

UNIVERSIDAD DE BURGOS

ESCUELA POLITÉCNICA SUPERIOR

PROGRAMA DE DOCTORADO: "INVESTIGACIÓN EN INGENIERÍA"



**OPTIMIZACIÓN MULTI OBJETIVO DEL
TRANSPORTE DE PERSONAS DISCAPACITADAS.
DISEÑO DE NUEVAS METODOLOGÍAS
METAHEURÍSTICAS.**

Tesis doctoral presentada por:

Lara Velasco Carrera

Bajo la dirección de los doctores:

Joaquín A. Pacheco Bonrostro

Hernán Gonzalo Orden

Burgos 2017

Quiero expresar mi agradecimiento:

A los Doctores Pacheco y Gonzalo, codirectores de este trabajo, gracias a los cuales se ha podido llevar a cabo el mismo.

A Joaquín, Nina, Daría, Alexandra, Irma, Yajaira, Juanjo, Iris, José Ramón y “El Maño”, por hacer de los últimos días de este periplo un momento único.

A mis incondicionales: mis padres, mi tía y Javi, por su apoyo, paciencia y ánimo constantes, que han sido esenciales a la hora de enfrentarme a este trabajo.

A todos ellos,

Tan sólo decir,

Gracias y Gracias.

ÍNDICE

INTRODUCCIÓN

1.- LA OPTIMIZACIÓN	15
1.1.- INTRODUCCIÓN.....	15
1.2.- TÉCNICAS DE OPTIMIZACIÓN.....	16
1.3.- TÉCNICAS EXACTAS.....	17
1.4.- MÉTODOS DE APROXIMACIÓN HEURÍSTICOS.....	17
1.5.- MÉTODOS DE APROXIMACIÓN METAHEURÍSTICOS.....	20
1.6.- CLASIFICACIÓN DE LOS MÉTODOS METAHEURÍSTICOS.....	22
1.6.1.- <i>Basados en trayectoria</i>	22
1.6.1.1.- Recocido simulado [SA – Simulated Annealing].....	23
1.6.1.2.- Búsqueda con entorno variable [VNS – Variable Neighbourhood Search].....	24
1.6.1.3.- Búsqueda tabú [TS – Tabu Search].....	25
1.6.1.4.- Búsqueda ávida, aleatoria y adaptativa [GRASP – Greedy Randomized Adaptive Search Procedure].....	26
1.6.1.5.- Búsqueda local iterativa [ILS – Iterated Local Search].....	26
1.6.1.6.- Búsqueda voraz iterativa [IG – Iterated Greedy].....	26
1.6.1.7.- Re-encadenamiento de trayectorias [PR – Path Relinking].....	27
1.6.2.- <i>Basados en población</i>	27
1.6.2.1.- Algoritmos evolutivos [EA – Evolutionary Algorithms].....	27
1.6.2.2.- Optimización basada en colonias de hormigas [ACO – Ant Colony Optimization].....	28
1.6.2.3.- Optimización basada en cúmulos de partículas [PSO – Particle Swarm Optimization].....	29
1.6.2.4.- Algoritmos de estimación de la distribución [EDAs – Estimation of Distribution Algorithms].....	30
1.6.2.5.- Búsqueda dispersa [SS – Scatter Search].....	30
1.6.2.6.- Algoritmos genéticos [GA – Genetic Algorithms].....	31
1.6.2.7.- Algoritmos meméticos [MA – Memetic Algorithms].....	32
1.6.3.- <i>Otros</i>	32
2.- LA PROGRAMACIÓN MULTIOBJETIVO CON METAHEURÍSTICOS	37
2.1.- INTRODUCCIÓN.....	37
2.2.- FORMULACIÓN DEL PROBLEMA MULTIOBJETIVO.....	37
2.3.- FINALIDAD DE LA PROGRAMACIÓN MULTIOBJETIVO.....	39
2.4.- METAHEURÍSTICOS EN PROBLEMAS MULTIOBJETIVO.....	39
2.4.1.- <i>Algoritmos aplicados basados en trayectoria</i>	40
2.4.1.1.- Recocido Simulado.....	40
2.4.1.2.- Búsqueda Tabú.....	40
2.4.1.3.- GRASP.....	41
2.4.2.- <i>Algoritmos aplicados basados en población</i>	41
2.4.2.1.- Basados en el orden de Pareto.....	42
2.4.3.- <i>Otras tendencias aplicadas</i>	44
3.- LA BÚSQUEDA TABÚ	47
3.1.- INTRODUCCIÓN.....	47
3.2.- DESCRIPCIÓN Y CARACTERÍSTICAS.....	47
3.3.- LA LISTA TABÚ.....	49

3.4.- LOS CRITERIOS DE ASPIRACIÓN.....	50
3.5.- LA INTENSIFICACIÓN Y LA DIVERSIFICACIÓN.....	51
3.5.1.- <i>La intensificación</i>	51
3.5.2.- <i>La diversificación</i>	52
3.6.- LA MEMORIA.....	53
3.6.1.- <i>La Calidad</i>	53
3.6.2.- <i>La Influencia</i>	53
3.6.3.- <i>La Memoria a Corto Plazo</i>	54
3.6.4.- <i>La Memoria a Largo Plazo</i>	56
3.7.- OTRAS ESTRATEGIAS	57
3.7.1.- <i>Path Relinking</i>	57
3.7.2.- <i>Oscilación Estratégica</i>	58
4.- EL DIAL-A-RIDE PROBLEM (DARP).....	61
4.1.- INTRODUCCIÓN	61
4.2.- DESCRIPCIÓN DEL PROBLEMA.....	62
4.2.1.- <i>Los agentes implicados</i>	62
4.2.1.1.- El cliente.....	62
4.2.1.2.- El conductor	62
4.2.1.3.- La compañía de transporte	63
4.2.2.- <i>Los problemas de rutas de vehículos</i>	63
4.2.2.1.- VRP con carga y descarga	63
4.2.2.2.- VRP con ventanas de tiempo.....	64
4.2.3.- <i>Las características generales del DARP</i>	64
4.2.4.- <i>La formulación del problema</i>	66
5.- EL TRANSPORTE PÚBLICO COLECTIVO	71
5.1.- INTRODUCCIÓN	71
5.2.- EL TRANSPORTE PÚBLICO COMO RETO SOCIAL EN EUROPA.....	72
5.3.- EL TRANSPORTE PÚBLICO URBANO E INTERURBANO	73
5.4.- LAS APLICACIONES AL TRANSPORTE ESCOLAR.....	74
6.- EL TRANSPORTE DE PERSONAS DE MOVILIDAD REDUCIDA.....	77
6.1.- INTRODUCCIÓN	77
6.2.- LAS PERSONAS DE MOVILIDAD REDUCIDA.....	78
6.3.- LOS SERVICIOS ESPECÍFICOS DE TRANSPORTE PUERTA-A-PUERTA.....	79
6.4.- LOS MODELOS PLANTEADOS Y LAS TÉCNICAS APLICADAS.....	80
7.- EL PLANTEAMIENTO DEL PROBLEMA	85
7.1.- MOTIVACIÓN	85
7.2.- NOTACIÓN Y DESCRIPCIÓN DE OBJETIVOS.....	87
7.3.- FORMULACIÓN MATEMÁTICA DEL MODELO	90
7.4.- PROCEDIMIENTOS DE SOLUCIÓN.....	92
7.4.1.- <i>Procedimiento Constructivo</i>	93
7.4.2.- <i>Procedimiento de Búsqueda Tabú</i>	94

8.- LAS ESTRATEGIAS DE ACELERACIÓN.....	101
8.1.- INTRODUCCIÓN.....	101
8.2.- EL CHEQUEO DE LA FACTIBILIDAD.....	101
8.2.1.- <i>La restricción de capacidad máxima de los vehículos.....</i>	<i>102</i>
8.2.2.- <i>La restricción de tiempo máximo de duración de cada ruta.....</i>	<i>103</i>
8.2.3.- <i>La restricción de par completo (i +, i-) en la misma ruta.....</i>	<i>104</i>
8.2.4.- <i>La restricción de precedencia (i + antes que i-).....</i>	<i>105</i>
8.3.- EL CÁLCULO DE LAS FUNCIONES OBJETIVO.....	107
8.3.1.- <i>La función objetivo económica.....</i>	<i>107</i>
8.3.2.- <i>La función objetivo social.....</i>	<i>107</i>
9.- LAS METODOLOGÍAS DE RESOLUCIÓN.....	117
9.1.- INTRODUCCIÓN.....	117
9.2.- EL MÉTODO MOAMP.....	117
9.2.1.- <i>Los procedimientos GeneradorSocial y GeneradorCostes.....</i>	<i>120</i>
9.2.2.- <i>La aportación de diversidad al procedimiento.....</i>	<i>121</i>
9.3.- EL MÉTODO E-CONSTRAINT HEURÍSTICO (MHEC).....	122
10.- LOS EXPERIMENTOS COMPUTACIONALES.....	127
10.1.- INTRODUCCIÓN.....	127
10.2.- LA DESCRIPCIÓN DE LAS INSTANCIAS.....	127
10.3.- EL AJUSTE DE LOS PARÁMETROS.....	129
10.3.1.- <i>El parámetro tenure del procedimiento TabuSearch.....</i>	<i>129</i>
10.3.1.1.- <i>El tenure asociado a la función objetivo social.....</i>	<i>129</i>
10.3.1.2.- <i>El tenure asociado a la función objetivo económico.....</i>	<i>130</i>
10.3.2.- <i>Los parámetros α y β del procedimiento GeneradorSocial.....</i>	<i>130</i>
10.3.3.- <i>Los parámetros α y β del procedimiento GeneradorCostes.....</i>	<i>131</i>
10.4.- LA EVALUACIÓN DE LAS ESTRATEGIAS DE ACELERACIÓN.....	132
10.4.1.- <i>Constructive Sin y Con Aceleración.....</i>	<i>136</i>
10.4.2.- <i>TabuSearch Sin y Con Aceleración.....</i>	<i>138</i>
10.5.- LA RESOLUCIÓN COMPARADA EN INSTANCIAS PEQUEÑAS CON CPLEX.....	140
10.6.- LA COMPARACIÓN DE LOS MÉTODOS HEURÍSTICOS EN INSTANCIAS DE MAYOR TAMAÑO.....	145
10.7.- LA APLICACIÓN A UN CASO REAL.....	158
10.7.1.- <i>El contexto.....</i>	<i>158</i>
10.7.2.- <i>Las instancias reales y pseudoreales.....</i>	<i>159</i>
10.7.3.- <i>Los resultados.....</i>	<i>168</i>
11.- CONCLUSIONES, APORTACIONES, REFLEXIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN..	197
11.1.- CONCLUSIONES.....	197
11.2.- APORTACIONES.....	199
11.3.- REFLEXIONES.....	201
11.4.- FUTURAS LÍNEAS DE INVESTIGACIÓN.....	206
12.- BIBLIOGRAFÍA.....	209

ÍNDICE DE FIGURAS

FIGURA - 1: DIAGRAMA DE LAS TÉCNICAS DE OPTIMIZACIÓN	16
FIGURA - 2: REPRESENTACIÓN GRÁFICA DE ESTRATEGIA DE BÚSQUEDA LOCAL DE MÍNIMO (DOWSLAND, 1993).....	20
FIGURA - 3: TIPOS DE TÉCNICAS METAHEURÍSTICAS	22
FIGURA - 4: MOVIMIENTOS PERMITIDOS EN BÚSQUEDA TABÚ DE MÍNIMOS.....	48
FIGURA - 5: INTENSIFICACIÓN EN TS.....	51
FIGURA - 6: DIVERSIFICACIÓN EN TS.....	52
FIGURA - 7: COMPONENTES DE LA MEMORIA A CORTO PLAZO (GLOVER, 1990).....	55
FIGURA - 8: LÍNEA DE AUTOBÚS ORIGINAL (IZDA.) Y MODIFICADA/SUAVIZADA (DCHA.) [ÁLVAREZ ET AL. (2010)].....	73
FIGURA - 9: SOLUCIÓN S: A + - B + - B-- A-.....	88
FIGURA - 10: SOLUCIÓN S': A + - B + - A-- B-.....	88
FIGURA - 11: INSERCIÓN DE I + E I- CON ALGÚN PUNTO INTERCALADO.....	93
FIGURA - 12: INSERCIÓN DE I + E I- DE FORMA CONSECUTIVA	93
FIGURA - 13: INTERCAMBIO DE DOS CADENAS DE RUTAS DIFERENTES (CROSS)	96
FIGURA - 14: CAMBIO DE UNA CADENAS DE UNA RUTA A OTRA (MOVE).....	96
FIGURA - 15: INTERCAMBIO ENTRE DOS CADENAS CONSECUTIVAS DE UNA MISMA RUTA (OR).....	96
FIGURA - 16: INTERCAMBIO ENTRE DOS CADENAS NO CONSECUTIVAS DE UNA MISMA RUTA (OR-GENERALIZADO).....	96
FIGURA - 17: RUTA DE EJEMPLO RL.....	105
FIGURA - 18: RUTA CON PETICIÓN QUE PERTENECE A PTO (x)	108
FIGURA - 19: RUTA CON PETICIONES QUE PERTENECEN A PT1 x, y, PT2 x, y Y PT3 x, y.....	108
FIGURA - 20: JERARQUÍA DE PROCEDIMIENTOS EN MOAMP	119
FIGURA - 21: COMPARATIVA DE TIEMPOS MEDIOS DE COMPUTACIÓN DEL PROCEDIMIENTO CONSTRUCTIVE POR TAMAÑO	136
FIGURA - 22: COMPARATIVA DE TIEMPOS MEDIOS DE COMPUTACIÓN DEL PROCEDIMIENTO TABUSEARCH POR TAMAÑO	138

ÍNDICE DE TABLAS

TABLA - 1: POSIBLES LOCALIZACIONES DE LOS PARES DE PUNTOS DE LAS DIFERENTES SOLICITUDES DE UNA RUTA ANTE UN INTERCAMBIO OR FACTIBLE.....	111
TABLA - 2: POSIBLES LOCALIZACIONES DE LOS PARES DE PUNTOS DE LAS DIFERENTES SOLICITUDES DE UNA RUTA ANTE UN INTERCAMBIO OR-GENERALIZADO FACTIBLE	112
TABLA - 3: VALORES DE N, M Y LMAX PARA GENERACIÓN DE LAS INSTANCIAS.	128
TABLA - 4: SUBCONJUNTO TRAININGSET DE INSTANCIAS UTILIZADAS EN EL AJUSTE DE PARÁMETROS.....	129
TABLA - 5: VALORES DE MEANRATVALUE PARA DISTINTOS TENURE EN EL PROCEDIMIENTO TABUSEARCH CON F1	130
TABLA - 6: VALORES DE MEANRATVALUE PARA DISTINTOS TENURE EN EL PROCEDIMIENTO TABUSEARCH CON F2	130
TABLA - 7: VALORES DE MEANRATVALUEGS PARA TODAS LAS COMBINACIONES DE A Y B EN EL GENERADORSOCIAL.....	131
TABLA - 8: VALORES DE MEANRATVALUEGC PARA TODAS LAS COMBINACIONES DE A Y B EN EL GENERADORCOSTES	132
TABLA - 9: TIEMPOS COMPUTACIONALES EN SEGUNDOS PARA LA EVALUACIÓN DE LAS ESTRATEGIAS DE ACELERACIÓN.....	133
TABLA - 10: RESUMEN DE RESULTADOS DE LA COMPARATIVA CONSTRUCTIVE POR TAMAÑOS (TIEMPO COMPUTACIÓN EN S.)	136
TABLA - 11: TEST ESTADÍSTICO SOBRE EL VALOR MEDIO DE LAS DIFERENCIAS DE CONSTRUCTIVE POR TAMAÑOS.....	137
TABLA - 12: RESUMEN DE RESULTADOS DE LA COMPARATIVA TABUSEARCH POR TAMAÑOS (TIEMPO COMPUTACIÓN EN S.).....	138
TABLA - 13: TEST ESTADÍSTICO SOBRE EL VALOR MEDIO DE LAS DIFERENCIAS DE TABUSEARCH POR TAMAÑOS	139
TABLA - 14: EVOLUCIÓN DE CPLEX PARA INSTANCIAS PEQUEÑAS	140
TABLA - 15: MÉTRICAS COMPARATIVAS DE MOAMP Y MHEC EN INSTANCIAS DE TAMAÑO PEQUEÑO	141
TABLA - 16: COMPARATIVA DE SOLUCIONES ÓPTIMOS DE PARETO EN INSTANCIAS DE TAMAÑO PEQUEÑO.....	143
TABLA - 17: MÉTRICAS COMPARATIVAS DE MOAMP Y MHEC EN INSTANCIAS DE TAMAÑO MEDIANO Y GRANDE (N > 6).....	145

ÍNDICE DE PSEUDOCÓDIGOS

PSEUDOCÓDIGO - 1: PROCEDIMIENTO CONSTRUCTIVE	94
PSEUDOCÓDIGO - 2: PROCEDIMIENTO TABUSEARCH	95
PSEUDOCÓDIGO - 3: DESCRIPCIÓN DE LA ADAPTACIÓN DE MOAMP AL PROBLEMA	118
PSEUDOCÓDIGO - 4: PROCEDIMIENTO GENERADORSOCIAL	120
PSEUDOCÓDIGO - 5: PROCEDIMIENTO GENERADORCOSTES.....	120
PSEUDOCÓDIGO - 6: MODIFICACIÓN DEL PROCEDIMIENTO CONSTRUCTIVE.....	121
PSEUDOCÓDIGO - 7: MÉTODO HEURÍSTICO_E-CONSTRAINT (MHEC).....	123

INTRODUCCIÓN

En este trabajo de investigación se desarrolla una herramienta metodológica basada en técnicas metaheurísticas para resolver el problema de optimización de diseño de rutas para trasladar a personas de movilidad reducida, personas de avanzada edad o personas con algún tipo de discapacidad, dirigiendo la aplicabilidad de los resultados del estudio al sector público local y regional.

El interés por este problema surge tras haber formado parte como investigadora y consultora, especializada en transporte, en proyectos que analizaban la movilidad de las personas mayores en el contexto de la sociedad actual. La creciente demanda que está experimentando el transporte de personas de movilidad reducida demuestra la necesidad de planificar y poner en marcha servicios eficientes de transporte colectivo a disposición de este tipo de usuarios.

Se pretende concentrar la atención en aquellos traslados que el colectivo de personas de movilidad reducida realiza a centros de tratamiento, centros de día e incluso instalaciones hospitalarias y educativas. Se busca, por tanto, optimizar los viajes diarios que, las personas pertenecientes a dicho colectivo, necesitan realizar por motivo salud o educación, asistidos en el marco de un servicio de transporte público colectivo.

Con el fin de planificar eficientemente las rutas, no sólo se busca la minimización de los costes, sino también minimizar los inconvenientes generados a los usuarios del servicio de transporte. El modelo que se formule se resolverá bajo un enfoque biobjetivo, planteando el problema con un objetivo de carácter económico y otro, muy importante, de carácter social. A través de éste, se quiere mejorar la calidad del servicio de transporte minimizando el tiempo excedente de viaje para el usuario de movilidad reducida, que, en este caso, presenta más sensibilidad a este factor que otros colectivos de la sociedad.

Se ha decidido formular el modelo ajustando el Dial-A-Ride Problem (DARP) estático sin ventanas de tiempo con flota fija de vehículos adaptados. La metodología empleada en su resolución pasa por el empleo de la estrategia MultiObjective Adaptive Memory Procedure (MOAMP), cuyo principal motor es un procedimiento de búsqueda tabú y que resulta ser muy efectiva en problemas multiobjetivo, así como otras estrategias empleadas en problemas de este mismo tipo. Las publicaciones científicas encontradas con un enfoque similar del problema son muy escasas, como se indica más adelante.

La existencia del grupo de investigación GRINUBUMET, en el Departamento de Economía Aplicada de la Universidad de Burgos, liderado por el Dr. Joaquín A. Pacheco Bonrosto y especializado en el desarrollo de estrategias heurísticas y metaheurísticas aplicables a diversos tipos de problemas y con gran número de publicaciones, permite abordar el mismo en condiciones óptimas.

La consecución del presente trabajo de investigación se enmarca dentro de los proyectos que se enumeran a continuación, vinculados al grupo de investigación arriba citado:

Búsqueda de la eficiencia y sostenibilidad de las decisiones públicas: un enfoque multi-criterio. Subvencionado por el Ministerio de Economía y Competitividad y Fondos FEDER (Proyectos I+D+i orientada a los retos de la sociedad, referencia ECO2016-76567-C4-2-R) 2017-2020.

Metaheurísticas e hiperheurísticas para problemas de transporte público con varios criterios. Aplicaciones a problemas logísticos relacionados. Subvencionado por la Junta de Castilla y León y Fondos FEDER (proyecto BU062U16) 2016-2018.

Diseño de técnicas metaheurísticas para la toma de decisiones problemas con múltiples objetivos. Aplicaciones a problemas relacionados con transporte público y recogida de residuos. Subvencionado por la Junta de Castilla y León (proyecto BU329U14) 2014-2017.

Metaheurísticos para problemas de logística del transporte con múltiples criterios. Aplicaciones a problemas reales en transporte urbano y residuos. Subvencionado por el Ministerio de Economía y Competitividad y Fondos FEDER (Proyectos I+D+i orientada a los retos de la sociedad, referencia ECO2013-47129-C4-3-R) 2014-2017.

El trabajo se estructura de la manera siguiente:

- El bloque de revisión del estado del arte comienza en el Capítulo 1 abordando el estado actual de las técnicas de optimización, tanto heurísticas como metaheurísticas. En el Capítulo 2, se hace una revisión de las metaheurísticas en problemas multiobjetivo. En el Capítulo 3, se describe, de forma más profunda, la búsqueda tabú. En el siguiente capítulo, se hace una descripción del DARP y sus variantes, así como los diferentes métodos de resolución. En los Capítulos 5 y 6 se revisa el transporte público colectivo y el transporte de personas de movilidad reducida, respectivamente.
- El bloque metodológico comienza en el Capítulo 7 realizando la descripción y planteamiento del problema. En el capítulo siguiente se desarrollan las estrategias de aceleración diseñadas para complementar a las metodologías de resolución propuestas, que se describen en el Capítulo 9. En el Capítulo 10, a fin de validar y contrastar el algoritmo, se presentan las pruebas computacionales ejecutadas y se incluye la aplicación del problema a un caso real.
- El tercer bloque, de capítulo único, compila las conclusiones, aportaciones y reflexiones que se derivan del trabajo de investigación, así como las futuras líneas de investigación.
- En el último capítulo se listan las referencias bibliográficas utilizadas.

1.- LA OPTIMIZACIÓN

1.- LA OPTIMIZACIÓN

1.1.- INTRODUCCIÓN

Atendiendo rigurosamente a razones semánticas, optimizar es buscar la mejor manera de realizar una actividad y así lo recoge la Real Academia Española en su diccionario editado en 2014. Dentro del ámbito científico, cuando nos encontramos ante un problema de optimización tratamos de hallar la mejor solución posible, conocida como solución óptima, entre todas las soluciones existentes. Nos enfrentamos a este tipo de problemas identificando el objetivo que se pretende optimizar (minimizar o maximizar), el cuál depende de una serie de variables de decisión.

En la vida diaria vamos constantemente solucionando pequeños problemas de optimización, como el mejor camino para ir de casa al trabajo, la elección de la tienda donde realizar la compra, la organización de nuestra agenda, etc. En muchos casos, estos problemas son lo suficientemente sencillos como para poderlos resolver sin ayuda adicional. Pero existe una gran variedad de ámbitos y sectores en los que se aplican problemas de optimización, tal y como describen Laguna y Delgado (2007), y por tanto la complejidad asociada a cada uno de ellos es igualmente variada.

De manera formal; el problema de optimización *se puede definir como un par (S, f) , donde $S \neq \emptyset$ es el espacio de soluciones del problema y f es una función objetivo que se formaliza como:*

$$f : S \rightarrow \mathbb{R}$$

Resolver un problema de optimización pasa por hallar una solución, $i^ \in S$, que cumpla la siguiente desigualdad:*

$$f(i^*) \leq f(i), \quad \forall i \in S \quad \text{Caso concreto de minimización}$$

$$f(i^*) \geq f(i), \quad \forall i \in S \quad \text{Caso concreto de maximización}$$

La Investigación Operativa, y más concretamente la Programación Matemática, es la rama científica de la Matemática Aplicada que se encarga de resolver este tipo de problemas. Son las características de las variables y de las ecuaciones y/o inecuaciones que describen los modelos matemáticos las que concretan el tipo de programación necesaria en cada problema. Así, se habla de Programación Lineal si todas las ecuaciones son lineales y de Programación No Lineal en el resto de los casos.

Se emplea Programación Multiobjetivo si hay más de un objetivo a optimizar. Cuando las variables pueden tomar sólo valores enteros, estamos ante Programación Entera, diferente del caso en el que asumen valores continuos que establece la Programación Continua; si los parámetros que describen el problema tienen valores fijos, se habla de Programación Determinística; en el caso de tener valores aleatorios, se habla de Programación Estocástica. Otra ramificación de la Programación Matemática, la Optimización Combinatoria trata de resolver problemas de optimización en los que hay un número finito de soluciones.

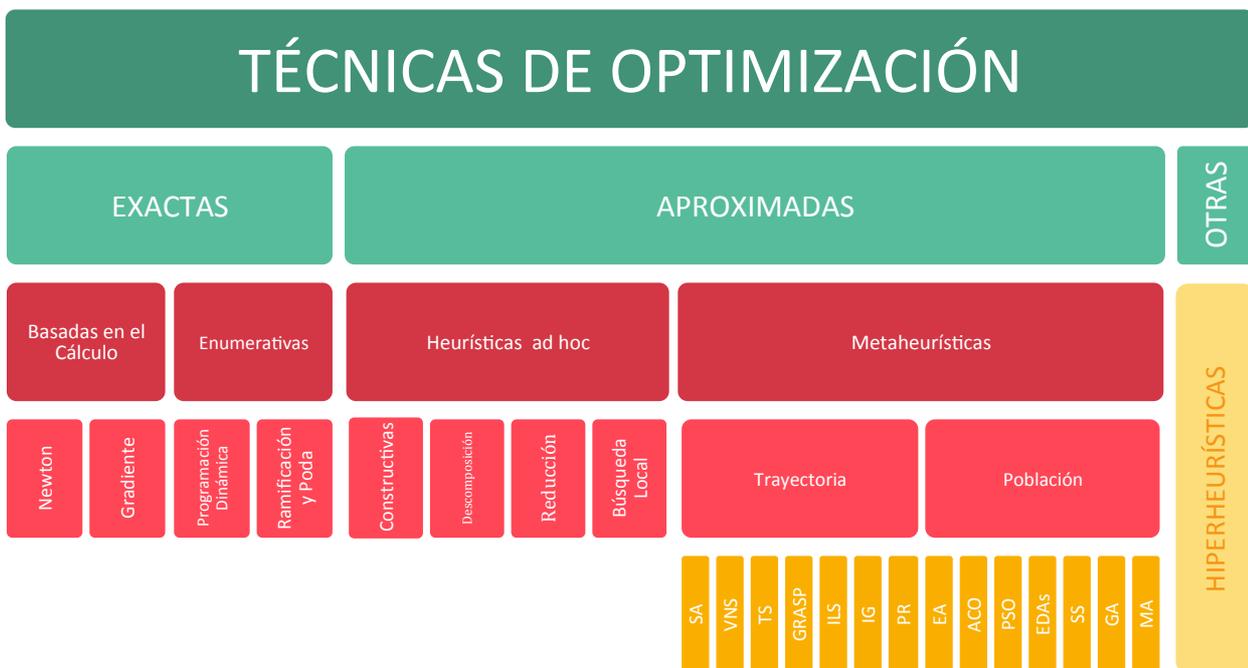
Algunos problemas de optimización son relativamente fáciles de resolver. Este es el caso, por ejemplo, de los problemas lineales continuos, en los que tanto la función objetivo como las restricciones son expresiones lineales y las variables son continuas. Esta tipología de problemas se pueden resolver, por ejemplo, con el método Simplex, publicado por Dantzig en 1947. Sin embargo, muchos otros tipos de problemas de optimización son muy difíciles de resolver porque no se ajustan a las condiciones de continuidad y tienen demasiadas variables. De hecho, la mayor parte de los que podemos encontrar en la práctica entran dentro de esta categoría (Crespo et al, 2007).

1.2.- TÉCNICAS DE OPTIMIZACIÓN

La resolución de los problemas de optimización es de gran importancia debido a que son clave para guiar a muchas organizaciones en la toma de decisiones en tareas tales como: la planificación de inversiones financieras, la gestión de recursos, la planificación y distribución de la producción, la planificación de cadenas de montaje, la planificación de la generación de energía, la planificación de los recursos hidráulicos, la agricultura, la política pública, el control del tráfico aéreo y la asignación de rutas de vehículos, están entre muchos otros campos de aplicación.

Las diversas técnicas existentes para resolver los problemas de optimización se puede clasificar de manera sencilla en técnicas exactas y aproximadas, tal y como se muestra en la Figura - 1.

Figura - 1: Diagrama de las técnicas de optimización



* [SA – Simulated Annealing], [VNS – Variable Neighbourhood Search], [TS – Tabu Search], [GRASP – Greedy Randomized Adaptive Search Procedure], [ILS – Iterated Local Search], [IG – Iterated Greedy], [PR – Path Relinking], [EA – Evolutionary Algorithms], [ACO – Ant Colony Optimization], [PSO – Particle Swarm Optimization], [EDAs – Estimation of Distribution Algorithms], [SS – Scatter Search], [GA – Genetic Algorithms] y [MA – Memetic Algorithms].

Las técnicas exactas (algoritmos óptimos o exactos) aseguran encontrar la solución óptima para cualquier problema en un tiempo acotado.

La principal desventaja de estos métodos es que el tiempo de resolución y la memoria computacional necesaria, aunque acotados, crecen exponencialmente con el tamaño del problema, ya que la mayoría de éstos son de la clase *NP-hard*, de dificultad no polinómica, (Garey, 1978). El hecho de no poder garantizar encontrar una solución en un tiempo razonable supone en muchos casos que el uso de técnicas exactas sea inviable. Por lo tanto, las técnicas aproximadas para resolver estos problemas están recibiendo una atención cada vez mayor por parte de la comunidad científica internacional en los últimos tiempos.

Los métodos aproximados (algoritmos de aproximación o heurísticos) obtienen una solución a los problemas en un tiempo razonable sin ofrecer garantías de que la solución encontrada sea la óptima.

1.3.- TÉCNICAS EXACTAS

Son métodos de resolución exactos aplicados a problemas *NP-Hard* (problema difícil no determinista en tiempo polinómico) aquellos que aseguran la obtención del óptimo del problema.

Existen algoritmos exactos basados en el cálculo que requieren herramientas de análisis matemático para determinar para qué valores del dominio la función presenta un máximo o un mínimo. Son métodos de optimización muy robustos, pero que requieren mucho esfuerzo computacional y que tanto la función objetivo como las restricciones presenten determinadas condiciones (por ejemplo, que estén definidas, que sean continuas, etc.). En general, debido a que muchos problemas no se ajustan a las condiciones de continuidad y tienen demasiadas variables no se suele apelar a estos métodos, ya que no presentan una buena eficiencia.

Entre las técnicas clásicas fundamentadas en enumeración destacan los métodos de programación dinámica y los métodos de enumeración implícita (ramificación, poda, acotación, planos de corte, etc.). La técnica de ramificación y poda, “branch-and-cut”, consiste en una relajación del problema empleando un método de ramificación y acotación, “branch-and-bound”, para resolver los problemas de tipo lineal con variables enteras (Nemhauser y Wolsey, 1988). Para resolver los distintos problemas que se van generando se incluye una técnica de hiperplanos de corte. El potencial de estas técnicas es que no realizan una exploración del conjunto de soluciones una a una, algo imposible de llevar a cabo incluso con problemas de tamaño pequeño, sino que van dividiendo el conjunto de soluciones en subconjuntos y evitan explorar algunos de los subconjuntos de los que, por alguna propiedad (por ejemplo valor de cotas calculadas), se tenga la certeza de que no contiene la solución óptima. Trabajos concretos sobre la técnica exacta de ramificación y poda se muestran en Padberg y Rinaldi (1991), Jünger et al. (1995a), Jünger et al. (1995b) y Caprara y Fischetti (1997).

Como se ha expuesto anteriormente, existen otros tipos de problemas de optimización relativamente fáciles de resolver; éste es el caso de los problemas lineales, en los que, tanto la función objetivo como las restricciones, son expresiones lineales. Estos problemas pueden resolverse mediante un método, elaborado por Dantzig en 1947 y conocido como método Simplex.

1.4.- MÉTODOS DE APROXIMACIÓN HEURÍSTICOS

La palabra heurística tiene su origen en el término griego *heuriskein*, que significa ‘hallar, inventar’, raíz etimológica que comparte con eureka. La heurística es la *manera de buscar la solución de un problema, en algunas ciencias, mediante métodos no rigurosos, como por tanteo, reglas empíricas, etc.* (RAE, 2014).

En el campo de la optimización matemática se define como técnica heurística de resolución de problemas aquella que no garantiza que la solución que encuentren sea óptima. Así en la definición recogida por Zanakis y Evans (1981) dice que los algoritmos heurísticos son: *procedimientos simples, a menudo basados en el sentido común, que se supone ofrecerán una buena solución (aunque no necesariamente la óptima) a problemas difíciles, de un modo fácil y rápido.*

Los heurísticos experimentaron un gran crecimiento a partir de los años 70, apoyados por la proliferación de resultados en el campo de la complejidad computacional, y prueba de ello es el gran número de artículos publicados en las revistas de investigación operativa que tratan sobre estas técnicas aproximadas. Incluso la editorial Springer ha apostado por la edición de una publicación específica, la revista *Journal of Heuristics*. En ella y en otras, tales como *Journal of Memetics* e *International Journal of Metaheuristics*, se recogen los últimos avances en esta disciplina científica.

Aunque los métodos exactos son relevantes desde un punto de vista científico y académico, en la práctica el tiempo que pueden precisar para resolver un problema NP-Hard es de un orden de magnitud muy superior al de los heurísticos, pudiendo darse el caso de no encontrar ninguna solución óptima en un tiempo razonable. Por lo tanto, en la mayoría de los resoluciones de problemas NP-Hard es aconsejable el uso de métodos heurísticos.

Algunas circunstancias que justifican la aplicación de técnicas heurísticas con el fin de encontrar una buena solución de un problema en lugar de la óptima pueden ser:

- El hecho de que en muchos problemas reales los datos de partidas no sean exactos sino que se trate de buenas aproximaciones. Por ello aplicar un costoso procedimiento exacto en estos casos no sería eficiente.
- Si en los modelos no es posible incluir todas las consideraciones que afectan al problema por ser éstas difícilmente objetivables y/o difíciles de modelar, circunstancia que ocurre muy frecuentemente. En estas situaciones, será más útil para el decisor contar con varias buenas soluciones entre las que optar en función de sus propios criterios más que obtener una solución exacta.
- Cuando se necesitan tomar decisiones en tiempo real, lo que interesa es una buena decisión inmediata. Se puede sacrificar la garantía de solución óptima si se puede conseguir una solución aceptable en mucho menos tiempo.

Queda por tanto patente la flexibilidad que aportan las técnicas heurísticas a la hora de modelizar problemas difíciles en diversos escenarios. Es por ello que, así como en las técnicas exactas existe, por lo general, un método claro y conciso, en las heurísticas los modelos aplicados son únicos y específicos de cada tipo de problema a resolver.

Autores como Silver et al. (1980) proponen algunas clasificaciones de los diferentes tipos de heurísticos que existen, en muchas ocasiones según el modo en que buscan y construyen sus soluciones. Una de las posibles categorizaciones de los heurísticos es la siguiente:

A. MÉTODOS HEURÍSTICOS CONSTRUCTIVOS

Suele tratarse de métodos que añaden paulatinamente componentes individuales hasta obtener una solución factible. La tipología más popular de estos métodos la constituyen los algoritmos golosos o devoradores, también conocidos como *greedy methods*. Estos algoritmos van construyendo paso a paso la solución, haciendo la mejor elección en cada una de las iteraciones según algún criterio.

B. MÉTODOS HEURÍSTICOS DE DESCOMPOSICIÓN

Emplean estrategias de división del problema en sub-problemas más pequeños. La técnica de resolución consiste en emplear el resultado de un sub-problema como dato de entrada del siguiente, de forma que al resolverlos todos se obtiene una solución para el problema global.

C. MÉTODOS HEURÍSTICOS DE MANIPULACIÓN DEL MODELO

Buscan modificar la estructura del modelo con el objetivo de simplificar su resolución, deduciendo, en función de la solución obtenida para el problema manipulado, la solución factible del problema original. El espacio de soluciones obtenido a través de estos métodos puede reducirse o aumentar como consecuencia de la alteración estructural del modelo.

D. MÉTODOS HEURÍSTICOS DE REDUCCIÓN

Tratan de simplificar el problema restringiendo el espacio de soluciones del mismo. La simplificación, que tiene la desventaja de poder estar obviando buenas soluciones, pasa por identificar alguna característica que presumiblemente deba poseer la solución óptima e introducirlas como restricciones del problema.

E. MÉTODOS HEURÍSTICOS DE BÚSQUEDA LOCAL

Parten de una solución inicial ya completa y, aplicando el concepto de vecindario o exploración del entorno, recorren parte del espacio de búsqueda hasta encontrar un óptimo local. Sustituyen la solución inicial por la mejor solución de éstas y repiten el proceso hasta no encontrar una solución mejor. Consideran, entonces, que se ha obtenido un mínimo/máximo local. La solución obtenida a través de este algoritmo es un óptimo local, que puede coincidir con el óptimo global del problema.

Un óptimo local es, por tanto, una solución mejor o igual que cualquier otra solución de su vecindario. Estos métodos, partiendo de una solución inicial, se mueven, examinan su entorno y se quedan con el mejor vecino, continuando el proceso hasta que encuentran un óptimo local o vecinal.

Formalmente si la solución del problema de optimización combinatoria es s , se denota como $N(s)$ al vecindario de la solución s . $N(s)$ representa el conjunto de soluciones que se pueden construir aplicando un operador específico de modificación (generalmente denominado movimiento).

Definiendo de manera formal el proceso de resolución expuesto en esta sección, un algoritmo de búsqueda local:

- Parte de una solución inicial s_0 y calcula su vecindario $N(s_0)$.
- Encuentra una nueva solución s_1 mejor que s_0 .
- Reemplaza s_0 por s_1 .
- Repite el proceso hasta que en el vecindario no se encuentre una solución mejor.

En muchos problemas, la exploración completa del vecindario es computacionalmente inviable y hay que aplicar distintas estrategias, dando lugar a diferentes variaciones dentro de la metodología de la búsqueda local. Según el operador de movimiento elegido, el vecindario cambia y el modo de explorar el espacio de búsqueda también, pudiendo simplificarse o complicarse el proceso general de búsqueda.

El proceso de búsqueda local queda indiscutiblemente determinado al definir tanto el vecindario como el criterio de selección dentro del mismo. A su vez el vecindario depende de la estructura del problema de optimización y de la función objetivo definida para el mismo.

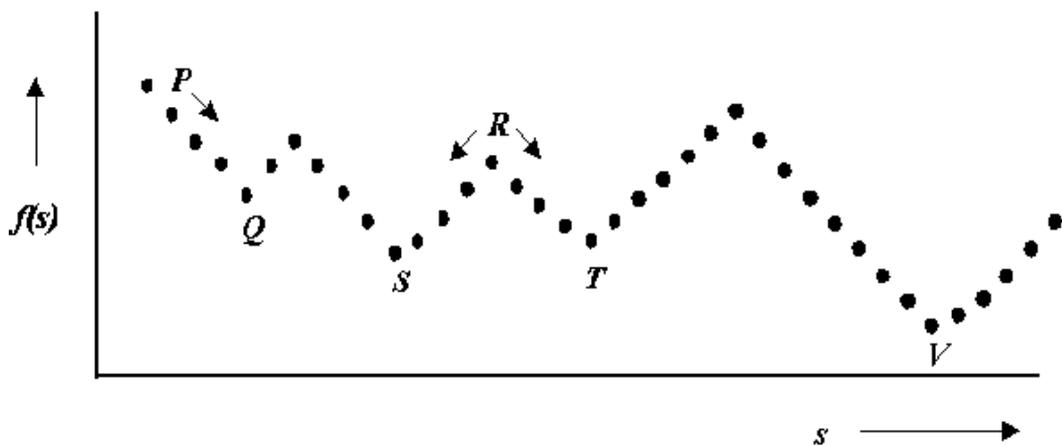
Para definir el criterio de selección de una nueva solución del vecindario, se puede, entre otras estrategias:

- buscar la mejor solución de $N(s)$ (estrategia *best improvement*) o
- tomar la primera solución de $N(s)$ que mejore la función objetivo (estrategia *first improvement*).

La búsqueda local es uno de los métodos heurísticos de resolución más aplicados tradicionalmente. No obstante, la gran desventaja de emplear esta metodología radica en el hecho de que la solución final obtenida es un óptimo local o vecinal que, en algunos casos puede coincidir con el óptimo global y que en otros, sin embargo, puede estar bastante alejado del valor de dicho óptimo global. Esto es debido a que la estrategia que usa no mira más allá de las soluciones vecinas (*estrategia miope*).

La Figura - 2 ilustra un problema de minimización donde la principal desventaja, antes explicada, de los métodos de búsqueda local y su gran dependencia de la solución inicial quedan patentes. En la figura, recogida en Dowsland (1993), cada solución del problema está representada por un punto de la gráfica. La búsqueda local en un caso concreto de minimización siempre nos lleva “al fondo del valle” donde se encuentra la solución inicial, como una pelota que rueda hacia abajo. Como se aprecia en la representación, si empezamos en la solución inicial P, siempre terminamos en Q, y no hay posibilidad de movernos ya que supondría empeorar. Se observa que el óptimo local Q, obtenido por este método, se aleja bastante del óptimo global V.

Figura - 2: Representación gráfica de estrategia de búsqueda local de mínimo (Dowsland, 1993)



Ahora bien, empeoramientos controlados podrían permitirnos salir del valle donde esta Q y llegar al valle donde esta S, y en definitiva alcanzar un mejor óptimo local. Esta idea de permitir empeoramientos controlados da pie a la aparición de nuevos algoritmos, denominados ‘metaheurísticos’, que evitan quedar atrapados en mínimos/máximos locales, permitiendo la realización de ciertos movimientos que empeoran temporalmente la función objetivo.

1.5.- MÉTODOS DE APROXIMACIÓN METAHEURÍSTICOS

Surgen como un paso más en el desarrollo de las estrategias de optimización, combinando diferentes métodos heurísticos a un nivel más elevado para conseguir una exploración más eficiente del espacio de búsqueda de soluciones.

Inicialmente estas técnicas eran conocidas como heurísticas modernas (Reeves, 1993) a pesar de que la denominación de metaheurísticas fue introducida por primera vez por Glover en 1986.

Los autores Osman y Kelly (1996) definen a este tipo de técnicas de optimización de la siguiente manera: “*Los procedimientos Metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que las heurísticas no son efectivas. Las Metaheurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos.*”

Otra definición que dieron Glover y Laguna (1997) dice que: “*Metaheurística se refiere a una estrategia maestra que guía y modifica otras heurísticas para producir soluciones más allá de aquellas que normalmente se generan en una búsqueda de óptimos locales.*”

En las publicaciones de Glover y Kochenberger (2003) así como en Blum y Roli (2003) se hacen revisiones detalladas de técnicas metaheurísticas. A continuación se enumeran las principales características que presentan este tipo de técnicas, resaltadas en las diferentes definiciones que de ellas se han consultado:

- Son algoritmos no exactos y, generalmente, no deterministas.
- Las estrategias metaheurísticas guían el proceso de búsqueda.
- Realizan una exploración eficiente del espacio de búsqueda.
- La estructura básica de cualquier metaheurística está predefinida.
- La estrategia de más alto nivel puede controlar heurísticos específicos que son añadidos a partir del conocimiento del problema que se trata de resolver.
- Pueden incorporar mecanismos para evitar las regiones menos favorables del espacio de búsqueda.
- Como en el caso de los algoritmos de aproximación heurísticos, son iterativos y disponen de un mecanismo de parada (número de iteraciones, número de iteraciones sin mejorar, acercarse al óptimo, previa fijación de un rango de acercamiento admisible, etc.)
- Su objetivo es encontrar soluciones cuasi-óptimas.

Condensando el listado anterior, una estrategia metaheurística conforma una plantilla general no determinista que, al completarse con los datos específicos que del problema se tienen, permite solucionar problemas cuyos espacios de búsqueda son casi inabarcables empleando otro tipo de técnicas.

Es muy relevante que el proceso de resolución consiga armonizar la exploración del espacio de búsqueda (evaluación de soluciones en regiones distantes) con la explotación del mismo (evaluación de soluciones en regiones acotadas). En dicho equilibrio reside la idoneidad y eficiencia de la estrategia metaheurística, que debe identificar rápidamente las regiones favorables del espacio de búsqueda sin emplear recursos en las regiones ya exploradas o carentes de soluciones de alta calidad.

Tal y como se representa en la Figura - 1, dentro de las estrategias de búsqueda metaheurísticas se suelen distinguir dos grandes tipos. Por un lado, las metaheurísticas basadas en trayectoria o búsqueda por entornos, diseñadas para evitar los mínimos locales y poder desplazarse a otras regiones favorables del espacio de búsqueda. El otro gran subgrupo consiste en las estrategias basadas en población o algoritmos evolutivos, que incorporan un mecanismo de aprendizaje para identificar las regiones del espacio de búsqueda con soluciones de alta calidad.

Sin embargo en la actualidad la mayoría de los métodos que se implementan son híbridos, de forma que los métodos basados en poblaciones incluyen búsquedas locales que por definición se realizan por medio de entornos.

Los autores Laguna y Delgado (2007) apuntan la reciente aparición de estrategias heurísticas que optan entre heurísticos para resolver problemas de optimización. Se refieren a este tipo de métodos como hiperheurísticos. El objetivo primero es diseñar estrategias de programación generales que puedan ser aplicadas a diferentes problemas. Los algoritmos metaheurísticos son técnicas ad hoc que funcionan bien para los problemas para los que se han diseñado. Al intentar aplicar dicha técnica a un tipo de problema diferente se deben realizar modificaciones en el método, lo que conlleva un consumo elevado de recursos. Otra característica propia de los hiperheurísticos, destacada por los autores, radica en que mientras un metaheurístico modifica las soluciones de forma directa, un hiperheurístico lo hace indirectamente, a través de un operador (heurístico de bajo nivel).

