991. The Application of Genetic Algorithms to Resource Scheduling. In R. K. Belew and L. B. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 502-508, San Mateo, California. Morgan Kaufmann Publishers.

Srinivas, N. and Deb, K., 1994. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221-248.

-T-

Tigan S., 1975. Sur le probleme de la programmation vectorielle fractionnaire, *Rev. Analyse Numer. Theor. Approx.*, 4, no.1, 99-103.

Ting, C. J., Schonfeld, P., 2005. Schedule coordination in a multiple hub transit network. *Journal of Urban Planning and Development* 131 (2), 112-124.

Tom, V. M., Mohan, S., 2003. Transit route network design using frequency coded genetic algorithm. *Journal of Transportation Engineering* 129 (2), 186-195.

-U-

-V-

Van Nes, R., 2002. Design of multimodal transport networks, a hierarchical approach. TRAIL Thesis Series T2002/5, DUP, Delft University, The Netherlands.

Van den Berg, J. H. A. V., Odijk, M. A., 1994. Dons: Computer aided design of regular service timetables. In *Proceedings of CompRail94, Madrid*, 109-116.

Van Nes, R., Hamerslag, R., Immers, B. H., 1988. Design of public transport networks. *Transportation Research Record* 1202, 74-83.

Veldhuizen D. A. H., 1999. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Ph.D thesis. Wright-Patterson AFB, OH.

-W-

Wong, R. C. W., Leung, J. M. Y., 2004. Timetable synchronization for mass transit railway. In *Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT)*. San Diego, California.

Wren, A., Rousseau, J.M., 1993. Bus driver scheduling - an overview. In *Daduna, J. R., Branco, I., and Paixao, J. M. P., editors, Computer-Aided Transit Scheduling*, 173-187. Springer, Berlin, Germany.

-X-

Xiong, Y., Schneider, J. B., 1993. Transportation network design using a cumulative algorithm and neural network. *Transportation Research Record* 1364, 37-44.

Yan, S., Chen, H. L., 2002. A scheduling model and a solution algorithm for inter-city bus carriers. *Transportation Research A* 36, 805-825.

-Y-

Yu, B., Yang, Z., Cheng, C., Liu, C., 2005. Optimizing bus transit network with parallel ant colony algorithm. In Proceedings of the Eastern Asia Society for Transportation Studies. Vol.5, 374-389.

-Z-

Zadeh, L. A., 1963. Optimality and Nonscalar-Valued Performance Criteria. IEEE Transactions on Automatic Control, AC-8(1):59-60.

Zeleny, M., 1973. Compromise Programming. In J. Cochrane and M. Zeleny, editors, Multiple Criteria Decision Making, pages 262-301. University of South Carolina Press, Columbia, South Carolina.

Zhao, F., Gan, A., 2003. Optimization of transit network to minimize transfers. Tech. Rep. BD015-02, Florida Department of Transportation, Center for Transportation Research, Florida international University.

Zhao, F., Ubaka, I., 2004. Transit network optimization- minimizing transfers and optimizing route directness. Journal of Public Transportation 7 (1), 67-82.

Zhao, F., Zeng, X., 2006a. Simulated annealing-genetic algorithm for transit network optimization. Journal of Computing in Civil Engineering 20 (1), 57-68.

Zhao, F., 2006b. Large-scale transit network optimization by minimizing user cost and transfers. Journal of Public Transportation 9(2), 107-129.

Zitzler, E., Künzli, S., 2004. Indicator-Based Selection in Multiobjective Search. In: X. Yao, E. K. Burke, J. Lozano, J. Smith, J. Merelo Guervós, J. Bullnaria, J. Rowe, P. Tiño, A. Kabán, H-P. Schwefel (Eds.): PPSN VIII. LNCS, vol. 3242, pp. 832--842. Springer.

Zitzler, E. and Thiele, L., 1998. An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach. Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland.

Zitzler, E., Thiele, L., 1999. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. Evol. Comput. 3 (4), 257-271.