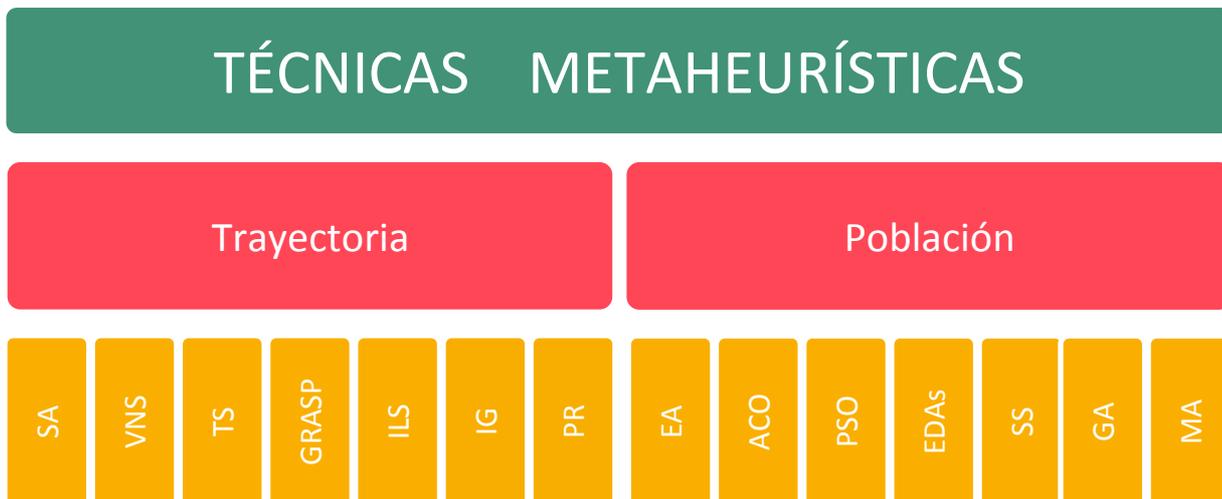
También en los artículos de Cowling et al. (2002) y Burke et al. (2003), se hacen investigaciones centradas en los denominados métodos hiperheurísticos.

1.6.- CLASIFICACIÓN DE LOS MÉTODOS METAHEURÍSTICOS

Existen diversas clasificaciones y ordenaciones de los métodos metaheurísticos, como los recogidos en Blum y Roli (2003), que van desde: basadas en la naturaleza y no basadas en la naturaleza, con memoria o sin ella, con una o varias estructuras de vecindario, etc.

La clasificación que se presenta en esta sección es una de las más extendidas y diferencia entre metaheurísticas basadas en trayectoria y las basadas en población. El rasgo más característico de las primeras es que manipulan en cada paso un único elemento del espacio de búsqueda, mientras que las segundas trabajan sobre un conjunto de ellos, también denominado población. En la Figura - 3 se presenta la anterior clasificación incluyendo las técnicas más representativas dentro de cada una de las dos categorías.

Figura - 3: Tipos de técnicas metaheurísticas



* [SA – Simulated Annealing], [VNS – Variable Neighbourhood Search], [TS – Tabu Search], [GRASP – Greedy Randomized Adaptive Search Procedure], [ILS – Iterated Local Search], [IG – Iterated Greedy], [PR – Path Relinking], [EA – Evolutionary Algorithms], [ACO – Ant Colony Optimization], [PSO – Particle Swarm Optimization], [EDAs – Estimation of Distribution Algorithms], [SS – Scatter Search], [GA – Genetic Algorithms] y [MA – Memetic Algorithms].

1.6.1.- Basados en trayectoria

La principal característica de esta tipología de metaheurísticos es que partiendo de una solución y, a través de la exploración del vecindario, van actualizando la solución inicial, configurando una trayectoria.

La mayor parte de estos algoritmos de optimización surgen como extensiones de los métodos simples de búsqueda local a los que se les añade algún mecanismo para explorar fuera de los mínimos locales. Es por ello que requieren de una condición de parada más elaborada que la de hallar un mínimo local. Habitualmente la búsqueda se termina en una de estas tres situaciones; cuando se encuentra una solución con una calidad aceptable, cuando se alcanza un número máximo predefinido de iteraciones, o cuando se detecta un estancamiento del proceso.

1.6.1.1.- Recocido simulado [SA – Simulated Annealing]

El recocido simulado es un algoritmo de búsqueda por entornos con un criterio probabilístico de aceptación de soluciones basado en termodinámica que fue inicialmente presentado por Kirkpatrick et al. (1983). El SA simula el tratamiento térmico conocido como recocido, en el cual, un material es calentado hasta el ablandamiento para ser luego enfriado lentamente, con el fin de mejorar algunas de sus propiedades, recuperar su estructura o eliminar las tensiones internas.

Es considerado como el primer algoritmo en emplear una estrategia explícita para escapar de los mínimos locales, siendo uno de los métodos metaheurísticos más antiguos. Debido a su simplicidad y versatilidad, al obtener buenos resultados en diferentes problemas, es un instrumento muy popular, con múltiples aplicaciones en los campos más variados. Caballero et al. (2007) indican que el recocido simulado tiene la característica de poder escapar fácilmente de óptimos locales, potenciando su transferibilidad y gran uso en aplicaciones prácticas en diferentes ámbitos de la ciencia y la ingeniería.

Los orígenes del algoritmo se encuentran en un mecanismo estadístico dentro del campo de la termodinámica, presente en el trabajo de Metropolis et al. (1953). En su trabajo modelizó el proceso de recocido simulando los cambios energéticos que se producen en un sistema de partículas al decrecer la temperatura, hasta que convergen a un estado estable, estado de congelación.

Según definen las leyes de la termodinámica a una temperatura T , la probabilidad de un incremento energético de valor δE se aproxima a $P(\delta E) = e^{(-\frac{\delta E}{kT})}$, siendo k la constante de Boltzmann. Teniendo en cuenta esta formulación, el algoritmo de Metropolis crea una perturbación aleatoria en el sistema y calcula los cambios de energía resultantes. Si se produce una caída energética, el cambio se acepta directamente; pero si se produce un incremento energético, el cambio se acepta con una probabilidad dada por la formulación anterior. Las perturbaciones se repiten hasta un número predeterminado de iteraciones en sentido decreciente de la temperatura, hasta que se produce el enfriamiento y termina el proceso de recocido.

La analogía con la optimización combinatoria radica en T , denominado normalmente ‘temperatura’, que pasa a ser un parámetro de control. Por tanto, una solución que implique el empeoramiento de la función objetivo, se aceptará según la probabilidad $e^{(-\frac{\delta E}{T})}$. Si T es bajo, será difícil que haya movimientos a peor y la función convergerá a un óptimo local. Si bien la termodinámica demuestra que si el enfriamiento es muy lento se converge a un óptimo global, aunque esto pueda conducir a tiempos de computación excesivamente largos. Diferentes trabajos demuestran que, aún empleando velocidades de enfriamiento elevadas, los resultados que se obtienen son muy eficientes.

Por consiguiente, para evitar quedar atrapado en un mínimo local, el algoritmo permite elegir con cierta probabilidad una solución cuyo valor sea peor que el de la solución actual. En cada iteración se elige, a partir de la solución actual s , una solución s_0 del vecindario $N(s)$. Si s_0 es mejor que s (es decir, tiene un mejor valor en la función), se sustituye s por s_0 como solución actual. Si la solución s_0 es peor, entonces es aceptada con una determinada probabilidad que depende de la temperatura actual T y de la diferencia entre ambas soluciones, $f(s_0) - f(s)$ (en el caso de minimización).

En líneas generales, se acepta que aunque es muy fácil hacer que el recocido simulado funcione, no es así de fácil hacer que funcione bien. Fueron Dowsland y Adenso (2001) los que, en su investigación, justificaron esta problemática por el hecho de que el recocido simulado no es propiamente un algoritmo, sino una estrategia heurística que necesita de varias decisiones para que quede totalmente diseñada.

Esta estrategia heurística necesita de varias decisiones para definirla que se pueden clasificar en genéricas y específicas. Las decisiones genéricas tienen que ver con los parámetros de la fase de enfriamiento: valores máximos y mínimos de la temperatura, la velocidad de enfriamiento y las condiciones de parada. Éstas se tratan detalladamente en Aarts y Lenstra (1997), Aarts y Korks (1989) y Hajek (1988).

Son las decisiones específicas las que definen el espacio de soluciones, la estructura del entorno, la función objetivo y la solución inicial. La interrelación de estas variables influye de forma considerable en la obtención de la solución final, factor que queda recogido en el trabajo de Hajek (1988).

Cualquier implementación de búsqueda local puede convertirse en una implementación SA al elegir elementos del entorno de modo aleatorio, aceptar automáticamente todos los movimientos hacia una mejor solución, y aceptar los movimientos a una solución peor de acuerdo con una probabilidad. Una investigación sobre las aplicaciones del SA se puede encontrar en Koulmas et al. (1994).

1.6.1.2.- Búsqueda con entorno variable [VNS – Variable Neighbourhood Search]

La búsqueda con entorno variable o Variable Neighborhood Search (VNS) es una metaheurística propuesta por Mladenovic (1995), posteriormente desarrollada en Hansen y Mladenovic (1997) y por Hansen y Mladenovic (2002). Esta técnica aplica una estrategia que permite el cambio entre diferentes vecindarios durante el proceso de búsqueda, evitando así el estancamiento en un óptimo local. El algoritmo es generalista y presenta muchos grados de libertad durante el diseño de variaciones e instancias concretas.

La metaheurística VNS, tal y como recogen Moreno y Mladenovic (2013), se basa en aprovechar metódicamente tres principios simples; 1. Un mínimo local con una estructura de entornos no lo es necesariamente con otra, 2. Un mínimo global es mínimo local con todas las posibles estructuras de entornos y 3. Para muchos problemas, los mínimos locales están relativamente próximos entre sí.

Los dos primeros principios proponen el empleo de varias estructuras de entornos en las búsquedas locales para resolver un problema de optimización. El tercer principio, con base empírica, indica que los óptimos locales aportan información sobre el óptimo global. Puede ser, por ejemplo, que tengan características comunes pero, generalmente, no se sabe cuáles son esas características. Es conveniente, por tanto, realizar un análisis de las proximidades de cualquier óptimo local buscando información que permita orientar la búsqueda hacia el óptimo global.

Se han planteado en la literatura diversas formas de extender la VNS para proporcionarla de algunas características adicionales y permitir dar solución a problemas de gran tamaño, pero intentando mantener la simplicidad del esquema básico. A continuación se enumeran algunas de las extensiones propuestas:

- la Búsqueda por Entornos Variables con descomposición (VNDS),
- la Búsqueda por Entornos Variables sesgada (SVNS),
- la Búsqueda por Entornos Variables paralela (PVNS).

Resulta común también encontrar estudios que consideran las extensiones por hibridación de la VNS. En algunos casos se han propuesto procedimientos heurísticos que en esencia explotan ideas del VNS y que pueden considerarse como casos específicos o extensiones de esta metaheurística como la búsqueda local iterada (Iterated Local Search ILS) o la búsqueda por entornos grandes (Large Neighbourhood Search, LNS).

1.6.1.3.- Búsqueda tabú [TS – Tabu Search]

La búsqueda tabú es una de las metaheurísticas que más exitosamente se han aplicado en la resolución de problemas de optimización combinatoria. Los fundamentos de este método fueron introducidos por Glover (1986), y están basados en las ideas formuladas en su trabajo anterior (Glover, 1977). Se enmarca dentro de la Programación mediante Memoria Adaptativa (Taillard et al., 2001), que se caracteriza por buscar en el vecindario, manejando información de los movimientos realizados anteriormente.

La idea esencial de la búsqueda tabú es el uso explícito de un historial de la búsqueda (una memoria a corto plazo), tanto para escapar de los mínimos locales como para implementar su estrategia de exploración y evitar buscar varias veces en la misma región. Esta memoria a corto plazo se efectúa con la composición de una lista tabú, donde se guardan atributos de las soluciones visitadas más recientemente para excluirlas de los próximos movimientos durante un número definido de iteraciones y así evitar que aparezcan ciclos.

Desde el punto de vista de la ejecución, guardar una lista de soluciones completas no es práctico, pues suele ser computacionalmente ineficiente. Por ello, generalmente, se suelen almacenar los movimientos que han llevado al algoritmo a generar esa solución o alguno de los atributos principales que definen la solución.

En cada iteración del algoritmo se elige la mejor solución entre las permitidas y la solución, o sus atributos, se añade a la lista tabú. Entonces, una posible nueva solución es declarada tabú si alguno de sus atributos está en dicha lista. Los elementos de la lista tabú permiten filtrar el vecindario, generando un conjunto reducido de soluciones elegibles.

Esta técnica debe incluir y definir un criterio de aspiración que permita aceptar una solución incluso si está prohibida debido a la lista tabú, aportando flexibilidad a la búsqueda. El criterio de aspiración más comúnmente usado es permitir soluciones que mejoren el valor de la mejor solución encontrada hasta el momento. Incluso se deben admitir movimientos que empeoren la solución actual, para ayudar a escapar de óptimos locales.

Con el fin de potenciar las propiedades de la TS y así ampliar su aplicabilidad en la resolución de problemas se han desarrollado estrategias complementarias como:

- La intensificación: concentrando la búsqueda dentro de una zona determinada del espacio de soluciones.
- La diversificación: explorando otras zonas del espacio de soluciones no visitadas con anterioridad.
- El re-encadenamiento de trayectorias: permitiendo la combinación de soluciones del espacio de soluciones factibles.
- La oscilación estratégica: no deteniendo la búsqueda en un punto donde, según las reglas usuales, se debería hacer como, por ejemplo, en la frontera del conjunto de soluciones factibles. Se acepta avanzar a partir de ese punto hasta una profundidad determinada y se retrocede en sentido contrario hasta alcanzar un nuevo punto de oscilación, desde donde volver a rebotar en sentido contrario, estableciendo el movimiento oscilatorio que da nombre al método. Esta estrategia combina elementos de la intensificación, al lado factible de la frontera, con rasgos de la diversificación, ya que de otra manera, el acceso a las nuevas regiones nos hubiera resultado imposible, en un proceso corto de movimientos.

1.6.1.4.- Búsqueda ávida, aleatoria y adaptativa [GRASP – Greedy Randomized Adaptive Search Procedure]

El procedimiento de búsqueda ávido, aleatorizado y adaptativo o Greedy Randomized Adaptive Search Procedure (GRASP) es una metaheurística simple que combina heurísticos constructivos con búsqueda local. Se dió a conocer a finales de los ochenta en el trabajo de Feo y Resende (1989) y una amplia descripción de su desarrollo se puede encontrar en trabajos posteriores de los mismos autores, Feo y Resende (1995, 1999). Un referencia completa y más reciente se encuentra en González-Velarde (2007).

GRASP es un procedimiento iterativo, compuesto de dos fases: primero la construcción de una solución mediante una función ávida y aleatoria (método multiarreglo) y después un proceso de mejora. La solución mejorada es el resultado de aplicar un procedimiento de búsqueda local a la solución construida.

El método constructivo usa una aleatoriedad controlada por una función ávida que orienta sobre cual puede ser el mejor elemento a introducir en la solución en cada paso. Va añadiendo paso a paso diferentes componentes a la solución parcial, que inicialmente está vacía. Los componentes que se añaden en cada paso son elegidos aleatoriamente de una lista restringida de candidatos. La segunda fase del algoritmo consiste en aplicar un método de búsqueda local para mejorar la solución generada. Este mecanismo de mejora puede tratarse de una técnica de mejora simple o incluir algoritmos más complejos como SA o TS.

1.6.1.5.- Búsqueda local iterativa [ILS – Iterated Local Search]

La búsqueda local iterativa o Iterated Local Search (ILS) es una metaheurística basada en un concepto, simple pero muy efectivo, de perturbación que se presentó inicialmente en los trabajos de Stützle (1999) y Lourenço et al. (2002). Los estudios más recientes sobre esta técnica se recogen en Lourenço et al. (2010), Juan et al. (2015), Domínguez et al. (2015) y Grasas et al. (2016).

En cada iteración, la solución actual es perturbada y, a esta nueva solución, se le aplica un método de búsqueda local para mejorarla. El mínimo local obtenido por el método de mejora puede ser aceptado como nueva solución actual si verifica un criterio de aceptación.

La importancia del proceso de perturbación es clara: si es demasiado pequeña puede que el algoritmo no sea capaz de escapar del mínimo local; por otro lado, si es demasiado grande, la perturbación puede hacer que el algoritmo sea como un método de búsqueda local con un reinicio aleatorio. Por ello, el método de perturbación debe generar una nueva solución que sirva como inicio a la búsqueda local, pero que no debe estar muy lejos del actual para que no sea una solución aleatoria.

El criterio de aceptación actúa como contra-punto, ya que filtra la aceptación de nuevas soluciones dependiendo de la historia de búsqueda y de las características del nuevo mínimo local.

1.6.1.6.- Búsqueda voraz iterativa [IG – Iterated Greedy]

La búsqueda voraz iterativa (IG, del inglés Iterated Greedy) se fundamenta en generar soluciones a distintos problemas de optimización iterando un proceso de destrucción parcial y reconstrucción voraz sobre una solución de partida.

Esta metodología, por su sencillez, ha sido presentada y reinventada en distintas ocasiones por varios autores con denominaciones diversas, siendo algunas de ellas: “Ruín and recreate”, “Iterative flattening” o “Strategic Oscillation” (Corberán et al., 2016).

Actualmente la IG se ha aplicado con éxito a un gran número de problemas de optimización. Entre los trabajos más relevantes y recientes cabe destacar Ruiz y Stützle (2007), Lozano et al. (2011), Bouamama et al. (2012), Fanjul y Ruiz (2010), García-Martínez et al. (2014) y Corberán et al. (2016).

A partir de una solución inicial se destruyen algunos elementos de esta solución para después volverlos a reconstruir, usando para ello una heurística voraz de alta calidad. La nueva solución se compara con la anterior y se aplica un criterio de aceptación.

Una destrucción y reconstrucción completa de la solución en cada iteración tendría muchas similitudes con GRASP mientras que una perturbación en vez de una destrucción-reconstrucción sería similar a ILS.

La metodología IG es sencilla de implementar si se conoce una heurística constructiva capaz y habitualmente deriva en algoritmos con muy pocos parámetros, fácilmente extensibles a otros problemas y con buenos rendimientos.

1.6.1.7.- Re-encadenamiento de trayectorias [PR – Path Relinking]

La PR, que fue propuesta originalmente como una estrategia dentro de la metaheurística búsqueda tabú, en los últimos años ha logrado alcanzar identidad propia. Entre los trabajos y autores que han investigado sobre esta técnica destacan: Glover (1989), (1994), Glover y Laguna (1993) y Glover et al. (2000).

El re-encadenamiento de trayectorias se basa en la generación de nuevas soluciones mediante la exploración de trayectorias que conectan soluciones de calidad elevada. Se inicia la búsqueda desde una de estas soluciones, llamada solución inicial, y se genera un camino en el espacio de vecindario que dirige la búsqueda hacia las otras soluciones, denominadas soluciones guía.

Su principio de funcionamiento es la exploración de un conjunto de soluciones en el camino entre pares de soluciones generadas previamente. Se pretende la incorporación de atributos de soluciones de alta calidad para crear una buena composición de atributos en la solución final.

Especial atención merece el método híbrido GRASP con PR que ha sido aplicado en numerosos problemas. Un ejemplo de esta hibridación GRASP-PR se puede encontrar en la investigación de Laguna y Martí (1999).

1.6.2.- Basados en población

Los métodos basados en población se caracterizan por trabajar con un conjunto de soluciones, usualmente denominado población, en cada iteración, a diferencia de los métodos basados en trayectoria, que únicamente manipulan una solución del espacio de búsqueda por iteración.

1.6.2.1.- Algoritmos evolutivos [EA – Evolutionary Algorithms]

A finales de los 60 y principios de los 70 distintos investigadores trataron de trasladar los principios de la evolución al campo de la algoritmia, dando lugar a lo que tradicionalmente se conocía como Evolutionary Computation y que ahora se llama Evolutionary Algorithms (Algoritmos Evolutivos). Como resultado de esta investigación se originaron diferentes modelos que pueden agruparse en tres grandes familias: Evolutionary Programming (Programación Evolutiva), Evolutions Strategies (EE, Estrategias de Evolución) y Genetic Algorithms (GA, Algoritmos Genéticos). Los GA se desarrollan más adelante en una sección específica debido a su relevancia.

Los algoritmos evolutivos son técnicas heurísticas avanzadas inspiradas en la teoría de la evolución natural. Esta tipología de técnicas sigue un proceso iterativo y estocástico que opera sobre una población de soluciones, denominadas en este contexto individuos, y que se basan en la abstracción de ciertos procesos de la teoría de la evolución darwiniana. Inicialmente, la población es generada típicamente de forma aleatoria, contando normalmente con la ayuda de un constructivo heurístico.

El esquema transversal de un algoritmo evolutivo comprende tres fases principales: selección, reproducción y reemplazo. El proceso completo es repetido hasta que se cumpla un criterio de terminación, que normalmente consiste en fijar la parada tras un número prefijado de iteraciones.

Es durante la fase de selección donde se escogen generalmente los individuos más aptos de la población actual que serán posteriormente recombinados en la fase de reproducción. Los individuos resultantes de la recombinación se alteran mediante un operador de mutación. Finalmente, a partir de la población actual y/o los mejores individuos generados se forma la nueva población, dando paso a la siguiente generación del algoritmo.

Los EA realizan una búsqueda global, ya que trabajan con una población de soluciones candidatas, y no con una sola solución candidata. Si a este factor se le añade que son de naturaleza estocástica, todo ello reduce la probabilidad de que estos algoritmos se queden atascados en óptimos locales e incrementan la probabilidad de encontrar el óptimo global.

Bajo la denominación genérica de EA se agrupa un cierto número de técnicas diferenciadas, las cuales parten todas de la misma idea fundamental, pero que suelen distinguirse tanto por sus características propias como por motivos históricos. Algunas referencias de trabajos sobre algoritmos evolutivos se encuentran en Bäck (1995) y en Alba y Cotta (2006).

1.6.2.2.- Optimización basada en colonias de hormigas [ACO – Ant Colony Optimization]

Los algoritmos de optimización basados en colonias de hormigas están inspirados en el comportamiento de las hormigas cuando buscan comida. Éstas son capaces de determinar la ruta óptima de ida y vuelta entre la colonia y la fuente de alimento.

Las hormigas son capaces de transmitirse entre ellas la información, a través del rastro (feromona) dejado por cada una en su trayectoria. Dicho rastro o feromona varía de intensidad dependiendo de la cantidad de hormigas que transiten por el mismo. Esta sustancia ayudará al resto de las hormigas a encontrar la comida. La comunicación indirecta entre las hormigas mediante el rastro de feromona las capacita para encontrar el camino más corto entre el nido y la comida. Este comportamiento es el que intenta simular este método para resolver problemas de optimización.

Explicado de forma más detallada el comportamiento de búsqueda de comida de una colonia de hormigas es el siguiente: inicialmente, las hormigas exploran el área cercana a su nido de forma aleatoria. Tan pronto como una hormiga encuentra comida, la lleva al nido. Mientras que realiza este camino, la hormiga va depositando una sustancia química denominada feromona. Otras hormigas seguirán el rastro dejado por la primera y, en caso de encontrar alimento, volverán por el mismo camino a depositarlo, con lo que vuelven a impregnar el mismo con más feromona, acrecentando la presencia de la misma.

Observando dicho fenómeno, se concluye que se produce una retroalimentación, y entonces la probabilidad de que una hormiga escoja una ruta, depende de la cantidad de hormigas que hayan transitado previamente por la misma. De esta forma, cuantas más hormigas sigan dicho trayecto, más atractivo se vuelve para ellas.

La técnica se basa en dos pasos principales: construcción de una solución basada en el comportamiento de una hormiga y actualización de los rastros de feromona artificiales. El algoritmo no fija ninguna planificación o sincronización a priori entre las fases, pudiendo ser incluso realizadas simultáneamente. La feromona depositada en cada ruta se ‘evapora’ según un parámetro, denominado coeficiente de evaporación, que constituye el porcentaje de feromona que permanece de una iteración a otra, lo que provoca que, en el caso de transcurrir un periodo de tiempo sin usar una ruta determinada, ésta pierda interés para las hormigas y surge con el objetivo principal de evitar la convergencia prematura del algoritmo con el paso del tiempo.

Las primeras referencias sobre este algoritmo se encuentran en Dorigo (1992). Éste mismo autor sigue desarrollando esta teoría en sus posteriores estudios Dorigo y Stützle (2003) y (2004) incluso lo aplica al problema del agente viajero (TSP por sus siglas en inglés) en Dorigo et al. (1996). Otros autores como Gravel et al. (2002) y Guntsch y Middendorf (2003) adaptaron la ACO a la resolución de problemas de Programación Multiobjetivo obteniendo muy buenos resultados.

1.6.2.3.- Optimización basada en cúmulos de partículas [PSO – Particle Swarm Optimization]

Esta metodología parte de experimentos realizados con algoritmos que modelizaban el comportamiento en vuelo de algunas especies de pájaros, el comportamiento de los bancos de peces e, incluso, las tendencias sociales en el comportamiento humano.

La PSO fue desarrollada por el psicólogo-sociólogo James Kennedy y por el ingeniero electrónico Russell Eberhart en 1995. Referencias completas sobre esta técnica de optimización se encuentran en Kennedy (1999) y en Kennedy y Eberhart (2001). Una publicación completa más reciente sobre esta estrategia metaheurística se puede encontrar en Poli et al. (2007).

El paralelismo que establecen este tipo de algoritmos con el comportamiento social se puede sintetizar de la siguiente forma: los individuos que son parte de una sociedad tienen una opinión influenciada por la creencia global compartida por todos los posibles individuos. Cada individuo, puede modificar su opinión (estado) dependiendo de tres factores:

- Su adaptación - el conocimiento del entorno,
- Su memoria - los cambios en la historia por los que ha pasado el individuo y
- La memoria del vecindario - los eventos en la historia de los individuos cercanos.

El algoritmo PSO mantiene un conjunto de soluciones, también llamadas partículas, que son inicializadas aleatoriamente en el espacio de búsqueda. Cada partícula posee una posición y velocidad que cambia conforme avanza la búsqueda y se va moviendo en un espacio multidimensional que representa su espacio social.

Cada movimiento de una partícula se compone de una velocidad, que inicialmente es aleatoria, y dos valores ponderados aleatoriamente: individual (tendencia de las partículas a preservar su mejor estado anterior) y social (tendencia a moverse hacia vecinos con mejor posición). El movimiento de las partículas por el espacio está guiado por las partículas óptimas en el momento actual. Cada partícula tiene memoria mediante la que retiene parte de su estado anterior.

En el contexto de PSO, el vecindario de una partícula se define como un conjunto de partículas del cúmulo. El vecindario de una partícula puede ser global, en el cual todas las partículas del cúmulo se consideran vecinas, o local, en el que sólo las partículas más cercanas se consideran vecinas.

Este tipo de algoritmos se adaptan muy bien a problemas matemáticos de carácter continuo, habiéndose también aplicado con éxito en problemas de naturaleza discreta. El PSO se ha empleado en programación multiobjetivo, con el ejemplo reciente en Santana et al. (2006) de hibridación de esta estrategia con búsqueda dispersa para ampliar el conjunto de soluciones no dominadas.

Comparte algunas similitudes con los algoritmos genéticos ya que comienza con un conjunto de soluciones aleatorias y busca la solución óptima actualizando generaciones. Sin embargo, PSO no emplea operadores evolutivos como el cruce o la mutación.

1.6.2.4.- Algoritmos de estimación de la distribución [EDAs – Estimation of Distribution Algorithms]

Son unos metaheurísticos que muestran un comportamiento similar a los algoritmos evolutivos, incluso encontrando muchos autores que consideran los EDAs como otro tipo de EA.

Los algoritmos evolutivos se desarrollan dependiendo de una serie de factores (operadores de cruce, operadores de mutación, tamaño de la población, número de generaciones, etc.). Determinar cuáles son esos factores se puede convertir, por sí mismo, en un problema de optimización. Esta problemática queda recogida por Grefenstette (1986). También se suma, a lo anteriormente expuesto, la dificultad de la predicción de los movimientos de la población de individuos en el espacio de búsqueda. Éste hecho se desarrolla en profundidad en el trabajo de Larrañaga et al. (2003).

Por todo lo anterior surge la necesidad de un nuevo tipo de algoritmos denominados algoritmos de estimación de distribuciones, inicialmente presentados por Mühlenbein y Paaß (1986). Investigaciones más exhaustivas sobre los EDAs se pueden localizar en Larrañaga et al. (1999) y Pelikan et al. (1999).

Los EDAs operan sobre una población de soluciones tentativas pero incluyen la distribución de probabilidad del conjunto seleccionado y generan nuevas soluciones que formarán parte de la población. Es decir, la nueva población de individuos es obtenida mediante la simulación de una distribución de probabilidad estimada a partir de datos que contienen atributos de la población anterior.

El algoritmo parte de la composición de manera aleatoria de una población de individuos y sobre ella se aplican tres tipos de operaciones:

- en primer lugar se genera un subconjunto de las mejores soluciones;
- en segundo lugar se realiza un proceso de aprendizaje de un modelo de distribución de probabilidad a partir del sub-conjunto generado,
- y, en tercer lugar, se generan nuevos individuos simulando dicha distribución.

Con el fin de representar eficientemente la distribución de probabilidad en el campo de los EDAs se emplean habitualmente los modelos gráficos probabilísticos. Para representar la distribución de probabilidad en dominios discretos algunos autores han propuesto las redes bayesianas, como por ejemplo Soto et al. (1999). Mientras que en los dominios continuos la literatura existente emplea usualmente las redes gaussianas (Whittaker, 1990).

1.6.2.5.- Búsqueda dispersa [SS - Scatter Search]

La SS es un método evolutivo que ha sido aplicado en la resolución de un gran número de problemas de optimización. Los principios fundamentales del método, fueron propuestos en la década de los setenta (Glover, 1977) y consisten en estrategias para combinar reglas de decisión, especialmente en problemas de secuenciación, así como en la combinación de restricciones o método de las restricciones subrogadas.

Se basa en el principio de que la información sobre la calidad o el atractivo de un conjunto de reglas, restricciones o soluciones puede ser adoptado mediante la combinación de éstas. En concreto, dadas dos soluciones, se puede obtener una nueva mediante su combinación de modo que mejore a las que la originaron.

Mantiene un conjunto de soluciones y realiza combinaciones con éstas; las elecciones son sistemáticas y estratégicas y se realizan sobre un conjunto relativamente pequeño de soluciones. El hecho de que las elecciones son sistemáticas y sobre un conjunto pequeño es lo que le diferencia de los métodos evolutivos, que se fundamentan en métodos aleatorizados.

La búsqueda dispersa opera sobre un conjunto de referencia, formado por un conjunto relativamente pequeño de soluciones tentativas llamado conjunto de referencia, caracterizado por contener soluciones de calidad y que son diversas, es decir, distantes en el espacio de búsqueda. Posteriormente se combinan estas soluciones entre sí a fin de encontrar otras soluciones que mejoren las iniciales. Con dicho objetivo se suele aplicar un procedimiento de mejora local, ya sea por búsqueda local o, incluso, por búsqueda tabú.

Para la definición completa de SS hay que concretar los cinco componentes o fases elementales que se enumeran a continuación:

- creación de la población inicial
- selección del conjunto de referencia
- generación de subconjuntos de soluciones
- aplicación del método de combinación de soluciones
- ejecución del método de mejora

Revisiones sobre el origen y las aplicaciones diversas de SS se pueden encontrar en Glover (1998), Laguna y Martí (2000), Laguna (2002), Laguna y Martí (2003), Glover et al. (2003a), Glover et al. (2003b), Casado y Martí (2007) y Martí et al. (2015), entre otros.

1.6.2.6.- Algoritmos genéticos [GA – Genetic Algorithms]

Los algoritmos genéticos son el representante más conocido de los algoritmos evolutivos, y aquellos cuyo uso está más extendido. Fueron concebidos originalmente por John Holland y presentadas en el *Adaptation in Natura and Artificial Systems* (Holland, 1975).

Los algoritmos genéticos nacieron con el objetivo de imitar los procesos adaptativos de los sistemas naturales mediante el diseño de sistemas artificiales que replican los mecanismos importantes de dichos sistemas naturales. Los algoritmos genéticos son métodos sistemáticos para la resolución de problemas de búsqueda y optimización que aplican a estos los mismos principios de la evolución biológica: selección basada en la población, reproducción y mutación.

Manipulan un conjunto de soluciones que se denomina población, denominando a cada uno de sus elementos individuo. Aplican, como mecanismo de búsqueda, una serie de operaciones de selección, mutación y cruce, inspiradas en la naturaleza tal y como se describe por Goldberg (1994). Hay que destacar que los algoritmos genéticos tienen una componente aleatoria muy significativa.

La característica distintiva de los algoritmos genéticos es el empleo de operadores genéticos sobre cromosomas (codificación, binaria normalmente, de una solución - genotipo), a diferencia de otras estrategias evolutivas: Estrategias de Evolución, Programación Evolutiva y Programación Genética, que enfatizan cambios a nivel de individuos y especies.

El algoritmo asigna a cada solución un cromosoma o genotipo y sobre los diferentes individuos de una población inicial, actúa un operador de cruce generando así un conjunto de soluciones hijo. En cada iteración, cada individuo o solución es evaluado mediante una función y tendrán mayor probabilidad de ser elegidos los individuos que presenten un mayor valor de la función. Si ese valor supone una mejora éste reemplazará a las peores soluciones de la iteración actual. Por último se incorpora un operador de mutación para poder explorar nuevas zonas del espacio de soluciones.

Una introducción sobre los GA se puede encontrar en Mitchell (1998). Más recientemente, en el trabajo de los autores Reeves (2003) y Affenzeller et al. (2009) se describen los algoritmos genéticos con gran profundidad.

1.6.2.7.- Algoritmos meméticos [MA – Memetic Algorithms]

Los orígenes de los algoritmos meméticos se remontan a finales de los años ochenta, presentes en el trabajo de Moscato (1989). La idea básica que sustenta a los MA consiste en combinar conceptos y estrategias de diferentes metaheurísticas para intentar aunar las ventajas de las mismas.

Los MA son un conjunto de metaheurísticos caracterizados por fusionar ideas y principios derivados de los algoritmos evolutivos y la búsqueda por entornos. El adjetivo ‘memético’ deriva del inglés *meme* para designar al equivalente del gen en el contexto de la evolución cultural, aunque ello no implica que se tengan que enmarcar junto a ese tipo de estrategias.

Fundamentalmente se basan en mejorar individualmente las soluciones junto con procesos de cooperación y competiciones de tipo poblacional. Los algoritmos meméticos se asocian a una estrategia de búsqueda en la que una población de agentes optimizadores compiten y cooperan sinérgicamente. Los agentes emplean el conocimiento que sobre el problema tienen para así poderlo resolver, teoría descrita en el estudio Culberson (1998).

A diferencia de otras técnicas de optimización, y como se indica en Cotta (2007), *los algoritmos meméticos fueron concebidos explícitamente como un paradigma ecléctico y pragmático, abierto a la integración de otras técnicas, metaheurísticas o no.*

En Cotta y Alba (2003) y en Cotta (2007) se indica la relevancia del rasgo ecléctico presente en estos algoritmos, siendo esta propiedad lo que permite al método crear las sinergias necesarias entre las diferentes técnicas metaheurísticas. Algunos de los últimos avances que se han dado en los MA se encuentran en Hart et al. (2004), en el que destacan como una herramienta exitosa y con gran potencial en el campo de la optimización.

1.6.3.- Otros

En este apartado se hace una revisión breve de otros metaheurísticos que también han resultado exitosos en la resolución de problemas de optimización combinatoria. A continuación se enumeran y describen brevemente, indicando los trabajos pioneros y destacados examinados para cada metodología:

- **Concentración heurística:** Consta de dos fases: en la primera, se generan múltiples soluciones y en la segunda, con los elementos de las mejores soluciones, se construye el llamado conjunto de concentración. Se resuelve el problema original, pero está restringido a los elementos del conjunto de concentración. Destaca el trabajo de Rosing (1999) y Pacheco y Delgado (2000).
- **Búsqueda reactiva:** Propone la integración de procedimientos de autoajuste de parámetro en algoritmos basados en búsqueda por entornos. Destaca el trabajo de Battiti (1996).

- **Búsqueda local guiada:** Implica un proceso iterativo de búsqueda local que se basa en el uso de memoria. Al culminar cada iteración, modifica la función objetivo y penaliza ciertos elementos que aparecen en el último óptimo local. De esta manera estimula la diversificación de la búsqueda y evita incurrir de nuevo en dichos óptimos locales. Destaca el trabajo de Voudouris y Tsang (1995) y Kilby (1997).
- **Búsqueda por ruido:** Comienza definiendo una vecindad y parte de una solución inicial. Genera nuevas soluciones mediante la introducción de algún tipo de perturbación de datos (ruido). Toma como solución actual la mejor de entre las nuevas soluciones y repite el ciclo. Al ir aumentando el número de iteraciones va disminuyendo la perturbación hasta alcanzar el valor de cero. Destaca el trabajo de Charon y Hudry (1993).
- **Búsquedas multiarranque:** Alternan una fase de generación de soluciones iniciales, con una fase de mejora de las mismas. Tanto en la primera como en la segunda fase, los procedimientos empleados pueden ser simples o complejos para obtener soluciones de calidad. El algoritmo genera una nueva solución inicial cuando se llega a un óptimo local en la fase de mejora de la solución anterior. La repetición de los procesos generar solución inicial y búsqueda local es el primer método multiarranque descrito en la literatura. Destacan los trabajos pioneros de Los y Lardinois (1982) y Boender y Rinnooy (1983). Y más recientemente sobresalen los trabajos de Glover (2000), Martí (2003) y Martí y Moreno (2003).
- **Búsqueda del núcleo:** El método se basa en la identificación de un conjunto restringido de valores prometedores (*kernel*) y en la resolución del problema de programación lineal entera mixta en dicho conjunto. Para identificar el núcleo inicial se relaja de forma continua el problema resuelto en el conjunto completo de valores disponibles y, entonces, se resuelve una secuencia de problemas enteros para identificar otros valores a insertar en el núcleo. La búsqueda del núcleo tiene la ventaja de ser una técnica fácilmente aplicable a una gran variedad de problemas combinatorios. Destacan los trabajos de Angelelli et al. (2010a, 2010b, 2012) y Guastaroba y Speranza (2012a, 2012b, 2014).
- **Algoritmos bionómicos:** Consisten en unas estrategias de optimización global evolutiva de búsqueda probabilística, afín a los algoritmos genéticos y a la búsqueda dispersa, que va actualizando la población de soluciones en cada iteración, pero difiere en la forma de búsqueda del espacio de soluciones. Primero se genera una población inicial de soluciones. A continuación, se mejora cada solución de la población inicial estimando el vector que mide la ‘no factibilidad’ de la solución. Si es ‘no factible’ se aplica un algoritmo de optimización local buscando la factibilidad. Si la solución es ‘factible’ se busca mejorar el valor de la misma con otro algoritmo de optimización local que mantenga la factibilidad. Destaca el trabajo de Christofides (1994) y han sido aplicados con éxito a problemas logísticos y financieros.
- **Evolución diferencial:** Expone la representación de cada variable de decisión en el cromosoma como un número real (codificación real) y genera de forma aleatoria, como otros algoritmos evolutivos, la población inicial. Durante el proceso de selección, elige tres padres (uno de ellos se considera el principal) que generan un hijo. Este hijo se crea sumando al principal la diferencia de los otros dos padres. Destaca el trabajo de Storn y Price (1997) y han sido aplicados en problemas de optimización sobre dominios continuos.

- **Algoritmos culturales:** Aprovechan la teoría socio-arqueológica que trata la evolución cultural como un proceso de herencia en dos niveles: el nivel micro-evolutivo, que consiste en el material genético heredado, de los padres a sus descendientes, y el nivel macro-evolutivo, que es el conocimiento adquirido por los individuos a través de las generaciones, y que una vez codificado y recopilado, sirve para guiar el comportamiento de los individuos de una población. El objetivo es incrementar las tasas de aprendizaje o convergencia, y de esta manera, que el sistema responda mejor a un gran número de problemas. Destacan los trabajos de Reynolds (1994) y Landa (2002).
- **Optimización extrema:** Presenta la evolución progresiva de la naturaleza de forma que las especies mejor adaptadas al medio son las que se reproducen más fácilmente (Darwin, 1859), y elimina componentes extremos indeseables que son los que conducen a la extinción. El proceso actúa sobre una única solución y elimina sucesivamente elementos no deseados en las soluciones intermedias. Se diferencia de los algoritmos genéticos en que éstos actúan sobre el conjunto de soluciones y no sobre una única. Destaca el trabajo de Boettcher y Percus (2000).
- **Sistema inmune artificial:** Emplea como punto de partida el sistema inmune apoyándose en la característica de ser un sistema de aprendizaje distribuido. El funcionamiento del sistema inmune se activa cuando detecta la existencia de un antígeno, molécula que se encuentra en la superficie de los microorganismos patógenos que invaden al organismo. El mecanismo de reconocimiento y respuesta del sistema inmune para intentar eliminar los patógenos es la analogía que intenta replicar esta metodología. Destaca el trabajo inicial de Nunes de Castro y Timmis (2002) y la aplicación a problemas multiobjetivo presentada en Coello y Cruz-Cortés (2002).
- **Entropía cruzada:** Aplica un proceso iterativo en el que cada iteración se compone de dos pasos. El primero es la generación de un conjunto aleatorio de datos según un mecanismo determinado. Y el segundo es la actualización de los parámetros de generación aleatoria basada en datos, para poder generar un mejor conjunto aleatorio de datos. En origen surge de la estimación de probabilidades de eventos raros en redes estocásticas complejas presente en el trabajo de Rubinstein (1997). Su aplicabilidad en problemas de optimización combinatoria se analiza en Rubinstein (2001) y González et al. (2006).
- **SimHeuristics:** Combinan metaheurísticas con técnicas de simulación para abordar problemas de optimización en escenarios con incertidumbre. Permiten resolver problemas en los que la función objetivo incluye variables aleatorias y aquellos en los que el modelo matemático contiene restricciones probabilísticas. La simulación se emplea con dos finalidades; para estimar el valor de una solución generada por la metaheurística en un escenario estocástico y para guiar el proceso de búsqueda de la propia metaheurística. La información obtenida de la simulación permite realizar un análisis del factor riesgo/fiabilidad de las mejores soluciones estocásticas encontradas. Destacan los trabajos recientes de Juan et al. (2011), Juan et al. (2013), Juan et al. (2015) y Grasas et al. (2016).

2.- LA PROGRAMACIÓN MULTIOBJETIVO CON METAHEURÍSTICOS

2.- LA PROGRAMACIÓN MULTI OBJETIVO CON METAHEURÍSTICOS

2.1.- INTRODUCCIÓN

La mayoría de los problemas de optimización que se plantean en la vida real son de naturaleza multiobjetivo, ya que intentan encontrar una solución que optimice dos o más criterios. Esto induce la necesidad de minimizar/maximizar varias funciones simultáneamente, que están muchas veces en conflicto entre sí. En los problemas multiobjetivo o Multiobjective Optimization Problems (MOPs) no es muy probable contar con una solución que optimice todos los criterios a la vez, y se debe analizar qué soluciones son las más adecuadas para el caso concreto que se esté estudiando.

Los problemas multiobjetivo comenzaron a resolverse como problemas monoobjetivo por la falta de soluciones metodológicas adaptadas. Las técnicas utilizadas para resolver MOPs difieren bastante de las mono-objetivo ya que no se restringen normalmente a encontrar una solución única, sino que aspiran a encontrar un conjunto de soluciones de compromiso entre los múltiples objetivos contrapuestos. Esto se debe en parte a que no suele existir la solución que optimiza a la vez todos los objetivos.

Al enfrentarse a un problema multiobjetivo se diferencian dos etapas claras: por un lado, la optimización de las distintas funciones objetivo implicadas y, por otro, la fase de toma de decisiones sobre qué solución de compromiso es la más apropiada, enunciado por Cohon y Marks (1975). Es en este último documento donde se clasifican las técnicas para resolver MOPs en función del orden en el que se presentan las dos etapas, siendo las técnicas:

- A priori: si la toma de decisiones se produce antes de iniciar la búsqueda de soluciones.
- A posteriori: si la búsqueda se realiza antes de la fase de toma de decisiones.
- Progresivas: cuando la búsqueda de soluciones y la toma de decisiones se integran en el proceso de optimización.

Según el caso concreto del problema que se esté optimizando se aplicará un tipo de técnica u otra ya que cuentan con ventajas e inconvenientes que las convierten en más adecuadas en según qué escenarios. Es necesario remarcar que, tanto en el caso de las técnicas de resolución ‘a priori’ como en las ‘progresivas’ el espacio de búsqueda se puede ver limitado por la influencia del experto decisor cuando otorgue más relevancia a uno de los objetivos, de tal forma que ello puede impedir la resolución óptima del problema. Las técnicas a posteriori están siendo muy aplicadas en el campo de las metaheurísticas ya que al contrario de lo recién expuesto; primero exploran intensamente el espacio de búsqueda, generando así tantas soluciones de compromiso como le es posible, y es después cuando interviene el experto decisor y selecciona entre las encontradas.

En general, se puede afirmar que todas estas técnicas requerirán encontrar un conjunto de puntos eficientes amplio y representativo como para poder localizar en él una opción que se ajuste a las preferencias del decisor.

Ejemplos de estas técnicas y sus características se pueden encontrar en los estudios de Coello et al. (2007) y Deb (2001).

2.2.- FORMULACIÓN DEL PROBLEMA MULTI OBJETIVO

El MOP se define como el problema de hallar un vector de variables de decisión que cumpla un conjunto de restricciones y que optimice un conjunto de funciones objetivo. Estas funciones están habitualmente en conflicto entre sí y, por ello y según Osyczka (1995), el término “optimizar” se refiere a buscar una solución que comprenda valores aceptables para todas las funciones objetivo.

La formulación matemática de un MOP considera la existencia de varias funciones objetivo y de un conjunto de soluciones. Este conjunto de soluciones se encuentra mediante la utilización de la Teoría de Optimalidad de Pareto desarrollado por Vilfredo Pareto en 1896, Pareto (1896). El MPO es formulado en la literatura de la siguiente manera:

Encontrar un vector $\vec{x}^* = [x_1^*, x_2^*, \dots, x_n^*]$ que satisfaga las m restricciones de desigualdad $g_i(\vec{x}) \geq 0, i = 1, \dots, m$, las p restricciones de igualdad $h_i(\vec{x}) = 0, i = 1, \dots, p$, y que minimice la función vector $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]^T$, donde $\vec{x} = [x_1, x_2, \dots, x_n]^T$ es el vector de decisión de las variables. Donde f_i corresponde a cada uno de los criterios u objetivos. El conjunto de todos los valores que satisfacen las restricciones define la región de soluciones factibles Ω y cualquier punto en $\vec{x} \in \Omega$ es una solución factible.

Dado el conflicto existente entre los diferentes criterios u objetivos que componen el vector, la optimización va a consistir en encontrar buenos compromisos entre dichas funciones objetivo y para ello es necesario introducir un nuevo concepto. Dicho concepto es la *Optimalidad de Pareto*, Pareto (1896), que considera que el óptimo ha de ser aquel para el cual no se pueda mejorar un objetivo sin empeorar los demás. La *optimalidad de Pareto* se formula a continuación:

Un punto $\vec{x}^* \in \Omega$ es un óptimo de Pareto si $\forall \vec{x} \in \Omega, I = \{1, 2, \dots, k\}$, o bien:

$$\forall_{i \in I} (f_i(\vec{x}) = f_i(\vec{x}^*)) \text{ o bien hay al menos un } i \in I \mid f_i(\vec{x}) > f_i(\vec{x}^*)$$

Esta definición expone que \vec{x}^* es un óptimo de Pareto si no existe ningún vector factible \vec{x} que mejore algún criterio sin causar simultáneamente un empeoramiento en al menos otro criterio, para el caso concreto de la minimización. La expresión de la *dominancia de Pareto* es:

Un vector $\vec{u} = (u_1, \dots, u_k)$ se dice que domina a otro $\vec{v} = (v_1, \dots, v_k)$ y se representa como $\vec{u} < \vec{v}$ si y sólo si \vec{u} es parcialmente menor que \vec{v} , es decir, $\forall i \in \{1, \dots, k\}, u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i$

El proceso de resolver un MOP pasará por encontrar el conjunto de soluciones que dominan a cualquier otra solución del espacio de soluciones, lo que significa que son las mejores para el problema y, por tanto, conforman su solución óptima. Dado $\vec{f}(\vec{x})$, el *conjunto óptimo de Pareto* se define como:

$$\mathcal{P}^* = \{\vec{x} \in \Omega \mid \nexists \vec{x}^t \in \Omega: \vec{f}(\vec{x}^t) \leq \vec{f}(\vec{x})\}$$

El *frente de Pareto* se formula según la siguiente expresión:

Para un MOP dado $\vec{f}(\vec{x})$ y su conjunto óptimo de Pareto \mathcal{P}^* , el *frente de Pareto* se define como

$$\mathcal{PF}^* = \{\vec{f}(\vec{x}), \vec{x} \in \mathcal{P}^*\}$$

Dicho de otra manera, el frente de Pareto se compone por los valores en el espacio de objetivos del conjunto óptimo de Pareto. Analíticamente no es fácil expresar la línea o superficie que contiene dichos puntos o valores, siendo en la mayor parte de los casos imposible.

2.3.- FINALIDAD DE LA PROGRAMACIÓN MULTIOBJETIVO

La Programación Multiobjetivo nació para la resolución de los problemas de optimización multiobjetivo. Los algoritmos de optimización empleados para su resolución, por lo expuesto en la sección inmediatamente anterior, tendrán por objeto hallar el frente de Pareto del PMO o, análogamente, el conjunto óptimo de Pareto del problema.

Una vez se obtienen múltiples soluciones Pareto-óptimas se debe escoger una solución sobre otra. Dicho proceso se complica al carecer de información adicional sobre el problema puesto que todas las soluciones obtenidas son igualmente importantes.

Es necesario fijar como finalidad de cualquier MOP la búsqueda de un número de soluciones no dominadas que pretendan los dos propósitos enumerados a continuación:

- El conjunto de soluciones deberá estar lo más cercano posible al frente de Pareto óptimo. Esto es la convergencia hacia la solución óptima presente en todo problema de optimización, independientemente de tener un objetivo o ser multiobjetivo.
- El conjunto de soluciones deberá ser lo más uniformemente diverso que sea viable. Esta segunda meta es específica de la programación multiobjetivo y busca que, conjuntamente con la convergencia hacia el frente óptimo, las soluciones estén repartidas uniformemente a lo largo de todo el frente.

La obtención de un conjunto diverso de soluciones que converja hacia el frente óptimo es la única forma de asegurar que se ha realizado una buena exploración del espacio de búsqueda y que se cuenta con un buen conjunto de soluciones de compromiso entre los diferentes objetivos facilitando la posterior toma de decisiones por parte del experto decisor (para las técnicas de resolución a posteriori de los MOP recomendadas en el campo de los metaheurísticos).

2.4.- METAHEURÍSTICOS EN PROBLEMAS MULTIOBJETIVO

El gran desarrollo de la Programación Multiobjetivo ha permitido que sus técnicas se empleen en la resolución de casos reales en los que nos encontraremos con las siguientes dificultades:

- Complejidad por el tamaño de un modelo real: tanto por el número de variables, como por las restricciones.
- Complicación por la naturaleza de las funciones de un problema real: suelen ser no lineales y no diferenciables.
- Inconveniente por la tipología de las variables de un caso real: pueden ser combinaciones de enteras y binarias simultáneamente.
- Y la multiplicidad de objetivos de los MOP: teniendo que encontrar un conjunto de soluciones y no un único punto.

Por todo lo anterior, a pesar de que la aplicación del enfoque multiobjetivo sea teóricamente correcta, puede que su tratamiento computacional no sea algorítmicamente válido por no contar con herramientas adaptadas que resuelvan el problema. Se justifica así la aplicación de metodologías metaheurísticas combinadas con las técnicas de programación multiobjetivo para resolver exitosamente una amplia tipología de problemas reales.

En los trabajos de Steuer (1986) y de Caballero et al. (2007) se encuentra tanto una revisión de las aplicaciones de los MOP como los diferentes enfoques estudiados a la hora de abordar este tipo de

problemas mediante algoritmos metaheurísticos. El efecto que el desarrollo de las técnicas metaheurísticas ha supuesto para el avance en la resolución de MOP se recoge en Jones et al. (2002).

A partir de las revisiones realizadas en la literatura existente y debido a la gran cantidad de metaheurísticas que se han aplicado con éxito en la resolución de problemas multiobjetivo se enumeran en esta sección las más destacadas clasificadas en sus dos grandes grupos: los basados en trayectoria y los basados en población. Finalmente se presentan métodos híbridos que están siendo aplicadas con gran éxito.

Aunque esta materia está permanentemente en desarrollo, ya en el trabajo de Jones et al. (2002), se estimó que, aproximadamente, el 70% de las metaheurísticas adaptadas en Programación Multiobjetivo se basan en algoritmos evolutivos (basados en población), un 24% en recocido simulado y un 6% en búsqueda tabú (estos dos últimos basados en trayectoria).

2.4.1.- Algoritmos aplicados basados en trayectoria

Como ya se ha explicado anteriormente estos métodos tienen como rasgo principal el uso de una operación básica denominada movimiento. El movimiento modifica atributos o características de una solución, creando un conjunto de posibles soluciones alternativas denominadas vecindario, entre las que se elegirá la que mejore a las demás, según unas reglas definidas previamente. El elemento seleccionado pasará a la siguiente iteración.

A continuación se describen aquellas metaheurísticas basadas en trayectoria o búsqueda por entornos que se han aplicado con éxito a la resolución de problemas multiobjetivo.

2.4.1.1.- Recocido Simulado

Técnica presentada en el campo de la optimización combinatoria por Kirkpatrick et al. (1983). En el capítulo previo ya se ha descrito que este metaheurístico es conocido en inglés como *Simulated Annealing* y en la literatura se ha demostrado que es una herramienta bastante eficaz para resolver una amplia gama de problemas de optimización combinatoria.

El recocido simulado (SA) se empleó por primera vez en el campo de la programación multiobjetivo en el trabajo de Serafini (1992), donde la principal idea para la adaptación de este método para problemas con criterios múltiples es la utilización de un criterio de aceptación de soluciones de peor calidad basado en la relación de dominancia entre dos soluciones dadas.

Otros estudios revisados plantean la utilización de funciones agregativas basadas en pesos y la variación de forma adecuada de dichos pesos, este es el caso de Teghem et al. (2000), y Ulungu et al. (1998). La resolución de programación multiobjetivo aplicada a distribución y planificación de redes se encuentra en Ponce y Matos (1999). Otros trabajos relevantes se pueden localizar en Erhgott y Gandibleux (2000) y en Jones et al. (2002).

Las aplicaciones del recocido simulado a problemas multiobjetivo se han mostrado en general muy eficientes y robustas, lo cual ha hecho que esta metaheurística sea muy popular para la resolución de problemas reales.

2.4.1.2.- Búsqueda Tabú

Tal y como se ha recogido en el capítulo anterior, la búsqueda tabú se presenta por primera vez en el trabajo de Glover (1986). Se trata del metaheurístico más importante dentro de lo que se conoce como

Programación mediante Memoria Adaptativa (AMP), que se caracteriza por ser un método de búsqueda por entornos en los cuales se utiliza información acerca de los movimientos realizados con anterioridad.

A partir de la búsqueda tabú se han desarrollado diferentes técnicas para su aplicación en problemas multiobjetivo, las cuales se basan en algún tipo de agregación de los diferentes criterios que forman parte del problema para convertirlos en problemas mono-objetivo que se resolverán empleando una búsqueda tabú.

En el trabajo presentado por Hertz et al. (1994) se soluciona un conjunto de funciones mono-objetivo resultado de considerar cada vez una de las funciones objetivo junto a una función de penalización. Otro ejemplo puede verse en el caso de en Dahl et al. (1995), donde se compone una familia de vectores de pesos y para cada uno de ellos se resuelve una función mono-objetivo que ha sido el resultado de incluir de forma ponderada cada uno de los criterios. En Gandibleux et al. (1997) se plantea el uso de una función escalada de logro, basada en las funciones objetivo del problema, que sirve de guía para la búsqueda tabú.

Es destacable el método denominado MOAMP (*MultiObjective Adaptive Memory Procedure*), que fue descrito en Caballero et al. (2003). Este procedimiento trata de adaptar una búsqueda tabú a un conjunto eficiente de un problema multiobjetivo. En el trabajo de Caballero et al. se dice “...es un hecho conocido que los puntos eficientes de un problema multiobjetivo, se encuentran ‘conectados’ entre sí...”, esto implica que:

- para problemas que cumplan unas mínimas condiciones de continuidad, si las variables son continuas, cualquiera de los puntos eficientes están conectados por una curva dentro del conjunto eficiente y,
- en el caso de variables enteras, cualquier punto está ‘suficientemente cerca’ de otro punto eficiente.

El MOAMP trata de aprovechar precisamente este principio de proximalidad, tratando de generar, por medio de una búsqueda tabú, un conjunto eficiente inicial de puntos y, a través un proceso de intensificación, una aproximación al resto de puntos eficientes del problema. Más ejemplos de autores que han publicado sobre esta temática son: Hansen (1997), Ben Abdelaziz et al. (1999), Gandibleux y Freville (2000), Alves y Clímaco (2000), Ehr Gott et al. (2004) y Gómez et al. (2015) entre otros.

2.4.1.3.- GRASP

En el capítulo anterior ya se ha presentado el procedimiento de búsqueda ávido, aleatorizado y adaptativo o Greedy Randomized Adaptive Search Procedure (GRASP) que se dio a conocer a finales de los ochenta en el trabajo de Feo y Resende (1989). Es un método iterativo en el que cada iteración aporta una solución al problema sobre el que se está trabajando y la mejor solución queda guardada como resultado final.

GRASP está clasificado como un metaheurístico constructivo de fácil implementación. En el campo de los problemas multiobjetivo se encuentran citas ya en los estudios de Gandibleux et al. (1998), así como en trabajos más recientes como el de Higgins et al. (2008).

2.4.2.- Algoritmos aplicados basados en población

La mayor parte de las aplicaciones de algoritmos basados en población al campo de la resolución de problemas multiobjetivo se concentran alrededor de los algoritmos genéticos. Inicialmente, recogido en Goldberg (1989), estas aplicaciones se fundamentan en el uso de funciones agregativas de criterios

mediante un escalamiento de los mismos. Es más tarde, con Wilson y Macleod (1993), donde se comienza a plantear un enfoque de programación mediante metas ponderadas.

El método VEGA (*Vector Evaluated Genetic Algorithms*), Schaffer (1985), se programó con el propósito de evitar las desventajas propias que la agregación de criterios producía. Este método está basado en *rankings* y se diferencia de un algoritmo genético habitual en la forma de llevar a cabo la selección para la fase de la reproducción. En cada iteración, la población se agrupa en sub-poblaciones (tantas como criterios) atendiendo en cada una de ellas al valor de una de las funciones objetivos. Dentro de cada sub-población se realiza la selección atendiendo al valor de ese criterio y, en la siguiente fase, las sub-poblaciones se mezclan de nuevo para aplicar los operadores de cruce y mutación, que dará lugar a la siguiente generación. Una de las debilidades de VEGA es que presenta problemas por no ser capaz de retener buenas soluciones eficientes (soluciones de compromiso entre criterios). En Caballero et al. (2007) se indica esta incapacidad de incorporar una solución que contenga la dominancia de Pareto.

Otro algoritmo aplicado que, estando basado en un ordenamiento lexicográfico, ordena las funciones objetivo según la importancia de cada objetivo, lo que se plantea por Fourman (1985). El algoritmo, antes de iniciar el proceso de búsqueda de soluciones, los ordena empezando por el más importante y procediendo según el orden de importancia asignado a cada uno de los objetivos. Este método es conveniente únicamente cuando la importancia de cada objetivo es conocida y ésta puede ser establecida, lo cual no suele ser común.

2.4.2.1.- Basados en el orden de Pareto

Dentro de los algoritmos aplicados basados en población surgió un subconjunto con un enfoque basado en el orden de Pareto, cuya potencialidad y ventajas se estudian en Hilliard et al. (1989). Según este enfoque, el valor de cada individuo no depende del valor de sus criterios, sino de su dominancia dentro del conjunto total de la población. Se fundamentan en la idea de encontrar individuos que no estén dominados por ningún otro, para poder extraerlos del conjunto y volver a repetir el proceso con el resto de individuos. El proceso acabará por incluir a todos los individuos dentro de un ranking con una posición determinada y así se podrá realizar la selección por ranking.

Enmarcados en la primera generación de los algoritmos basados en el principio de Optimalidad descrito por Pareto, se resumen a continuación los principales:

NSGA (Non-Dominated Sorting Genetic Algorithm): se basa en la clasificación previa de los puntos en función de su ‘no dominación’, antes de realizar la selección de los individuos. A todos los puntos no dominados de la población, se les asigna un mismo valor de aptitud, que es proporcional al tamaño de la población. Una vez eliminados estos, se repite el proceso hasta la total clasificación de los puntos. Gracias a ello, por un lado, aquellos puntos con mayor clasificación tienen mayor probabilidad de reproducirse en la siguiente generación y, además, la existencia de conjuntos de puntos (*niching*) que comparten el mismo valor de la función de aptitud garantiza la diversidad del conjunto de puntos eficientes. Fue presentado en el trabajo de Srinivas y Deb (1994).

NPGA (Niche Pareto Genetic Algorithm): adopta un esquema de selección por torneo, también basado en la dominancia de Pareto. Toma dos individuos de forma aleatoria, del conjunto de la población, que son comparados con un subconjunto de la misma. Si uno de ellos es dominado y el otro no, el no dominado gana; si ambos son dominados, pierden los dos, y si ninguno es dominado, entonces ganan los dos. El resultado del torneo se decide a través de

fitness sharing o método de proporción (intenta formar subconjuntos de elementos vecinos en la población, llamados ‘nichos’, y así reducir la aptitud de los individuos por la presencia de otros muy parecidos). Fue propuesto en la publicación de Horn y Nafpliotis (1993).

MOGA (*Multiobjective Genetic Algorithm*): se da un ranking a cada individuo según el número de cromosomas en la población actual por los cuales es dominado. Si se considera un individuo x_i en la iteración t que es dominado por $p_i^{(t)}$ individuos de la generación actual, el ranking de x_i viene dado por $rank(x_i, t) = 1 + p_i^{(t)}$. De esta forma, los individuos no dominados tiene un ranking igual a 1, mientras que los dominados son penalizados según el número de individuos en sus correspondientes regiones.

A medida que estos algoritmos se fueron desarrollando apareció una segunda generación en la que se introduce el concepto de *elitismo*. Dicho concepto, dentro del contexto de la optimización multiobjetivo, se refiere al uso de una población externa que almacena los individuos no dominados encontrados hasta esta generación y que participa en el proceso de selección.

En el trabajo de Caballero et al. (2007) se hace la siguiente enumeración de los algoritmos basados en el orden de Pareto de segunda generación más representativos:

SPEA (*Strength Pareto Evolutionary Algorithm*): trabaja con dos poblaciones, incorporando el concepto de elitismo mediante el almacenamiento de las soluciones no dominadas en una población externa, la cual participa del proceso de selección. El ranking de selección se realiza por un procedimiento basado en la asignación de un valor de fuerza (*strength*) a todos los elementos de la población externa. Este método introduce la formación de nichos a partir del concepto de dominancia Pareto, llamado *niching por strength*. A todo lo anterior se le agrega un procedimiento de clustering para reducir el número de soluciones la población externa para preservar la diversidad. Fue presentado en el trabajo de Zitle y Thiele (1999).

SPEA 2: presenta tres diferencias respecto al anterior: en primer lugar presenta una estrategia de asignación de aptitud de ‘grano fino’ (para cada individuo se tiene en cuenta el número de individuos que domina y el número de individuos por el que es dominado); en segundo lugar, utiliza una técnica para la estimación de densidad de su vecindario, que guía la búsqueda más eficientemente y, en tercer lugar, incorpora un método para truncar el archivo, garantizando el mantenimiento de soluciones extremas. Fue publicado en Zitle et al. (2001).

PAES (*Pareto Archived Evolution Strategy*): incorpora una estrategia del tipo (1+1), un único padre genera un único hijo y compiten entre ellos, junto con una búsqueda local y el empleo de un archivo histórico (concepto de elitismo) que almacena soluciones no dominadas encontradas a lo largo de la ejecución del algoritmo. Fue propuesto por Knowles y Corne (2000).

NSGA-II: aumenta su eficiencia al incorporar el concepto de elitismo y un operador de comparación (*crowding*) en función de la proximidad de soluciones alrededor de cada uno de los puntos de la población. Su mecanismo de selección consiste en la selección de los mejores padres con los mejores hijos. Fue desarrollado en el trabajo de Deb et al. (2000).

Controlled Elitist NSGA-II: difiere del NSGA-II en la forma de seleccionar al número de individuos pertenecientes al mejor frente actual de no-dominados, que en este algoritmo aplicado se realiza de manera adaptativa.

NPGA2: incorpora el ranking de Pareto y es considerado como una versión revisada del algoritmo NPGA porque mantiene la selección por torneo. No hace uso de memoria externa y aplica un mecanismo de elitismo similar al presente en el NSGA-II. Fue introducido por Erickson et al. (2001).

2.4.3.- Otras tendencias aplicadas

Existen varios nuevos algoritmos que, combinando los dos enfoques anteriormente descritos, de búsqueda por trayectoria y de búsqueda por población, con otros metaheurísticos que han resultado eficientes en problemas mono-objetivo, ya están dando buenos resultados en MOP. Brevemente y entre otros, cabe destacar:

Búsqueda Dispersa: destacan los trabajos de Molina et al. (2005) y Beausoleil (2006) aplicados a la resolución de problemas de optimización multiobjetivo no lineales. En publicaciones más recientes, como Pérez et al. (2007), se introducen variaciones al reemplazar la búsqueda local aplicada en la mejora de las soluciones, por un mecanismo de mejora guiado por direcciones proporcionadas por los vectores gradiente de las funciones objetivo.

Evolución Diferencial: sobresale el PDE (*Pareto Differential Evolution*) en el trabajo de Coello y Lamont (2005). Este método emplea una única población, en fase de la reproducción toma sólo soluciones no dominadas e incorpora una métrica de distancia para favorecer la diversidad. También resaltar PDEA (*Pareto Differential Evolution Approach*) en Madavan (2002), que combina la DE con distintos elementos de la NSGA-II; VEDE (*Vector Evaluated Differential Evolution*) en Parsopoulos et al. (2004), que se inspira en VEGA manejando múltiples poblaciones en paralelo y DEMORS (*Differential Evolution MultiObjective Rough Sets*) recogido en la publicación de Hernández-Díaz et al. (2006), donde se combina la evolución diferencial con *Rough Sets*, herramienta de la Inteligencia Artificial.

3.- LA BÚSQUEDA TABÚ

3.- LA BÚSQUEDA TABÚ

3.1.- INTRODUCCIÓN

La búsqueda tabú surgió a finales de la década de los años 70 por el trabajo de Glover (1977), aunque fue más tarde cuando se dio a conocer formalmente con ese nombre, Tabu Search – TS, en Glover (1989).

Una descripción de los orígenes de esta metodología se encuentra en la publicación de Glover y Laguna (1997), en la que se dice: *“La Búsqueda Tabú tiene sus orígenes en procedimientos combinatorios aplicados a problemas de cubrimiento no lineales, en los finales de los años 70, y aplicada subsecuentemente a una diversa colección de problemas que van desde secuenciación y balance de canales de computación hasta análisis de clústeres y planteamiento de espacio”*.

Etimológicamente la palabra tabú procede de la Polinesia, recogido en la Real Academia Española (2014), donde es usado por los aborígenes de la isla Tonga para referirse a cosas que no pueden ser tocadas o ingeridas por ser sagradas, otra acepción la define como *“Condición de las personas, instituciones y cosas a las que no es lícito censurar o mencionar”*. En el trabajo de Glover se hace referencia al término *tabú* como *“...un tipo de inhibición a algo debido a connotaciones culturales o históricas y que puede ser superada en ciertas condiciones...”*, esta acepción es la que está más cerca de la esencia del método.

Originalmente, la búsqueda tabú se desarrolló como una búsqueda local enfocada a resolver problemas de programación entera encontrando buenas soluciones. El éxito de su aplicación radica en sus características principales, que se citan brevemente a continuación y serán desarrolladas en secciones posteriores:

- actualiza la lista de mejores soluciones en cada proceso iterativo
- implementa un proceso de prioridades que aspiran a mejores posiciones al cabo de un tiempo dado
- incorpora la lista tabú que funciona como un mecanismo de control aceptando o renegando la entrada de posibles candidatos vecinos a optar a ser solución.
- aplica técnicas de diversificación e intensificación

3.2.- DESCRIPCIÓN Y CARACTERÍSTICAS

Incluido en el estudio de Melián y Glover (2004) se define la búsqueda tabú como *“...un procedimiento metaheurístico cuya característica distintiva es el uso de memoria adaptativa y de estrategias especiales de resolución de problemas...”*.

La búsqueda tabú emplea conceptos de la Inteligencia Artificial permitiendo así resolver problemas combinatorios de gran complejidad. Su planteamiento general se basa en la idea de explotar diversas estrategias inteligentes aplicadas a la resolución de problemas, todas ellas basadas en procedimientos de aprendizaje.

La metaheurística de la TS es una búsqueda local guiada hacia el óptimo. La forma de conseguir dirigirse al óptimo, consiste en evitar que la búsqueda se estanque en mínimos locales a favor de encontrar el óptimo global. Aquí es donde entra en juego el concepto de memoria, extraído de la inteligencia artificial, necesario para conseguir que la búsqueda no se pare en el óptimo local. Es decir, el algoritmo dirige la búsqueda teniendo en cuenta los movimientos anteriores convirtiéndose en una búsqueda inteligente.

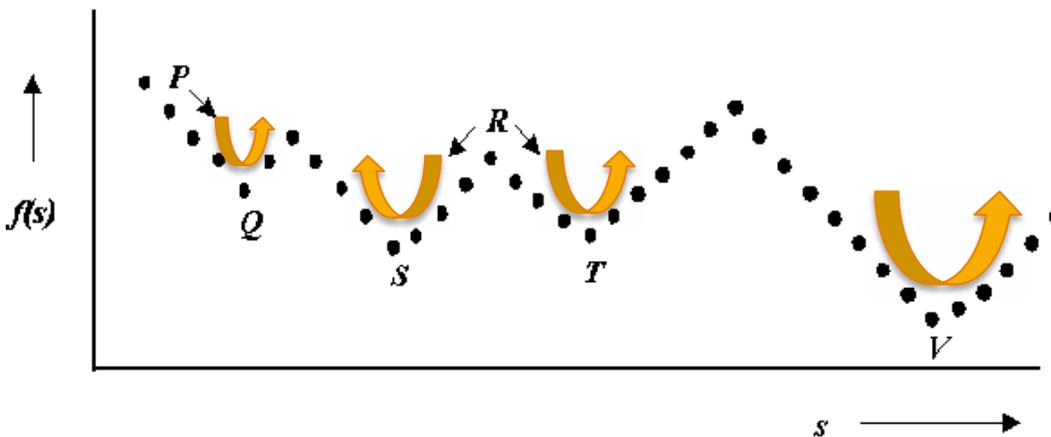
A diferencia de otros métodos, la búsqueda tabú elimina el carácter estocástico de un procedimiento, al incorporar estrategias de búsqueda inteligentes y sistemáticas, por lo que es considerada como una metodología esencialmente ‘determinista’.

La búsqueda tabú, como se ha enunciado anteriormente, se fundamenta en la técnica de búsqueda local cuyo algoritmo es el siguiente:

- I. Seleccionar una solución del conjunto de soluciones
- II. Seleccionar un vecino tal que mejore el valor inicial
 - II a. Si no existe uno mejor, es un óptimo local
 - II b. Si existe, guardar la nueva solución y volver al punto II
- III. Repetir hasta que se cumpla la condición final

La principal desventaja de las búsquedas locales es que el proceso pare en un óptimo local, circunstancia que puede ocurrir también en la búsqueda tabú aunque es menos probable por la implementación que propone sobre dicha heurística. El perfeccionamiento planteado tiene la ventaja de permitir movimientos que no mejoren la solución seleccionada para poder salir de mínimos locales y comenzar la exploración de otro conjunto de soluciones. Partiendo como base de la Figura - 2 , Dowsland (1993), en la que se mostraba la representación gráfica de una heurística de búsqueda local del mínimo, a continuación se representa en la Figura - 4, los movimientos que permite la búsqueda tabú para explorar y mejorar las soluciones encontradas, Casado (2005).

Figura - 4: Movimientos permitidos en búsqueda tabú de mínimos



Otro rasgo importante incluido en la TS es la memoria temporal que evita regresar a soluciones anteriormente visitadas mediante el concepto de lista tabú. De esta forma se evita ciclar durante un determinado número de iteraciones y además, tal y como indican Melián y Glover (2004), la lista tabú tiene el fin último de “...continuar estimulando el descubrimiento de soluciones de alta calidad”. Si bien, la condición tabú puede ser ignorada bajo unas circunstancias determinadas, lo que da lugar a la aparición de los llamados criterios de aspiración. Estos dos conceptos, de lista tabú y de criterios de aspiración, aporta gran flexibilidad al proceso de búsqueda.

Por todo lo anterior se puede decir que, la búsqueda tabú supera la búsqueda local utilizando una estrategia de modificación del entorno $N(s)$ a medida que el proceso de búsqueda avanza, reemplazándolo por un nuevo entorno $N^*(s) = N(s) - T$.

Dentro de la literatura, más en concreto en Casado (2005), se encuentra la representación en pseudocódigo de la estrategia de la búsqueda tabú que se muestra a continuación:

PROCEDIMIENTO BÁSICO DE UNA BÚSQUEDA TABÚ (problema de mínimos)

- Seleccionar una solución inicial $s_0, s^* = s_0$
- Inicializar $T = \emptyset$
- Repetir
 - Determinar $N^*(s_0) \subset N(s_0)$
 - Seleccionar $f(s') = \min \{f(s) / s \in N^*(s_0)\}$
 - Si $f(s') < f(s^*)$ entonces $s^* = s'$
 - Hacer $s_0 = s'$
 - Actualizar T

Hasta alcanzar una condición de parada.

donde $N(s)$ es el entorno; $N^*(s)$ es el entorno modificado y T es el conjunto de soluciones tabú.

Con el fin de poder evaluar la calidad de la solución encontrada, será necesario definir una función objetivo que dependa de diferentes parámetros a optimizar (coste, tiempo, distancia, etc.). Entonces, en cada iteración, una solución seleccionada dentro del entorno de la solución actual, es analizada y evaluada mediante dicha función objetivo y, si mejora la solución y no es tabú o, siendo tabú, cumple un criterio de aspiración, sustituirá a la solución inicial. Se deberá actualizar el entorno de las soluciones candidatas y repetir todo el ciclo anterior; y así continuamente hasta que se cumpla una condición de parada preestablecida..

Se puede compendiar, por lo anteriormente mencionado, que la búsqueda tabú se fundamenta en dos estrategias que se alternan secuencialmente: la intensificación y la diversificación; y que el uso que se hace de una u otra depende, principalmente, de la información almacenada en la memoria. Estas características esenciales de la TS, junto con otras tales como la lista tabú y los criterios de aspiración, van a ser expuestas en las siguientes secciones de este capítulo.

3.3.- LA LISTA TABÚ

Dentro de la programación la lista tabú es un inventario donde se registran las soluciones, o atributos de las mismas, que no deben ser elegidas.

Un ejemplo simple de generación de una lista tabú consiste en que cada vez que se ejecuta un movimiento, su inverso se introduce en una lista circular, de forma que los elementos de dicha lista están penalizados durante un cierto tiempo. El tiempo dependerá del tamaño de la lista. A mayor tamaño más tiempo permanecerán los movimientos restringidos en la lista ya que estos no se eliminan hasta que no se ha llenado el resto de posiciones libres.

Por definición, un movimiento que está en la lista tabú no será aceptado hasta que desaparezca de ella, incluso aunque se tratase de una mejor solución que la solución actual. Esto no sería eficiente y es por ello que en la búsqueda tabú se permiten excepciones en caso de mejorar la solución, pero con ciertas restricciones (concepto de criterios de aspiración que se tratan más adelante).

La lista tabú se forma a partir de los movimientos realizados recientemente. Esta circunstancia implica que la lista tabú se componga tanto de las soluciones visitadas recientemente como de los atributos o características presentes en dichas soluciones visitadas.

El tamaño de la lista tabú (*tabu size*) se suele definir a través del número de iteraciones que movimiento o atributo permanecen en la lista tabú. La lista también puede estar formada de elementos de un mismo tamaño y por tanto se clasificará por la longitud de la lista tabú, es decir, por el número de elementos en ella.

En el caso de una lista tabú variable puede suceder que un elemento que entró a la lista tabú antes que otro pueda salir tiempo o iteraciones después. Valga de ejemplo de lo anterior el siguiente caso; si un movimiento de la lista tabú tiene una espera asociada de 5 iteraciones y entra a la lista tabú en la iteración 15, no saldrá de ella hasta la iteración 20. Puede existir otro elemento con una espera de 2 iteraciones que entre justo después en la iteración 16 y entonces saldrá antes, en la iteración 18.

3.4.- LOS CRITERIOS DE ASPIRACIÓN

La prohibición de visitar soluciones que contengan atributos tabú no tiene por qué ser absoluta. Al igual que las costumbres sociales pueden cambiar con el tiempo, las soluciones tabú pueden dejar de serlo justificado en el hecho de la existencia de una memoria cambiante, se debe contemplar una forma de “olvido estratégico”, es decir que una solución o atributo pueda salir de la lista tabú antes de que se cumpla su plazo. Es por ello necesaria la introducción de criterios de aspiración que pueden determinar en qué casos ignorar una restricción tabú, eliminando así la clasificación de tabú asociada a un movimiento. Resultando que cualquier solución de más calidad que las encontradas previamente, logre ser analizada, aunque para ello se deba realizar un movimiento prohibido.

Los criterios de aspiración presentes en la búsqueda tabú son de dos clases según a qué tipo de elemento vayan asociadas: del movimiento o del atributo.

Las aspiraciones del movimiento, al cumplirse anulan la condición tabú del movimiento.

Las aspiraciones del atributo, al satisfacerse invalidan el status tabú del atributo. Su movimiento correspondiente, en este caso, puede o no cambiar su condición de tabú, dependiendo de si la restricción tabú puede activarse por más de un atributo.

En la siguiente enumeración se presentan brevemente los criterios de aspiración que se suelen incluir en las TS:

- Aspiración Standard: Si todos los movimientos posibles se encuentran clasificados como tabú, entonces se seleccionará el movimiento “menos tabú”, es decir, el movimiento que se encuentre menos penalizado. Sirva de ejemplo el siguiente caso, si un movimiento (mov_1) está penalizado en la lista tabú durante 5 iteraciones y el movimiento (mov_2) está penalizado durante sólo 3 iteraciones, se considera que (mov_2) es menos tabú que (mov_1).
- Aspiración por Objetivo: En funciones objetivo que se busca minimizar o maximizar un resultado, se permitirá aquel movimiento que aún estando en la lista tabú mejore la función objetivo.
- Aspiración por Dirección de Búsqueda: Un atributo de aspiración para una solución se cumple si la dirección en la solución proporciona una mejora del atributo y además el movimiento actual es un movimiento de mejora. Entonces dicha solución se considera como una candidata.

De manera resumida se concluye que la lista tabú influye sobre el vecindario. En cada iteración, para poder evolucionar hacia otras soluciones, se seleccionan éstas en un entorno del vecindario $N(s)$, pero asegurando que no estén en la lista Tabú, ($N(s) - \{Lista\ Tabú\}$), evaluando cada una de las soluciones en el entorno y quedándose con la mejor. Así se realizarán movimientos de baja influencia mientras existan probabilidades de mejora y en el momento en que se agote esta posibilidad, se modifican los criterios de aspiración para dar entrada a movimientos más influyentes. Generalmente, una vez que se ha dado un salto a otra región del entorno (diversificación), se deben eliminar las restricciones tabú establecidas previamente.

3.5.- LA INTENSIFICACIÓN Y LA DIVERSIFICACIÓN

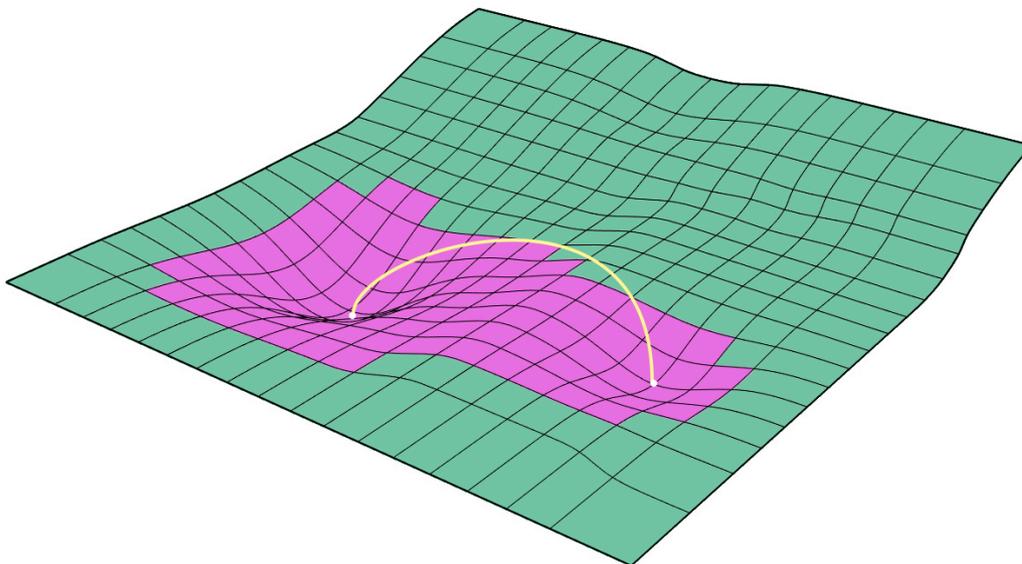
Son muchas las aplicaciones de búsqueda tabú que refuerzan el procedimiento básico con estrategias de intensificación y diversificación para así conseguir una búsqueda más eficiente y flexible del entorno.

3.5.1.- La intensificación

La idea de intensificación consiste en explorar más en porciones del espacio que aparentan ser mejores o más prometedoras mediante la modificación de las reglas de selección.

La Figura - 5 muestra una representación gráfica tridimensional de un proceso de intensificación. Cada cierto tiempo se puede realizar un proceso de intensificación re-iniciando la búsqueda a partir de la mejor solución actual.

Figura - 5: Intensificación en TS



Las funciones de memoria a corto plazo sirven para registrar y comparar atributos de las mejores soluciones obtenidas durante un período de búsqueda. Los atributos comunes pueden servir para guiar nuevas soluciones a espacios en donde existan tales atributos.

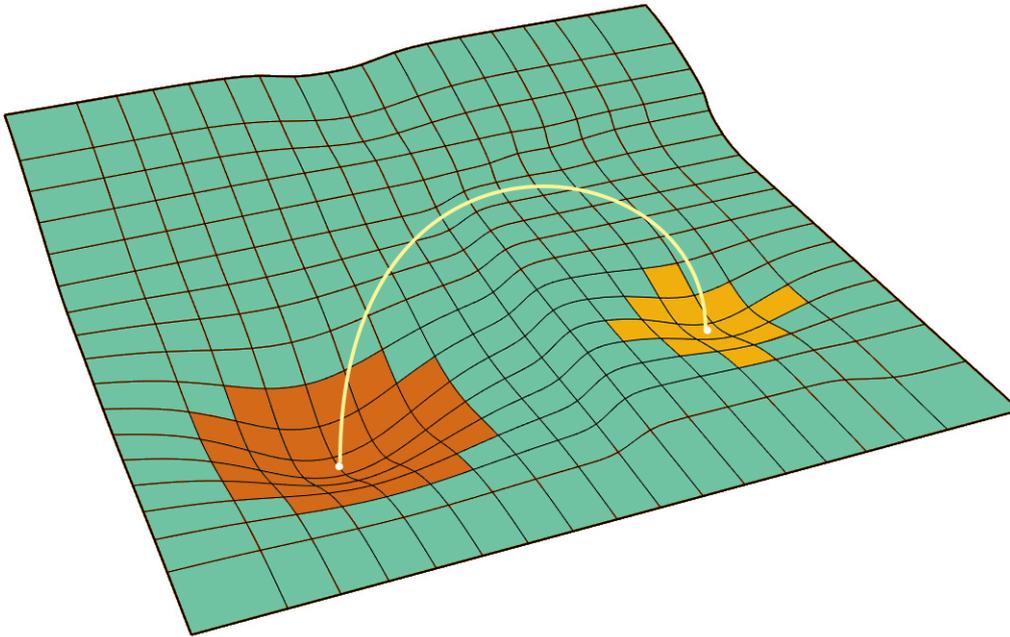
Una extensión es considerar los comportamientos de los patrones producidos por una lista tabú. Con esto se puede por ejemplo especificar un número de movimientos necesarios en una cierta dirección antes de permitir algún retorno.

3.5.2.- La diversificación

El concepto de diversificación se basa en buscar más en áreas no exploradas. Lo que trata de aplicar y estimar es una distancia de escape del óptimo local.

La Figura - 6 muestra una representación gráfica tridimensional de un proceso de diversificación, el cual se suele realizar después de la intensificación obteniendo resultados, en el nuevo espacio, que difieren significativamente de las soluciones ya evaluadas.

Figura - 6: Diversificación en TS



El proceso trata de explorar regiones que contrastan fuertemente con las regiones exploradas hasta el momento pero sin hacerlo de manera aleatoria, sino teniendo en cuenta el proceso de búsqueda usado hasta ese momento. Son las funciones de memoria a largo plazo las que se encargan de la labor de diversificar la búsqueda, empleando principios más o menos contrarios a los de memoria a corto plazo.

Para escapar de puntos muy atractivos se requieren ciertas acciones, entre las que se pueden destacar, por ejemplo:

- Imponer restricciones más estrictas a las condiciones tabú: lo que permitirá excluir un número más elevado de movimientos.
- Usar información acumulada de direcciones hacia las que el resultado estaba mejorando: empleando dicha información se tratará de favorecer los movimientos que se dirijan en esa dirección.
- Penalizar movimientos que contengan atributos muy usados en el pasado.
- Incorporar elementos probabilísticos: a través de ello se podrán preferir movimientos con cierta probabilidad.

No hay que olvidar que una vez que se sale de una zona de fuerte atracción las condiciones anteriormente descritas se pueden relajar para poder así explorar más intensamente la nueva zona del espacio de búsqueda.

3.6.- LA MEMORIA

La búsqueda tabú, como se ha explicado, se caracteriza porque emplea una estrategia basada en el uso de estructuras de memoria para escapar de los óptimos locales, en los que se puede caer al desplazarse por el espacio de soluciones de una solución a otra.

El uso de las estructuras de memoria empleadas en la TS es habitualmente de dos tipos:

- Explícito: la solución se almacena de manera completa, normalmente son soluciones élite (que dan un óptimo local) ya visitadas durante la búsqueda. En ocasiones se aplica una extensión de esta memoria explícita, registrando vecindarios muy atractivos e inexplorados de las soluciones élite.
- Implícito (de atributos): se recoge información sobre ciertos atributos de las soluciones visitadas, para poder orientar la búsqueda. Se registra información acerca de los atributos o características que cambian al moverse de una solución a otra con el propósito de guiar la búsqueda.

Los dos tipos de memoria son complementarios, puesto que el primer tipo permiten expandir los entornos de búsqueda mediante la inclusión de soluciones élite, mientras que el segundo tipo los reduce prohibiendo determinados movimientos.

Es importante tener en cuenta que los métodos basados en búsqueda local precisan de la exploración de un gran número de soluciones en poco tiempo, por ello es fundamental reducir el tiempo computacional de operaciones recurrentes, y es entonces cuando la orientación de la búsqueda mediante el registro de los atributos se convierte en una herramienta de gran utilidad.

Las estructuras de memoria en las metaheurísticas de búsqueda tabú operan en cuatro direcciones principales, tal y como se recoge en Glover y Laguna (1997): calidad, influencia, tiempo y frecuencia. Estas dos últimas se complementan y emplean en las estructuras de memoria a corto y a largo plazo.

3.6.1.- La Calidad

La calidad se refiere a la habilidad para diferenciar entre buenas y malas soluciones. Teniendo en cuenta la función objetivo, identifica qué las hace tan buenas e incentiva la búsqueda para reforzar las acciones que conducen a una buena solución y penaliza las acciones que conducen a soluciones malas.

La flexibilidad de la estructura de memoria hace posible que la búsqueda sea guiada en un contexto multiobjetivo, donde la bondad de una dirección de búsqueda particular puede estar determinada por más de una función.

El concepto de calidad en la búsqueda tabú es más amplio que el usado en otros métodos de optimización, en los cuales se considera que un movimiento es de mejor calidad que otro porque produce una mejora, y puede ser adaptado a la naturaleza del problema.. Según el enfoque de la búsqueda tabú un movimiento, por ejemplo, puede ser de mejor calidad si su frecuencia de ocurrencia en el pasado es baja o no ha ocurrido antes y nos permite explorar nuevas regiones.

3.6.2.- La Influencia

La influencia considera el impacto de las elecciones realizadas durante la búsqueda y mide el grado de cambio provocado en la estructura de la solución, no sólo en relación a su calidad sino también en su estructura.

El hecho de almacenar información sobre las elecciones de un elemento de una solución particular aporta un nivel adicional de aprendizaje, al registrar qué elementos o atributos crean ese impacto.

Se puede privilegiar a los movimientos definidos como influyentes, pero sin restringir la opción de poder optar en alguna iteración por otros menos influyentes. Éstos pueden ser admitidos si proporcionan mejores valores, hasta que las mejoras a partir de ellos no sean significativas. A partir de ese momento, y en ausencia de movimientos de mejora, los criterios de aspiración se varían para dar a los movimientos influyentes un rango mayor y que así puedan salir de la lista tabú antes del plazo establecido en su tamaño tabú.

3.6.3.- La Memoria a Corto Plazo

La memoria a corto plazo es la memoria reciente, en la que se almacenan los últimos movimientos realizados, y que puede ser utilizada para recordar aquellos movimientos que hacen caer en soluciones ya exploradas.

Las memorias a corto y a largo plazo presentan sus propias estrategias y es importante diferenciarlas, pero ambas producen una modificación de la estructura del entorno de la solución actual. El efecto de las estructuras de memoria, al hacer que la búsqueda tabú mantenga un historial selectivo (H) de los estados encontrados durante la búsqueda, consiste en el reemplazo del entorno actual, $N(s_{actual})$, por un entorno modificado $N(H, s_{actual})$.

Respecto a la memoria a corto plazo, $N(H, s_{actual})$ es un subconjunto de $N(s_{actual})$ y es precisamente la clasificación tabú quien determina los elementos de $N(s_{actual})$ excluidos de $N(H, s_{actual})$. En las estrategias de memoria a largo plazo, por el contrario, $N(H, s_{actual})$ puede contener soluciones que no estén incluidas en $N(s_{actual})$ (que serán, generalmente, óptimos locales).

La memoria a corto plazo, según se define en Casado (2005), constituye un método agresivo de exploración, que tiene por objeto realizar siempre el mejor movimiento, sujeto a las posibles soluciones factibles fruto de la aplicación de las restricciones tabú. Su cometido es permitir ir más allá de los óptimos locales, realizando siempre, movimientos de alta calidad.

A diferencia de la memoria explícita que guarda la solución entera, esta memoria no almacena soluciones completas, sino que se basa en el almacenamiento de atributos siendo por tanto un tipo de memoria implícita, definida anteriormente.

Un atributo de un movimiento o de una solución visitada, es cualquier característica asociada a dicho movimiento o solución. Los atributos seleccionados que se presentan en soluciones recientemente visitadas son designados como *tabú-activos*, lo que provoca que posibles soluciones que contengan atributos *tabú-activos* pasen a ser *soluciones-tabú*, prohibidas durante un periodo determinado de tiempo o por un número concreto de iteraciones.

Para que las listas que contienen los atributos tabú no sean demasiado grandes se requiere su actualización tras cada movimiento. Así sólo permanecerá en la memoria la información más reciente y es, por ello, posible que el estado tabú de cada atributo cambie en cada iteración.