Zitzler, E., Laumanns, M., Thiele, L., 2002. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In Giannakoglou, Tsahalis, Periaux, Papailiou, and Fogarty (eds), Evolutionary Methods for Design, Optimisations and Control, 19-26.

Lista de figuras

Figura 1.1 Elementos de la red de tránsito.....	15
Figura 2.1 Clasificación de subproblemas del TNDSP.....	24
Figura 2.2 Árbol de relaciones entre problemas TN.....	25
Figura 2.3 Posición del problema de Diseño (TNDP) en el contexto de TNDSP.....	27
Figura 2.4 TNFSP en el contexto de TNDSP.....	30
Figura 2.5 TNTP en el contexto de TNDSP.....	32
Figura 2.6 Esquema del problema TNDFSP en el contexto del TNDSP.....	36
Figura 2.7 Esquema del problema TNDFSP en el contexto del TNDSP.....	38
Figura 3.1 Espacio de decisión (en color rosa) vs. Espacio de la función objetivo (en color celeste).....	41
Figura 3.2 Dos espacios de soluciones.....	42
Figura 4.1 Pasos esenciales de un EA.....	57
Figura 4.2 Una permutación (solución) para el problema del viajante de comercio (TSP).....	61
Figura 4.3 Recorrido y orden de una solución del TSP.....	62
Figura 4.4 Dos individuos para el TSP.....	64
Figura 4.5 Ejemplo de cruzamiento OX entre los padres p_1 y p_2	64
Figura 4.6 Ejemplo de cruzamiento PMX entre los padres p_1 y p_2	65
Figura 4.7 un recorrido a_1	66
Figura 4.8 mutación de a_1	66
Figura 5.1 La máxima extensión no revela la distribución real de las soluciones.....	83
Figura 5.2 El hipervolumen encerrado en las soluciones no dominadas.....	83
Figura 5.3 Visión global de PISA para el caso particular del TNDSP.....	88
Figura 6.1 Pasos de un estudio utilizando simulación.....	96
Figura 7.1 Esquema general del algoritmo híbrido dinámico multi-objetivo.....	106
Figura 7.2 Gráfica del procedimiento de búsqueda local.....	109
Figura 7.3 Ejemplo de un individuo con 40 paradas y 5 líneas.....	110
Figura 7.4 Estructura de una red con dos líneas, varias paradas, usuarios y buses.....	114
Figura 7.5 Padres seleccionados para cruzamiento.....	116
Figura 7.6 Hijos resultantes del crossover.....	117
Figura 7.7 Red de tránsito a mutar.....	118
Figura 7.8 Diferentes operadores de mutación.....	118
Figura 7.9 Estructura general de AHDM utilizando la herramienta PISA.....	120
Figura 8.1 Ubicación de los nodos en el barrio residencial.....	125
Figura 8.2 Mapa de una ciudad Suiza (Mandl, 1980).....	127
Figura 8.3 Individuos no dominados obtenidos con los diferentes tamaños de generaciones.....	130
Figura 8.4 Diagramas de caja (boxplots) para cada indicador unario.....	134
Figura 8.5 Gráfica de 4 conjuntos de aproximación generados con IBEA, NSGA-II y SPEA2.....	135
Figura 8.6. Tres soluciones no dominadas para el problema de Mandl.....	137
Figura 8.7 Población final obtenida luego de 200 generaciones para el problema de Mandl.....	138

Lista de algoritmos

Algoritmo 4.1 Esquema de un Algoritmo Evolutivo	57
Algoritmo 7.1 Esquema de GRASP	108
Algoritmo 7.2 Procedimiento de búsqueda local	109
Algoritmo 7.3 Pseudo-código del AHDM para el evolutivo inspirado en NSGA-II	120
Algoritmo 7.4 Pseudo-código del AHDM para la implementación de SPEA2	121
Algoritmo 7.5 Pseudo-código del AHDM para la implementación de IBEA	121