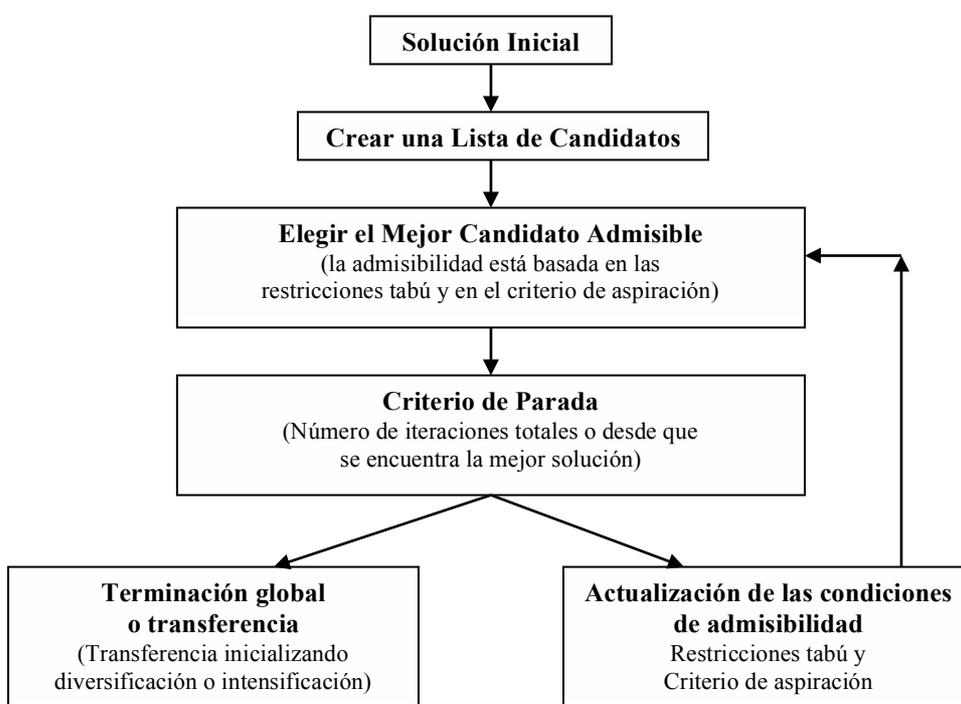
A medida que se itera, la búsqueda transita por regiones distintas y en dicho paso se libera de la condición tabú a ciertos movimientos. Se denomina periodo tabú o *tabu tenure* al periodo de tiempo en el que un atributo permanece *tabú-activo*.

El periodo tabú puede ser desigual según los tipos de atributos que se traten, o cambiar según diferentes estados de proceso de búsqueda.

Es necesario definir funciones de memoria que permitan almacenar y usar eficientemente los estados tabú de los diferentes atributos, a fin de determinar cuándo son aplicables las restricciones tabú. También es importante la duración del periodo tabú, abundando los estudios realizados sobre esta cuestión que demuestran que un buen valor de este periodo depende del tamaño del problema que se aborda. Periodos excesivamente cortos, generan problemas de ciclado y periodos demasiado largos, pueden originar un deterioro en la calidad de las soluciones encontradas.

En Glover (1990) se muestra un diagrama de los componentes de la memoria a corto plazo presentado más abajo en la Figura - 7. El diagrama esquematiza cómo una vez iniciado el algoritmo, se parte de una solución inicial de buena calidad obtenida por otros métodos, se crea una lista de candidatos y se procede a elegir el mejor candidato posible. En la creación de la lista de candidatos será determinante la existencia de listas tabú así como de los criterios de aspiración.

Figura - 7: Componentes de la memoria a corto plazo (Glover, 1990)



En búsqueda tabú se han desarrollado diferentes estrategias para confeccionar estas listas de candidatos en un periodo de tiempo razonable, entre otras destacar: *Aspiration Plus*, lista de élite, filtros sucesivos...

En la elección del mejor candidato, y puesto que normalmente el número de movimientos tabú es inferior en comparación con todos los movimientos posibles, se comprueba inicialmente el estado tabú del movimiento y, en caso de no ser tabú, se admite de forma inmediata; en caso contrario, el criterio de aspiración puede dar una segunda oportunidad para calificar el movimiento como admisible. Es habitual que, en la etapa de memoria a corto plazo, el algoritmo se detenga tras un número de iteraciones sin producirse mejora alguna.

La condición de terminación del algoritmo puede ser muy amplia, aunque usualmente se suelen dar condiciones del siguiente tipo:

- Tiempo máximo de aplicación del algoritmo.
- Número máximo de vecinos generados durante la aplicación del algoritmo.
- Un valor mínimo de tiempo combinado con una ley de variación de tiempo.
- Número máximo de iteraciones sin variar la solución mejorada.
- Alguna combinación de las anteriores.

3.6.4.- La Memoria a Largo Plazo

Es patente el hecho de que a veces, tan sólo usando estrategias basadas en memoria a corto plazo las soluciones obtenidas tienen buena calidad. Pero incluir memoria a largo plazo, así como las estrategias asociadas a la misma, hacen de la búsqueda tabú una estrategia más robusta. Se aplicará memoria a largo plazo en aquellos problemas en los que se desee refinar los resultados obtenidos de aplicar memoria a corto plazo.

La memoria a largo plazo es una memoria basada en la frecuencia y proporciona un tipo de información que complementa la información suministrada por la memoria basada en el corto plazo, de forma que amplía la base para seleccionar movimientos preferidos. En esta estructura de memoria se registra la frecuencia de ocurrencias de los movimientos, las soluciones o sus atributos en las soluciones visitadas, tratando de descubrir nuevas regiones.

La medida de la frecuencia, generalmente, se expresa como proporciones, en el numerador se representa el número de ocurrencias de un evento particular, y en el denominador se enumera el número total de ocurrencias, la suma de los numeradores, el máximo valor del numerador o la media del valor del numerador, según el problema que se esté estudiando.

Se diferencian dos tipologías dentro de la medida de las frecuencias en las estrategias de memoria a largo de la búsqueda tabú, y estas son:

- Frecuencias de transiciones: número de veces que un atributo cambia de valor
- Frecuencias de residencia: número de iteraciones durante la cuales un atributo determinado pertenece a la solución generada

Ambos tipos de frecuencias amplían la información cualitativa de las soluciones. Por ejemplo, un valor alto de la frecuencia de residencia, puede indicar que un atributo es altamente atractivo en una secuencia de soluciones de alta calidad, o puede indicar lo contrario dentro de una secuencia de soluciones de baja calidad. Y es más, si la frecuencia de residencia en una secuencia con soluciones tanto de alta como de baja calidad, puede apuntar a que se trate de un atributo fortalecido (o excluido) que restringe al espacio de búsqueda, y que necesita ser desechado (o incorporado) para fomentar la diversificación. El caso de un valor elevado de la frecuencia de transición, puede indicar la capacidad de un atributo determinado de incorporarse en una solución a fin de realizar un buen ajuste en un momento concreto y volver a salir en la siguiente iteración.

La memoria a largo plazo es el fundamento de las estrategias de intensificación y de diversificación, aunque sus elementos fundamentales también se presentan en las estrategias de memoria a corto plazo. La intensificación en el corto plazo se da al obligar a elegir soluciones atractivas en las listas tabú, y la diversificación al extender la búsqueda a nuevas soluciones cuyos movimientos no estén suprimidos.

Estas estrategias de memoria a largo plazo permiten introducir componentes penalizadoras en función de la frecuencia de la ocurrencia de un evento. Se pueden conformar así listas tabú graduadas, cosa que no sucede en el caso de la memoria basada en lo reciente (a corto plazo) que obliga a movimientos de la forma todo-nada. La penalización a introducir, en muchos problemas, suele expresarse como un múltiplo lineal de una medida de frecuencia.

La revisión realizada de la literatura indica que, en la mayoría de las experiencias comprobadas de problemas en los que se ha aplicado búsqueda tabú, la utilización de estrategias de memoria a largo plazo no requiere necesariamente recorrer demasiadas soluciones antes de que sus beneficios se hagan perceptibles.

3.7.- OTRAS ESTRATEGIAS

La búsqueda tabú, como se ha explicado, aplica estrategias de intensificación y diversificación. Además de esas estrategias básicas nuevos elementos y técnicas se están incorporando en los procedimientos de TS. A continuación se analizan dos de ellos con más detalle: el Path Relinking y la Oscilación estratégica.

3.7.1.- Path Relinking

La estrategia Path Relinking o re-encadenamiento de trayectorias surge como una integración de las estrategias de intensificación y diversificación. Cabe destacar los estudios precursores de Glover (1989) y Glover (1993).

Originalmente, Path Relinking se explotó como un método de intensificación dentro de la búsqueda tabú, pero en la actualidad se ha desarrollado hasta adquirir identidad propia, Ho y Gendreau (2006). El re-encadenamiento de trayectorias se puede interpretar como un método evolutivo por cuanto las soluciones son generadas por combinación de otras soluciones previas, pero prescinde del carácter aleatorio ya que utiliza reglas sistemáticas y deterministas para combinar las soluciones.

Esta estrategia genera soluciones nuevas, al explorar trayectorias que conectan buenas soluciones conocidas como soluciones élite (óptimos locales). Se comienza desde una de esas soluciones, llamada solución inicial, y se genera un camino en el vecindario que conduce a otras soluciones élite conocidas, llamadas soluciones guía.

En un conjunto dado de soluciones, el papel de la solución inicial y de una solución guía se pueden alternar. Esto es, se puede generar simultáneamente un conjunto de soluciones actuales, extendiendo caminos diferentes, y permitiendo que una solución inicial sea reemplazada (como una solución orientadora para las otras) por otra, si ésta satisface un criterio de aspiración suficientemente fuerte. Debido a que sus papeles son intercambiables, la solución inicial y la guía son llamadas de forma global soluciones de referencia.

La potencialidad de la estrategia radica en el hecho de que entre dos soluciones élite, se pueda trazar un camino que las una, y que dicho camino está formado por soluciones que contengan atributos de ellas. El proceso de generar caminos entre soluciones de referencia está acompañado por movimientos de selección que incorporan atributos contenidos en las soluciones que operan como soluciones guía. Si se requiere el proceso también puede continuar más allá de una solución de referencia, cambiando el criterio de selección de movimiento para que estratégicamente introduzca atributos que no están en las soluciones guías.

En ambos todo caso, los atributos están ponderados para establecer qué movimientos tienen una prioridad superior. La formación de los caminos dentro de un vecindario vuelve a asociar puntos previos pero de una forma diferente a las registradas en la historia previa de búsqueda.

Las estrategias de diversificación y de intensificación se benefician también cuando la búsqueda agrega información acerca de las soluciones adicionales evaluadas durante el examen de los movimientos no adoptados, es decir de las soluciones evaluadas pero no visitadas.

3.7.2.- Oscilación Estratégica

En un inicio fue considerada como una estrategia de refinamiento muy específica, que ha resultado ya ser un complemento muy eficiente en diferentes metodologías metaheurísticas.

La oscilación estratégica se basa en guiar los movimientos hasta que se llegue a un límite que por lo general representaría un punto donde el método debe parar. Pero en lugar de detenerse, se le permite al procedimiento cruzar ese límite modificando la definición de entorno y el criterio de evaluación.

Se perturban las normas, de forma que pueda avanzar desde un punto límite hasta una profundidad determinada y luego volverá en sentido contrario, hasta alcanzar un nuevo punto de oscilación, iniciando de nuevo el proceso de rebote; es este movimiento cíclico el que da nombre a la estrategia. La repetición de este proceso oscilatorio de cruce de límite proporciona un marco propicio para combinar estrategias de intensificación y diversificación a corto y largo plazo.

Este procedimiento combina la intensificación, en el lado admisible de la frontera, con la diversificación, ya que de otra manera hubiera sido difícil acceder a nuevas regiones en un proceso de movimientos cortos. El control sobre estos movimientos cíclicos se establece generando evaluaciones modificadas y reglas de movimiento, dependiendo de la zona en la que se esté explorando.

4.- EL Dial-A-Ride PROBLEM (DARP)

4.- EL Dial-A-Ride PROBLEM (DARP)

4.1.- INTRODUCCIÓN

El Dial-A-Ride es una forma de responder a la demanda de transporte público que funciona con una flota de vehículos que operan en las rutas flexibles, es una modalidad de transporte a la demanda.

En el momento en el que los clientes llaman a un servicio de Dial-A-Ride solicitando el transporte desde un lugar especificado a un destino concreto, la operadora del servicio planifica y distribuye los vehículos para poder satisfacer la demanda de los clientes. Esta modalidad de servicio puede describirse como un sistema de transporte público colectivo en el que se planifican las rutas y los horarios para hacer posible el transporte de varios pasajeros, con necesidades distintas, viajando juntos durante un día normal de funcionamiento. En este transporte a la demanda se requieren traslados puerta-a-puerta con una restricción temporal, es decir, cada usuario especifica el lugar donde debe ser recogido, el lugar a donde necesita desplazarse y el horario requerido para el traslado.

Los problemas que tienen como finalidad el diseño y la generación de rutas que satisfagan la demanda de los clientes se denominan problemas de Dial-A-Ride (DARPs). Desde el punto de vista de la programación, el DARP generaliza el problema de rutas de vehículos con carga y descarga (*Pickup and Delivery Vehicle Routing Problem - PDVRP*) y el problema de rutas de vehículos con ventanas de tiempo (*Vehicle Routing Problem with Time Windows - VRPTW*). Una revisión de estos tipos de problemas se puede encontrar en los estudios de Desaulniers et al. (1995) y de Desrosiers et al. (2002).

Lo diferencia principal del DARP frente a otros problemas de rutas es la inclusión de la perspectiva humana. Tal y como se recoge en el trabajo de Cordeau y Laporte (2003a), el hecho de transportar personas hace que se pretenda encontrar un equilibrio entre la minimización de los costes del transporte y la reducción de los inconvenientes ocasionados a los usuarios. Es por ello, que los tiempos de viaje son fundamentales en la resolución de este tipo de problemas, puesto que los clientes desean pasar el menor tiempo posible dentro del vehículo y llegar a tiempo a su destino.

Debido a los múltiples objetivos contrapuestos y a las restricciones a considerar en el problema, los DARPs son, en su mayoría, difíciles de resolver. El DARP con múltiples vehículos, que se desarrollará en profundidad más adelante en este capítulo, es considerado de la clase *NP-hard*, (Garey, 1978).

En las últimas décadas, la aplicación exitosa de metodologías metaheurísticas para resolver muchos problemas combinatorios ha hecho que sean consideradas adecuadas para su empleo en la resolución del DARP. Además, aunque es recomendable el uso de métodos exactos para resolver DARPs, las limitaciones de tiempo y el elevado número de solicitudes en los servicios reales de Dial-a-Ride, ponen en evidencia la ineficiencia del uso de algoritmos exactos para su resolución. Por consiguiente, se justifica la elección de técnicas metaheurísticas para la solución del DARP.

Aplicaciones prácticas reales del DARP pueden encontrarse operando actualmente en forma de taxis compartidos, servicios de ambulancias, e incluso algunos tipos de servicios de mensajería y paquetería. El colectivo social sobre el que habitualmente se requiere de la resolución de DARPs para permitir su transporte, generalmente, se compone por personas de edad avanzada o aquellas personas que tienen algún tipo de discapacidad que impide su movilidad de manera independiente. Consecuentemente, la utilización de los DARPs está esencialmente ligada a los servicios de transporte de personas de movilidad reducida.

4.2.- DESCRIPCIÓN DEL PROBLEMA

En el trabajo de Tang et al. (2010) se describe de forma resumida el problema de Dial-A-Ride de la siguiente manera: *Existiendo un número de solicitudes en las que un cliente se debe recoger de una ubicación específica (punto de recogida) y entregar en un destino específico (punto de entrega), el problema debe determinar el conjunto de rutas y horarios que acomoden dichas peticiones de forma que la distancia total sea mínima y se cumplan las restricciones de nivel de servicio. Entre el conjunto de restricciones comunes destacan las limitaciones máximas de tiempo de espera, tiempo máximo de espera permisible en una parada antes de la salida, y las limitaciones máximas de tiempo de viaje, tiempo máximo permitido entre la recogida y la entrega.*

Un caso típico de DARP se presenta en el estudio de Hunsaker y Savelsbergh (2002), cuyo extracto se muestra a continuación: *Una compañía de transporte que oferta un servicio de taxi colectivo a la demanda tiene el cometido de transportar a los clientes desde su origen hasta su destino. Diferentes clientes pueden compartir el mismo vehículo durante su trayecto, pero teniendo siempre en cuenta, que puede llevar un número máximo limitado de pasajeros. El objetivo es minimizar la distancia total recorrida por el vehículo en el transporte de todos los pasajeros teniendo en cuenta las restricciones relacionadas con el nivel de servicio.*

4.2.1.- Los agentes implicados

Siguiendo con el ejemplo de un caso típico de problema de Dial-A-Ride presentado en el párrafo anterior y con el fin de comprender mejor la programación de los DARPs, será conveniente examinar las preocupaciones de los tres principales elementos que intervienen: los clientes, los conductores y la compañía de transporte que oferta el servicio.

4.2.1.1.- El cliente

Un cliente prefiere un servicio tipo taxi tradicional en el que se ofrecen conexiones directas desde el origen al destino, sin desvíos innecesarios.

Básicamente, se puede decir que un cliente quiere que el tiempo de viaje sea corto y el servicio rápido. Por ejemplo, un cliente no quiere viajar durante cincuenta minutos cuando existe una ruta directa por la que llega a su destino en tan sólo quince minutos. Del mismo modo, un cliente no entiende que le digan que el horario de llegada del vehículo que espera puede ser en cualquier momento entre las 10 y las doce de la mañana, si en cambio cabe la opción de realizar una reserva de su viaje en la que le aseguren que entre las diez y diez minutos después de las diez va ser recogido. Es elemental pues que cada cliente desea realizar viajes bajo las mejores condiciones de comodidad y fiabilidad a precios asequibles.

4.2.1.2.- El conductor

Desde la perspectiva de un conductor que opere en un servicio de transporte colectivo a la demanda, puede que ciertas rutas sean preferidas sobre otras.

La preferencia del conductor puede estar influenciada por el número de clientes a los que atender en cada ruta y por las condiciones del tráfico que se encuentre, tales como: la congestión, el número de intersecciones o cruces semafóricos y los límites de velocidad de las vías por las que circule. La equidad de la carga de trabajo debe darse para evitar una percepción de injusticia. Además, un conductor puede preferir tener un único descanso largo en un día a varios cortos (o viceversa). Cada conductor tiene, por tanto, una utilidad individual a maximizar. Así, no es sencillo generar horarios para complacer a todos.

4.2.1.3.- La compañía de transporte

Dados los recursos disponibles, tales como vehículos y conductores, una compañía de transporte tiene como objetivo cubrir todas las demandas de los clientes a un bajo costo operativo.

Sin embargo, una agencia se enfrenta con cuestiones contrapuestas. Por un lado, le gustaría proporcionar un servicio rápido y fiable a los clientes a precios asequibles. Por otro lado, tiene que lidiar con costes de operación y recursos escasos. La singularidad y el valor social añadido que un sistema de transporte colectivo a la demanda del tipo Dial-A-Ride posee puede en ocasiones proporcionar oportunidades para que la compañía que opera el servicio reciba financiación externa. En estos casos las políticas de explotación adoptadas por la empresa estarán encaminadas a demostrar que los fondos recibidos son bien utilizados y, al mismo tiempo, que los clientes están satisfechos con el servicio, es decir tendrá que rendir cuentas ante la comunidad y las fuentes de financiación.

4.2.2.- Los problemas de rutas de vehículos

Como ha sido introducido anteriormente, en lo que respecta a la programación, el DARP generaliza dos tipos concretos de problemas de rutas de vehículos (Vehicle Routing Problems o VRPs). En todo modelo VRP se dispone de un conjunto de vehículos, un grupo de clientes a visitar y uno o varios depósitos desde donde se inician las rutas; los vehículos deben satisfacer la demanda de los clientes. Son las distintas ordenaciones entre clientes, vehículos y depósitos, así como las particularidades respecto a entregas y recogidas, las que dan pie a diferentes variantes del problema de rutas de vehículos.

A continuación se describen las dos tipologías de VRP íntimamente relacionadas con el DARP desde un punto de vista del modelizado y la programación.

4.2.2.1.- VRP con carga y descarga

También conocido como PDVRP (*Pick-up and Delivery Vehicle Routing Problem*). Para que la recogida sea efectiva, es necesario disponer de espacio dentro del vehículo, que hace que la solución del problema sea mucho más compleja.

En el marco de la revisión de la literatura científica realizada se ha encontrado que los primeros trabajos se atribuyen a Wilson et al. (1971), donde se trataron de resolver problemas aplicados al transporte de personas. En posteriores publicaciones de Wilson y Colvin (1977) y en Wilson y Weissberg (1977) se amplió la investigación en el intento de optimizar redes tanto en Haddonfield como en Rochester, Estados Unidos. Fue en estos trabajos en los que se introdujo el concepto de creación de rutas a partir de inserción secuencial de vértices.

En el estudio descrito en Toth y Vigo (1997) se incorpora tanto una implementación de búsqueda tabú en el proceso de inserción de vértices en la solución, como otro de mejora.

Años más tarde y ya dentro del campo de los procedimientos exactos, se pueden encontrar algunas referencias en los trabajos de Kohl et al. (1999), donde incorporan el concepto de las desigualdades válidas, y la publicación de Du Merle et al. (1999), donde se emplean variables de perturbación acotadas.

El principal campo de aplicación de esta variante del VRP ha sido el transporte aéreo y marítimo de personas, el transporte de personas discapacitadas y el transporte escolar. Estos problemas reales llevan aparejadas, en muchas ocasiones, restricciones temporales (ventanas de tiempo) e, incluso a veces, condiciones por tiempo máximo de permanencia en el vehículo (caso del transporte escolar), problemática analiza en Pacheco et al. (2013).

4.2.2.2.- VRP con ventanas de tiempo

En esta variante, conocida como VRPTW (*Vehicle Routing Problem with Time Windows*), además de una capacidad en el vehículo, cada cliente tiene asociada una franja de tiempo en el que está permitida la recogida o la llegada. Es habitual considerar dos casos:

- el primero: los vehículos no pueden realizar un servicio después de una hora determinada, pero sí pueden esperar en caso de llegar antes de tiempo
- el segundo: se consiente la violación de las restricciones asociadas a las ventanas de tiempo, siempre y cuando se pague por ello un determinado coste.

Los primeros estudios de este tipo de problemas se basaron en la aplicación de casos concretos, como los que se recogen por Pullen y Webb (1967), Knight y Hofer (1968) y Madsen (1976). Fue más adelante cuando se comenzaron a presentar estudios basados en el desarrollo de nuevos algoritmos de resolución, también con aplicaciones a casos reales e, incluso, a variantes más simplistas con ventanas de tiempo, como las que se encuentran en Desoiers et al. (1984).

Las publicaciones iniciales en las que se empleaban algoritmos exactos para la resolución de este tipo de problemas se deben a Kolen et al. (1987) y a Desrochers et al. (1992). Gracias al carácter precursor de dichos estudios surgieron nuevos análisis basados en técnicas de *Branch and Cut*, que presentaron resultados muy prometedores, tal y como se recoge en Cook y Rich (1999) y en el trabajo de Kohl et al. (1999).

Es destacable la existencia de unas librerías, llamadas Instancias de Solomon, presentes en el trabajo de Solomon (1987). En ellas se recogen una serie de casos, los cuales se dividen en seis grupos (R1, R2, C1, C2, RC1 y RC2) y se emplean como ejemplo en las distintas implementaciones propuestas para poder comprobar la bondad de las mismas.

El empleo con éxito de algoritmos metaheurísticos para la resolución de VRPTW se presentan en las publicaciones de Homberger y Gehring (1999), en Cordeau et al. (2000) y, más tarde, en el estudio de Taillard et al. (2001). Más recientemente, se pueden encontrar investigaciones en las que la hibridación de métodos como metodología para solucionar estos problemas toma más relevancia, este es el caso de trabajos de Kallehauge et al. (2007) y Kallehauge (2008).

4.2.3.- Las características generales del DARP

Las representaciones que del DARP se encuentran en la literatura no muestran una clara y única definición del problema.

En el caso de las investigaciones presentadas en Fu (2002) y en Cordeau y Laporte (2003a), el DARP debe incluir ventanas de tiempo o, al menos, un intervalo de tiempo durante el cual el cliente espera la llegada del vehículo que le recogerá y dará comienzo a su viaje. Con el fin de remarcar la inclusión de ventanas de tiempo en los problemas estudiados, en Psarafatis (1980 y 1983) y, posteriormente, en Madsen et al. (1995), se hace hincapié en la resolución del problema de Dial-A-Ride con ventanas de tiempo, surgiendo la denominación DARPTW. Por el contrario, se han encontrado autores cuyos trabajos presentan el DARP sin ningún tipo de ventanas de tiempo, este es el caso de Healy y Moll (1995).

El hecho de que muchos de los estudios encontrados sobre el DARP deriven del análisis de servicios de transporte reales, como los presentados por Alfa (1986), Madsen et al. (1995), Toth y Vigo (1997) y Baugh et al. (1998), y puesto que éstos son función de la disponibilidad de recursos y de la política de

gestión de los organismos de transporte, diferentes en cada caso, se producen variaciones en los objetivos del modelo y en sus restricciones, complicando aún más la clasificación única de los DARPs.

En general, el DARP puede ser clasificado según la naturaleza del sistema de Dial-A-Ride y según las consideraciones de las políticas de gestión que operen. Con el fin de reflejar la diversidad problemática que se da en el mundo real, la naturaleza de un sistema de transporte a la demanda puede permitir la clasificación del problema como sigue:

- Según el patrón de orígenes y destinos:
 - De un origen a muchos destinos
 - De múltiples orígenes a múltiples destinos
 - De muchos orígenes a un solo destino
- Según el proceso de reserva del viaje:
 - Reserva con antelación (modelos estáticos)
 - Reserva en tiempo real (modelos dinámicos)
- Según el número de: terminales de transporte (depot):
 - Con terminal única
 - Con múltiples terminales
- Según el tipo de solicitud:
 - Con condiciones temporales en la partida
 - Con condiciones temporales en la llegada
 - Con indicaciones sobre ambas
- Según la estimación del tiempo de viaje:
 - Modelos estáticos
 - Modelos dinámicos

La segunda clasificación, indicada arriba, tiene en cuenta las diversas políticas de gestión que se pueden considerar en la formulación de un DARP y éstas pueden ser:

- Por los requisitos del sistema de transporte:
 - Según las ventanas de tiempo a considerar:
 - Con ventana de tiempo en la recogida
 - Con ventana de tiempo en la llegada
 - Con ventanas de tiempo en recogida y llegada
 - Con ventanas de tiempo específicas de la solicitud realizada (intermedias)
 - Según el máximo tiempo de viaje permitido:
 - Con tiempo de viaje fijo
 - Con tiempo de viaje variable
 - Según el momento de planificación:
 - Con horizonte fijo
 - Con horizonte variable
 - Según el tiempo de espera permitido:
 - Sin espera
 - Con espera en el punto de origen
 - Con espera en el punto de destino
 - Con espera en origen y destino
- Por los objetivos de la empresa que oferta el servicio de transporte:

- Con minimización del tiempo de viaje para el cliente
- Con minimización del tiempo de espera para el cliente
- Con minimización del tiempo de recorrido del vehículo
- Con minimización del número de vehículos utilizados
- Con maximización del número total de viajes realizados por cada vehículo

Se considera relevante señalar que el objetivo último de la empresa operadora de un servicio de transporte es minimizar el coste total del sistema, en este caso, de Dial-A-Ride y ésto no es fácilmente cuantificable. Consecuentemente, los objetivos mencionados en la lista deben ser convertidos y analizados por su coste en el modelo del problema.

4.2.4.- La formulación del problema

Existen varias formas de formular este tipo de problemas; en este caso, se va a presentar el modelo de tres índices de Cordeau (2006), que ha servido de base en la literatura para modelos con características similares. El objetivo perseguido en el modelo es minimizar el coste. El modelo considera lo siguiente:

Sea: $G = (V, A)$ un grafo dirigido

V el conjunto de vértices que se divide en $\{0, 2n + 1, P, D\}$ donde:

0 y $2n + 1$ son dos copias del depósito,

$P = \{1, \dots, n\}$ es el conjunto de vértices de recolección de personas y

$D = \{n + 1, \dots, 2n\}$ es el conjunto de vértices de destino

Una solicitud es una pareja $(i, n + i)$ donde:

$i \in P$ y

$n + i \in D$

A cada vértice $v_i \in V$ se le asocia:

una demanda q_i , con:

$$q_0 = q_{2n+1} = 0,$$

$$q_i \geq 0 \text{ para } i = 1, \dots, n \text{ y}$$

$$q_i = -q_{i-n} \text{ para } i = n + 1, \dots, 2n$$

y una duración de servicio $d_i \geq 0$ con:

$$d_0 = d_{2n+1} = 0$$

El conjunto de arcos se define como:

$$A = \{(i, j): i = 0, j \in P \text{ o } i, j \in P \cup D, i \neq j \text{ y } i \neq n + j \text{ o } i \in D, j = 2n + 1\}$$

La capacidad de un vehículo k es Q_k

La máxima duración de la ruta del vehículo $k \in K$ se denota por T_k

El coste de utilizar el arco (i, j) por el vehículo k es igual a c_{ij}^k

El tiempo de viaje del arco (i, j) es t_{ij}

El tiempo máximo de viaje se denota por L

La ventana de tiempo del vértice i es $[e_i, l_i]$

El problema del Dial-A-Ride consiste, según definió Cordeau, en diseñar m rutas de vehículos en G tales que:

- Cada ruta inicie y finalice en el depósito
- Para cada solicitud i , los vértices v_i y v_{i+n} pertenezcan a la misma ruta y el vértice v_{i+n} se visita después que el vértice v_i
- La carga del vehículo k no exceda en cualquier momento la cota Q_k
- La duración total del viaje del vehículo k no exceda la cota T_k
- El servicio en el vértice v_i inicie en el intervalo $[e_i, l_i]$ y cada vehículo deje el depósito y regrese al depósito en el intervalo $[e_0, l_0]$
- El tiempo de viaje de cualquier usuario no exceda L
- El costo total de ruteo de todos los vehículos sea minimizado

El modelo utiliza variables binarias de tres índices x_{ij}^k igual a 1 si, y sólo si, el arco (i, j) es utilizado por el vehículo $k \in K$.

Además, sea u_i^k el tiempo en el que el vehículo k inicia el servicio en el vértice i , w_i^k la carga del vehículo k al partir del vértice i y r_i^k el tiempo de viaje del usuario i . La formulación del modelo es la siguiente:

Minimizar

$$\sum_{k \in K} \sum_{i \in V} \sum_{j \in V} c_{ij}^k x_{ij}^k \quad (1)$$

Sujeto a

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \quad i \in P \quad (2)$$

$$\sum_{i \in V} x_{0i}^k = \sum_{i \in V} x_{i,2n+1}^k = 1, \quad k \in K \quad (3)$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{n+1,j}^k = 0, \quad i \in P, k \in K \quad (4)$$

$$\sum_{j \in V} x_{ji}^k - \sum_{j \in V} x_{ij}^k = 0, \quad i \in P \cup D, k \in K \quad (5)$$

$$u_j^k \geq (u_i^k + d_i + t_{ij})x_{ij}^k \quad i, j \in V, k \in K \quad (6)$$

$$w_j^k \geq (w_i^k + q_j)x_{ij}^k \quad i, j \in V, k \in K \quad (7)$$

$$r_i^k \geq u_{n+i}^k - (u_i^k + d_i) \quad i \in P, k \in K \quad (8)$$

$$u_{2n+1}^k - u_0^k \leq T_k \quad k \in K \quad (9)$$

$$e_i \leq u_i^k \leq l_i \quad i \in V, k \in K \quad (10)$$

$$t_{i,n+i} \leq r_i^k \leq L \quad i \in P, k \in K \quad (11)$$

$$\max\{0, q_i\} \leq w_i^k \leq \min\{Q_k, Q_k + q_i\} \quad i \in V, k \in K \quad (12)$$

$$x_{ij}^k \in \{0,1\} \quad i, j \in V, k \in K \quad (13)$$

Analizando paso a paso esta formulación se observa que las restricciones (2) y (4) aseguran que cada solicitud es atendida y llevada a su destino por el mismo vehículo. El hecho de plantear las restricciones (3) y (5) garantiza que cada vehículo inicia y finaliza su ruta en el depósito. Es la restricción (6) la que define los tiempos de inicio de servicio del vehículo. La restricción (7) está asociada a las cargas del vehículo. Por medio de la restricción (8) se definen los tiempos de viaje de los usuarios. Gracias a las restricciones (9), (10), (11) y (12) se asegura la factibilidad del modelo en cuanto a cargas y duraciones máximas. Y por último, la restricción (13) define la naturaleza binaria de las variables.

Esta es la formulación propuesta por Cordeau (2006), en la que tiene en cuenta sólo el objetivo económico de minimizar costes, pero donde se formula el modelo matemático del DARP con condiciones de precedencia, capacidad, ventanas de tiempo, tiempo de viaje y, además, se diseña un algoritmo de ramificación y corte para resolver el modelo en instancias aleatorias. Su trabajo se encuentra relacionado con el anterior planteamiento presentado en Cordeau y Laporte (2003b) donde se plantea un algoritmo basado en búsqueda tabú para resolver el DARP con ventanas de tiempo, restricciones de capacidad y condiciones en la duración máxima de la ruta y el tiempo total de viaje de cada usuario. Más adelante, ellos mismos en su publicación Cordeau y Laporte (2007) hacen una revisión de modelos y algoritmos para la resolución del DARP. El análisis de la problemática de las medidas de calidad en el servicio dentro del DARP y el estudio de un caso real de una ciudad canadiense se ve Paquette et al. (2012). A continuación y para darle solución, en Paquette et al. (2013) se desarrolla un modelo multicriterio con búsqueda tabú donde se tienen en cuenta la minimización de costes junto a otros criterios de calidad del servicio. Se ha encontrado más recientemente, en el trabajo publicado por Guerriero et al. (2014), el uso de una optimización multiobjetivo en transporte público Dial-A-Ride donde optimizan conjuntamente el tiempo máximo de viaje y el tiempo total de espera del cliente, haciendo una resolución con técnicas de ε -restricción (ε -constraint).

5.- EL TRANSPORTE PÚBLICO COLECTIVO

5.- EL TRANSPORTE PÚBLICO COLECTIVO

5.1.- INTRODUCCIÓN

El transporte público colectivo comprende aquellos medios que permiten el traslado masivo de personas de un lugar a otro, y que son, en muchos casos, operados o regulados por un organismo estatal. Se considera público desde una perspectiva jurídica ya que es un servicio de interés para la sociedad en general, independientemente de quién realice la prestación del mismo. Esta visión queda recogida en el trabajo de García-Schilardi (2014).

Las circunstancias actuales que han fomentado la búsqueda y desarrollo de sistemas eficientes de transporte colectivo han sido primordialmente:

- La movilidad sostenible: enfocada principalmente al ámbito urbano, siendo un elemento fundamental para el desarrollo de las ciudades.
- La gestión racional de los recursos: incluida dentro de los compromisos que tiene la Administración para con la sociedad. Entre los que también se haya la mejora del nivel de servicio dado a los usuarios.

Los factores arriba mencionados han acrecentado la necesidad de implementar sistemas de planeamiento, decisión y de gestión de recursos más eficientes que los tradicionalmente manejados.

En todo problema de transporte público de personas las decisiones a tomar deben recoger diferentes restricciones, únicas para cada estudio real, y considerar múltiples objetivos, que en la mayoría de los casos se encuentran enfrentados unos con otros.

Entre los objetivos que normalmente persiguen los problemas de optimización del transporte público colectivo, y que son resultado de los factores que provocan su desarrollo, están:

- Aquellos que abarcan aspectos de protección del medioambiente: se pueden incluir en esta categoría el intento de reducir el consumo de carburantes y de emisiones contaminantes,
- Aquellos que comprenden aspectos económicos: el ejemplo más claro es el que persigue la reducción del coste total de la explotación del servicio,
- Aquellos que engloban matices sociales: como es el caso de la búsqueda de la mejora del nivel de servicio ofertado a todo tipo de clientes,
- Aquellos que implican consideraciones laborales: el máximo representante dentro de este grupo está el propósito de las mejoras en las condiciones de trabajo de los conductores.

El hecho de enfrentarse a problemas reales actuales fomenta la aparición de nuevos modelos específicos, cada vez más complejos, para la optimización y toma de decisiones en el campo del transporte público colectivo.

Una vez planteado el modelo, la correcta elección o, en algunos casos, el diseño ad hoc de un método de resolución de estos modelos es una labor crítica, puesto que se da la circunstancia de que no todos son igual de eficaces y, por norma general, a mayor complejidad del problema a resolver, más diferencias surgen entre aplicar unas u otras técnicas para su resolución.

Algunas técnicas que incluyen estrategias metaheurísticas han demostrado ser, en muchos casos, mejores que algunos métodos tradicionales. Los beneficios de abordar estos problemas con técnicas adecuadas son importantes tanto para las administraciones como para los usuarios del transporte.

5.2.- EL TRANSPORTE PÚBLICO COMO RETO SOCIAL EN EUROPA

Desde la demarcación europea de la Unión Internacional de Transporte Público (UITP) se lanza el siguiente mensaje: “*Un servicio de transporte público de calidad es la piedra angular de las ciudades habitables, cuyos habitantes puedan prosperar tanto en el plano económico como en el social*”. Y se presta especial atención a su desarrollo en un futuro próximo, de aquí a dos tres años.

Una de las cuatro prioridades que persigue la política europea en materia de trasportes se centra en ‘*la investigación y la innovación*’ como herramientas para alcanzar los objetivos que se ha fijado la Unión Europea en cuanto a movilidad urbana sostenible, UITP (2014). Dentro de esta visión, se menciona que la industria proveedora y los operadores del transporte público deben trabajar en estrecha colaboración para desarrollar soluciones de movilidad más inteligentes y ecológicas y *optimizar los sistemas integrados de transporte*. También se incluye, como parte de esta prioridad, que los responsables políticos deben garantizar los medios económicos necesarios para llevar a cabo actividades de investigación e innovación del transporte público.

Con el fin de aunar los conceptos de investigación, innovación, retos sociales, Europa y futuro inmediato se presenta la comunicación de la Comisión Europea de 30 de noviembre de 2011 sobre el Programa Marco de Investigación e Innovación 2014-2020 (Horizonte 2020). Es reseñable el hecho de que los objetivos que persiguen los problemas de optimización del transporte público colectivo se vinculan de manera directa con cinco de los retos sociales expuestos en CE (2014) y estos son:

- “*Transporte inteligente, ecológico e integrado*”: ya que se desarrollan métodos para racionalizar el transporte, transporte inteligente, y se consideran diferentes objetivos entre los que se incluye la reducción del consumo combustible y la correspondiente reducción de emisiones, transporte ecológico.
- “*Energía segura, limpia y eficiente*”: debido a lo ya comentado de que entre los objetivos de estos problemas se busca racionalizar el uso de los combustibles, energía eficiente, mejorando el medioambiente y favoreciendo la sostenibilidad, energía limpia.
- “*Acción por el clima, eficiencia de recursos y materias primas*”: puesto que se busca racionalizar los costes del conjunto de las operaciones, eficiencia de recursos, y el uso de combustible, materias primas, con la consiguiente reducción del impacto medio-ambiental, acción por el clima.
- “*Sociedades inclusivas innovadoras y reflexivas*”: ya que el transporte público tiene como objetivo la mejora del nivel de servicio de todo tipo de clientes. Entre los clientes están las personas de movilidad reducida (personas discapacitadas, mayores,...) y se busca mejorar el nivel de servicio de estos usuarios, sociedades inclusivas.
- “*Sociedades seguras*”: puesto que dentro de los problemas del transporte público colectivo también se encuentra el caso particular del transporte escolar y como objetivo de mejora de calidad de un servicio de estas características se incluye la reducción de los tiempos de estancia de los alumnos en los vehículos, evitando rutas excesivamente largas, sociedades seguras.

Finalmente, en UITP (2009), se resalta que debido a la gran diversidad de campos integrados en el transporte público la metodología actual de evaluación no puede recoger todas las ventajas del transporte público y recomienda los análisis multicriterio para complementar a los clásicos análisis coste-beneficio.

5.3.- EL TRANSPORTE PÚBLICO URBANO E INTERURBANO

En la revisión bibliográfica realizada se han encontrado diversas referencias recientes sobre aplicaciones de problemas de transporte urbano. A continuación se resumirán los casos que, por su consideración de múltiples objetivos, por las técnicas de resolución de los modelos o incluso por su ámbito de aplicación, se encuentran estrechamente relacionados con el presente trabajo de investigación y, por ello, se consideran relevantes.

En los trabajos de Liu et al (2010), Bigotte et al (2010) y Moura et al (2010) se contemplan, de manera integrada, el transporte urbano con algunos aspectos del diseño de las infraestructuras viarias urbanas.

La publicación de Sharma y Mathew (2011) trata un problema de diseño de transporte urbano multiobjetivo en que se incluye la sostenibilidad como uno de los objetivos para la optimización de la red. Una perspectiva similar se adopta en el trabajo de Li et al (2011).

Con respecto a la metodología destacar, el diseño de un algoritmo de búsqueda tabú para la determinación de las frecuencias y las capacidades óptimas de los autobuses en servicios de transporte urbano, en la investigación de Ruisánchez et al (2012).

Otros ejemplos en los que se aplican métodos basados en la metaheurística búsqueda tabú en problemas reales son los elaborados por Pacheco et al (2009) y Álvarez et al (2010). En ellos se analiza el problema del transporte urbano en la ciudad de Burgos. El objetivo del trabajo radicaba en reducir los tiempos medios de trayecto por pasajero, considerando tiempos de espera más tiempos de viaje, para lo que se proponían modificaciones en las líneas y se permitían reasignaciones de autobuses a estas líneas. En el modelo se modificaban las paradas intermedias de cada línea, dejando fijas la parada inicial y final de cada línea. Se obtuvieron reducciones considerables, en algunos casos superiores a cinco minutos, en los tiempos medios de trayecto de los usuarios. Un ejemplo de los resultados del modelo se presenta en la Figura - 8 donde se observan tanto la línea del servicio original como la obtenida tras la optimización.

Figura - 8: Línea de autobús original (izda.) y modificada/suavizada (dcha.) [Álvarez et al. (2010)]



En un ámbito internacional resulta destacable, como caso concreto de transporte urbano, el sistema de autobuses de tránsito rápido (BRT o Bus Rapid Transit), en los que se habilitan los carriles centrales de grandes avenidas para el uso exclusivo de autobuses con características especiales. Este innovador sistema de transporte público urbano ha sido estudiado y aplicado con éxito en ciudades como: Brasilia, Bogotá, Ciudad de México, Sao Paulo o Shanghai entre otras.

Entre las referencias literarias más recientes que tratan el BRT con aplicación a casos reales caben destacar Schmid (2014), en el que se analiza el re-diseño de la red de BRT en la ciudad de Bogotá y García de la Parra (2015) que lo plantea para el caso concreto de Ciudad de México. En ambos estudios se intenta mejorar el nivel de servicio por medio de la reducción de los tiempos medios de espera, de transbordo y de viaje para cada usuario. Para su resolución emplean diferentes metaheurísticas: búsqueda por entorno variable (VNS) y búsqueda tabú (TS).

En lo que respecta a estudios que versen sobre el transporte interurbano no se han encontrado demasiadas referencias recientes. En el trabajo de Chen y Jang (2007) se emplea una metodología basada en redes neuronales para resolver un modelo de transporte colectivo entre ciudades teniendo en cuenta objetivos económicos y sociales.

5.4.- LAS APLICACIONES AL TRANSPORTE ESCOLAR

Los últimos trabajos que se centran en la problemática del transporte escolar son diversos, por ejemplo cabe reseñar la publicación de Park y Kim (2010) en la cual se presenta una amplia recolección de problemas de esta temática.

Es más adelante, en la revisión de Dragan et al. (2011) donde se profundiza en un problema simultáneo de rutas y localización de paradas. Un interesante caso real en ámbito internacional se trata en el estudio de Kim y Soh (2012), en el que se diseña una red de transporte escolar para una universidad ubicada en Corea. Abordando un modelo con localización en áreas rurales es destacable el presentado por Mandujano et al. (2012).

Aunando el análisis de casos reales con el problema de transporte escolar en áreas rurales dispersas, caben destacar varios trabajos en los que se han tratado datos de la provincia de Burgos. En dos de ellos, Delgado y Pacheco (2001) y Delgado (2002), abordan el problema de diseño de rutas escolares teniendo en cuenta dos objetivos, de forma no simultánea, el primer objetivo consiste en la minimización del coste de las operaciones, objetivo económico, siendo el segundo minimizar el tiempo máximo de estancia de los alumnos en los vehículos, objetivo de carácter social. El planteamiento de este objetivo social surgió como respuesta a la inquietud mostrada por las diferentes asociaciones de padres de alumnos por el excesivo tiempo que, en algunos casos, los alumnos permanecían en el vehículo escolar.

Fundamentándose en dicho problema con función objetivo social de minimización del tiempo máximo de los escolares en el vehículo de transporte, se analizaron posteriormente varios problemas, destacando los trabajos de: Corberán et al. (2002) Pacheco y Martí (2006) y Alabas-Uslu (2008). Es finalmente en el trabajo de Pacheco et al. (2013) donde se modeliza y resuelve este problema considerando simultáneamente los dos objetivos, el económico y el social.

En todos los trabajos mencionados sobre el problema del transporte escolar se proponen diferentes técnicas metaheurísticas para su resolución. Entre las más aplicadas destacar; la búsqueda tabú (TS), la búsqueda dispersa (SS), el path relinking (PR) e incluso los algoritmos genéticos (GA).

6.- EL TRANSPORTE DE PERSONAS DE MOVILIDAD REDUCIDA

6.- EL TRANSPORTE DE PERSONAS DE MOVILIDAD REDUCIDA

6.1.- INTRODUCCIÓN

Uno de los modos de integrar a todos los colectivos sociales es mejorando su movilidad y el acceso que tienen a los servicios urbanos, por lo que el transporte público colectivo se concibe como una herramienta clave en esta tarea.

Es necesario comprender en toda su extensión el término movilidad y su interrelación con el transporte para así poder ahondar en el caso concreto que se está tratando es el presente trabajo. En Gutiérrez (2009), se propone concebir al transporte como el elemento material de la movilidad, en tanto es un componente técnico de ésta. Y entiende que, a su vez, la movilidad se expresa en el transporte.

En consecuencia, la movilidad es un concepto más amplio que el de transporte. En la publicación de Dangond et al. (2010) se dice que: *“El transporte es un eslabón más en la esfera de la movilidad, un medio para que los desplazamientos ocurran. Este enfoque, obliga a comprender las necesidades individuales dentro de una estructura social. No se trata sólo de mejorar el sistema de transporte, sino de crear condiciones seguras, eficientes, igualitarias e inclusivas de acuerdo a las necesidades y deseos de las personas.”*

En el trabajo de García-Schilardi (2014) se presenta un caso concreto de la incidencia que tiene el transporte público colectivo de pasajeros en los procesos de inclusión y exclusión social.

Es preciso citar como en Stanley et al. (2010) se hace referencia a la exclusión social y el valor de la movilidad, describiendo la existencia de barreras que hacen difícil o imposible que las personas participen completamente en la sociedad. Consideran, en su trabajo, que la escasa movilidad puede ser una barrera para la integración social.

De manera comparada, en el estudio de Gutiérrez (2008) sobre geografía, transporte y movilidad, se define la inclusión social como *“la ampliación de la estructura de oportunidades hacia aquellos grupos de población que poseen ínfimas posibilidades de elección para la satisfacción de las necesidades de producción de sus vidas”*.

La movilidad junto con una de sus partes integrantes, el transporte público colectivo, son considerados elementos facilitadores en el proceso de expansión de oportunidades de desarrollo local, ya que permiten acceder, desde un punto de vista espacial, a un puesto de trabajo, un lugar donde recibir educación, unas instalaciones sanitarias y a todo otro tipo de servicios, García-Schilardi (2014). Y por ello, también desde la Unión Internacional de Transporte Público, en UITP (2007), se indica que: *“al proporcionar movilidad a todos los ciudadanos el transporte público contribuye al éxito de las políticas de inclusión social”*.

Se puede entonces compendiar que, el transporte público colectivo facilita la conectividad entre las personas y el territorio, siendo un instrumento de promoción de la cohesión, la integración y la identidad. Por tanto, uno de los requisitos fundamentales para que se produzca un proceso exitoso de integración social consistirá en planificar y poner en marcha servicios eficientes de transporte colectivo a disposición de todo tipo de usuarios.

Recogido en las reflexiones que hace Gutiérrez (2003) sobre el transporte público y la exclusión social, aunque enmarcadas en un territorio y época concretas, es interesante la visión presentada sobre la necesidad de un modelo de desarrollo para el transporte urbano que tenga en cuenta una nueva movilidad para una nueva ciudad y una nueva sociedad. Así mismo, menciona que la movilidad sostenible no debe ser sólo desde una perspectiva ambiental, sino también económica y social.

Concentrándonos en el concepto de movilidad sostenible, y en esa misma línea, Lizárraga (2006) entiende que es aquella que posibilita la satisfacción de las necesidades de las personas de modo eficiente y equitativo.

Como conclusión, teniendo en cuenta los objetivos que persiguen los problemas de transporte público colectivo, y tal y como se recoge en Acevedo y Bocarejo (2009), se entiende que los planes de movilidad no se deben limitar únicamente al desarrollo de sistemas que minimicen los tiempos y costes de desplazamiento de las personas, sino que también deben analizar su aportación al desarrollo social, al uso racional de bienes escasos, tales como la energía y el espacio urbano, y a los impactos sobre el medio ambiente. De aquí la importancia en desarrollar una investigación sobre la optimización de rutas de transporte del colectivo de personas de movilidad reducida desde un enfoque multiobjetivo.

6.2.- LAS PERSONAS DE MOVILIDAD REDUCIDA

Se puede definir como personas de movilidad reducida (PMR) aquellas que tienen permanente o temporalmente limitada la capacidad de moverse de manera completamente autónoma.

La definición de PMR puede variar según la normativa en la que se enmarque. Incluso dentro del marco legislativo europeo se encuentran diversas definiciones, similares todas ellas pero que presentan ligeros matices diferenciadores. A continuación se muestran tres ejemplos:

- En COM (1997) 276 de la Comisión Europea se define como viajeros de movilidad reducida a *“todos los viajeros que tengan dificultades especiales para utilizar el transporte público, en particular las personas de edad avanzada y los discapacitados. La movilidad reducida no implica necesariamente problemas de salud.”*
- En COM (2001) 217 aprobado por el Consejo Europeo se redefine como viajeros de movilidad reducida a *“todas las personas que tengan dificultades para utilizar el transporte colectivo, como, por ejemplo, las personas con discapacidad (incluidas las personas con deficiencias sensoriales y psíquicas y los usuarios de sillas de ruedas), las personas con discapacidades en las extremidades, las personas de baja estatura, las personas que lleven equipaje pesado, las personas de edad, las mujeres embarazadas, las personas con carritos de la compra y los padres que lleven niños (incluidos niños sentados en cochecitos).”*
- En CE (2006) del Parlamento y Consejo Europeo sobre los derechos de las personas con discapacidad o movilidad reducida en el transporte aéreo se describe como persona de movilidad reducida a *“toda persona cuya movilidad para utilizar el transporte se halle reducida por motivos de discapacidad física (sensorial o locomotriz, permanente o temporal), discapacidad o deficiencia intelectual, o cualquier otra causa de discapacidad, o por la edad, y cuya situación requiera una atención adecuada y la adaptación a sus necesidades particulares del servicio puesto a disposición de los demás pasajeros”*

Independientemente de las sutiles diferencias presentes en su definición legislativa, es patente la intención que, desde organizaciones internacionales como la UITP, se tiene de conseguir que los sistemas de transporte público ofrezcan soluciones seguras de transporte a las personas mayores o con movilidad reducida.

También en el ámbito europeo, en el dictamen CE (2011) del Comité Económico y Social Europeo (CESE), se hace referencia a las medidas complementarias a incluir en los planes de acción de movilidad urbana, entre las que se encuentra la mejora de la accesibilidad de las personas de movilidad reducida.

El CESE estimaba para 2011 que el número de ciudadanos europeos de movilidad reducida, incluyendo en dicho colectivo a personas de edad avanzada, personas inválidas o con discapacidad, así como mujeres embarazadas o personas con silla de niño, superaba ampliamente los 100 millones. Así mismo, reconocía que la accesibilidad de las personas de movilidad reducida se debía integrar en la estrategia comunitaria para las personas con discapacidad 2010-2020.

En el caso de España, se comienza a tomar conciencia de las necesidades de los distintos colectivos integrantes de la sociedad principalmente gracias al apoyo expresado en la Constitución Española de 1978, en sus artículos 9.2 y 14 en los que de manera pionera se establece la “*obligación de los poderes públicos de fomentar la igualdad y el desarrollo individual de la persona dentro de la esfera política, económica y social.*”

Pero tiene que transcurrir 25 años, año 2003, para que se apruebe la Ley 51/2003 de Igualdad de Oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad y cuando se desarrolla el Primer Plan Nacional de Accesibilidad 2004-2012. La Línea Estratégica 17 sobre la “Incorporación de la accesibilidad a los medios de transporte público” incluida en dicho plan, motivó el trabajo de Vega (2006), en el que se analiza la accesibilidad del transporte en autobús.

El Libro Verde de la Accesibilidad en España (2002) presentaba una estimación sobre el número de personas de movilidad reducida en el año de su publicación de casi 16 millones de personas, lo que suponía aproximadamente el 39,1% de la población. Se incluía dentro de PMR a tres grandes colectivos: las personas con discapacidades permanentes – representando un 8,8%, las de edad avanzada no discapacitadas – siendo un 10,9% y las afectadas por circunstancias transitorias que reducían su movilidad: accidentes, enfermedades, embarazos avanzados o actividades diversas, como compras, mudanzas, viajes, etc. – constituyendo un 19,5%. En dicho documento se presenta que la tendencia previsible es que el número de PMR aumente de forma paralela al envejecimiento demográfico que se da en la población española, lo que podría suponer que el número se incrementase en casi 500.000 personas para el año 2020.

Se concluye por tanto, que el colectivo de personas de movilidad reducida va en aumento y es una obligación de la sociedad el tomar conciencia de las dificultades que encuentran en su día a día para desarrollarse plenamente como personas y más concretamente, en relación con el tema del presente trabajo, en lo que a su transporte se refiere.

6.3.- LOS SERVICIOS ESPECÍFICOS DE TRANSPORTE PUERTA-A-PUERTA

El transporte de personas de movilidad reducida es un servicio cuya demanda está experimentado un gran crecimiento en los últimos tiempos y se espera, en gran parte debido a las expectativas en las cifras de población integrante de este colectivo en un futuro, que siga aumentando.

Debido a las necesidades específicas que deben ser ofertadas para una realización adecuada de este servicio, el transporte de personas de movilidad reducida presenta unas características idóneas para ser gestionado como un sistema de transporte público colectivo a la demanda.

Históricamente, fue la inaccesibilidad que tenían las personas con movilidad reducida a los transportes públicos colectivos existentes lo que hizo aparecer los primeros sistemas de transporte especial “puerta-a-puerta” en la década de los años 60. La mayor parte de los vehículos de transporte colectivo empleados en dicha época no estaban adaptados para ser usados de modo seguro por usuarios sin capacidad de moverse de manera completamente autónoma.

En la actualidad, y gracias a los avances tecnológicos, se dispone de vehículos que permiten el traslado simultáneo de varias personas teniendo en cuenta las necesidades específicas de cada una de ellas. Es el caso, por ejemplo, de los “Paratransit”, muy extendidos en Estados Unidos, que son minibuses con plazas habilitadas para el transporte de más de un usuario en silla de ruedas.

Por todo lo anterior, para el transporte de personas de movilidad reducida se requieren traslados puerta-a-puerta en vehículos adaptados, donde cada usuario especifica el lugar donde debe ser recogido, el lugar a donde necesita desplazarse y el horario requerido para el traslado.

El diseño de rutas de transporte satisfaciendo las necesidades de los usuarios con estas características específicas se puede tratar empleando el, anteriormente descrito, “Dial-A-Ride Problem”, DARP por sus siglas en inglés. Se considera entonces como una auténtica necesidad, poder ser capaces de desarrollar métodos efectivos para resolver DARPs, en donde el objetivo principal al diseñar las rutas no sólo consista en minimizar costes, sino también traten de minimizar los inconvenientes generados en los usuarios.

6.4.- LOS MODELOS PLANTEADOS Y LAS TÉCNICAS APLICADAS

En lo que respecta a tratados específicos sobre problemas de transporte de personas de movilidad reducida la bibliografía existente es más bien escasa. En concreto, hay que remontarse al trabajo de Ioachim et al. (1995) para encontrar referencias de trabajos en revistas destacadas que citen este tipo de problema de forma explícita. Posteriormente, en el estudio presentado en Toth y Vigo (1997) se encuentra la primera denominación del problema de transporte de las personas discapacitados por sus siglas en inglés, HTP Handicapped Transportation Problem.

Tal y como se ha descrito anteriormente, muchos de los problemas de transporte de personas de movilidad reducida se ajustan al modelo del Dial-A-Ride Problem y a variantes del mismo. En este caso, sobre este problema, sí existen referencias actuales que merecen ser mencionadas por su interés y relevancia para esta investigación.

Aunque se desarrolla con más detalle en los capítulos que siguen, se anticipa que en este trabajo investigador se estudia un DARP estático, en el que todas las solicitudes de traslados se conocen con anticipación y se dispone, en el depósito, de una flota fija de vehículos para poder realizar los traslados. El problema se aborda con un enfoque de optimización multiobjetivo, en el que se tratan de encontrar buenos frentes de Pareto minimizando dos objetivos a la vez.

La mayor parte de los estudios encontrados que versan sobre el DARP se basan en la optimización de una función que contiene varios objetivos, que finalmente resuelven con una única función objetivo en la que se optimiza una suma ponderada. Es menor el número de ejemplos donde se resuelve un problema multiobjetivo buscando soluciones eficientes asociadas al frente de Pareto para optimizar varios objetivos de manera simultánea.

Una contribución que ha sido de gran utilidad se encuentra en el trabajo de Cordeau (2006) donde se formula el modelo matemático del DARP con condiciones de precedencia, capacidad, ventanas de tiempo, tiempo de viaje; además se diseña un algoritmo de ramificación y corte para resolver este modelo en instancias aleatorias. En Mauri y Nogueira Lorena (2006) se presenta un modelo multiobjetivo y una aplicación de Recocido Simulado para resolverlo, trabajan una suma ponderada que contiene distancia total recorrida, número de vehículos, duración de las rutas, tiempo de viaje de los usuarios y tiempo de espera.

El DARP con variantes mutiobjetivos es analizado en el trabajo de Parragh et al. (2009) y en Zidi et al. (2012). El mismo tipo de problema en sus variantes con funciones estocásticas se presenta en Ho y Haugland (2011) y en la publicación de Schilde et al. (2011). En Chevrier et. al. (2012) se muestra un método evolutivo híbrido para un problema con tres objetivos: minimizar número de vehículos, duración total de los viajes y el tiempo de espera; comparan el desempeño de su algoritmo con otros métodos evolutivos para problemas multiobjetivo. En Muelas et al. (2013) se estudia un algoritmo distribuido basado en búsquedas por entornos variables para problemas DARP a gran escala, minimizando el costo total de las rutas. En el trabajo de Ritzinger et al. (2016) se aborda el problema del DARP minimizando la distancia total recorrida con un algoritmo híbrido que combina programación dinámica con Large Neighborhood Search. En Chassaing et al. (2016) se resuelve el DARP con un algoritmo de búsqueda local evolutiva, consideran la minimización de la distancia total y obtienen resultados competitivos.

Algunas variantes del problema de Dial-A-Ride en las que se integran estudios de casos reales están presentes en los trabajos de Paquette et al. (2012), Parragh et al. (2012) y en el posterior estudio de Parragh y Schmid (2013).

Por último citar, por similitud por el tema tratado, otro trabajo basado en el modelo del DARP que extiende los objetivos clásicos al considerar, la consistencia del conductor como una componente para la evaluación de la calidad del servicio de transporte, presentado en Braekers y Kovacs (2016).

Debido a la escasa literatura encontrada que resuelva problemas de transporte de PMR, por similitud con el tema investigado, se referencian a continuación algunas de las metodologías de resolución empleadas recientemente en problemas específicos de transporte de pacientes en centros hospitalarios. En Zhang et al. (2015) presentan un modelo DARP con multi-viaje para el caso real del transporte de pacientes en Hong Kong que resuelven empleando un algoritmo memético. Y por último, en el trabajo de Molenbruch et al. (2017) se presenta una búsqueda local multi-direccional para un problema de dos objetivos, en el que se minimizan simultáneamente el tiempo total de viaje de todos los usuarios y la distancia total recorrida; se presenta un caso real y se obtienen frentes de Pareto.

7.- EL PLANTEAMIENTO DEL PROBLEMA

7.- EL PLANTEAMIENTO DEL PROBLEMA

7.1.- MOTIVACIÓN

En este trabajo se desarrolla una herramienta metodológica para resolver el problema de diseño de rutas para trasladar a personas de movilidad reducida, personas de avanzada edad o con algún tipo de discapacidad que impida su completa independencia a la hora de trasladarse.

La demanda en el transporte de personas de movilidad reducida está experimentando un gran crecimiento en los últimos tiempos y se espera, debido a las expectativas en las cifras de población integrante de este colectivo, que siga aumentando en un futuro. Se demuestra, entonces, la necesidad de planificar y poner en marcha servicios eficientes de transporte colectivo a disposición de este tipo de usuarios.

Asimismo, dentro de la Encuesta de Movilidad de las Personas Residentes (MOVILIA 2006/2007), en la que se recoge información sobre las pautas de movilidad de la población residente en España, se han encontrado estadísticas que alertan sobre la necesidad de mejorar el transporte de las personas mayores. En concreto, se indica que los residentes españoles mayores de 64 años representan un 36% sobre el total de personas encuestadas con motivos para no desplazarse en un día laborable medio. Analizando los motivos revelados para la no realización del desplazamiento se ha encontrado que el 35% se debe a incapacidad, impedimento o enfermedad temporal. Todo esto pone de manifiesto la existencia de una demanda que tiene que ser satisfecha y justifica que, debido a las circunstancias del colectivo que hará uso del transporte, ésta se satisfaga ofertando un servicio de calidad lo más óptimo posible.

La idea del problema surge al concentrar la atención en aquellos traslados que el colectivo de personas de movilidad reducida realiza a centros de tratamiento, centros de día e incluso instalaciones hospitalarias y educativas. Se busca, por tanto, optimizar los viajes diarios que, las personas pertenecientes a dicho colectivo, necesitan realizar por motivo salud o educación, asistidos en el marco de un servicio de transporte público colectivo.

Se está, pues, estudiando escenarios en que, desde un punto de vista geográfico, se transportan personas desde diferentes orígenes a distintos destinos. Pero es importante resaltar que, todos estos puntos, por el carácter social que tiene este tipo de servicio de transporte, se van a localizar en ámbitos locales, provinciales o, a lo sumo, dentro del contexto de una misma Comunidad Autónoma.

La motivación de los trayectos para los que se oferta el transporte, hace que los destinos sean sedes, instituciones o centros que dan servicio a una comunidad de personas concreta. Es por ello, que se estudia el problema en un medio en el que la ordenación del territorio asume una distribución geográfica con unas isócronas de acceso que permiten dar servicios completos a dicha comunidad. Entonces, en el planteamiento del problema se tratan situaciones en las que los tiempos de traslado directo entre el origen y el destino de las personas pueden ser diferentes, presentando una ligera heterogeneidad, pero no van a alcanzar en ningún caso valores muy elevados, ya que el estudio se plantea en contextos con una relativamente buena y equitativa distribución de dotaciones y servicios sociales, sanitarios y educativos.

El problema que se aborda en este trabajo persigue el diseño de un sistema de transporte colectivo puerta a puerta para el traslado de PMR mediante la optimización y planificación de las rutas del servicio. Durante el estudio del problema, se deben tener en cuenta, para cada usuario, el lugar en el que debe ser recogido, el lugar a donde necesita desplazarse, el horario requerido para el traslado, así como, las características específicas que debe presentar el vehículo para adaptarse a las necesidades concretas del usuario.

El diseño de las rutas de vehículos, satisfaciendo todas las particularidades mencionadas, se ha propuesto a través de un DARP estático sin ventanas de tiempo, en el que todas las solicitudes de traslados se conocen con anticipación, disponiendo en el depósito de una flota fija de vehículos adaptados para su asignación a las diferentes rutas.

Con el fin de planificar eficientemente las rutas, no sólo se va a buscar la minimización de los costes, sino también el minimizar los inconvenientes de los usuarios. Por consiguiente, el primer objetivo del problema es de **carácter económico**, en el que se busca minimizar el total de la distancia recorrida, y se plantea un segundo objetivo de **carácter social**, por el que se quiere mejorar la calidad minimizando el máximo tiempo excedente de viaje que tiene un usuario, medido como la diferencia entre el tiempo de viaje al compartir el vehículo con otros usuarios y el tiempo al realizar el recorrido de forma directa, es decir, sin paradas intermedias.

El criterio de calidad que se pretende dar en el servicio va a tener en cuenta, entre otros factores, la duración y longitud de la ruta, el tiempo de espera de los usuarios, el tiempo total de viaje de los usuarios y la diferencia entre el tiempo actual de llegada y el tiempo deseado de llegada. Algunos de estos criterios se van a tratar como restricciones o como parte de las funciones objetivo del problema.

La mayoría de los trabajos científicos, presentados hasta el momento, que tratan el DARP con varios objetivos, los abordan como sumas ponderadas entre los objetivos o consideran alguno de los objetivos como una restricción en el modelo. Son pocos los trabajos que abordan el problema del DARP bajo el enfoque de optimización multiobjetivo y, al menos desde nuestro conocimiento, no existe ninguna referencia en la literatura que verse sobre el DARP con estos dos objetivos concretos. Al plantearse un modelo biobjetivo no va a existir una solución óptima única, sino que, se persigue encontrar el conjunto de soluciones eficientes, frentes de Pareto, o, al menos, una aproximación a dicho conjunto, que sirva de apoyo para la adecuada toma de decisiones respecto al traslado de los usuarios.

Es pues, objeto de esta tesis, **la optimización de rutas de transporte de personas de movilidad reducida a fin de minimizar el coste total del servicio y mejorar la calidad del mismo desde un enfoque multiobjetivo.**

A través de la estrategia empleada en la resolución del problema, se obtienen buenos frentes de Pareto que van a proporcionar diversas soluciones al tomador de decisiones. Así, según la situación existente en el momento de la toma de decisión, se puede optar y adoptar la más adecuada.

7.2.- NOTACIÓN Y DESCRIPCIÓN DE OBJETIVOS

Se presenta, a continuación, la notación que se va a emplear en esta investigación, a fin de ilustrar con mayor facilidad el trabajo propuesto.

Sea;

n : número de solicitudes.

q_i : cantidad de personas en la solicitud i -ésima, $i = 1, \dots, n$.

V : conjunto de localizaciones del problema, $V = \{0, 1, \dots, n, n + 1, \dots, 2n\}$.

El origen (y final) de cada ruta viene representado por 0. El subconjunto $P = \{1, \dots, n\}$ representa el conjunto de puntos de recogida y el subconjunto $D = \{n + 1, \dots, 2n\}$ representa el conjunto de puntos de destino. Así para la solicitud i -ésima el punto de recogida es i , y el de destino $i + n$. Para simplificar, se puede identificar a P con el conjunto de solicitudes. En algunas ocasiones, también para simplificar, para cada solicitud $i \in P$ el punto de recogida se denota por i^+ y el de destino i^- .

m : número de vehículos.

Q : capacidad de cada vehículo.

$Lmax$: tiempo máximo entre la primera recogida y la última descarga para cada ruta.

t_{ij}, d_{ij} : matrices de tiempo y distancia, respectivamente, entre cada par de puntos $i, j, \forall i, j \in V$; se hace notar que en los tiempos t_{ij} ya están incluidos los tiempos de carga en el punto i y de descarga en el punto j .

El problema que se plantea consiste en el **diseño de m rutas factibles**, esto implica que respeten la capacidad máxima del vehículo Q y el tiempo máximo de cada ruta $Lmax$, **que den servicio a todas las solicitudes**, de forma que los usuarios de una solicitud viajen en el mismo vehículo, **y que optimicen** los siguientes objetivos:

- minimizar la máxima diferencia, en el conjunto de solicitudes, entre el tiempo en el vehículo de los usuarios de la solicitud i y t_{i+i^-} (**objetivo social**)
- minimizar la distancia total de todas las rutas (**objetivo económico**)

Sea S una solución el objetivo social se puede definir más formalmente de la forma siguiente:

$$\max \{\Delta_i(S) : i \in P\}$$

Donde $\Delta_i(S) = T_i(S) - t_{i+i^-}$, y $T_i(S)$ es el tiempo de viaje de la solicitud i en la solución S .

Obsérvese que para cada solicitud $i \in P$, el tiempo t_{i+i^-} representa el “tiempo ideal”, es decir, el tiempo de ir directamente del punto de recogida i^+ al punto de destino i^- .

El incremento sobre el “tiempo ideal” (valor de Δ_i) puede ser entendido como una medida de “no-calidad” para la petición i . Por otra parte, el hecho de minimizar el máximo de los valores Δ_i , en lugar de la media, intenta evitar el riesgo de que existan algunos clientes no satisfechos con el servicio aunque el resto sí estén satisfechos y la media de los incrementos sea buena. Con el fin de ilustrar dicha circunstancia, seguidamente, se describe un ejemplo explicativo.

Se plantea un problema con dos solicitudes A y B , donde se dispone de un vehículo con capacidad suficiente para cargar a los usuarios de ambas solicitudes a la vez. Los correspondientes puntos de carga y descarga se pueden localizar en el espacio en las coordenadas siguientes:

- Para la solicitud A : el punto A^+ $(0,0)$ y el punto A^- $(2,0)$
- Para la solicitud B : el punto B^+ $(1,2)$ y el punto B^- $(3,1)$

Se considera, finalmente, que los tiempos de viaje entre los diferentes puntos coinciden con las distancias euclídeas.

A continuación se representan gráficamente dos posibles soluciones, S y S' , que se ilustran en la Figura - 9 y Figura - 10, respectivamente.

Figura - 9: Solución S : $A^+ - B^+ - B^- - A^-$

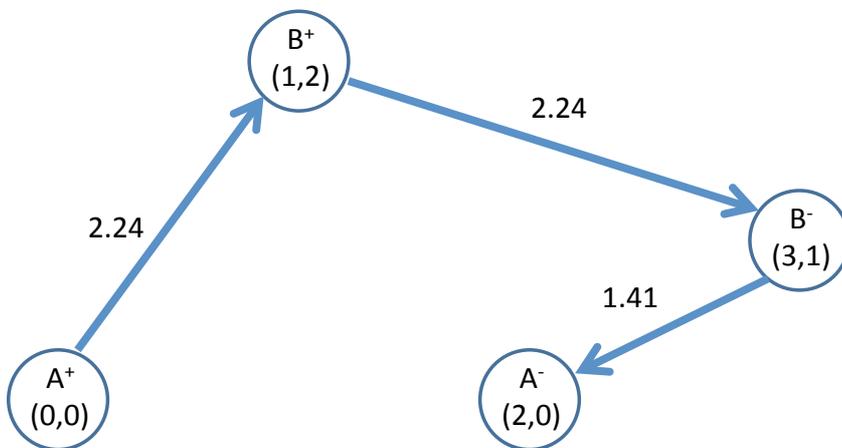
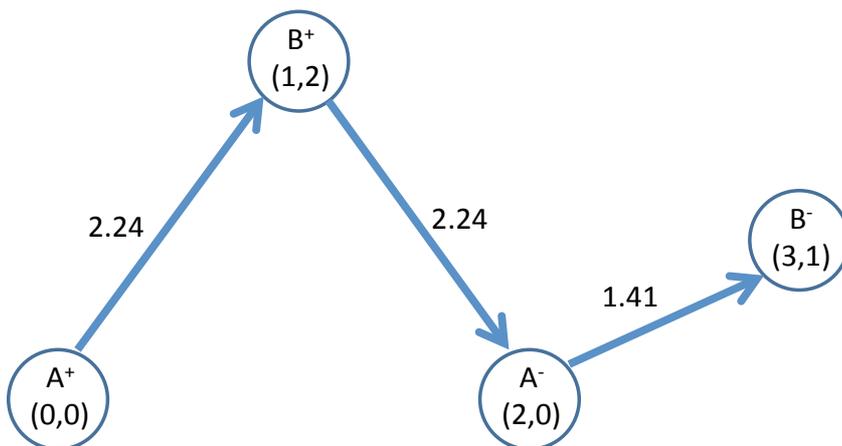


Figura - 10: Solución S' : $A^+ - B^+ - A^- - B^-$



La primera solución mostrada, S , consiste en la ruta con secuencia de visita $A^+ - B^+ - B^- - A^-$. Se observa que en este caso $\Delta_A(S) = 3,89$ y $\Delta_B(S) = 0$. La media de ambos valores es de 1,94 y el máximo es 3,89. Se puede apreciar como los usuarios de la solicitud A tienen que dar un gran rodeo para

viajar entre sus respectivos puntos de carga y descarga, mientras que los usuarios de la solicitud B viajan de forma directa. Es decir, se manifiesta una gran descompensación entre las calidades de los servicios ofertados a ambas solicitudes.

Por otra parte, la segunda solución considerada, S' , consiste en la ruta con secuencia de visita $A^+ - B^+ - A^- - B^-$. En este caso $\Delta_A(S) = 2,48$ y $\Delta_B(S) = 1,41$, siendo la media de ambos valores de 1,94 y el máximo de 2,48. Se observa que los usuarios de ambas solicitudes no viajan de forma directa, sino que tienen que realizar un cierto rodeo al trasladarse entre sus respectivos puntos de carga y descarga. El incremento sobre el tiempo ideal es algo mayor en la solicitud A. Comparando ambas soluciones, queda patente que en la solución S' existe un mayor equilibrio entre las calidades de los servicios ofertados a las dos solicitudes.

En el ejemplo ilustrado, ambas soluciones presentan los mismos valores medios para Δ_i , sin embargo, el valor máximo de estos Δ_i es menor en S' que en S . Por todo lo anterior, parece claro que el minimizar el máximo valor de los Δ_i puede ser una forma más adecuada de conseguir un buen nivel de servicio y que, a su vez, este nivel de servicio esté equilibrado entre los diferentes usuarios.

Prosiguiendo con la descripción de los objetivos, se denota con f_1 y f_2 las funciones objetivo, social y económico respectivamente, a minimizar. Así, dada una solución S :

Objetivo social: $f_1(S) = \max \{\Delta_i : i \in P\}$

Objetivo económico: $f_2(S) = \text{distancia total recorrida por las } m \text{ rutas en la solución } S$

Cabe destacar que ambas funciones objetivo presentan un comportamiento confrontado; la minimización del máximo del incremento sobre el “*tiempo ideal*”, implica un aumento de la distancia total recorrida por las m rutas y por extensión del coste de prestación del servicio, por lo que se intenta buscar soluciones que equilibren ambos aspectos.

Se plantea aquí, y como ya se ha adelantado, un problema biobjetivo, en el que existen dos criterios a optimizar y donde se requiere de un análisis para determinar qué alternativa o alternativas son las más idóneas. Ello implica la obtención de un conjunto eficiente de puntos que recoja aquellas soluciones no dominadas, es decir, que no existan otras soluciones que sean mejores que éstas, en ninguno de los objetivos que se deben optimizar.

Por consiguiente, se trata de un problema de optimización multiobjetivo donde las soluciones no siempre se pueden comparar. En esta tipología de problemas, sean dos soluciones S y S' , se dice que la solución S domina a la solución S' , si:

- a) $f_1(S) \leq f_1(S')$,
- b) $f_2(S) \leq f_2(S')$ y
- c) o bien $f_1(S) < f_1(S')$ o bien $f_2(S) < f_2(S')$

Una solución se considera eficiente si no existe otra que la domine. El problema que se plantea en este trabajo trata de encontrar un conjunto de soluciones lo más cercanas posible al conjunto de soluciones eficientes del problema y que se conoce como “curva de eficiencia”.

7.3.- FORMULACIÓN MATEMÁTICA DEL MODELO

Con el fin de formular este problema se plantea un modelo de tres índices, adaptando el propuesto por Cordeau (2006) en su trabajo sobre el DARP.

Para ello se definen:

$$q_0 = 0 \text{ y } q_i = -q_{i-n} \text{ para } i = n + 1, \dots, 2n$$

Así mismo se definen las siguientes variables de decisión binarias:

$$x_{ij}^k : \text{ es igual a 1 si y sólo si el vértice } j \text{ es visitado justo después del vértice } i \text{ por el vehículo } k, \\ k = 1, \dots, m$$

Además se definen las siguientes variables auxiliares reales;

$$u_i^k : \text{ se interpreta como el tiempo en el que el vehículo } k \text{ sale del vértice } i,$$

$$w_i^k : \text{ la carga del vehículo } k \text{ al partir del vértice } i,$$

$$r_i^k : \text{ el tiempo de viaje del usuario } i \text{ en el vehículo } k, \text{ y}$$

$$E : \text{ valor de la función objetivo social.}$$

Finalmente, y teniendo en cuenta todo lo anteriormente descrito, la formulación del modelo se expresa de la manera siguiente:

$$\min \quad E \quad (1)$$

$$\min \quad \sum_{k=1}^m \sum_{i \in V} \sum_{j \in V} d_{ij} \cdot x_{ij}^k \quad (2)$$

Sujeto a

$$r_i^k - t_{i,n+i} \leq E \quad i \in P, k = 1, \dots, m \quad (3)$$

$$\sum_{k=1}^m \sum_{\substack{j \in V \\ j \neq i}} x_{ij}^k = 1 \quad i \in P \quad (4)$$

$$\sum_{\substack{j \in V \\ j \neq 0}} x_{0j}^k \leq 1 \quad k = 1, \dots, m \quad (5)$$

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ij}^k - \sum_{\substack{j \in V \\ j \neq i+n}} x_{n+i,j}^k = 0 \quad i \in P, k = 1, \dots, m \quad (6)$$

$$\sum_{\substack{j \in V \\ j \neq i}} x_{ji}^k - \sum_{\substack{j \in V \\ j \neq i}} x_{ij}^k = 0 \quad i \in V, k = 1, \dots, m \quad (7)$$

$$u_j^k \geq (u_i^k + t_{ij}) - (1 - x_{ij}^k) \cdot M \quad i, j \in V, j \neq 0, j \neq i, \quad k = 1, \dots, m \quad (8)$$

$$w_j^k \geq (w_i^k + q_j) - (1 - x_{ij}^k) \cdot Q \quad i, j \in V, j \neq 0, j \neq i, \quad k = 1, \dots, m \quad (9)$$

$$u_{n+i}^k - u_i^k \geq 0 \quad i \in P, k = 1, \dots, m \quad (10)$$

$$r_i^k \geq u_{n+i}^k - u_i^k \quad i \in P, k = 1, \dots, m \quad (11)$$

$$\sum_{i \in V - \{0\}} \sum_{\substack{j \in V - \{0\} \\ j \neq i}} t_{ij} \cdot x_{ij}^k \leq Lmax \quad k = 1, \dots, m \quad (12)$$

$$\max\{0, q_i\} \leq w_i^k \leq \min\{Q, Q + q_i\} \quad i \in V, k = 1, \dots, m \quad (13)$$

$$x_{ij}^k \in \{0, 1\} \quad i, j \in V, j \neq i, k = 1, \dots, m \quad (14)$$

En esta formulación, (1) y (2) son respectivamente los objetivos social y económico. Las restricciones (3) y (11) se plantean para dar linealidad a la función objetivo social. La restricción (4) asegura que cada solicitud sea atendida exactamente por un vehículo. La restricción (5) cerciora que no se usen más de m vehículos. La restricción (6) asegura que para cada solicitud los correspondientes puntos de carga y descarga son visitados por el mismo vehículo o ruta. La restricción (7) garantiza que a un punto llegue un arco en una ruta si y sólo si sale de él exactamente un arco. La restricción (8) sirve para evitar ciclos y junto con la restricción (10) asegura que para cada solicitud el punto de recogida se visite antes que el punto de llegada. M es una constante lo suficientemente grande, puede igualarse a $2 \cdot Lmax$, para que la restricción se cumpla siempre que $x_{ij}^k = 0$. Las restricciones (9) y (13) sirven para respetar la capacidad máxima de los vehículos. La restricción (12) garantiza que el tiempo entre la primera recogida y la última descarga no supere la $Lmax$ de cada ruta. Por último, la restricción (14) define la naturaleza binaria de las variables.

Más adelante, en el Capítulo 10 posterior, se muestran las pruebas computacionales realizadas en las que se resuelve este modelo de forma exacta con software comercial (Cplex). En concreto, se analiza y determina cuál es el tamaño de instancias máximo que se puede resolver con dicho software de manera razonable.

7.4.- PROCEDIMIENTOS DE SOLUCIÓN

Tal y como se ha descrito anteriormente, los problemas de rutas, incluido el DARP entre ellos, son *NP-Hard* ‘problema difícil no determinista en tiempo polinómico’. En consecuencia, el modelo planteado en esta investigación, al tratarse de una variante biobjetivo del DARP, también será *NP-Hard*.

Esto significa, como se manifiesta en las pruebas computacionales recogidas en el Capítulo 10, que el modelo sólo se puede resolver de forma exacta si los problemas son de tamaño moderado o pequeño. Para resolver instancias de mayor tamaño se hace conveniente el uso de métodos de solución heurísticos. Concretamente, para la resolución heurística del problema se van a proponer los dos métodos siguientes:

- Un método basado en la estrategia MOAMP y
- uno basado en la adaptación del método ε -constraint (ε -restricción) para el problema biobjetivo. Se denomina Método Heurístico_ ε -constraint (MHeC).

Como base de ambas metodologías de resolución se proponen los dos procedimientos denominados:

- **Constructive** – procedimiento constructivo
- **TabuSearch** – procedimiento de búsqueda tabú

Con el fin de describir ambos procedimientos se requiere definir primeramente la siguiente serie de conceptos:

- Se dice que una solución S es completa si todas las solicitudes están programadas: es decir, si para cada solicitud i el par de puntos i^+ e i^- están asignados en alguna ruta; en caso contrario se dice que la solución es parcial.
- Una solución S viene representada por un conjunto de m rutas, una por vehículo. Es decir, $S = \{ R_1, R_2, \dots, R_m \}$.
- A su vez cada ruta R_k se representa por una secuencia de puntos que empiezan y acaban en el origen 0. Por ejemplo, si la ruta R_k es de la forma $R_k = 0 - x - y - z - t - 0$, entonces, dicha ruta desde el origen 0 visita secuencialmente los puntos x , y , z y t para finalmente regresar al origen 0.
- Una ruta vacía se representa como $0 - 0$.
- Tanto en el planteamiento como en la formulación del problema ni los valores de t_{i0} ni los de $t_{0i}, \forall i \in V, i \neq 0$, influyen en los objetivos ni en las restricciones del problema. Por tanto, se pueden redefinir estos valores como $t_{0i} = t_{i0} = 0$ sin que alteren el problema. De esta forma, la restricción que obliga a que el tiempo máximo entre la primera recogida y la

última descarga para cada ruta no supere $Lmax$ (la restricción (9) de la formulación), equivale a que el tiempo total de cada ruta no debe ser superior a $Lmax$. Por tanto, en el diseño de las heurísticas se redefinen $t_{0i} = t_{i0} = 0$, y se emplea esta restricción equivalente - el tiempo total de cada ruta no debe ser superior a $Lmax$ - en lugar de la original, ya que se considera una restricción más fácil de chequear.

7.4.1.- Procedimiento Constructivo

La forma detallada en que el procedimiento **Constructive** trabaja es la siguiente:

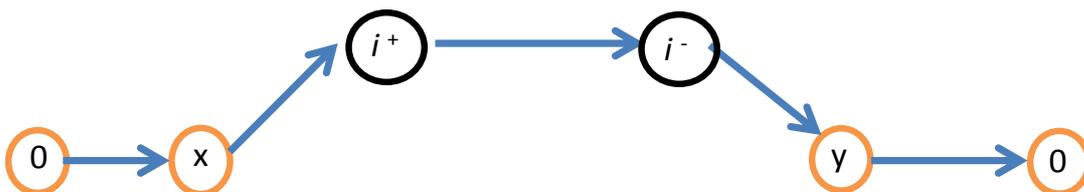
Inicialmente las solicitudes se ordenan de forma decreciente según los valores de $d_{0i^+} + d_{i^+i^-} + d_{i^-0}$ y se inicializan las rutas, esto consiste en definir las m rutas vacías. En cada iteración se inserta cada uno de las solicitudes i siguiendo el orden anterior, es decir, los pares de puntos i^+ e i^- . Para ello se determina el conjunto L de todas posibles inserciones factibles, o sea las que deben cumplir restricciones del modelo. Cada inserción factible, ins , es evaluada según el valor de la función objetivo que estemos considerando, f_1 o f_2 en cada caso.

En las figuras que siguen se ilustran las dos posibilidades que existen para realizar inserciones de una solicitud i en una ruta. En la primera, Figura - 11, los puntos i^+ e i^- se insertan con al menos un punto intermedio, y en la segunda, Figura - 12, los puntos i^+ e i^- se insertan de forma consecutiva.

Figura - 11: Inserción de i^+ e i^- con algún punto intercalado



Figura - 12: Inserción de i^+ e i^- de forma consecutiva



Sea g la función objetivo ($g = f_1, f_2$). La formulación en pseudocódigo del procedimiento **Constructive** se describe a continuación.

Pseudocódigo - 1: Procedimiento Constructive

Procedimiento **Constructive** (g ; $var\ factible$; $var\ S$)

1. Ordenar las solicitudes según los valores de $d_{0i^+} + d_{i^+i^-} + d_{i^-0}$ en orden decreciente (para simplificar considerar la ordenación natural, i.e., $1, 2, \dots, n$)
2. Hacer $R_t = 0 - 0, \forall l = 1, \dots, m; i = 0$ y $factible = TRUE$

Repetir

3. Hacer $i = i + 1$
4. Construir el conjunto L de todas las inserciones factibles de la solicitud i
5. Si $L = \emptyset$ entonces: hacer $factible = FALSE$ y finalizar (EXIT)
6. Calcular $\Delta g(ins) =$ la variación del valor de la función objetivo g si se ejecuta la inserción $ins, \forall ins \in L$
7. Determinar $ins^* = argmin \{\Delta g(ins): ins \in L\}$
8. Ejecutar ins^*

Hasta $i = n$

9. Denotar por S la solución obtenida
-

7.4.2.- Procedimiento de Búsqueda Tabú

Con el fin de explicar el procedimiento **TabuSearch**, es necesario volver a mencionar que la Búsqueda Tabú es una estrategia dada a conocer en los trabajos de Glover (1989) y (1990), que está teniendo grandes éxitos y mucha aceptación en las últimas décadas en el campo de la investigación operativa. Según su creador, es un procedimiento que "*explora el espacio de soluciones más allá del óptimo local*", permitiendo movimientos hacia arriba o que empeoran la solución una vez que se llega a un óptimo local. Simultáneamente, los últimos movimientos realizados se califican como *tabú* durante las siguientes iteraciones para evitar volver a soluciones anteriores y que el algoritmo cicle. Amplios tutoriales sobre Búsqueda Tabú que incluyen todo tipo de aplicaciones pueden encontrarse en Glover y Laguna (1997).

Sea g la función objetivo. La formulación en pseudocódigo del procedimiento **TabuSearch** usado se describe a continuación.

Pseudocódigo - 2: Procedimiento TabuSearch

Procedimiento **TabuSearch** ($g; var S$)

1. Hacer $S^* = S, niter = 0, iterbest = 0,$
2. Iniciar *matriztabu*: $matriztabu(i, j) = -tenure, \forall i, j \in V$

Repetir

3. $niter = niter + 1$
4. Buscar S' la mejor solución en $N(S)$ con respecto a la función g tal que S' o no es “tabú” o mejora S^*
5. Hacer $S = S'$
6. Actualizar *matriztabu*
7. Si $g(S) < g(S^*)$ hacer $S^* = S$ e $iterbest = niter$

Hasta $niter > maxitertabu + iterbest$

8. Hacer $S = S^*$

Seguidamente, para la mejor comprensión del pseudocódigo anterior, se describen algunos conceptos importantes.

La función objetivo genérica g puede ser $g = f_1, f_2$ o F_λ , donde F_λ es una función mixta de las funciones f_1 y f_2 que se definirá más adelante.

En la variable auxiliar S^* se guarda la mejor solución encontrada durante la búsqueda, $niter$ es un contador de iteraciones y $iterbest$ la iteración donde se ha encontrado a S^* .

El conjunto $N(S)$ se define como el conjunto de *soluciones vecinas* factibles de S_0 . Una solución vecina es aquella a la que se llega por un “movimiento” o cambio sencillo (o relativamente sencillo). Esencialmente, estos movimientos consisten en cambios e intercambios de cadenas de puntos. Concretamente se consideran cuatro tipos:

- Dos cambios “entre-rutas” : El primero de los dos cambios “entre-rutas” que se usan en este trabajo consiste en intercambios de dos cadenas de rutas diferentes, también denominados CROSS intercambios - Taillard et al. (1997) - ; y el segundo tipo es el cambio de una cadena de una ruta a otra diferente, denominado MOVE.
- Dos intercambios “intra-rutas” (dentro de una misma ruta): En este trabajo dentro de los intercambios “intra-rutas” se consideran los intercambios tipo Or que consisten en intercambiar cadenas consecutivas - Or (1976) - y los intercambios Or-Generalizados, que intercambian cadenas no consecutivas.

La Figura - 13, Figura - 14, Figura - 15 y Figura - 16 ilustran respectivamente los cuatro tipos de movimientos: CROSS, MOVE, Or, y Or-Generalizado.

Figura - 13: Intercambio de dos cadenas de rutas diferentes (CROSS)

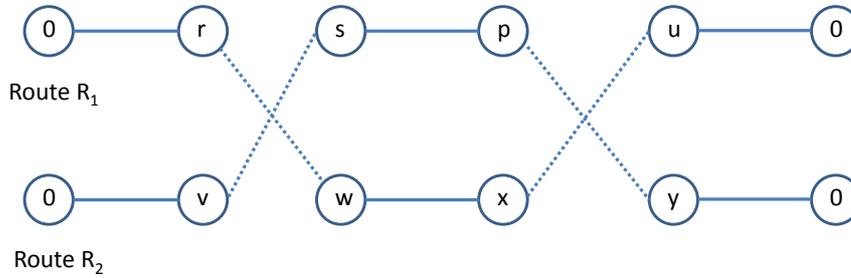


Figura - 14: Cambio de una cadenas de una ruta a otra (MOVE)

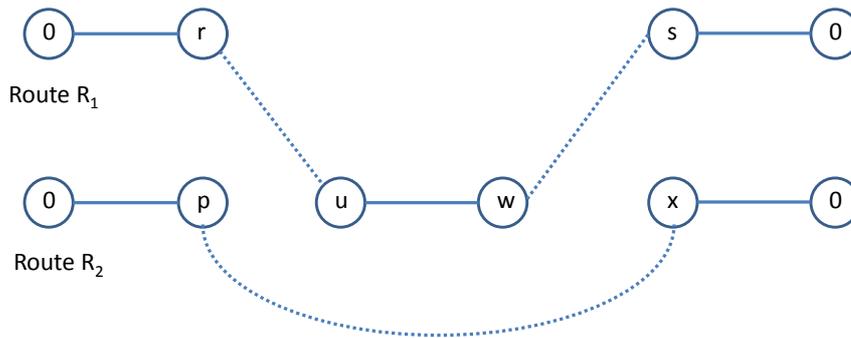


Figura - 15: Intercambio entre dos cadenas consecutivas de una misma ruta (Or)

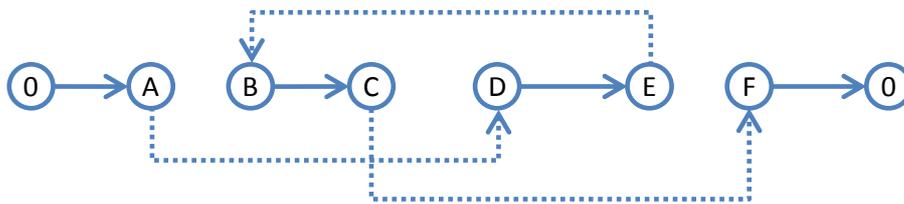
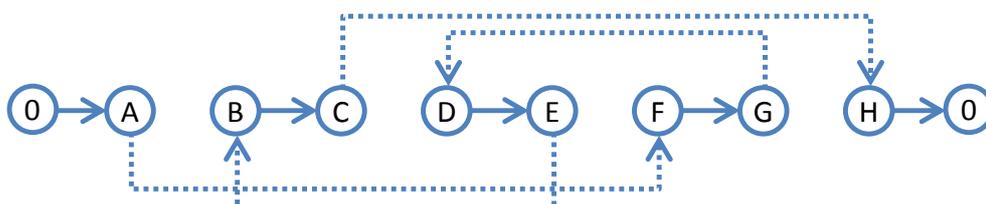


Figura - 16: Intercambio entre dos cadenas no consecutivas de una misma ruta (Or-Generalizado)



Se hace notar que todos los tipos de movimientos enunciados implican la eliminación de un conjunto de arcos y la incorporación de arcos nuevos. Con el propósito de que el algoritmo no cicle, se impide que los arcos que han sido eliminados recientemente de la solución actual S_0 vuelvan a ella. Para lo cual se hace uso de *matriztabu* que se define como:

$matriztabu(i, j) =$ iteración en que el arco (i, j) ha sido eliminado; $\forall i, j \in V; i \neq j$

Un arco (i, j) es tabú si: $niter \leq matriztabu(i, j) + tenure$, donde *tenure* es un parámetro que indica durante cuántas iteraciones un arco es tabú.