Lista de tablas

Tabla 1.1 El Proceso de Planeamiento del Tránsito (Ceder y Wilson, 1986). Los números incorporados en las actividades corresponden a las secciones de este capítulo que las describen	16
Tabla 4.1 Diferencias clave entre los diferentes algoritmos evolutivos.	58
Tabla 5.1 Relaciones entre distintos conjuntos de soluciones	85
Tabla 6.1 Notación para sistemas de colas	100
Tabla 8.1 Distancias del mapa de cada nodo desde el nodo 1	126
Tabla 8.2 Matriz OD para la Figura 8.1 (viajes de la mañana)	126
Tabla 8.3 Matriz OD: Valores del parámetro α de la distribución de probabilidad exponencial que modela los arribos (viajes de la mañana)	126
Tabla 8.4 Matriz OD de la ciudad de Mandl (1979)	128
Tabla 8.5 Matriz OD con los parámetros α de las distribuciones de probabilidad para los arribos de los clientes (jornada completa)	129
Tabla 8.6 Frecuencia de los buses para cada escenario	129
Tabla 8.7 Ejemplos de cuatro soluciones no dominadas para los escenarios de la Tabla 8.6	131
Tabla 8.8 Resultados estadísticos	132
Tabla 8.9 Test de Fisher aplicado a los indicadores de calidad	134
Tabla 8.10 Resultados estadísticos de los diferentes algoritmos para resolución del problema de Mandl	135

Abreviaturas

AHDM Algoritmo Híbrido Dinámico Multiobjetivo
DM decision maker
DPGA Algoritmo genético Pareto basado en la distancia
EA Evolutionary Algorithm
ES Estrategia Evolutiva
EP Evolutionary Programming
ER Error Ratio
GA Genetic Algorithm
GRASP Greedy Randomized Adaptive Search Procedure
IBEA indicator-based evolutionary algorithm
JSSP Job Shop Scheduling Problem
MINLP Mixed Integer Non-Linear Problem
MOEA Multi-objective Evolutionary Algorithm
MOGA Algoritmo genético con múltiples objetivos
MOOP Multi-Objective Optimization Problem
NP Non-Deterministic Polynomial-time
NSGA Algoritmo genético de ordenamiento no dominado
NSGA-II Algoritmo genético de ordenamiento no dominado elitista
OD origen-destino
OX Cruzamiento por orden
PISA Platform and Programming Language Independent Interface for Search Algorithms
PMX Cruzamiento por emparejamiento parcial
QSAP Quadratic Semi Assignment Problem
SPEA Strength Pareto Evolutionary Algorithm
SPEA2 Improving the Strength Pareto Evolutionary Algorithm 2
TNP Transit Network Problem
TN red de tránsito
TNDP Transit Network Design Problem
TNDSP Transit Network Design Scheduling Problem
TNDFSP Transit Network Frequency and Design Scheduling Problem
TNFSP Transit Network Frequency Setting Problem
TNSP Transit Network Frequency Setting and Timetabling Problem

TNTP Transit Network Timetabling Problem

TSP Travel Salesman Problem

VEGA algoritmo genético con vector de evaluación

WBGA Algoritmo genético basado en pesos

Nomenclatura

x solución a un problema

y solución a un problema

u número de padres

λ número de descendientes

I espacio de individuos

$f(x)$, $g(x)$ valor de la función de evaluación de x o *fitness* para un problema mono-objetivo

$f_i(x)$, $g_i(x)$ valor del objetivo i -ésimo para x en un problema multi-objetivo

$F(x)$ vector de funciones objetivo para un problema multi-objetivo

Ω el espacio de soluciones

$x \leq y$ x domina parcialmente a y .

$x_1 \prec x_2$ x_1 domina fuertemente a x_2

P^* conjunto Pareto óptimo

PF^* Pareto Frontier

P conjunto de soluciones

P' conjunto de soluciones débilmente dominadas

p_i individuo p_i utilizado como padre

Q conjunto de N soluciones

h_i individuo hijo i

k número de objetivos

Φ función de *fitness* multi-objetivo

w coeficiente de peso

f_i^{\min} y f_i^{\max} los valores mínimo y máximo respectivamente para la función i -ésima

σ_{sharing} factor de participación

A , B conjuntos de soluciones

ω frecuencia de los buses