Un movimiento, y por tanto la correspondiente solución en $N(S)$, es tabú si supone la incorporación de al menos un arco tabú. Un movimiento tabú puede ser admitido si da lugar a una solución mejor que S^* , que se conoce en la terminología científica como ‘criterio de aspiración’.

El procedimiento finaliza cuando trascurren *maxitertabu* iteraciones sin producirse mejora.

8.- LAS ESTRATEGIAS DE ACELERACIÓN

8.- LAS ESTRATEGIAS DE ACELERACIÓN

8.1.- INTRODUCCIÓN

En el presente trabajo de investigación se han incorporado estrategias adaptadas tanto para realizar chequeos de la factibilidad como para el cálculo de las funciones objetivo del problema. El empleo de dichas estrategias tiene por objeto acelerar de manera significativa los procedimientos **Constructive** y **TabuSearch**. En los párrafos siguientes se explican detalladamente dichas estrategias.

8.2.- EL CHEQUEO DE LA FACTIBILIDAD

Tal y como se ha señalado anteriormente, al describir los dos procedimientos que se emplean en el problema, se debe chequear la factibilidad de las inserciones en cada iteración del procedimiento **Constructive** y la factibilidad de los movimientos en cada iteración del procedimiento **TabuSearch**. Más concretamente hay que comprobar que se cumplan:

- a) La restricción de capacidad máxima de los vehículos;
- b) La restricción de tiempo máximo de duración de cada ruta;
- c) Los puntos de carga (i^+) y descarga (i^-) de una solicitud deben estar en la misma ruta;
- d) Las restricciones de precedencia (i^+ antes que i^-).

Es importante indicar en qué procedimiento y, de manera específica, en qué casos dentro de él se debe o no realizar cada chequeo. Para ello se presenta la siguiente lista:

- Las restricciones a) y b) deben ser chequeadas para las inserciones del procedimiento **Constructive** y todos los movimientos del procedimiento **TabuSearch**.
- Según se han diseñado las inserciones del procedimiento **Constructive** ya cumplen con las restricciones c) y d). Esto puede verse tanto en la Figura - 11 como en la Figura - 12.
- Los movimientos “entre-rutas” del procedimiento **TabuSearch** (CROSS y MOVE) pueden separar los puntos de carga y descarga (i^+ e i^- respectivamente) de una misma solicitud. Consecuentemente hay que chequear para estos movimientos que cumplan la restricción c). Más en concreto, se debe asegurar que las cadenas que cambian de ruta estén formadas por pares completos de puntos (i^+ , i^-). Esto es, que si contienen un punto de recogida i^+ se tiene que comprobar que también contengan el correspondiente punto de recogida i^- y viceversa. No obstante y dado que, una vez que esto se cumple, el orden de los puntos que componen la cadena no cambia, como se puede ver en la Figura - 13 y en la Figura - 14, en estos movimientos no es necesario comprobar la restricción d).
- Por su parte los dos movimientos “intra-rutas” considerados en el procedimiento **TabuSearch** (Or y Or-Generalizado) no alteran el conjunto de puntos de cada ruta sino exclusivamente su orden. Es por ello que no se requiere chequear en este tipo de movimientos la restricción c) aunque sí es necesario chequear la restricción d).

El tiempo de cálculo computacional que se requiere para chequear estas restricciones puede ser excesivo. Por ello, se explican seguidamente las herramientas que se han incorporado en la resolución del problema para acelerar el proceso de chequeo de la factibilidad.

8.2.1.- La restricción de capacidad máxima de los vehículos

En el caso de la restricción de capacidad, identificada como a) en la enumeración anterior, se precisa indicar primeramente que $\forall x \in V$ se define:

$qac(x)$ como el número de pasajeros presentes en el vehículo correspondiente tras visitar el punto x .

Así mismo, sean y y z dos puntos de la misma ruta donde y precede a z , con $z \neq 0$, se define $maxq(y, z)$ como el máximo valor de $qac(x)$ para los puntos x de la cadena que empieza en y y terminan en z (ambos incluidos).

Se muestran a continuación, para los distintos tipos de inserciones y movimientos que se recogen en este trabajo, ejemplos de cómo las variables qac y $maxq$ se emplean para chequear la factibilidad de la restricción a).

- En las inserciones de pares i^+ e i^- con algún punto intercalado, como la ilustrada en la Figura - 11, éstas son factibles si:

$$maxq(x, z) + q(i) \leq Q$$

- En el caso de inserciones de pares i^+ e i^- de forma consecutiva, siguiendo el ejemplo que se muestra en la Figura - 12, éstas son factibles si se cumple la siguiente condición:

$$qac(x) + q(i) \leq Q$$

- En lo que a un movimiento tipo CROSS se refiere, como el que se detalla en la Figura - 13, éste es factible si se cumplen las siguientes condiciones:

$$maxq(s, p) - qac(r) + qac(v) \leq Q$$

$$maxq(w, x) - qac(v) + qac(r) \leq Q$$

Se precisa indicar, como se ha explicado anteriormente y siguiendo con el mismo ejemplo, que las cadenas que se intercambian, $s - \dots - p$ y $w - \dots - x$, están formadas por pares de puntos (i^+, i^-) , donde $q(i^-) = -q(i^+)$.

Es por ello que el número de pasajeros en los puntos de las cadenas $u - \dots - 0$ y $y - \dots - 0$ no varía con el intercambio.

- Por otra parte, en el caso del cambio de una cadena de una ruta a otra - tipo MOVE -, como el que se representa en la Figura - 14, éste es factible si:

$$maxq(u, w) - qac(p) + qac(r) \leq Q$$

- En intercambios entre dos cadenas consecutivas de una misma ruta - tipo Or -, como el que se define en la Figura - 15, éste es factible si se cumplen las siguientes condiciones:

$$\max q(D, E) - qac(C) + qac(A) \leq Q$$

$$\max q(B, C) - qac(C) + qac(E) \leq Q$$

- En el caso de movimientos tipo Or-Generalizado de intercambio entre dos cadenas no consecutivas de una misma ruta, siguiendo el ejemplo que se muestra en la Figura - 16, éste es factible si se cumplen las siguientes condiciones:

$$\max q(F, G) - qac(E) + qac(A) \leq Q$$

$$\max q(D, E) - qac(C) + qac(G) - qac(E) + qac(A) \leq Q$$

$$\max q(B, C) - qac(C) + qac(G) \leq Q$$

Es importante señalar que, con estas variables auxiliares qac y $\max q$ el número de operaciones que se deben realizar para chequear la factibilidad de cada inserción o movimiento no depende del tamaño del problema.

8.2.2.- La restricción de tiempo máximo de duración de cada ruta

Para el tipo de restricción de tiempo máximo de cada ruta, identificada como b) en la enumeración inicial, indicar que $\forall x \in V, x \neq 0$ se define:

$tac(x)$ como el tiempo total de conducción del vehículo correspondiente tras visitar el punto x .

Así mismo, para cada ruta $R_l, l = 1, \dots, m$ se define $tpRt(l)$ como el tiempo total de la ruta R_l .

Se muestran a continuación, para los distintos tipos de inserciones y movimientos que se recogen en este trabajo, ejemplos de cómo las variables tac y $tpRt$ se emplean para chequear la factibilidad de la restricción b).

- De esta forma una inserción de pares i^+ e i^- con algún punto intercalado, como la ilustrada en la Figura - 11, es factible con respecto a la restricción citada si, siendo R_l la ruta donde se realiza la inserción:

$$tpRt(l) + t_{xi^+} + t_{i^+y} - t_{xy} + t_{zi^-} + t_{i^-p} - t_{zp} \leq L_{max}$$

- De manera análoga en el caso de inserciones de pares i^+ e i^- de forma consecutiva, siguiendo el ejemplo que se muestra en la Figura - 12, éstas son factibles si se cumple la condición que sigue:

$$tpRt(l) + t_{xi^+} + t_{i^+i^-} + t_{i^-y} - t_{xy} \leq L_{max}$$

- En el caso de un movimiento tipo CROSS, como el que se describe en la Figura - 13, éste es factible si se cumplen las siguientes condiciones:

$$tpRt(1) + t_{rw} + tac(x) - tac(w) + t_{xu} - t_{rs} - tac(p) + tac(s) - t_{pu} \leq L_{max}$$

$$tpRt(2) + t_{vs} + tac(p) - tac(s) + t_{py} - t_{vw} - tac(x) + tac(w) - t_{xy} \leq L_{max}$$

- Para movimientos que implican el cambio de una cadena de una ruta a otra - tipo MOVE -, como el que se representa en la Figura - 14, éste es factible si se cumplen las condiciones siguientes:

$$tpRt(1) + t_{ru} + tac(w) - tac(u) + t_{ws} - t_{rs} \leq L_{max}$$

$$tpRt(2) + t_{px} - t_{pu} - tac(w) + tac(u) - t_{wx} \leq L_{max}$$

Es importante señalar que si se verifica la desigualdad triangular la comprobación de ésta segunda condición sería innecesaria.

- En el caso del intercambio entre dos cadenas dentro de una misma ruta es suficiente con calcular las sumas de los tiempos de los arcos que se añaden y la suma de los tiempos de los arcos que se eliminan. Así en un intercambio de dos cadenas consecutivas - tipo Or -, como el que se define en la Figura - 15, éste es factible si:

$$tpRt(l) + t_{AD} + t_{EB} + t_{CF} - t_{AB} - t_{CD} - t_{EF} \leq L_{max}$$

- Para movimientos tipo Or-Generalizado de intercambio entre dos cadenas no consecutivas de una misma ruta, siguiendo el ejemplo que se muestra en la Figura - 16, éste es factible si se cumple la siguiente condición:

$$tpRt(l) + t_{AF} + t_{GD} + t_{EB} + t_{CH} - t_{AB} - t_{CD} - t_{EF} - t_{GH} \leq L_{max}$$

Siendo R_l la ruta donde se realiza el movimiento de intercambio “intra-rutas” en los dos últimos casos presentados.

Es significativo remarcar que también, con estas variables auxiliares tac y $tpRt$ el número de operaciones que se deben realizar para chequear la factibilidad de cada inserción o movimiento no depende del tamaño del problema.

8.2.3.- La restricción de par completo (i^+ , i^-) en la misma ruta

Ya se ha comentado que la restricción identificada como c), en el listado de chequeo de la factibilidad inicial, tan sólo necesita ser chequeada para los movimientos de intercambio “entre-rutas”, estos son los tipo CROSS y tipo MOVE.

Con el fin de que esta restricción, que obliga a que los puntos de carga (i^+) y descarga (i^-) de una solicitud estén en la misma ruta, se cumpla es necesario que:

Las cadenas que se mueven de una ruta a otra estén formadas por solicitudes completas, es decir, pares de puntos de recogida y descarga (i^+ , i^-).

De esta forma, para cada ruta R_l , $l = 1, \dots, m$ se define y se calcula el conjunto $SetFeasibleChain(l)$ como el conjunto de cadenas de la ruta R_l que se pueden mover sin violar la restricción c). En otras palabras, el conjunto de rutas formadas por pares enteros de puntos de carga y descarga.

Así por ejemplo, para la ruta R_l que se muestra en la Figura - 17 el conjunto $SetFeasibleChain(l)$ es el siguiente:

$$SetFeasibleChain(l) = \begin{aligned} & A^+ - B^+ - B^- - A^+, \\ & A^+ - B^+ - B^- - A^+ - C^+ - C^-, \\ & B^+ - B^-, \\ & C^+ - C^-. \end{aligned}$$

Figura - 17: Ruta de ejemplo R_l



Por otra parte, para así ahorrar memoria y poder agilizar los cálculos, las cadenas que componen cada conjunto $SetFeasibleChain$ pueden ser definidas por las posiciones de sus elementos inicial y final.

8.2.4.- La restricción de precedencia (i^+ antes que i^-)

Como se ha comentado anteriormente las restricciones del tipo d), identificadas como tal en el listado de chequeo de la factibilidad inicial, sólo se chequean para los movimientos de intercambio “intra-rutas”, estos son los tipo Or y tipo Or-Generalizado.

Para que esta restricción, que obliga a que el punto de carga (i^+) preceda al punto de descarga (i^-) en la ruta, se cumpla es necesario que:

Sean y y z dos puntos de la misma ruta donde y precede a z , con $y, z \neq 0$, se define $maxpos(y, z)$ como el máximo número de posiciones que puede trasladarse hacia la derecha la cadena que empieza en y y termina en z , en su misma ruta, sin violar las restricciones de precedencia d).

Por ello en el ejemplo de la ruta R_l que se muestra en la Figura - 17 se obtienen los siguientes valores de $maxpos$:

$$maxpos(A^+, A^+) = 2;$$

En efecto, la cadena formada únicamente por el punto A^+ de recogida puede trasladarse dos posiciones hasta poner justo delante del punto de descarga A^- . No puede trasladarse más posiciones hacia la derecha porque entonces esto haría que el punto A^+ se colocase por detrás del punto A^- , lo que violaría la restricción d).

De igual forma se tiene que:

$$\begin{array}{lll} maxpos(A^+, B^+) = 0, & maxpos(A^+, B^-) = 0, & maxpos(A^+, A^-) = 2, \\ maxpos(A^+, C^+) = 0, & maxpos(A^+, C^-) = 0, & maxpos(A^+, C^-) = 0, \\ maxpos(B^+, B^+) = 0, & maxpos(B^+, B^-) = 3, & maxpos(B^+, A^-) = 2, \\ maxpos(B^+, C^+) = 0, & maxpos(B^+, C^-) = 0, & maxpos(B^-, A^-) = 2, \\ maxpos(B^-, C^+) = 0, & maxpos(B^-, C^-) = 0, & maxpos(A^-, A^-) = 2, \\ maxpos(A^-, C^+) = 0, & maxpos(A^-, C^-) = 0, & maxpos(C^+, C^+) = 0, \\ maxpos(C^+, C^-) = 0 & y & maxpos(C^-, C^-) = 0. \end{array}$$

Todas estas variables auxiliares (qac , $maxq$, tac , $tpRt$, $SetFeasibleChain$ y $maxpos$) definidas para acelerar el cálculo se inicializan al principio de cada procedimiento. Cuando una ruta es modificada, esto es la ejecución de una inserción en el procedimiento **Constructive** o de un movimiento en el procedimiento **TabuSearch**, sólo es necesario actualizar los valores correspondientes a la ruta o rutas modificadas.

Así mismo, para facilitar el cálculo y la actualización de estas variables auxiliares, se hace uso de otros dos variables auxiliares rut y pos que se definen como sigue:

$$\forall x \in V - \{0\} \quad \begin{array}{l} rut(x) \text{ es el índice de la ruta a la que pertenece } x, \text{ y} \\ pos(x) \text{ es la posición que ocupa } x \text{ en dicha ruta.} \end{array}$$

Las variables rut y pos también se calculan al principio de cada procedimiento y, posteriormente, tan sólo es necesario actualizar los valores correspondientes a las rutas modificadas.

8.3.- EL CÁLCULO DE LAS FUNCIONES OBJETIVO

Al igual que para el chequeo de las restricciones, el cálculo de las funciones objetivos de las soluciones a las que dan lugar las inserciones y movimientos que se evalúan en los procedimientos **Constructive** y **TabuSearch** puede conllevar excesivo tiempo de resolución si no se emplean las herramientas adecuadas. Es necesario, por tanto, definir una serie de estrategias y variables auxiliares que ayuden a agilizar estos cálculos.

8.3.1.- La función objetivo económica

La estrategia de aceleración que se propone en el caso concreto de la función objetivo económica es sencilla y se basa en lo siguiente:

El valor de este objetivo es la suma de la distancia total recorrida, es decir, la suma de los arcos que componen las rutas de cada solución. Como las inserciones y movimientos consisten en añadir una serie de arcos y eliminar otros, la distancia total recorrida por la posible nueva solución se puede calcular como la distancia de la solución actual más la suma de los arcos que se añaden, menos la suma de los arcos que se eliminan en la nueva solución.

De esta forma, el cálculo de la función objetivo económica es relativamente fácil y se puede hacer en un tiempo de cálculo que no depende del tamaño de problema.

8.3.2.- La función objetivo social

Resulta más complicado plantear estrategias que aceleren el tiempo de cálculo de la función objetivo social. Hay para ello que recordar que:

Sea S una solución, por tanto formada por m rutas, el objetivo social se puede calcular formalmente de la forma siguiente:

$$\max \{\Delta_i(S) : i \in P\}$$

donde $\forall i \in P$, $\Delta_i(S) = T_i(S) - t_{i^+i^-}$ y $T_i(S)$ es el tiempo de viaje de la solicitud i en la solución S .

Según eso, ante una inserción o movimiento se trata de poder determinar los valores de T_i correspondientes a la nueva solución y más en concreto: determinar qué solicitudes van a estar afectadas por dicha inserción o movimiento, modificando su valor T_i , y de qué forma va a cambiar dicho valor.

Para ello es necesario definir un conjunto de variables auxiliares que intentan agilizar esos cálculos:

- $\forall x \in V - \{0\}$ se define $PT0(x)$ como el conjunto de solicitudes $i \in P$ cuyos puntos de recogida y descarga están en la misma ruta que x de forma que i^+ es anterior a x , e i^- es el mismo x o posterior a x . Más formalmente se expresa como:

$$PT0(x) = \{i \in P : rut(i^+) = rut(x), pos(i^+) < pos(x), pos(i^-) \geq pos(x)\}$$

- $\forall x, y \in V - \{0\}$ tales que $rut(x) = rut(y)$ y $pos(x) \leq pos(y)$ se define $PT1(x, y)$ como el conjunto de solicitudes $i \in P$ cuyos puntos de recogida y descarga están en la misma ruta que x de forma que i^+ es anterior a x , e i^- está en la cadena que empieza en x y termina en y . Se puede formular como:

$$PT1(x, y) = \{i \in P: rut(i^+) = rut(x), pos(i^+) < pos(x), pos(x) \leq pos(i^-) \leq pos(y)\}$$

- $\forall x, y \in V - \{0\}$ tales que $rut(x) = rut(y)$ y $pos(x) \leq pos(y)$ se define $PT2(x, y)$ como el conjunto de solicitudes $i \in P$ cuyos puntos de recogida y descarga están en la misma ruta que x de forma que i^+ está en la cadena que empieza en x y termina en y e i^- es posterior a y . De manera formal esto es:

$$PT2(x, y) = \{i \in P: rut(i^+) = rut(x), pos(x) \leq pos(i^+) \leq pos(y), pos(i^-) > pos(y)\}$$

- $\forall x, y \in V - \{0\}$ tales que $rut(x) = rut(y)$ y $pos(x) \leq pos(y)$ se define $PT3(x, y)$ como el conjunto de solicitudes $i \in P$ cuyos puntos de recogida y descarga están en la misma ruta que x de forma que i^+ es anterior a x e i^- es posterior a y . Más formalmente esto se expresa como:

$$PT3(x, y) = \{i \in P: rut(i^+) = rut(x), pos(i^+) < pos(x), pos(i^-) > pos(y)\}$$

En las figuras siguientes se muestran ejemplos de rutas y de elementos identificables dentro de los conjuntos $PT0$, $PT1$, $PT2$ y $PT3$.

Figura - 18: Ruta con petición que pertenece a $PT0(x)$

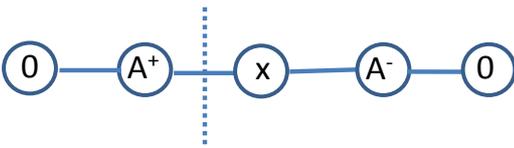
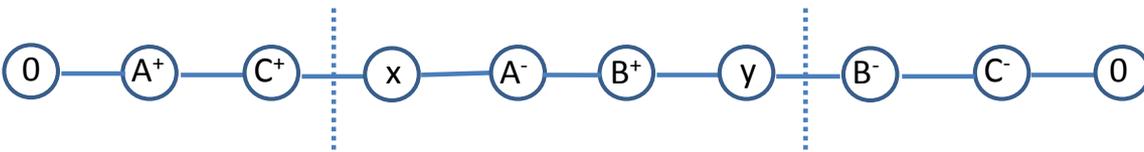


Figura - 19: Ruta con peticiones que pertenecen a $PT1(x, y)$, $PT2(x, y)$ y $PT3(x, y)$



Por consiguiente, en la Figura - 18 la solicitud A está incluida en el conjunto $PT0(x)$, ya que A^+ está antes del punto x , y A^- está después de dicho punto.

De igual forma, en la Figura - 19 se observa que la solicitud A pertenece a $PT1(x, y)$, la solicitud B pertenece a $PT2(x, y)$ y la solicitud C pertenece a $PT3(x, y)$, al cumplir en cada caso con las formulaciones anteriores.

A continuación se citan algunos ejemplos de conjuntos PT0, PT1, PT2 y PT3 que se derivan de la ruta R_l representada en la Figura - 17:

$$\begin{array}{lll}
 PT0(A^+) = \emptyset, & PT0(B^+) = \{A\}, & PT0(B^-) = \{A, B\}, \\
 PT0(A^-) = \{A\}, & PT0(C^+) = \emptyset, & PT0(C^-) = \{C\}, \\
 PT1(A^+, A^+) = \emptyset, & PT2(A^+, A^+) = \{A\}, & PT3(A^+, A^+) = \emptyset, \\
 PT1(B^+, B^+) = \emptyset, & PT2(B^+, B^+) = \{B\}, & PT3(B^+, B^+) = \{A\}, \\
 PT1(A^+, B^+) = \emptyset, & PT2(A^+, B^+) = \{A, B\}, & PT3(A^+, B^+) = \emptyset, \\
 PT1(A^+, B^-) = \emptyset, & PT2(A^+, B^-) = \{A\}, & PT3(A^+, B^-) = \emptyset, \\
 PT1(B^+, C^+) = \{A\}, & PT2(B^+, C^+) = \{C\}, & PT3(B^+, C^+) = \emptyset, \text{ etc.}
 \end{array}$$

Una vez que se han definido formalmente los conjuntos de las cuatro variables auxiliares se puede determinar, ante una inserción o movimiento, que solicitudes $i' \in P$ van a quedar “alteradas” (es decir, aquellas cuyos valores $T_{i'}$ van a cambiar), y cómo van a ser los cambios de los correspondientes $T_{i'}$. Así, posteriormente se puede calcular el valor de la función objetivo social de la nueva solución completa o el parcial, correspondiente a dicha inserción o movimiento de manera aislada.

Ahora se detalla concretamente, para los dos tipos de inserciones y los cuatro tipos de movimientos que se recogen en los procedimientos **Constructive** y **TabuSearch** respectivamente, como se calculan las solicitudes alteradas y su subsecuente impacto en el valor $T_{i'}$.

- I. En las inserciones de pares i^+ e i^- con algún punto intercalado, como la que se muestra en la Figura - 11, se tiene que las solicitudes alteradas son las que componen $PT1(y, z)$, $PT2(y, z)$ y $PT3(y, z)$, además de la propia solicitud i que se inserta. Indicar que los valores de los $T_{i'}$ correspondientes cambian de la forma siguiente:

- $\forall i' \in PT1(y, z)$ el valor de $T_{i'}$ varía en $t_{xi^+} + t_{i^+y} - t_{xy}$
- $\forall i' \in PT2(y, z)$ el valor de $T_{i'}$ varía en $t_{zi^-} + t_{i^-p} - t_{zp}$
- $\forall i' \in PT3(y, z)$ el valor de $T_{i'}$ varía en $t_{xi^+} + t_{i^+y} - t_{xy} + t_{zi^-} + t_{i^-p} - t_{zp}$
- Para la solicitud i que se inserta el valor T_i es $T_i = t_{i^+y} + tac(z) - tac(y) + t_{zi^-}$

II. En el caso de las inserciones de pares $i^+e i^-$ de forma consecutiva, como la que se representa en la Figura - 12, se tiene que las solicitudes alteradas son las que componen $PT0(y)$, además de la propia solicitud i que se inserta. Señalar que el valor del $T_{i'}$ correspondiente cambian de la forma siguiente:

- $\forall i' \in PT0(y)$ el valor de $T_{i'}$ varía en $t_{xi^+} + t_{i^+i^-} + t_{i^-y} - t_{xy}$
- Para la solicitud i que se inserta el valor de T_i es $T_i = t_{i^+i^-}$ y por tanto $\Delta_i = 0$

III. En un movimiento tipo CROSS, como el que se ejemplifica en la Figura - 13, las solicitudes alteradas son las que componen $PT3(s, p)$ y $PT3(w, x)$. Reseñar que los valores de los T_i correspondientes cambian de la forma siguiente:

- $\forall i \in PT3(s, p)$ el valor de T_i varía en

$$t_{rw} + tac(x) - tac(w) + t_{xu} - t_{rs} - tac(p) + tac(s) - t_{pu}$$
- $\forall i \in PT3(w, x)$ el valor de T_i varía en

$$t_{vs} + tac(p) - tac(s) + t_{py} - t_{vw} - tac(x) + tac(w) - t_{xy}$$

IV. En el caso de un movimiento tipo MOVE, como el mostrado en la Figura - 14, las solicitudes alteradas son las que componen $PT0(s)$ y $PT3(u, w)$. Indicar que los valores de los T_i correspondientes cambian de la forma siguiente:

- $\forall i \in PT0(s)$ el valor de T_i varía en $t_{ru} + tac(w) - tac(u) + t_{ws} - t_{rs}$
- $\forall i \in PT3(u, w)$ el valor de T_i varía en $t_{px} - t_{pu} - tac(w) + tac(u) - t_{wx}$

V. En movimientos “intra-rutas” de tipo Or de intercambio entre dos cadenas consecutivas de una misma ruta, como en el ejemplo que se representa en la Figura - 15, se tienen cuatro cadenas: $0 - A$, $B - C$, $D - E$ y $F - 0$. Las solicitudes i cuyos puntos de recogida y descarga estén incluidos ambos en alguna de estas cadenas no van a estar alterados por el intercambio. Por otra parte como el intercambio entre las cadenas $B - C$ y $D - E$ es factible no puede haber ninguna petición i cuyo punto de recogida esté en $B - C$ y cuyo punto de descarga esté en $D - E$. El conjunto de localizaciones posibles de los pares de puntos de las diferentes solicitudes se resume en la Tabla - 1. Cada fila representa una posible situación. Las cuatro primeras columnas indican en qué cadena está situado cada punto de la solicitud: la recogida se señala con “+” y la descarga con “-”. La última columna indica, en función de estas localizaciones, si la petición se altera o no por el intercambio.

Tabla - 1: Posibles localizaciones de los pares de puntos de las diferentes solicitudes de una ruta ante un intercambio Or factible.

0 - A	B - C	D - E	F - 0	Alterada
+ -				NO
	+ -			NO
		+ -		NO
			+ -	NO
+	-			Sí
+		-		Sí
+			-	Sí
	+		-	Sí
		+	-	Sí

Entonces, las solicitudes alteradas son las que componen $PT1(B, C)$, $PT1(D, E)$, $PT3(B, E)$, $PT2(B, C)$ y $PT2(D, E)$. Señalar que los valores de los T_i correspondientes cambian de la forma siguiente:

- $\forall i \in PT1(B, C)$ el valor de T_i varía en $t_{AD} + tac(E) - tac(D) + t_{EB} - t_{AB}$
- $\forall i \in PT1(D, E)$ el valor de T_i varía en $t_{AD} - t_{AB} - tac(C) + tac(B) - t_{CD}$
- $\forall i \in PT3(B, E)$ el valor de T_i varía en $t_{AD} + t_{EB} + t_{CF} - t_{AB} - t_{CD} - t_{EF}$
- $\forall i \in PT2(B, C)$ el valor de T_i varía en $t_{CF} - t_{CD} - tac(E) + tac(D) - t_{EF}$
- $\forall i \in PT2(D, E)$ el valor de T_i varía en $t_{EB} + tac(C) - tac(B) + t_{CF} - t_{EF}$

VI. En el caso de movimientos “intra-rutas” de tipo Or-Generalizado de intercambio entre dos cadenas no consecutivas de una misma ruta, como en el ejemplo que se representa en la Figura - 16, el razonamiento es muy similar al tipo anterior teniendo en este caso cinco cadenas: $0 - A$, $B - C$, $D - E$, $F - G$ y $H - 0$. Igual que antes, las solicitudes i cuyos puntos de recogida y descarga estén incluidos ambos en alguna de estas cadenas no van a estar alterados por el intercambio. Teniendo en cuenta que el intercambio debe ser factible, entonces no puede haber ninguna petición i cuyo punto de recogida se encuentre en $B - C$ y cuyo punto de descarga esté en $D - E$ o en $F - G$. Tampoco puede existir ninguna solicitud i cuyo punto de recogida se sitúe en $D - E$ y cuyo punto de descarga esté en $F - G$. El conjunto de localizaciones posibles de los pares de puntos de las diferentes solicitudes se resume en la Tabla - 2 que sigue.

Tabla - 2: Posibles localizaciones de los pares de puntos de las diferentes solicitudes de una ruta ante un intercambio Or-Generalizado factible.

0 - A	B - C	D - E	F - G	H - 0	Alterada	
+	-				NO	
	+	-			NO	
		+	-		NO	
			+	-	NO	
				+	-	NO
+					Sí	
+			-		Sí	
+					Sí	
+				-	Sí	
	+			-	Sí	
		+		-	Sí	
			+	-	Sí	

Entonces, las solicitudes alteradas son las que componen $PT1(B, C)$, $PT1(D, E)$, $PT1(F, G)$, $PT3(B, G)$, $PT2(B, C)$, $PT2(D, E)$ y $PT2(F, G)$. Los valores de los T_i correspondientes cambian de la forma siguiente:

- $\forall i \in PT1(B, C)$ el valor de T_i varía en

$$t_{AF} + tac(G) - tac(F) + t_{GD} + tac(E) - tac(D) + t_{EB} - t_{AB}$$

- $\forall i \in PT1(D, E)$ el valor de T_i varía en

$$t_{AF} + tac(G) - tac(F) + t_{GD} - t_{AB} - tac(C) + tac(B) - t_{CD}$$

- $\forall i \in PT1(F, G)$ el valor de T_i varía en

$$t_{AF} - t_{AB} - tac(C) + tac(B) - t_{CD} - tac(E) + tac(D) - t_{EF}$$

- $\forall i \in PT3(B, G)$ el valor de T_i varía en

$$t_{AF} + t_{GD} + t_{EB} + t_{CH} - t_{AB} - t_{CD} - t_{EF} - t_{GH}$$

- $\forall i \in PT2(B, C)$ el valor de T_i varía en

$$t_{CH} - t_{CD} - tac(G) + tac(D) - t_{GH}$$

- $\forall i \in PT2(D, E)$ el valor de T_i varía en

$$t_{EB} + tac(C) - tac(B) + t_{CH} - t_{EF} - tac(G) + tac(F) - t_{GH}$$

- $\forall i \in PT2(F, G)$ el valor de T_i varía en

$$t_{GD} + tac(E) - tac(D) + t_{EB} + tac(C) - tac(B) + t_{CH} - t_{GH}$$

Como en el apartado anterior de estrategias de aceleración para el chequeo de la factibilidad, las variables auxiliares PT_0 , PT_1 , PT_2 y PT_3 se calculan al comienzo de cada procedimiento y, posteriormente, tras producirse cada inserción o movimiento tan sólo es necesario actualizar los valores correspondientes a las rutas modificadas.

En este caso, de estrategias de aceleración para el cálculo de las funciones objetivo, para el cálculo de PT_0 , PT_1 , PT_2 y PT_3 también se hace uso de las variables auxiliares rut y pos definidas en el apartado anterior.

En el cálculo de las variaciones de los T_i se usan las variables auxiliares tac , ya definidas en el anteriormente para el chequeo de la factibilidad del tiempo máximo de duración de cada ruta.

Por otra parte, el cálculo de las funciones objetivo es conveniente hacerlo sobre las inserciones y movimientos factibles para evitar cálculos innecesarios.

Todo lo arriba expuesto se debe tener en cuenta a la hora de establecer el orden de actualización de las variables auxiliares. En el Capítulo 10 se analiza el efecto del uso de estas estrategias de aceleración, descritas a lo largo del presente capítulo, en el tiempo de computación de los procedimientos **Constructive** y **TabuSearch**.

9.- LAS METODOLOGÍAS DE RESOLUCIÓN

9.- LAS METODOLOGÍAS DE RESOLUCIÓN

9.1.- INTRODUCCIÓN

En este trabajo de investigación se proponen dos métodos de solución heurísticos cuyos resultados se comparan y recogen dentro del capítulo dedicado a las pruebas computacionales. Para la resolución heurística del problema se van a proponer, y explicar a continuación, las dos metodologías siguientes:

- La primera basada en la estrategia MOAMP
- La segunda consiste en el MHeC

Los dos procedimientos expuestos en el Capítulo 7, **Constructive** y **TabuSearch**, se emplean en la resolución del problema al incluirse en ambas metodologías. En los párrafos siguientes se explican de manera detallada dichas metodologías.

9.2.- EL MÉTODO MOAMP

Uno de los métodos que se propone en este trabajo para resolver el modelo formulado, es una adaptación del procedimiento o estrategia MOAMP para problemas multiobjetivo.

Este procedimiento genera, esencialmente, una aproximación a la curva de eficiencia, enlazando varias ejecuciones de un procedimientos de búsqueda tabú, en este caso se emplea **TabuSearch**. Es decir, hace uso de ejecuciones que utilizan, como solución inicial, la solución final obtenida en la ejecución anterior. Posteriormente, se realiza una exploración alrededor de las soluciones obtenidas hasta ese momento.

Las ideas en las que se basa MOAMP son:

- 1) El principio de proximidad de puntos eficientes, según el cual, en un entorno o vecindario de una solución eficiente se puede encontrar otra solución eficiente.
- 2) La solución que minimiza la distancia L_∞ normalizada y/o ponderada al punto ideal es también eficiente.

Para la descripción de este procedimiento se define la función:

$$F_\lambda(S) = \max \left\{ \lambda \cdot \frac{f_1(S) - f_1^{\min}}{f_1^{\max} - f_1^{\min}}, (1 - \lambda) \cdot \frac{f_2(S) - f_2^{\min}}{f_2^{\max} - f_2^{\min}} \right\}$$

donde $\lambda \in (0,1)$ y f_i^{\min} y f_i^{\max} son el mínimo y máximo valor encontrado para f_i en el conjunto de soluciones no dominadas $i = 1, 2$.

Indicar que, precisamente, la función F_λ representa la distancia ponderada multiplicada por λ y $1 - \lambda$. Se ha observado, igualmente, que las soluciones que minimizan F_λ son también eficientes. Se va a denotar por *SetND* el conjunto de soluciones no dominadas encontradas durante la búsqueda. El conjunto *SetND* final será la aproximación a la curva de eficiencia obtenida.

El Pseudocódigo - 3 describe la adaptación de la estrategia MOAMP al caso concreto del problema analizado en esta investigación. Como se puede comprobar en este pseudocódigo, esta metodología consta de 3 fases, que se explican con detalle más adelante.

Pseudocódigo - 3: Descripción de la adaptación de MOAMP al problema

FASE I

1. Hacer $SetND = \emptyset$
2. Ejecutar **GeneradorSocial**(S) para obtener una solución S que minimice el objetivo social (f_1)
3. Ejecutar **TabuSearch**(f_2, S), tomando f_2 como función objetivo ($f_1 \rightarrow f_2$)
4. Ejecutar **GeneradorCostes** (S) para obtener una solución S minimizando la distancia, objetivo económico (f_2)
5. Ejecutar **TabuSearch**(f_1, S), tomando f_1 como función objetivo ($f_2 \rightarrow f_1$)
6. Actualizar $SetND$ con todas las soluciones no-dominadas encontradas

FASE II

Repetir

7. Generar $\lambda \in (0,1)$ aleatoriamente
8. Ejecutar **TabuSearch**(F_λ, S), tomando F_λ como función objetivo
9. Actualizar $SetND$

Hasta que trascurren *maxiterfaseii* iteraciones sin cambios en $SetND$

FASE III

Repetir

10. Identificar las soluciones S de $SetND$ cuyos vecindarios no han sido explorados (todas en la primera iteración)
11. Para cada una de estas soluciones S explorar sus vecindarios y actualizar $SetND$ con cada una de las soluciones vecinas

Hasta que $SetND$ se estabilice (no cambie)

En la primera fase se persigue obtener las mejores soluciones con respecto a cada una de las funciones objetivo originales. En otras palabras, se obtienen los extremos de la curva de eficiencia y por tanto los valores de f_1^{min} , f_1^{max} , f_2^{min} y f_2^{max} que son usados en el cálculo de las funciones mixtas F_λ en la fase siguiente.

No obstante, como se ha mencionado, en la FASE I en realidad se obtienen aproximaciones a estos valores que no siempre tienen por qué coincidir con el real. Incluso a veces estas aproximaciones son mejoradas en fases posteriores.

Por tanto, en la descripción del método general, f_1^{min} y f_2^{max} se redefinen respectivamente como los valores de los objetivos f_1 y f_2 de la mejor solución con respecto al objetivo social de *SetND*. Por su parte f_2^{min} y f_1^{max} se redefinen respectivamente como los valores de los objetivos f_2 y f_1 de la mejor solución en *SetND* con respecto al coste.

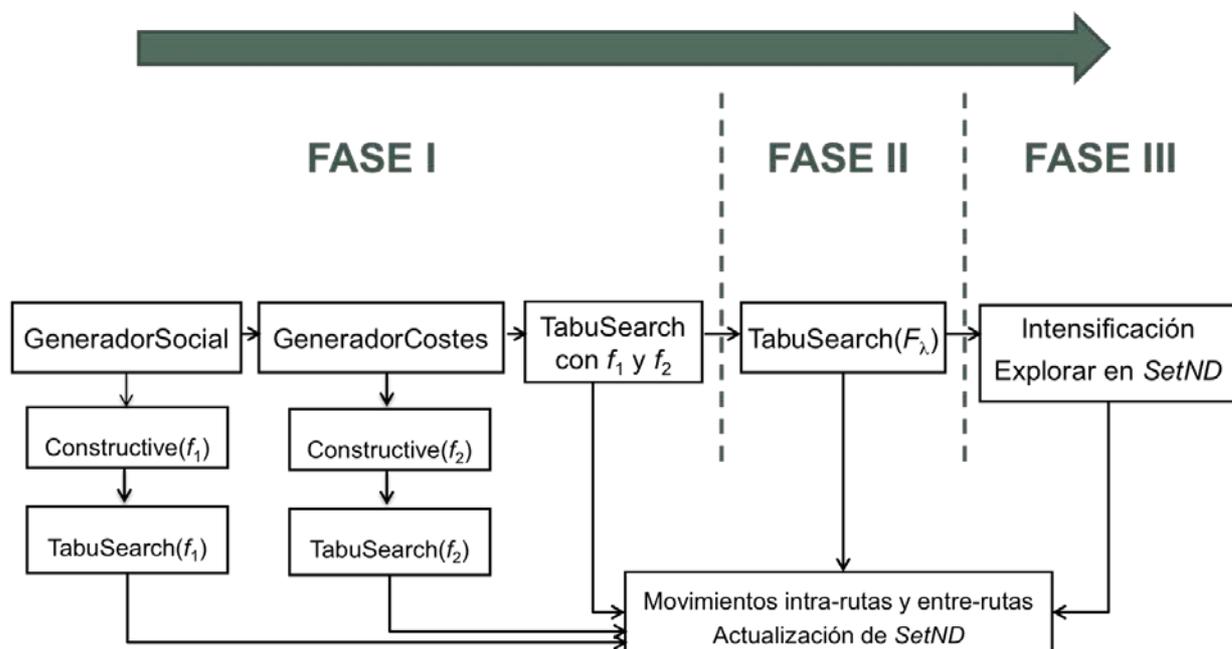
En la segunda fase, se obtiene una muestra diversa de soluciones ‘de compromiso’; es decir, no se trata de las mejores soluciones con respecto a algún objetivo concreto, pero sí son buenas soluciones en conjunto. Esta fase finaliza cuando transcurre un cierto número de iteraciones (*maxiterfaseii*) sin cambios en *SetND*.

Hay que indicar que, dentro de cada procedimiento de búsqueda tabú, se comprueba si cada solución visitada se incorpora a *SetND*. En otras palabras, el conjunto de soluciones no dominadas se actualiza con todas las soluciones visitadas durante cada ejecución de la búsqueda tabú en estas dos primeras fases.

Finalmente, en la tercera fase se completa el conjunto de soluciones no dominadas explorando el vecindario de las encontradas hasta ese momento. Los vecindarios de cada solución $S \in SetND$ que se exploran en el paso 11 de la FASE III, son precisamente los mismos vecindarios $N(S)$ usados en el procedimiento **TabuSearch**.

La Figura - 20 representa las tres fases y la jerarquía de procedimientos de la metodología de solución MOAMP adaptada al problema que se estudia.

Figura - 20: Jerarquía de procedimientos en MOAMP



A continuación se describen los procedimientos **GeneradorSocial** y **GeneradorCostes**, incluidos en la FASE I de MOAMP. Así mismo, se explica cómo se favorece la diversidad del proceso a través de la modificación del procedimiento **Constructive** explicado en capítulos precedentes.

9.2.1.- Los procedimientos **GeneradorSocial** y **GeneradorCostes**

El procedimiento **GeneradorSocial** consiste en una estrategia iterativa de tipo multi-arranque. A continuación, en Pseudocódigo - 4 se describe este procedimiento.

Pseudocódigo - 4: Procedimiento **GeneradorSocial**

Procedimiento **GeneradorSocial** (*var S*)

1. Hacer $f_{best} = \infty$ y $freq(i, j) = 0, \forall i, j \in V, i \neq j$

Repetir

2. **Constructive**($f_1, factible, S$)
3. Si *no factible* entonces ir al final del bucle (2-6)
4. Ejecutar **TabuSearch**(f_1, S)
5. Actualizar *freq* con las soluciones visitadas en el paso 4
6. Si $f_1(S) < f_{best}$ entonces hacer: $S_{best} = S$ y $f_{best} = f_1(S)$

Hasta que transcurran *maxitersocial* iteraciones sin cambios en *f_{best}*

7. Hacer $S = S_{best}$
-

El procedimiento **GeneradorCostes** sigue el mismo esquema, variando únicamente la función objetivo, esto es, f_1 por f_2 . Seguidamente, en Pseudocódigo - 5 se describe dicho procedimiento.

Pseudocódigo - 5: Procedimiento **GeneradorCostes**

Procedimiento **GeneradorSocial** (*var S*)

1. Hacer $f_{best} = \infty$ y $freq(i, j) = 0, \forall i, j \in V, i \neq j$

Repetir

2. **Constructive**($f_2, factible, S$)
3. Si *no factible* entonces ir al final del bucle (2-6)
4. Ejecutar **TabuSearch**(f_2, S)
5. Actualizar *freq* con las soluciones visitadas en el paso 4
6. Si $f_2(S) < f_{best}$ entonces hacer: $S_{best} = S$ y $f_{best} = f_2(S)$

Hasta que transcurran *maxitersocial* iteraciones sin cambios en *f_{best}*

7. Hacer $S = S_{best}$
-

9.2.2.- La aportación de diversidad al procedimiento

Según se puede observar en los descrito en la sección anterior, en cada iteración se construye una solución que es mejorada posteriormente por el procedimiento **TabuSearch**. Y entonces, cada vez que se construye una solución se tienen en cuenta los arcos de las soluciones visitadas anteriormente, de forma que se penalizan los arcos que han aparecido más veces y se favorecen los menos.

Teniendo por objeto el favorecer la diversidad del proceso se hace uso de la matriz auxiliar $freq(i, j)$, donde se registra el número de veces que el arco (i, j) aparece en las soluciones visitadas en las ejecuciones del procedimiento **TabuSearch**. Para poder obtener soluciones diversas haciendo uso de la matriz $freq$ el procedimiento **Constructive**, recogido en el Pseudocódigo - 1, se modifica como se describe en el Pseudocódigo - 6 que sigue.

Pseudocódigo - 6: Modificación del procedimiento Constructive

Modificación del Procedimiento **Constructive** (g ; $var\ factible$; $var\ S$)

Sustituir el paso 6 por los pasos 6.a – 6.d siguientes:

6.a Calcular $\forall ins \in L$:

$\Delta f(ins)$: Variación de la suma de las frecuencias ($freq$) de los arcos que aparecen en la solución si se ejecuta la inserción ins

$\Delta g(ins)$: Variación de la función objetivo g de la solución si se ejecuta la inserción ins

6.b Calcular:

$$ming = \max\{\Delta g(ins): ins \in L\}$$

$$ming = \min\{\Delta g(ins): ins \in L\}$$

$$maxf = \max\{\Delta f(ins): ins \in L\}$$

$$minf = \min\{\Delta f(ins): ins \in L\}$$

6.c Calcular $\forall ins \in L$:

$$val(ins) = \beta \left(\frac{\Delta g(ins) - ming}{maxg - ming} \right) + (1 - \beta) \left(\frac{\Delta f(ins) - minf}{maxf - minf} \right) \quad \text{si } maxf \neq minf$$

$$val(ins) = \left(\frac{\Delta g(ins) - ming}{maxg - ming} \right) \quad \text{si } maxf = minf$$

6.d Calcular:

$$maxval = \max\{val(ins): ins \in L\}$$

$$minval = \min\{val(ins): ins \in L\}$$

Sustituir el paso 7 por los pasos 7.a y 7.b siguientes:

7.a Determinar $L' = \{ins \in L : val(ins) \leq \alpha \cdot minval + (1 - \alpha) \cdot maxval\}$

7.b Elegir $ins^* \in L'$ aleatoriamente

Como se observa en el paso 6.c del Pseudocódigo - 6 $\forall ins \in L$ se calcula un valor $val(ins)$ que tiene en cuenta tanto la variación del valor de la función objetivo (Δg) como la variación de la suma de las frecuencias de los arcos (Δf), si se ejecuta ins . El parámetro $\beta \in (0,1)$ indica el peso que se otorga a cada uno de estos dos criterios.

Finalmente, no se elige necesariamente la inserción correspondiente al mayor valor de val . En su lugar, se construye una lista (L' o lista de candidatos) con las inserciones correspondientes a los mayores valores y se elige ins^* de L' de manera aleatoria. El parámetro $\alpha \in (0,1)$ sirve para establecer el tamaño de la lista y por tanto el grado de aleatoriedad del procedimiento. El valor de $\alpha = 0$ hace que L' coincida con L (todas las inserciones posibles), y $\alpha = 1$ hace que L' coincida solo con la inserción (o inserciones) correspondiente al menor valor de val .

La idea de formar una lista de candidatos L' con las mejores inserciones y elegir una aleatoriamente de L' sigue la estrategia GRASP que se desarrolla en Feo y Resende (1989) y en Feo y Resende (1995). La aplicación de lo anterior permite construir soluciones diversas y de calidad.

Se concluye, por tanto, que se aportan dos fuentes de diversidad en la metodología MOAMP planteada, una a través de una penalización por frecuencias y otra por medio de la aleatoriedad, que se controlan con los parámetros β y α respectivamente. Posteriormente, se analiza cuál es la combinación de valores más adecuada para estos dos parámetros dentro del capítulo dedicado a las pruebas computacionales.

9.3.- EL MÉTODO ϵ -CONSTRAINT HEURÍSTICO (MHEC)

El segundo método de solución que se propone es una variante heurística del método ϵ -constraint. Para una mejor descripción del mismo se requiere presentar las definiciones y notaciones siguientes:

Se define *MonoSocial*(Dmx) como el problema mono-objetivo definido por la función objetivo social (1) del apartado 7.3, por las restricciones (3) a (14), y donde se añade la restricción de que la distancia total no supere Dmx , es decir:

$$\sum_{k=1}^m \sum_{i \in V} \sum_{j \in V} d_{ij} \cdot x_{ij}^k \leq Dmx \quad (15)$$

o lo que es lo mismo: una solución S del problema *MonoSocial*(Dmx) debe verificar:

$$f_2(S) \leq Dmx$$

además de las otras restricciones del problema original.

Sea $Dmax$ un valor suficientemente alto inicial de Dmx .

El MHeC adaptado al problema que se estudia en la presente investigación se describe en el Pseudocódigo - 7 que sigue.

Pseudocódigo - 7: Método Heurístico_ε-constraint (MHeC)

Método Heurístico_ε-constraint

1. Determinar un valor alto inicial de Dmx que se denota como $Dmax$
2. Hacer $Dmx = Dmax$ y $SetND = \emptyset$

Repetir

3. Resolver $MonoSocial(Dmx)$
4. Si en 3 se ha obtenido una solución factible S hacer:
 $SetND = SetND \cup \{S\}$ y $Dmx = f_2(S)-1$

Hasta no encontrar soluciones factibles en 3

5. Extraer de $SetND$ aquellas soluciones que estén dominadas por alguna otra
-

En la primera iteración, el método resuelve el problema $MonoSocial$ para un valor lo suficientemente alto de Dmx , de forma que la restricción (15) no se sature. O lo que es lo mismo, se intenta obtener el óptimo de la función objetivo social (f_1).

En las siguientes iteraciones se resuelve el problema $MonoSocial$ para un valor de Dmx que corresponde justo con valor de la función de costes f_2 de la solución obtenida en la iteración anterior menos una unidad. Al usar valores enteros, Dmx es justo el mayor valor posible en f_2 que podría tener una solución que mejore a la solución obtenida en la iteración anterior en f_2 . De esta forma en cada iteración se obtienen mejores soluciones con respecto a f_2 . Hay que tener en cuenta que, al usar cada vez menores valores de Dmx , el espacio de soluciones de $MonoSocial(Dmx)$ también es cada vez menor. Por tanto, es de esperar, que se obtenga en cada iteración peores, o a lo sumo iguales, soluciones con respecto a f_1 . En cualquier caso, como en la resolución de $MonoSocial(Dmx)$ se usa un procedimiento heurístico, esto no siempre está garantizado, ya que tampoco se garantiza la obtención del óptimo del problema $MonoSocial(Dmx)$.

El procedimiento planteado para la resolución del problema mono-objetivo $MonoSocial(Dmx)$, en el paso 3 del método, es un proceso iterativo donde en cada iteración se genera una solución mediante un procedimiento constructivo que posteriormente es mejorada por un procedimiento de búsqueda tabú. Por tanto, el procedimiento coincide con el procedimiento **GeneradorSocial** con pequeñas modificaciones adicionales para incluir la restricción (15) tanto en el procedimiento **Constructive** como en el **TabuSearch**.

Como se ha expuesto anteriormente, las inserciones del procedimiento **Constructive** y los movimientos del procedimiento **TabuSearch** consisten en la eliminación de una serie de arcos y la incorporación de otros nuevos. Es por ello que, para determinar la variación que en el valor de f_2 producen estas inserciones y movimientos, y así poder chequear la restricción (15), únicamente se necesita calcular la suma de las distancias de los arcos que se añaden y restar la suma de las distancias de los arcos que se eliminan.

Finalmente, hay que indicar que la actualización de Dmx en el paso 4, ($Dmx = f_1(S) - 1$) se ha elegido de esta manera ya que tanto las distancias d , como los tiempos t son valores enteros, y por extensión, los objetivos f_1 y f_2 también lo son. Se escogen unidades suficientemente pequeñas que permitan que la discretización de los valores de distancias y tiempos se ajuste a la realidad. Además, desde un punto de vista computacional, las matrices enteras usan menos memoria y permiten alcanzar más rapidez durante la ejecución, si se compara con los valores reales.

10.- LOS EXPERIMENTOS COMPUTACIONALES

10.- LOS EXPERIMENTOS COMPUTACIONALES

10.1.- INTRODUCCIÓN

En este capítulo se muestran los resultados de diferentes pruebas computacionales realizadas a fin de chequear el funcionamiento del modelo propuesto en la sección 7.3. Para ello se emplean las tres formas siguientes:

- con software comercial Cplex,
- el método MOAMP y
- el método MHeC.

Además, se incluyen en el capítulo los experimentos para realizar los ajustes de parámetros de los procedimientos que componen estos métodos. También se incluyen los experimentos computacionales para analizar la eficacia de las estrategias propuestas en el Capítulo 8 para acelerar el chequeo de la factibilidad y del cálculo de las funciones objetivo.

Concretamente, tras esta introducción, en el segundo apartado se describen las instancias usadas en estos experimentos; en el tercero se especifican las pruebas realizadas para el ajuste de los parámetros incluidos en algunos de los procedimientos descritos; el cuarto apartado recoge las pruebas de evaluación de las estrategias de aceleración para el chequeo de la factibilidad y el cálculo de la función objetivo en los procedimientos **Constructive** y **TabuSearch**; en el quinto se muestra la evolución de los tiempos de computación usado por el Software comercial Cplex para resolver el modelo propuesto en instancias pequeñas y se incluye, también, una comparación de los resultados obtenidos por Cplex con los obtenidos por los métodos MOAMP y MHeC propuestos; en el sexto apartado se comparan los resultados obtenidos por los métodos MOAMP y MHeC en instancias de mayor tamaño; en el último apartado se presenta la aplicación del problema al caso real del traslado de pacientes entre las ciudades de Saltillo y Monterrey, en México.

Todas las pruebas han sido realizadas en un ordenador con procesador Intel i7 4790, 3.6GHz y 16 GB RAM. El lenguaje de programación para la implementación y compilación de los algoritmos ha sido Object Pascal con el entorno de desarrollo Rad Studio 10 Seattle. La versión de Cplex usada ha sido Cplex 12.6.3 versión académica.

10.2.- LA DESCRIPCIÓN DE LAS INSTANCIAS

Con el fin de realizar los experimentos computacionales, primeramente, se han generado un conjunto de instancias simuladas de la manera que se describe a continuación:

- Se consideran 15 tamaños, o valores de n , diferentes. Dichos tamaños se muestran en la Tabla - 3 que sigue. Para cada tamaño se generan $2n+1$ puntos correspondientes a las localizaciones del problema. El punto correspondiente al depósito 0 se coloca en el punto (500,500) y el resto de los puntos toman valores enteros de forma aleatoria en el cuadrado $[0,1000] \times [0,1000]$. A partir de estas localizaciones se obtienen las distancias $d_{ij}, \forall i, j \in V$ que se calculan como las distancias euclídeas redondeadas al entero más próximo. De igual forma se hace $t_{ij} = d_{ij}, \forall i, j \in V$.
- Se considera el valor de la capacidad máxima $Q = 50$.

- Para cada valor de n se generan diferentes vectores de valores q_i . Concretamente cada vector, definido por q_i siendo $i = 1, \dots, n$, toma un valor entero de forma aleatoria entre 1 y $Qmax$. Se consideran los siguientes valores de $Qmax = 5, 10, 25, 40$ y 50 . De esta forma se generan cinco vectores de valores q_i .
- Finalmente para cada valor de n , se consideran dos valores de m , o número de vehículos, y un valor de $Lmax$. Los valores de m y $Lmax$ correspondientes a cada valor de n se resumen en la Tabla - 3.

Recapitulando indicar que, para cada valor de n se considera una matriz de distancias, una matriz de tiempos, cinco vectores de valores q_i , dos valores de m y un valor de $Lmax$. También se considera un valor fijo de Q para todas las instancias. De esa forma, combinando los diferentes datos de entrada, se cuenta con 150 instancias ($15 \times 1 \times 1 \times 5 \times 2 \times 1$), esto es 10 por cada valor de n . La Tabla - 3, como se ha indicado anteriormente, muestra los valores de n con sus valores de m y $Lmax$ correspondientes. Cada valor de $Lmax$ se ha calculado para asegurar que la instancia tenga soluciones factibles.

Tabla - 3: Valores de n , m y $Lmax$ para generación de las instancias.

n	m		$Lmax$
5	2	3	2400
6	2	3	3000
8	2	3	3600
10	3	5	3600
12	3	5	4500
15	3	5	4500
20	4	6	4500
30	5	7	5000
50	9	11	5400
70	12	15	5400
100	13	17	7200
150	19	24	7200
200	25	30	7200
300	30	40	9600
500	50	75	9600

Las instancias se denotan por $InstXXX_YY_ZZZ$, donde XXX es el valor de n , YY el valor de $Qmax$ y ZZZ el valor de m . Así por ejemplo, $Inst070_25_015$ hace referencia a la instancia de tamaño $n = 70$, $Qmax = 25$ y $m = 15$. Y, siguiendo la tabla anterior, en este caso $Lmax = 5400$.

10.3.- EL AJUSTE DE LOS PARÁMETROS

Se detallan, a continuación, las pruebas computacionales que se han realizado para el ajuste de los parámetros presentes en algunos de los procedimientos que forman parte de los métodos MOAMP y MHeC planteados en este estudio. Más en concreto, se van a analizar parámetros de los procedimientos **TabuSearch**, **GeneradorSocial** y **GeneradorCostes**.

Es importante señalar que, para llevar a cabo este análisis de parámetros se ha utilizado el subconjunto de 15 instancias que se muestra en la Tabla - 4 siguiente. Este subconjunto de instancias, al que se denomina *TrainingSet*, ha sido seleccionado para representar cada uno de los quince tamaños de problema distintos, representando tres veces cada uno de los cinco valores distintos asignados a Q_{max} y alternando el mínimo y el máximo de los valores que presenta m en la Tabla - 3 para cada uno de los quince valores de n .

Tabla - 4: Subconjunto TrainingSet de instancias utilizadas en el ajuste de parámetros.

1	<i>Inst005_25_002</i>	6	<i>Inst015_50_005</i>	11	<i>Inst100_40_013</i>
2	<i>Inst006_40_003</i>	7	<i>Inst020_25_004</i>	12	<i>Inst150_50_024</i>
3	<i>Inst008_50_002</i>	8	<i>Inst030_40_007</i>	13	<i>Inst200_25_025</i>
4	<i>Inst010_25_005</i>	9	<i>Inst050_50_009</i>	14	<i>Inst300_40_040</i>
5	<i>Inst012_40_003</i>	10	<i>Inst070_25_015</i>	15	<i>Inst500_50_050</i>

10.3.1.- El parámetro *tenure* del procedimiento **TabuSearch**

Los valores del parámetro *tenure* del procedimiento **TabuSearch** se analizan con respecto a las dos funciones objetivo del problema, de forma análoga aunque independiente.

10.3.1.1.- El *tenure* asociado a la función objetivo social

El proceso llevado a cabo para el ajuste del parámetro *tenure* es el que se describe a continuación:

- Se consideran los siguientes valores *tenure*: $tenure = n/2, n, 2n$ y $5n$.
- Se ejecuta el método **TabuSearch** para cada uno de los valores anteriores combinado con cada instancia de la Tabla - 4, resultando un total de 60 ejecuciones.
- La solución inicial que lee **TabuSearch** es la obtenida por el procedimiento **Constructive**.
- Se considera, en todos los casos, el procedimiento aplicado a la función objetivo social f_1 y el valor de $maxitertabu = 10n$.
- Para simplificar la descripción de los resultados obtenidos se definen $\forall ins \in TrainingSet$ y para cada valor *tenure*:

$value(ins, tenure) =$ valor de la solución obtenida por el procedimiento **TabuSearch** al ejecutarse para la instancia *ins* y el correspondiente valor *tenure*;

$bestvalue(ins) = \min\{ value(ins, tenure) : tenure = n/2, n, 2n \text{ y } 5n \}$ y

$Ratvalue(ins, tenure) = value(ins, tenure)/bestvalue(ins)$.

De ello se coliga que, cuanto menor es el valor de *Ratvalue* mejor es el valor del parámetro *tenure* para la instancia *ins* ejecutada. Es más, un valor de *Ratvalue* igual a 1 indicará que nos encontramos ante la mejor combinación para dicha instancia. Para cada valor de *tenure* se va a calcular el siguiente valor:

$$MeanRatvalue(tenure) = mean\{ Ratvalue(ins, tenure) : \forall ins \in TrainingSet \}$$

Entonces, un menor valor de *MeanRatvalue* indica un mejor valor del parámetro *tenure*. En la Tabla - 5 se muestran los valores de *MeanRatvalue* obtenidos para los valores de *tenure* considerados. De los resultados obtenidos, se observa que el mejor valor del parámetro *tenure* para el procedimiento **TabuSearch** con la función objetivo social corresponde a $2n$.

Tabla - 5: Valores de *MeanRatvalue* para distintos *tenure* en el procedimiento **TabuSearch** con f_1

<i>tenure</i>	$n/2$	n	$2n$	$5n$
<i>MeanRatvalue</i>	1,03294	1,02874	1,02631	1,02990

10.3.1.2.- El *tenure* asociado a la función objetivo económico

El mismo proceso llevado a cabo para el ajuste del parámetro *tenure* descrito en el sub-apartado anterior se realiza, pero aplicado esta vez a la función objetivo económico f_2 para todos los casos.

A continuación, en la Tabla - 6, se muestran los valores de *MeanRatvalue* obtenidos para los valores de *tenure* considerados en el procedimiento **TabuSearch** con la función objetivo económico. Observando los resultados alcanzados, se concluye que el mejor valor del parámetro *tenure* también corresponde a $2n$, como en el caso del ajuste realizado en relación a la función objetivo social.

Tabla - 6: Valores de *MeanRatvalue* para distintos *tenure* en el procedimiento **TabuSearch** con f_2

<i>tenure</i>	$n/2$	n	$2n$	$5n$
<i>MeanRatvalue</i>	1,04004	1,03141	1,02231	1,02798

10.3.2.- Los parámetros α y β del procedimiento **GeneradorSocial**

Una vez se ha ajustado el valor de *tenure* en el procedimiento **TabuSearch** con la función objetivo social ya se pueden analizar los valores de los parámetros α y β en el procedimiento **GeneradorSocial**.

- Concretamente se consideran los siguientes valores para ambos parámetros:

$$\alpha = 0, 0,5, 0,9, 0,99, 1$$

$$\beta = 0, 0,3, 0,5, 0,7, 1$$

- Se consideran todas las combinaciones de valores posibles, excepto para $\alpha = 0$. En este caso, cuando $\alpha = 0$, como se puede observar en el paso 7.a de la modificación del procedimiento **Constructive**, recogido en la sección 9.2.2, la “lista de candidatos” L' coincide con el conjunto L , independientemente del valor de β , y por tanto éste no tiene ninguna influencia. Por todo ello se consideran 21 combinaciones distintas de parámetros.

- Se ejecuta el procedimiento **GeneradorSocial** para cada una de las combinaciones y para cada una de las 15 instancias de la Tabla - 4, resultando un total de 315 ejecuciones.
- En todos los casos ejecutados se considera el valor de $maxitersocial = 10n$ y $maxitertabu = 10n$.
- Se define *Combinations* como el conjunto de las 21 combinaciones distintas de los parámetros α y β .
- Para la mejor interpretación de los resultados obtenidos se definen $\forall ins \in TrainingSet$ y $\forall com \in Combinations$:

$valueGS(ins, com)$ = valor de la solución obtenida por el procedimiento **GeneradorSocial** al ejecutarse para la instancia *ins* y el correspondiente valor *com*;

$bestvalueGS(ins) = \min\{valueGS(ins, com) : \forall com \in Combinations\}$,

$RatvalueGS(ins, com) = valueGS(ins, com)/bestvalueGS(ins)$ y

$MeanRatvalueGS(com) = \text{mean}\{RatvalueGS(ins, com) : \forall ins \in TrainingSet\}$.

Por consiguiente, un menor valor de *MeanRatvalueGS* indica un mejor valor de la combinación de los parámetro α y β ejecutada. En la Tabla - 7 se muestran los valores de *MeanRatvalueGS* obtenidos para todas las combinaciones consideradas.

Tabla - 7: Valores de *MeanRatvalueGS* para todas las combinaciones de α y β en el GeneradorSocial

	$\beta = 0$	$\beta = 0,3$	$\beta = 0,5$	$\beta = 0,7$	$\beta = 1$
$\alpha = 0$	1,06036				
$\alpha = 0,5$	1,06085	1,05582	1,05139	1,04616	1,04351
$\alpha = 0,9$	1,06058	1,02963	1,02452	1,01144	1,00657
$\alpha = 0,99$	1,05664	1,01150	1,01021	1,00680	1,00038
$\alpha = 1$	1,05426	1,01127	1,01048	1,00684	1,00385

Como se puede observar en la tabla anterior la mejor combinación se da para los valores de $\alpha = 0,99$ y $\beta = 1$, indicado en negrita. Además, para los diferentes valores de α se puede observar que a medida que crece el valor de β se obtienen mejores valores. Por tanto se puede concluir, que al menos en este caso, la penalización por frecuencias no es fuente de soluciones diversas de calidad.

10.3.3.- Los parámetros α y β del procedimiento **GeneradorCostes**

Un proceso similar se ha llevado a cabo para el ajuste de los valores de los parámetros α y β en el procedimiento **GeneradorCostes**, esta vez teniendo en cuenta el resultado obtenido para el valor de *tenure* en el procedimiento **TabuSearch** con la función objetivo económico.

A continuación, en la Tabla - 8, se muestran los valores de *MeanRatvalueGC* obtenidos para todas las combinaciones consideradas.

Tabla - 8: Valores de *MeanRatvalueGC* para todas las combinaciones de α y β en el *GeneradorCostes*

	$\beta = 0$	$\beta = 0,3$	$\beta = 0,5$	$\beta = 0,7$	$\beta = 1$
$\alpha = 0$	1,06080				
$\alpha = 0,5$	1,06034	1,05664	1,05043	1,04733	1,04212
$\alpha = 0,9$	1,06010	1,03052	1,02342	1,01287	1,00545
$\alpha = 0,99$	1,05756	1,01234	1,01198	1,00780	1,00192
$\alpha = 1$	1,05324	1,01112	1,01102	1,00591	1,00416

Se observa, en la tabla anterior, que el mejor resultado se obtiene para la combinación de parámetros $\alpha = 0,99$ y $\beta = 1$, resaltado en negrita. Al igual que en el caso anterior, se puede concluir, que en el procedimiento *GeneradorCostes*, la penalización por frecuencias no es fuente de soluciones diversas de calidad.

10.4.- LA EVALUACIÓN DE LAS ESTRATEGIAS DE ACELERACIÓN

En los siguientes párrafos se describen las pruebas realizadas para evaluar el efecto que en el tiempo de computación tienen las estrategias propuestas en el Capítulo 8. Se recuerda que con dichas estrategias se busca acelerar el chequeo de la factibilidad de las inserciones y de los movimientos descritos en las secciones 7.4.1 y 7.4.2, así como para calcular más rápidamente las funciones objetivo tras producirse una inserción o movimiento, más concretamente la función objetivo social f_1 .

Estas estrategias, como se ha visto anteriormente, definen y emplean una serie de variables auxiliares con las que se puede chequear la factibilidad y el cálculo de la función objetivo en un número reducido de operaciones. En muchos casos este número de operaciones no depende del tamaño del problema.

Las operaciones realizadas con estas variables globales, que son por ejemplo: el cálculo inicial, la actualización, el chequeo de la factibilidad y el cálculo de f_1 , suponen un desarrollo de código adicional.

La alternativa más sencilla es no usar todas estas variables auxiliares, y tan sólo utilizar una variable auxiliar para grabar en ella la solución a la que daría lugar la inserción o el movimiento en cuestión. O también, si se prefiere, se podría usar una o dos variables auxiliares para grabar en ellas la ruta o rutas resultantes afectadas al hacer esa inserción o ese movimiento. Ello permitiría analizar las rutas afectadas resultantes y con ello se podría determinar su factibilidad e, igualmente, la de la inserción o movimiento correspondiente. Posteriormente, y en caso de factibilidad, se podría determinar el valor de las funciones objetivo. Esto es relativamente sencillo desde un punto de vista de programación y requiere de mucho menos código.

Por todo ello, se pretende analizar la evolución de los tiempos de computación en ambos casos, es decir, sin y con implementación de estrategias de aceleración. Con dicha finalidad, se han ejecutado los procedimientos *Constructive* y *TabuSearch* para todas las instancias descritas en la sección 10.1, sin y con estrategias de aceleración. La Tabla - 9 muestra los tiempos de computación, en segundos, obtenidos para cada una de las 150 instancias analizadas. Se quiere indicar que la función objetivo usada es f_1 . Por otra parte, con el fin de evitar tiempos de computación excesivos, se ha tomado $maxitertabu = 0$, es decir, que *TabuSearch* se comporta como una búsqueda local.

Tabla - 9: Tiempos computacionales en segundos para la evaluación de las estrategias de aceleración

n	Qmax	m	Const. Sin A.	Constructive Con A.	TabuS. Sin A.	TabuSearch Con A.
5	5	2	0,001	0,000	0,000	0,000
5	5	3	0,000	0,000	0,001	0,000
5	10	2	0,000	0,001	0,001	0,000
5	10	3	0,001	0,000	0,000	0,000
5	25	2	0,000	0,000	0,001	0,000
5	25	3	0,000	0,000	0,001	0,001
5	40	2	0,000	0,000	0,001	0,000
5	40	3	0,000	0,000	0,001	0,000
5	50	2	0,000	0,000	0,000	0,001
5	50	3	0,000	0,000	0,001	0,000
6	5	2	0,000	0,000	0,002	0,001
6	5	3	0,001	0,000	0,001	0,000
6	10	2	0,000	0,000	0,001	0,001
6	10	3	0,000	0,000	0,002	0,000
6	25	2	0,000	0,000	0,001	0,001
6	25	3	0,000	0,000	0,002	0,000
6	40	2	0,001	0,000	0,001	0,000
6	40	3	0,000	0,001	0,001	0,000
6	50	2	0,000	0,000	0,002	0,000
6	50	3	0,001	0,000	0,001	0,000
8	5	2	0,000	0,000	0,002	0,001
8	5	3	0,000	0,000	0,003	0,000
8	10	2	0,000	0,000	0,002	0,001
8	10	3	0,000	0,000	0,003	0,001
8	25	2	0,001	0,000	0,002	0,000
8	25	3	0,000	0,000	0,002	0,001
8	40	2	0,000	0,000	0,002	0,001
8	40	3	0,000	0,000	0,002	0,001
8	50	2	0,000	0,000	0,002	0,001
8	50	3	0,000	0,000	0,001	0,001
10	5	3	0,000	0,001	0,002	0,000
10	5	5	0,000	0,000	0,002	0,000
10	10	3	0,000	0,000	0,002	0,000
10	10	5	0,000	0,000	0,002	0,000
10	25	3	0,000	0,000	0,002	0,000
10	25	5	0,000	0,000	0,002	0,000
10	40	3	0,000	0,000	0,002	0,000
10	40	5	0,000	0,000	0,002	0,000
10	50	3	0,000	0,000	0,002	0,000
10	50	5	0,000	0,000	0,001	0,000
12	5	3	0,000	0,000	0,003	0,000
12	5	5	0,000	0,000	0,003	0,001
12	10	3	0,001	0,000	0,003	0,000
12	10	5	0,000	0,000	0,003	0,000
12	25	3	0,000	0,000	0,003	0,000
12	25	5	0,000	0,000	0,003	0,001
12	40	3	0,000	0,000	0,002	0,001
12	40	5	0,000	0,000	0,003	0,000
12	50	3	0,000	0,000	0,002	0,001
12	50	5	0,000	0,000	0,002	0,001

n	Qmax	m	Const. Sin A.	Constructive Con A.	TabuS. Sin A.	TabuSearch Con A.
15	5	3	0,001	0,000	0,006	0,000
15	5	5	0,000	0,000	0,006	0,001
15	10	3	0,000	0,000	0,006	0,001
15	10	5	0,000	0,000	0,006	0,001
15	25	3	0,001	0,000	0,005	0,000
15	25	5	0,001	0,001	0,006	0,000
15	40	3	0,000	0,000	0,005	0,001
15	40	5	0,001	0,000	0,005	0,000
15	50	3	0,000	0,000	0,006	0,000
15	50	5	0,000	0,000	0,005	0,001
20	5	4	0,000	0,001	0,016	0,001
20	5	6	0,000	0,000	0,015	0,002
20	10	4	0,001	0,000	0,015	0,001
20	10	6	0,001	0,000	0,015	0,001
20	25	4	0,001	0,000	0,016	0,001
20	25	6	0,001	0,001	0,015	0,001
20	40	4	0,001	0,000	0,013	0,001
20	40	6	0,001	0,000	0,012	0,001
20	50	4	0,001	0,000	0,012	0,001
20	50	6	0,000	0,000	0,012	0,001
30	5	5	0,001	0,001	0,052	0,002
30	5	7	0,001	0,001	0,044	0,003
30	10	5	0,002	0,001	0,050	0,002
30	10	7	0,001	0,000	0,044	0,003
30	25	5	0,001	0,000	0,050	0,003
30	25	7	0,001	0,000	0,046	0,003
30	40	5	0,001	0,000	0,044	0,003
30	40	7	0,001	0,000	0,038	0,003
30	50	5	0,001	0,000	0,042	0,003
30	50	7	0,001	0,000	0,037	0,002
50	5	9	0,003	0,003	0,176	0,006
50	5	11	0,003	0,003	0,179	0,007
50	10	9	0,003	0,003	0,179	0,007
50	10	11	0,003	0,002	0,176	0,008
50	25	9	0,003	0,002	0,175	0,007
50	25	11	0,003	0,003	0,175	0,007
50	40	9	0,003	0,001	0,156	0,006
50	40	11	0,003	0,001	0,157	0,008
50	50	9	0,003	0,002	0,145	0,006
50	50	11	0,003	0,002	0,142	0,007
70	5	12	0,007	0,006	0,897	0,018
70	5	15	0,006	0,006	0,446	0,016
70	10	12	0,006	0,005	0,901	0,017
70	10	15	0,006	0,005	0,449	0,016
70	25	12	0,006	0,005	0,902	0,018
70	25	15	0,006	0,006	0,458	0,016
70	40	12	0,006	0,004	0,764	0,016
70	40	15	0,006	0,005	0,394	0,015
70	50	12	0,006	0,004	0,710	0,016
70	50	15	0,006	0,003	0,360	0,014

n	Qmax	m	Const. Sin A.	Constructive Con A.	TabuS. Sin A.	TabuSearch Con A.
100	5	13	0,019	0,014	1,716	0,040
100	5	17	0,019	0,013	1,686	0,046
100	10	13	0,019	0,013	1,716	0,042
100	10	17	0,019	0,014	1,701	0,044
100	25	13	0,019	0,013	1,675	0,038
100	25	17	0,018	0,014	1,700	0,045
100	40	13	0,017	0,010	1,411	0,036
100	40	17	0,017	0,010	1,390	0,042
100	50	13	0,016	0,009	1,297	0,034
100	50	17	0,017	0,009	1,320	0,039
150	5	19	0,123	0,109	5,524	0,089
150	5	24	0,131	0,118	5,478	0,101
150	10	19	0,121	0,109	5,444	0,084
150	10	24	0,131	0,118	5,466	0,100
150	25	19	0,121	0,111	5,350	0,083
150	25	24	0,134	0,119	5,385	0,099
150	40	19	0,119	0,101	4,603	0,077
150	40	24	0,128	0,111	4,665	0,094
150	50	19	0,117	0,098	3,970	0,071
150	50	24	0,126	0,107	4,016	0,090
200	5	25	0,258	0,241	11,079	0,128
200	5	30	0,286	0,27	11,075	0,181
200	10	25	0,259	0,241	11,059	0,128
200	10	30	0,287	0,269	11,009	0,180
200	25	25	0,264	0,243	10,881	0,128
200	25	30	0,284	0,270	10,732	0,184
200	40	25	0,255	0,227	9,033	0,114
200	40	30	0,282	0,248	8,834	0,164
200	50	25	0,255	0,217	8,055	0,112
200	50	30	0,277	0,242	7,982	0,161
300	5	30	0,877	0,817	58,558	0,653
300	5	40	0,973	0,914	59,744	0,796
300	10	30	0,890	0,850	58,700	0,671
300	10	40	0,950	0,913	59,043	0,774
300	25	30	0,877	0,818	57,150	0,653
300	25	40	0,943	0,894	57,573	0,773
300	40	30	0,861	0,762	44,740	0,594
300	40	40	0,935	0,834	45,591	0,710
300	50	30	0,872	0,752	39,601	0,566
300	50	40	0,917	0,799	39,830	0,684
500	5	50	4,008	4,063	239,794	1,921
500	5	75	4,353	4,398	241,914	2,468
500	10	50	4,038	4,101	239,531	1,918
500	10	75	4,374	4,436	242,082	2,473
500	25	50	4,005	4,110	234,364	1,912
500	25	75	4,387	4,437	236,884	2,462
500	40	50	4,026	3,950	186,497	1,766
500	40	75	4,357	4,279	194,910	2,299
500	50	50	4,010	3,872	159,005	1,671
500	50	75	4,342	4,194	160,861	2,202

10.4.1.- Constructive Sin y Con Aceleración

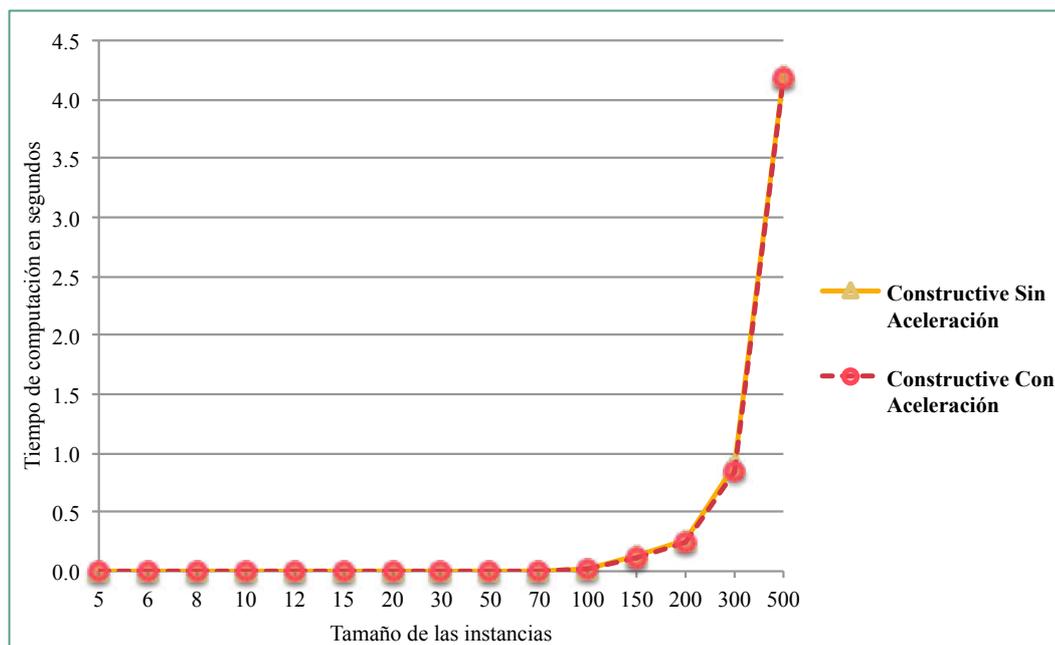
A continuación, Tabla - 10, se muestran los resultados de los tiempos de computación empleados por el procedimientos **Constructive** “Sin y Con Aceleración” agrupados por tamaño de instancia. En concreto, se muestran: mínimo, máximo, desviación típica y media para cada uno de ellos.

Tabla - 10: Resumen de resultados de la comparativa Constructive por tamaños (tiempo computación en s.).

Instancias Tamaño	Mínimo		Máximo		Desviación Típica		Media	
	Sin Aceleración	Con Aceleración	Sin Aceleración	Con Aceleración	Sin Aceleración	Con Aceleración	Sin Aceleración	Con Aceleración
5	0,000	0,000	0,001	0,001	0,000	0,000	0,000	0,000
6	0,000	0,000	0,001	0,001	0,000	0,000	0,000	0,000
8	0,000	0,000	0,001	0,000	0,000	0,000	0,000	0,000
10	0,000	0,000	0,000	0,001	0,000	0,000	0,000	0,000
12	0,000	0,000	0,001	0,000	0,000	0,000	0,000	0,000
15	0,000	0,000	0,001	0,001	0,001	0,000	0,000	0,000
20	0,000	0,000	0,001	0,001	0,000	0,000	0,001	0,000
30	0,001	0,000	0,002	0,001	0,000	0,000	0,001	0,000
50	0,003	0,001	0,003	0,003	0,000	0,001	0,003	0,002
70	0,006	0,003	0,007	0,006	0,000	0,001	0,006	0,005
100	0,016	0,009	0,019	0,014	0,001	0,002	0,018	0,012
150	0,117	0,098	0,134	0,119	0,006	0,007	0,125	0,110
200	0,255	0,217	0,287	0,270	0,014	0,018	0,271	0,247
300	0,861	0,752	0,973	0,914	0,039	0,058	0,910	0,835
500	4,008	3,872	4,387	4,437	0,178	0,200	4,195	4,184

Se observa que los tiempos medios en el procedimiento que incluye estrategias de aceleración son ligeramente menores, aunque muy similares, que cuando no se usan. La Figura - 21 muestra la comparativa en la evolución de los tiempos medios de computación según el tamaño de la instancia.

Figura - 21: Comparativa de tiempos medios de computación del procedimiento Constructive por tamaño



Con el fin de analizar si los tiempos empleados por el procedimiento **Constructive** Con Aceleración son significativamente menores estadísticamente hablando, se ha realizado un test para cada tamaño de instancia por el que se puede comprobar si el valor medio de la diferencia de tiempos de computación, (Sin Aceleración - Con Aceleración), es significativamente mayor que 0. En la Tabla - 11 se muestran los resultados del test realizado, concretamente la media de la diferencia de tiempos, su desviación típica, el valor del estadístico *t* y la cola de probabilidad. Se señalan las diferencias significativas para los niveles de 0,05 (“**”) y 0,10 (“*”).

Tabla - 11: Test estadístico sobre el valor medio de las diferencias de Constructive por tamaños

Instancias	Diferencia (s.)	Desviación (s.)	Estadístico t	Probabilidad
Tamaño				
5	0,000	0,001	0,557	0,591
6	0,000	0,001	1,000	0,343
8	0,000	0,000	1,000	0,343
10	0,000	0,000	-1,000	0,343
12	0,000	0,000	1,000	0,343
15	0,000	0,000	1,964	0,081*
20	0,001	0,001	2,236	0,052*
30	0,001	0,000	6,000	0,000**
50	0,001	0,001	3,207	0,011**
70	0,001	0,001	4,129	0,003**
100	0,006	0,001	16,112	0,000**
150	0,015	0,003	15,169	0,000**
200	0,024	0,009	8,410	0,000**
300	0,074	0,032	7,338	0,000**
500	0,010	0,089	0,375	0,717

A partir de los datos plasmados en la tabla anterior, se puede concluir que la incorporación de las estrategias de aceleración planteadas al procedimiento **Constructive**, en general no suponen una gran reducción en los tiempos de computación obtenidos. De forma más detallada, aplicar las estrategias en el proceso analizado produce una reducción significativa de los tiempos de computación en las instancias más pequeñas, hasta tamaño 20 aproximadamente. En las instancias de tamaños entre 30 y 300 presenta una cierta relevancia y significatividad. Sin embargo, en las instancias de mayor tamaño, tamaño 500, las diferencias de los tiempos en uno y otro escenario vuelven a presentar valores de significatividad bajos.

La conclusión, por tanto, en cuanto a la relevancia en la incorporación de estrategias de aceleración específicas y adaptadas al problema dentro del procedimiento **Constructive** es que es prácticamente igual de relevante para los distintos tamaños de las instancias, siendo para todos ellos moderadamente baja. No obstante, la significación de la reducción del tiempo de computación por el uso de las estrategias de aceleración, en este caso concreto, sí se muestra ligeramente influenciado por el tamaño de la instancia que se analiza.

10.4.2.- TabuSearch Sin y Con Aceleración

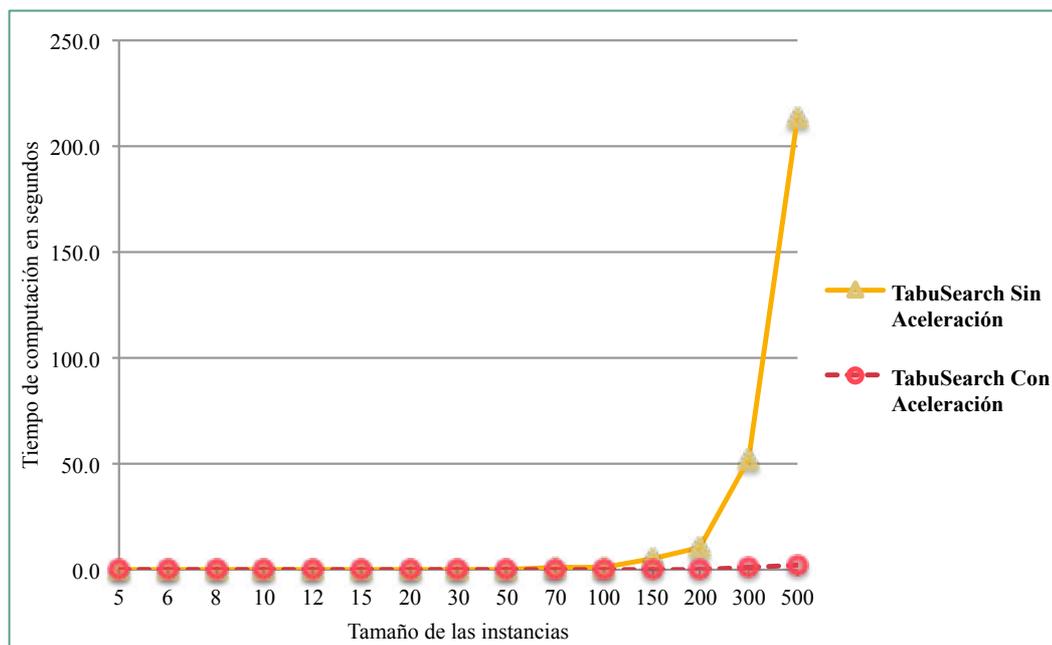
Es a continuación, en la Tabla - 12, donde se muestran los resultados de los tiempos de computación empleados por los procedimientos **TabuSearch** “Sin y Con Aceleración” agrupados por tamaño de las instancias. En la tabla se muestran: mínimo, máximo, desviación típica y media para cada grupo.

Tabla - 12: Resumen de resultados de la comparativa TabuSearch por tamaños (tiempo computación en s.).

Instancias Tamaño	Mínimo		Máximo		Desviación Típica		Media	
	Sin Aceleración	Con Aceleración	Sin Aceleración	Con Aceleración	Sin Aceleración	Con Aceleración	Sin Aceleración	Con Aceleración
5	0,000	0,000	0,001	0,001	0,000	0,000	0,001	0,000
6	0,001	0,000	0,002	0,001	0,001	0,000	0,001	0,000
8	0,001	0,000	0,003	0,001	0,001	0,000	0,002	0,001
10	0,001	0,000	0,002	0,000	0,000	0,000	0,002	0,000
12	0,002	0,000	0,003	0,001	0,000	0,001	0,003	0,001
15	0,005	0,000	0,006	0,001	0,001	0,001	0,006	0,001
20	0,012	0,001	0,016	0,002	0,002	0,000	0,014	0,001
30	0,037	0,002	0,052	0,003	0,005	0,000	0,045	0,003
50	0,142	0,006	0,179	0,008	0,014	0,001	0,166	0,007
70	0,360	0,014	0,902	0,018	0,228	0,001	0,628	0,016
100	1,297	0,034	1,716	0,046	0,181	0,004	1,561	0,041
150	3,970	0,071	5,524	0,101	0,622	0,010	4,990	0,089
200	7,982	0,112	11,079	0,184	1,330	0,029	9,974	0,148
300	39,601	0,566	59,744	0,796	8,503	0,077	52,053	0,687
500	159,005	1,671	242,082	2,473	34,626	0,307	213,584	2,109

Se observa que los tiempos medios en el procedimiento que incluye estrategias de aceleración son considerablemente menores, con una diferencia que crece casi exponencialmente con el tamaño. La Figura - 22 muestra la evolución de los tiempos medios de computación según el tamaño de la instancia.

Figura - 22: Comparativa de tiempos medios de computación del procedimiento TabuSearch por tamaño



De igual forma, se ha realizado un test de significatividad con el fin de analizar si los tiempos empleados por el procedimiento **TabuSearch** Con Aceleración son significativamente menores estadísticamente hablando. Por medio del test agrupado se puede comprobar si el valor medio de la diferencia de tiempos de computación, (Sin Aceleración - Con Aceleración), es significativamente mayor que 0. En la Tabla - 13 se muestran los resultados del test realizado, concretamente la media de la diferencia de tiempos, su desviación típica, el valor del estadístico *t* y la cola de probabilidad. Se señalan las diferencias significativas para los niveles de 0,05 (“**”) y 0,10 (“*”).

Tabla - 13: Test estadístico sobre el valor medio de las diferencias de TabuSearch por tamaños

Instancias	Diferencia (s.)	Desviación (s.)	Estadístico t	Probabilidad
Tamaño				
5	0,001	0,001	2,236	0,052*
6	0,001	0,001	4,714	0,001**
8	0,001	0,001	4,993	0,001**
10	0,002	0,000	19,000	0,000**
12	0,002	0,001	7,571	0,000**
15	0,005	0,001	21,857	0,000**
20	0,013	0,002	25,174	0,000**
30	0,042	0,005	25,937	0,000**
50	0,159	0,014	35,008	0,000**
70	0,612	0,227	8,526	0,000**
100	1,521	0,179	26,932	0,000**
150	4,901	0,617	25,125	0,000**
200	9,826	1,322	23,497	0,000**
300	51,366	8,457	19,206	0,000**
500	211,475	34,500	19,384	0,000**

Interpretando los datos presentados en la tabla anterior, se puede concluir que la incorporación de las estrategias de aceleración adaptadas al procedimiento **TabuSearch** suponen, en general, una importante reducción en los tiempos de computación obtenidos. De forma más detallada, introducir las estrategias de aceleración produce una reducción que se puede considerar relativamente significativa de los tiempos de computación en las instancias más pequeñas, de tamaño 5, al presentar un nivel de significación que se haya en un valor límite. Mientras que, para el resto de los tamaños de las instancias analizadas presenta de moderada a gran relevancia y elevada significatividad.

La conclusión, por tanto, en lo que respecta a la relevancia en la incorporación de estrategias de aceleración específicas y adaptadas al problema dentro del procedimiento **TabuSearch** es que crece en función del tamaño de las instancias.

Por todo lo anterior, y a pesar de que la incorporación de estrategias de aceleración en el método de resolución propuesto supone un desarrollo de código adicional, se considera que éstas agilizan considerablemente la obtención de resultados, sobretodo para **TabuSearch** y a la hora de resolver problemas de gran tamaño, que son precisamente los más exigentes con respecto a tiempos de cálculo. Las estrategias de aceleración, por tanto, se han incluido en ambos procedimientos para ejecutar todos los experimentos computacionales de este trabajo, los cuales se detallan en las secciones que siguen.

10.5.- LA RESOLUCIÓN COMPARADA EN INSTANCIAS PEQUEÑAS CON CPLEX

En esta sección se muestra la evolución de los tiempos de computación para obtener el frente de Pareto óptimo usando el software comercial Cplex y el modelo propuesto en el Capítulo 7 en instancias de diferente tamaño. También se comparan los resultados obtenidos por Cplex, con los obtenidos por los métodos MOAMP y MHeC. La estrategia usada para obtener los puntos del frente de Pareto óptimo mediante Cplex ha sido el método ϵ -constraint, es decir, un método análogo al método MHeC descrito en el Capítulo 9, pero resolviendo el problema *MonoSocial* de forma exacta mediante Cplex.

La Tabla 14 muestra los resultados medios obtenidos por Cplex para cada valor de n . Concretamente, se muestran las medias de los tiempos de computación en segundos (T.C.) y el número de soluciones pertenecientes al frente de Pareto ($|P^*|$).

Tabla - 14: Evolución de Cplex para instancias pequeñas

n	T.C.	$ P^* $
5	9,305	7,50
6	150,417	6,00

No se reportan los resultados de las instancias correspondientes a $n > 6$ ya que, en las instancias correspondientes a $n = 8$, una vez transcurridos 28.800 segundos (8 horas) se detectó que ninguna de ellas había concluido. En consecuencia, se entiende que $n = 6$ es el máximo tamaño de instancias, de este modelo biobjetivo, que pueden resolverse de forma exacta.

Para las instancias anteriores ($n \leq 6$) se muestran, en la Tabla 15 los resultados medios (para cada valor de n) de tres métricas para comparar los resultados obtenidos por los métodos MOAMP y MHeC. Así mismo, también se indica el tiempo medio de computación en segundos de cada uno de los métodos.

La primera de las métricas, denominada *Cardinalidad*, se recoge por Martí et al. (2015). En este caso, se define como el número de soluciones obtenidas por cada método que forman parte del frente de Pareto (P^*).

La segunda métrica se basa en el concepto de *Cubrimiento* de los dos conjuntos, que se desarrolla en el trabajo de Zitzler y Thiele (1999). A través de ella, se compara el porcentaje de un conjunto que es dominado por otro conjunto. En concreto, si $R(X, Y)$ es el porcentaje de soluciones encontradas por el método Y que están dominadas por una solución obtenida por el método de X para una instancia determinada, entonces, se define:

$$dif = R(MOAMP, MHeC) - R(MHeC, MOAMP)$$

En consecuencia, los valores positivos de dif indican un mayor dominio de las soluciones obtenidas por MOAMP sobre aquellas obtenidas empleando MHeC. Por el contrario, valores negativos indican mayor dominancia de las soluciones obtenidas por MHeC. Cuanto más alejado de 0 se encuentre el valor dif , mayor se considera el grado de dominancia, ya sea en una dirección u otra.

Por último, la tercera métrica se conoce como *Hipervolumen*, Zitzler y Thiele (1999). Para cada conjunto de soluciones, este indicador mide el porcentaje de volumen de la región que está dominada por este conjunto, en el espacio de la función objetivo. El porcentaje se calcula con respecto al hipercubo que tiene por extremos los puntos ideal y anti-ideal. El punto ideal y anti-ideal, se obtienen mediante el

cálculo de los mínimos y los máximos valores para cada función objetivo de entre todas las soluciones que forman los conjuntos considerados en el análisis. En los trabajos de Beume et al (2009) y de Bradstreet (2011) se encuentra recogido, con más detalle, cómo realizar el cálculo del *Hipervolumen*.

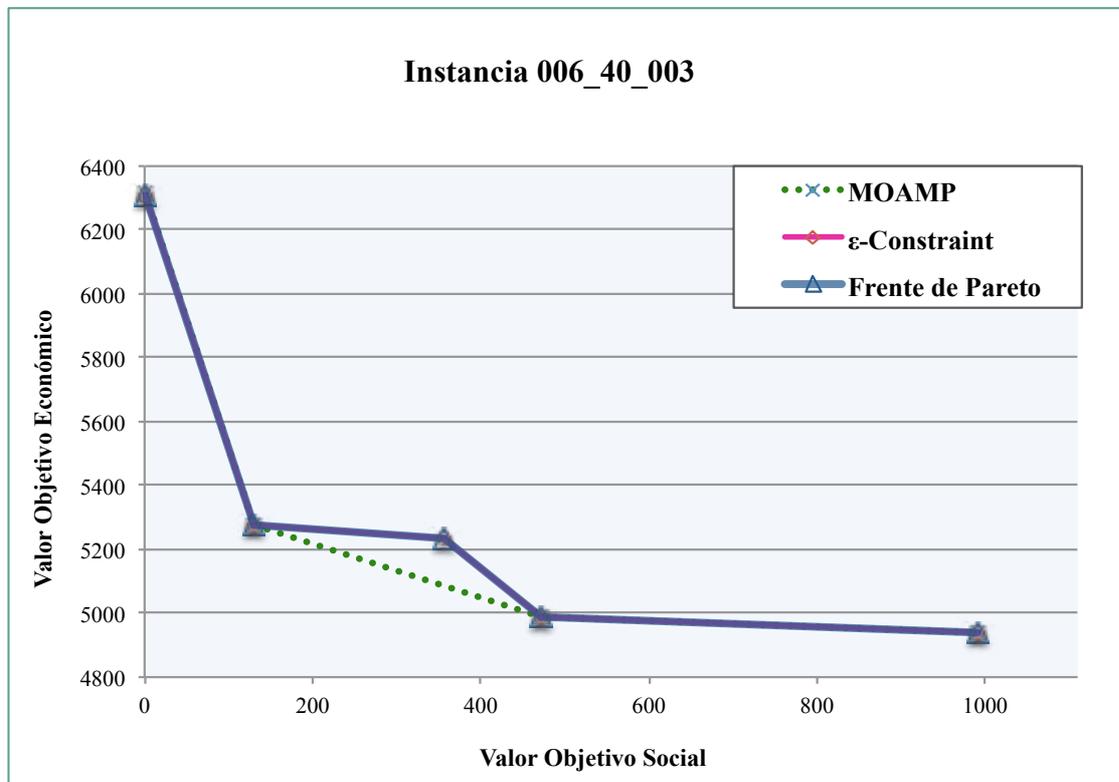
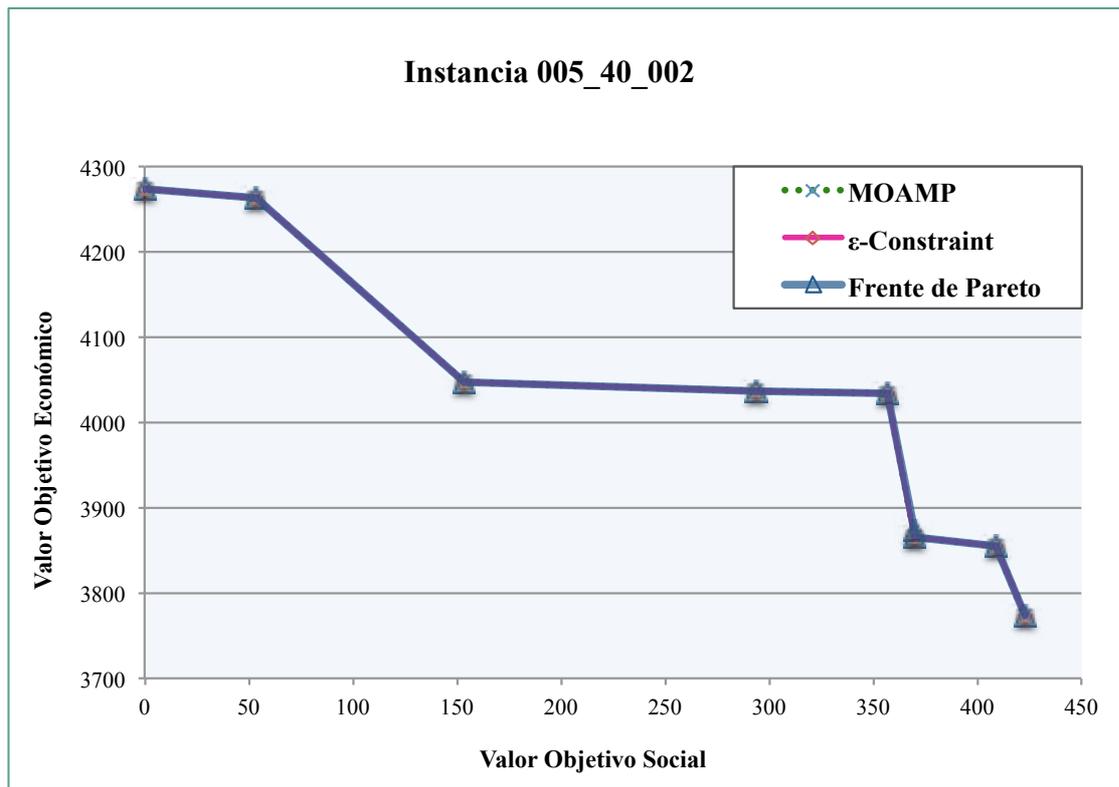
Tabla - 15: Métricas comparativas de MOAMP y MHeC en instancias de tamaño pequeño

n	T.C.		Cardinalidad		Cubrimiento <i>dif</i>	Hipervolumen	
	MOAMP	MHeC	MOAMP	MHeC		MOAMP	MHeC
5	0,053	0,010	93.39 %	97.92 %	0,049	0,444	0,441
6	0,039	0,018	55.85 %	48.06 %	0,420	0,772	0,716

De la tabla anterior se puede concluir que, al menos en las instancias de tamaño pequeño, el método MOAMP consigue mejores resultados que el método MHeC. Fijándose en las instancias de tamaño 5 se observan resultados muy similares: el valor de *Cardinalidad* obtenido por el método MHeC es ligeramente mejor que MOAMP (97% frente a 93%), mientras que MOAMP obtiene levemente mejores resultados en *Cubrimiento* (+0,049) y en *Hipervolumen* (0,444 frente a 0,441). Por otra parte, en las instancias de tamaño 6 el método MOAMP obtiene claramente mejores resultados en las tres métricas: 55% frente a 48% en *Cardinalidad*, +0,420 en *Cubrimiento* y 0,772 frente a 0,716 en *Hipervolumen*.

No obstante, el método MHeC usa ligeramente menos tiempo de computación que el método MOAMP. Cabe destacar que, ambos métodos usan mucho menos tiempo de computación que el empleado por Cplex. Así, a modo de ejemplo, los métodos heurísticos emplean para $n = 6$ apenas 0,039 segundos (MOAMP) y 0,018 segundos (MHeC), mientras que Cplex usa más de 150 segundos. Consecuentemente, los métodos heurísticos, como se mostrará en el siguiente sección, podrían permitir solucionar problemas de mucho mayor tamaño. Por su parte, no parece que se pueda garantizar resolver de forma exacta instancias de tamaño mayor que $n = 6$ en un tiempo aceptable.

A continuación se muestran, como ejemplo, una serie de gráficas correspondientes a diferentes instancias analizadas, una por tamaño de n ($n \leq 6$), con el frente de Pareto obtenido por Cplex y los conjuntos de soluciones no dominadas obtenidos por los métodos MOAMP y MHeC.



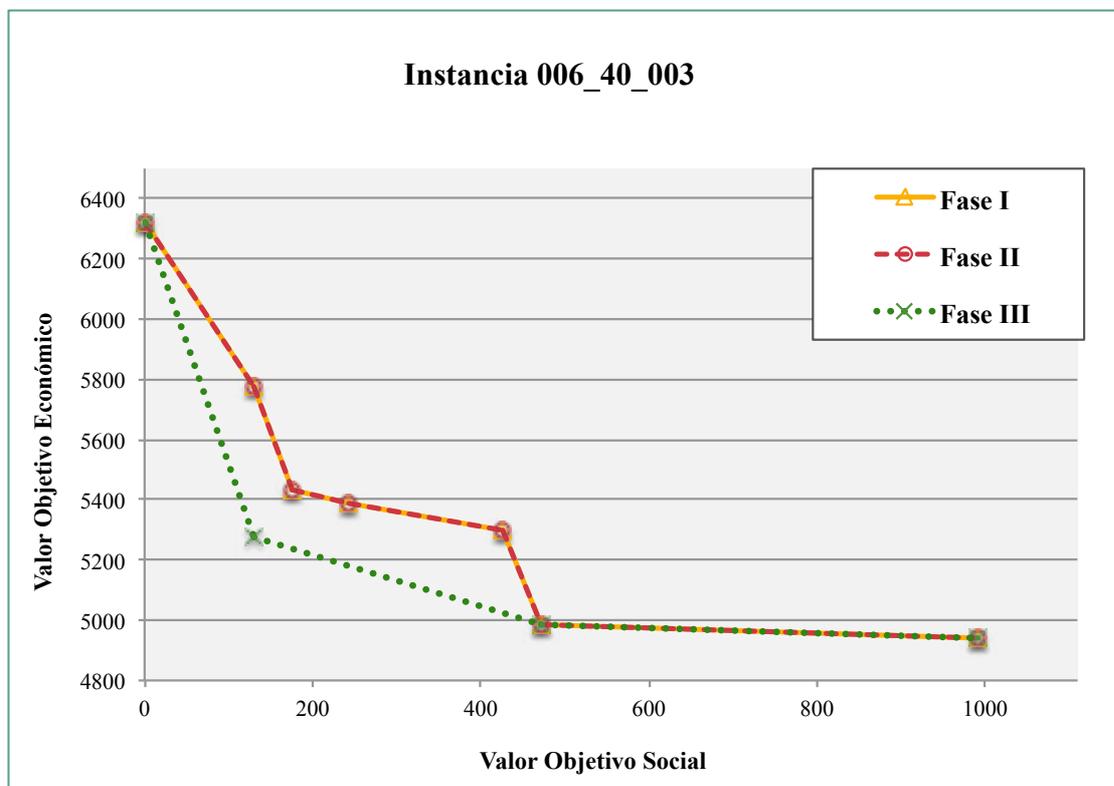
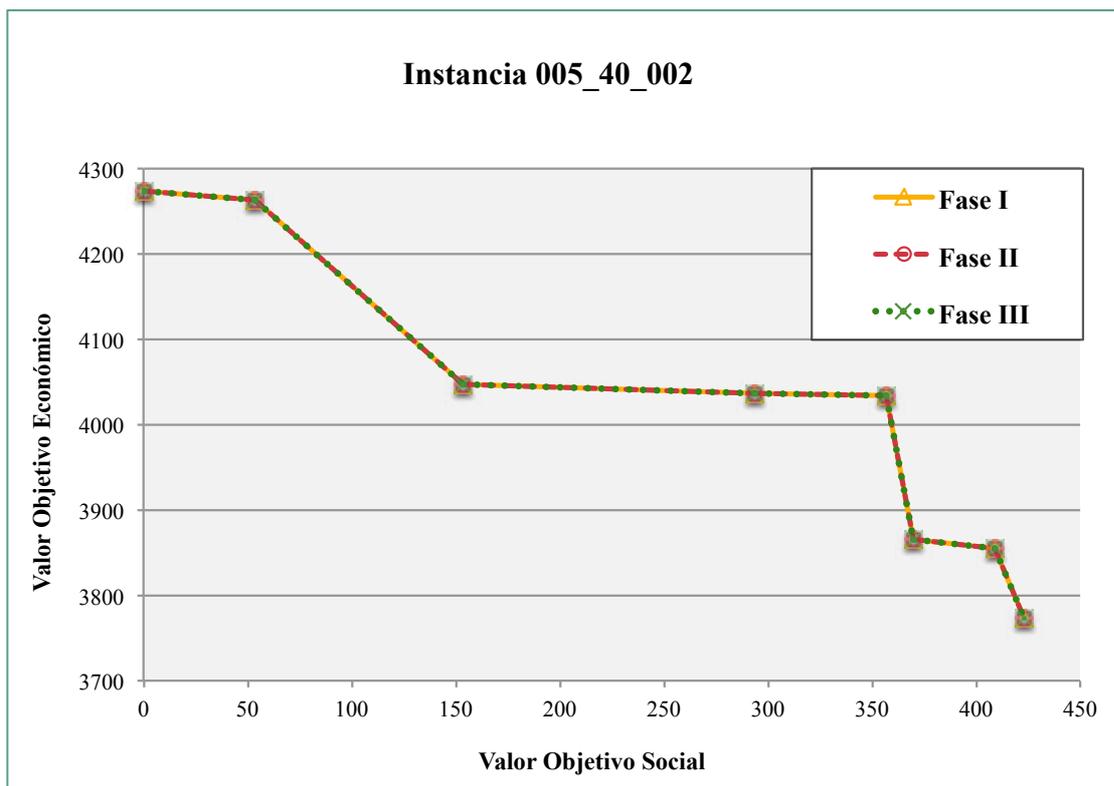
De las gráficas anteriores se observa que, en las instancia correspondiente a $n = 5$ los dos métodos heurísticos obtienen conjunto de soluciones que coinciden con el frente de Pareto. En la instancia correspondiente a $n = 6$, no todos las óptimos de Pareto pertenecen al conjunto de soluciones generadas por MOAMP. Es decir, todas las soluciones obtenidas por MOAMP son óptimos de Pareto, pero no todos los óptimos de Pareto han podido ser hallados por MOAMP. En este sentido, se muestra la Tabla 16 que, para cada valor de n , indica el número medio de soluciones que son óptimos de Pareto y el número medio de ellas que son obtenidas por cada método heurístico.

Tabla - 16: Comparativa de soluciones óptimos de Pareto en instancias de tamaño pequeño

n	P*	Número de soluciones de P* obtenidas por métodos heurísticos	
		MOAMP	MHeC
5	7,50	7,17	7,17
6	6,00	3,50	3,17

De la tabla anterior se extrae que, el método MOAMP se comporta ligeramente mejor cuando se trata de encontrar mayor número de óptimos de Pareto, ya que el número medio de soluciones obtenidas es igual para $n = 5$ pero, para $n = 6$, es algo mejor MOAMP (3,50 frente a 3,17).

Finalmente, para las dos instancias mostradas en las gráficas anteriores se presentan las correspondientes gráficas detalladas, que incorporan el conjunto de soluciones no dominadas obtenido en cada una de las tres fases del método MOAMP. En general, en ellas se puede observar como durante las FASES I y II se obtienen conjuntos muy similares, idénticos en la mayoría de los casos. Es más, de las dos figuras analizadas, en ninguna de ellas se observan diferencias entre estos dos conjuntos de soluciones no dominadas. Por otra lado, se aprecia que durante la FASE III se producen mejoras más destacables sobre los conjuntos obtenidos en las otras dos fases, circunstancia que queda patente en la gráfica representativa de instancias de tamaño $n = 6$.



10.6.- LA COMPARACIÓN DE LOS MÉTODOS HEURÍSTICOS EN INSTANCIAS DE MAYOR TAMAÑO

A continuación, en esta sección, se comparan los resultados obtenidos por los métodos MOAMP y MHeC en problemas de mayor tamaño ($n > 6$). Para ello, se utilizan las tres métricas descritas en el apartado anterior: *Cardinalidad*, *Cubrimiento* e *Hipervolumen*. Las instancias analizadas en el presente apartado alcanzan el tamaño máximo $n = 150$, con los correspondientes valores de m definidos en la Tabla 3 y $Q_{max} = 25, 40$ y 50 .

La Tabla 17 muestra los resultados medios, para cada valor de n , de estas tres métricas que comparan ambos métodos heurísticos. También se incluyen los tiempos medios de computación empleados, expresados en segundos.

En el caso de la *Cardinalidad* se precisa hacer la siguiente puntualización: como no se ha obtenido el frente óptimo de Pareto P^* , se sustituye éste por una aproximación E^* . La aproximación al frente óptimo se consigue mediante la combinación de todas las soluciones obtenidas por ambos métodos y, la posterior, identificación y selección del conjunto de soluciones no dominadas. De esta forma, la *Cardinalidad* de cada método se define como el número de soluciones obtenidas por cada método que forman parte de la aproximación al frente óptimo (E^*).

Tabla - 17: Métricas comparativas de MOAMP y MHeC en instancias de tamaño mediano y grande ($n > 6$)

n	T.C.		Cardinalidad		Cubrimiento <i>dif</i>	Hipervolumen	
	MOAMP	MHeC	MOAMP	MHeC		MOAMP	MHeC
8	0,127	0,062	88,38%	44,12%	0,4427	0,6495	0,5222
10	0,319	0,192	87,50%	60,47%	0,2703	0,5900	0,5793
12	0,550	0,346	88,45%	38,68%	0,4977	0,5529	0,5147
15	0,938	0,558	89,86%	34,07%	0,5579	0,8172	0,7707
20	2,594	1,632	81,39%	41,93%	0,3946	0,7886	0,7836
30	7,552	6,807	64,47%	65,18%	-0,0070	0,7231	0,7717
50	42,173	51,185	51,75%	75,72%	-0,2397	0,6829	0,8069
70	125,965	195,642	35,91%	80,67%	-0,4476	0,7623	0,8398
100	550,717	1.204,949	35,89%	88,86%	-0,5297	0,6744	0,8202
150	2.083,751	6.280,392	41,44%	86,54%	-0,4510	0,7133	0,8420

Observando la tabla que precede, se puede señalar que en las instancias de menor tamaño ($n = 8, 10, 12, 15$ y 20) el método MOAMP obtiene, en general, mejores resultados que el método MHeC, esto es valores mayores en *Cardinalidad* e *Hipervolumen* y valores positivos en *Cubrimiento*.

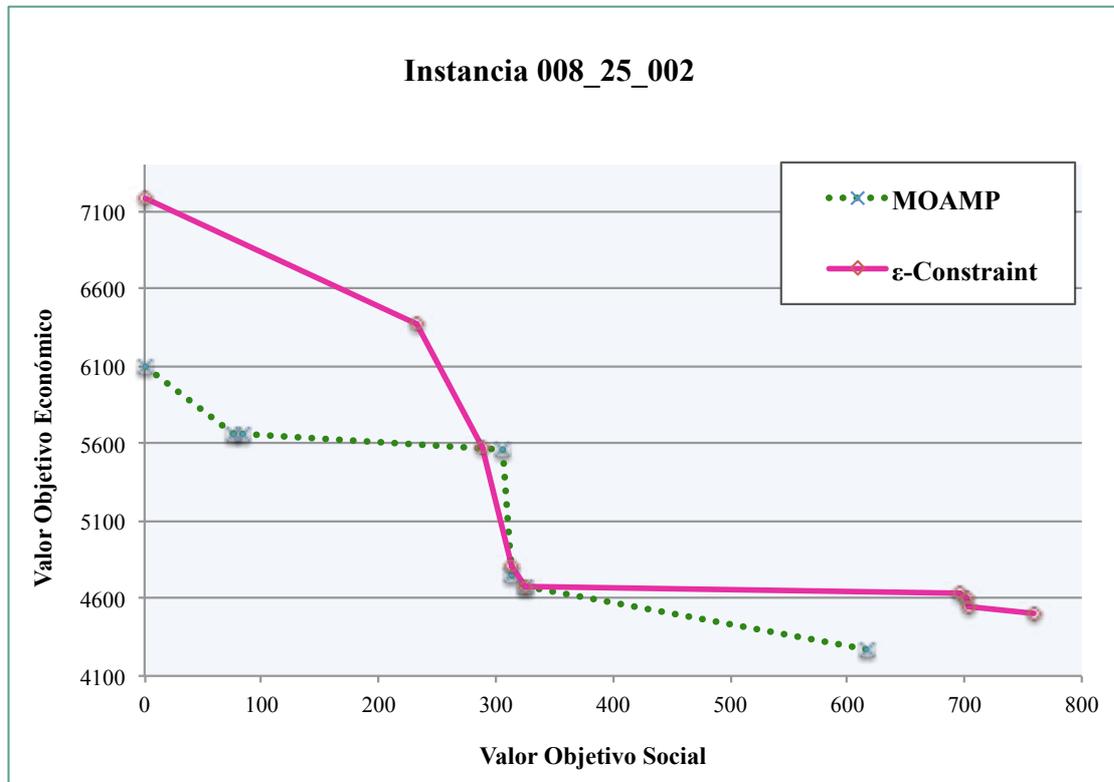
Es, concretamente, para las instancias de tamaño $n = 30$ donde ambos métodos presentan resultados muy similares e incluso se empieza a observar que el método MHeC obtiene levemente mejores valores: 65% frente a 64% en *Cardinalidad*, -0,007 en *Cubrimiento* y 0,77 frente a 0,72 en *Hipervolumen*.

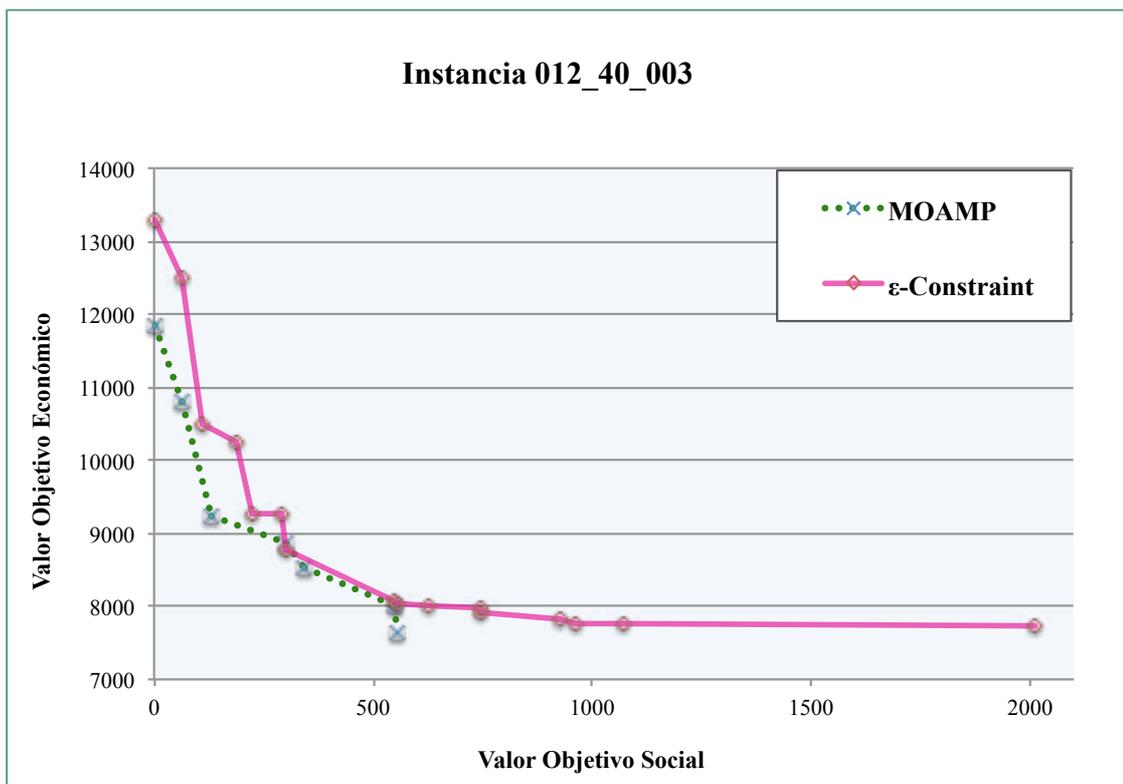
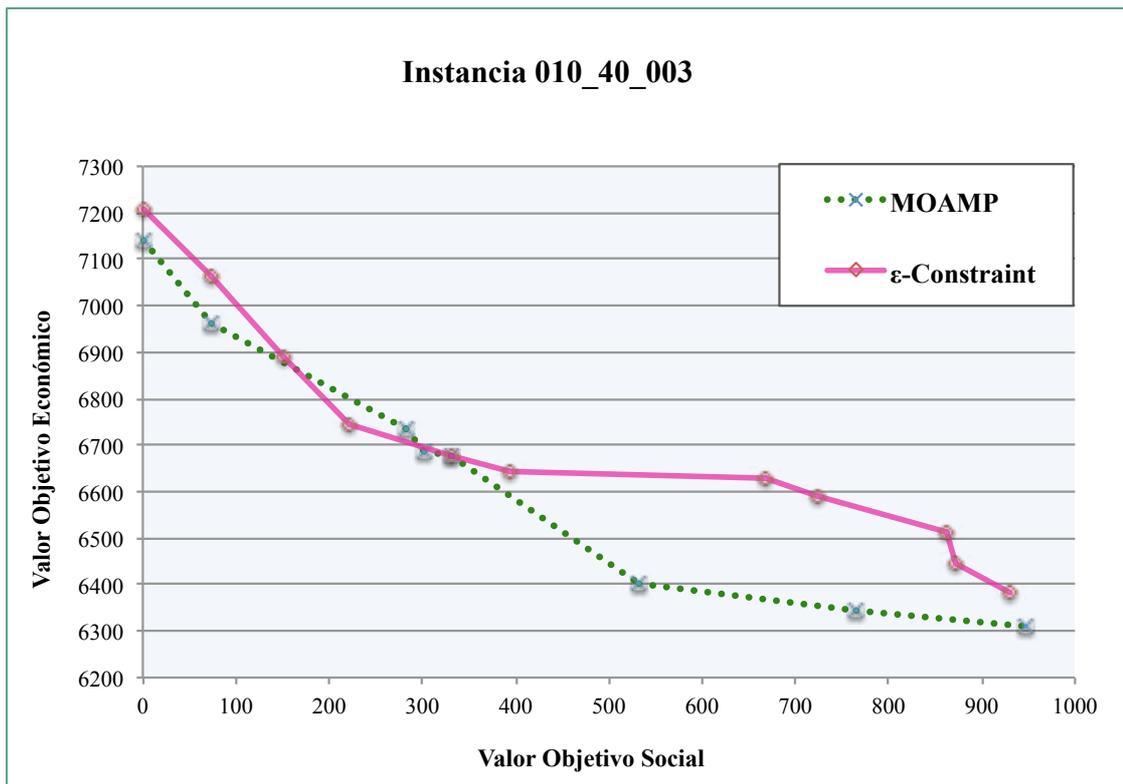
Analizando las instancias de mayor tamaño, aquellas de $n > 30$, se aprecia que los resultados obtenidos por el método MHeC son visiblemente mejores en las tres métricas estudiadas. Destacar que las mayores diferencias se dan en los dos tamaños mayores. En el caso concreto de $n = 100$ el método MHeC consigue: 88% frente a 35% en *Cardinalidad*, -0,53 en *Cubrimiento* y 0,82 frente a 0,67 en

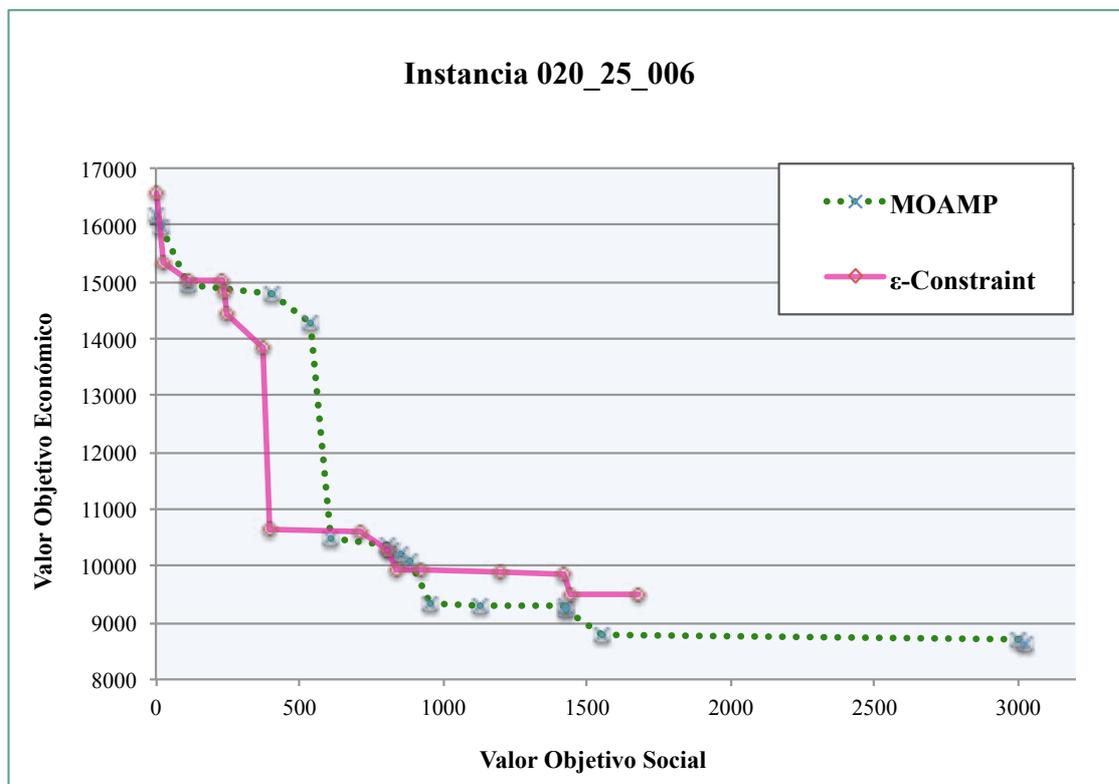
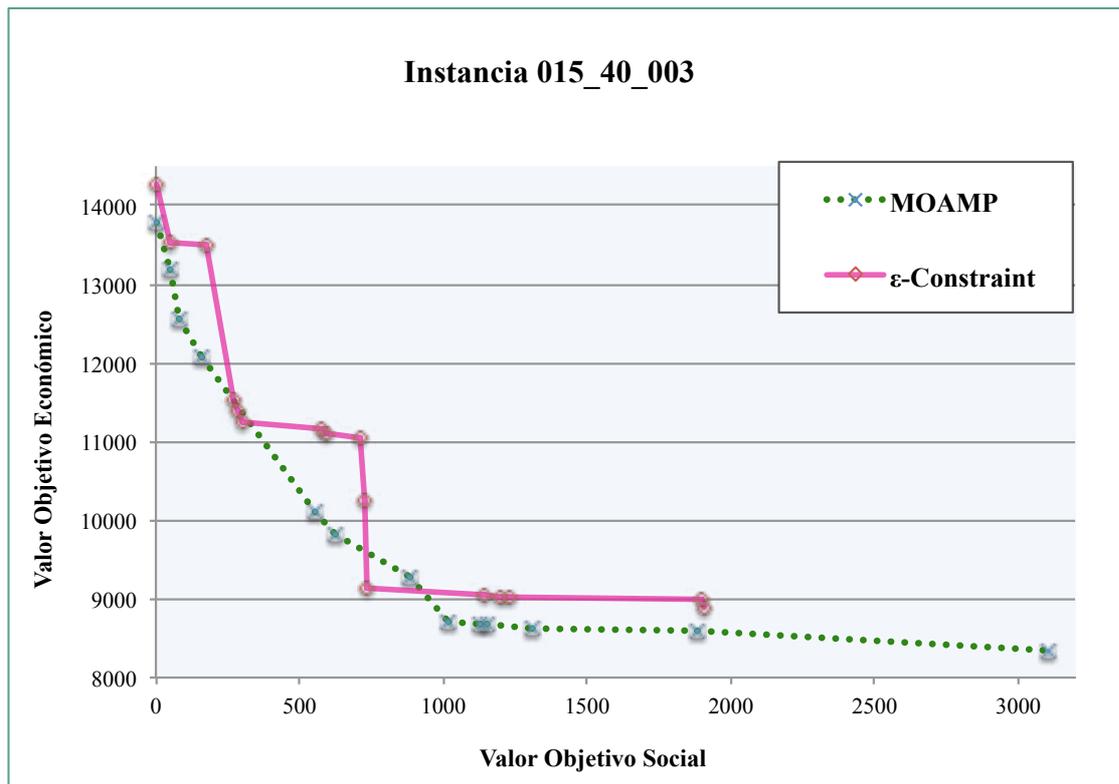
Hipervolumen. Finalmente, en instancias de $n = 150$ el método MHeC obtiene: 86% frente a 41% en *Cardinalidad*, -0,45 en *Cubrimiento* y 0,84 frente a 0,71 en *Hipervolumen*.

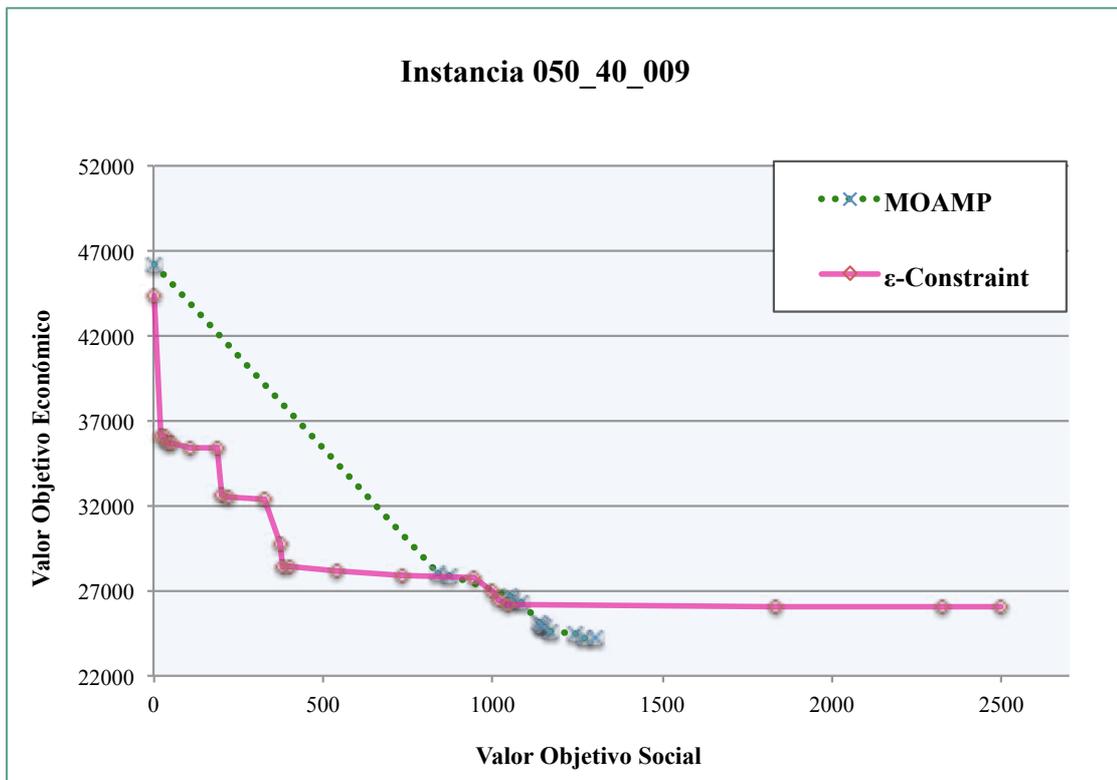
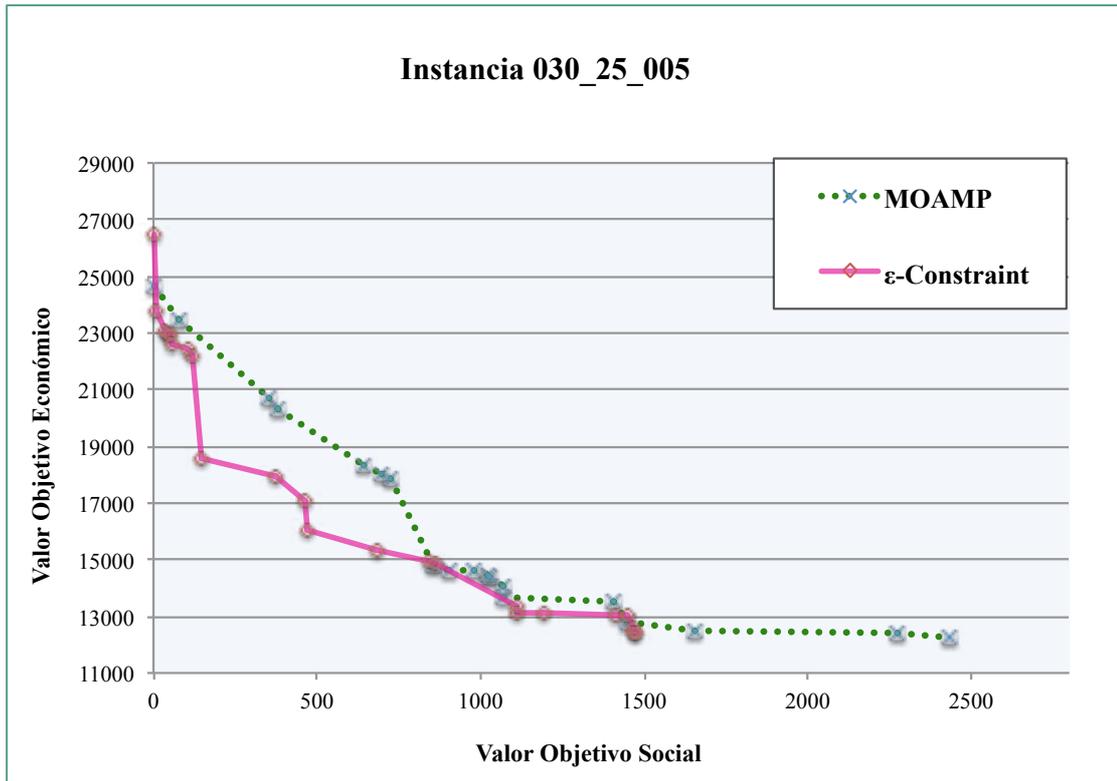
Además, se puede observar como, precisamente, el tiempo de computación empleado por ambos métodos evoluciona de forma similar a la calidad de sus resultados. En las instancias de menor tamaño ($n < 30$), para las cuales el método MOAMP obtiene mejores resultados, se advierte que el tiempo de computación empleado por dicho método es también mayor. De forma más detallada, se percibe que en las instancias de tamaño $n = 8$ el tiempo empleado por MOAMP es el doble que el empleado por el método MHeC, y en las instancias de tamaño $n = 10, 12, 15$ y 20 , MOAMP emplea aproximadamente un 66% más de tiempo. En las instancias de tamaño $n = 30$, para las que ambos métodos obtenían resultados muy similares, el tiempo de cálculo es también parecido, 7,5 segundos empleados por MOAMP frente a los 6,8 segundos del MHeC. Por último, en las instancias de mayor tamaño ($n > 30$), donde el método MHeC consigue mejores resultados, la tendencia prosigue y, también, su tiempo de cálculo es claramente mayor. Se quiere destacar que, las mayores diferencias en el tiempo de cálculo se dan, precisamente, en las instancias de tamaño $n = 100$, para las que el método MHeC emplea más del doble de tiempo del empleado MOAMP, y en las de tamaño $n = 150$, precisando tres veces más de tiempo.

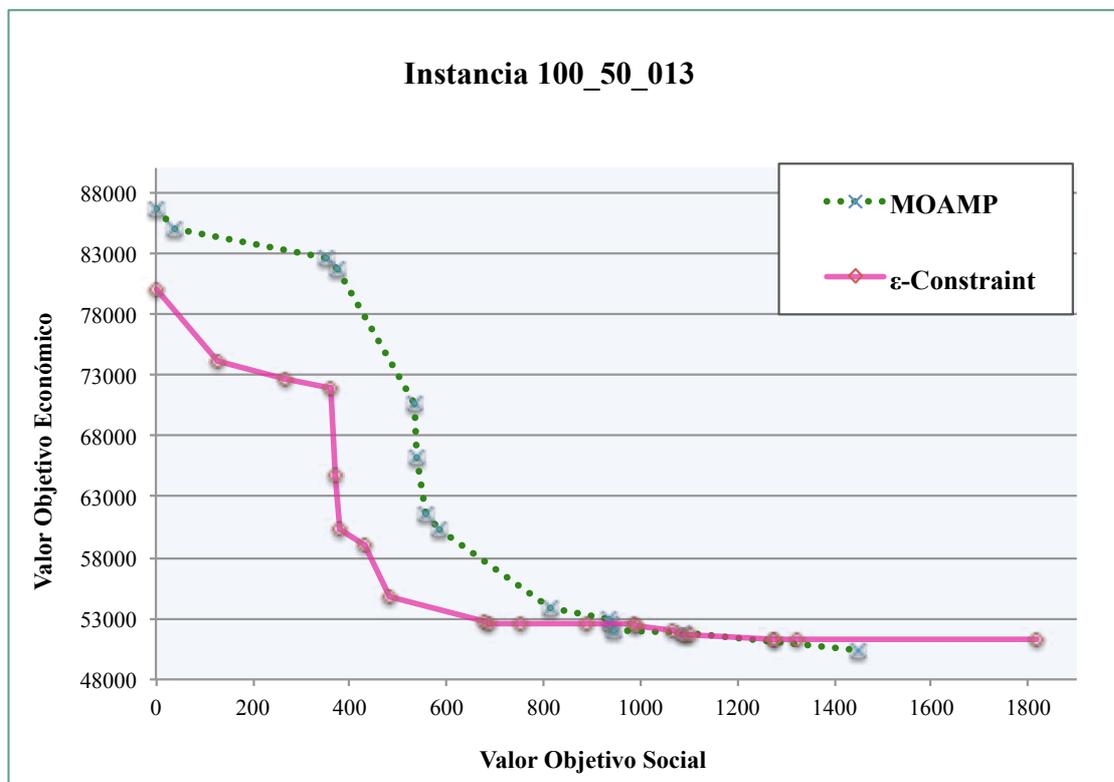
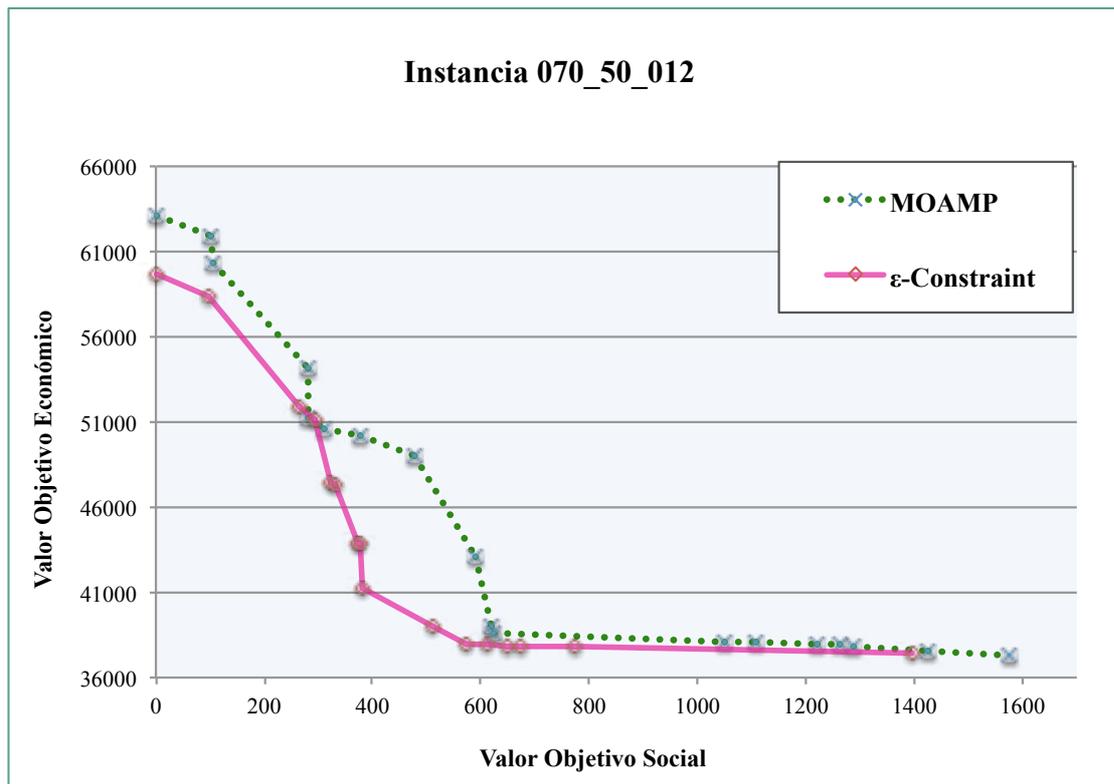
A continuación se muestran, a modo de ejemplo, una serie de gráficas correspondientes a diferentes instancias analizadas, una por tamaño de n ($n > 6$), con los conjuntos de soluciones no dominadas obtenidos por los métodos MOAMP y MHeC.

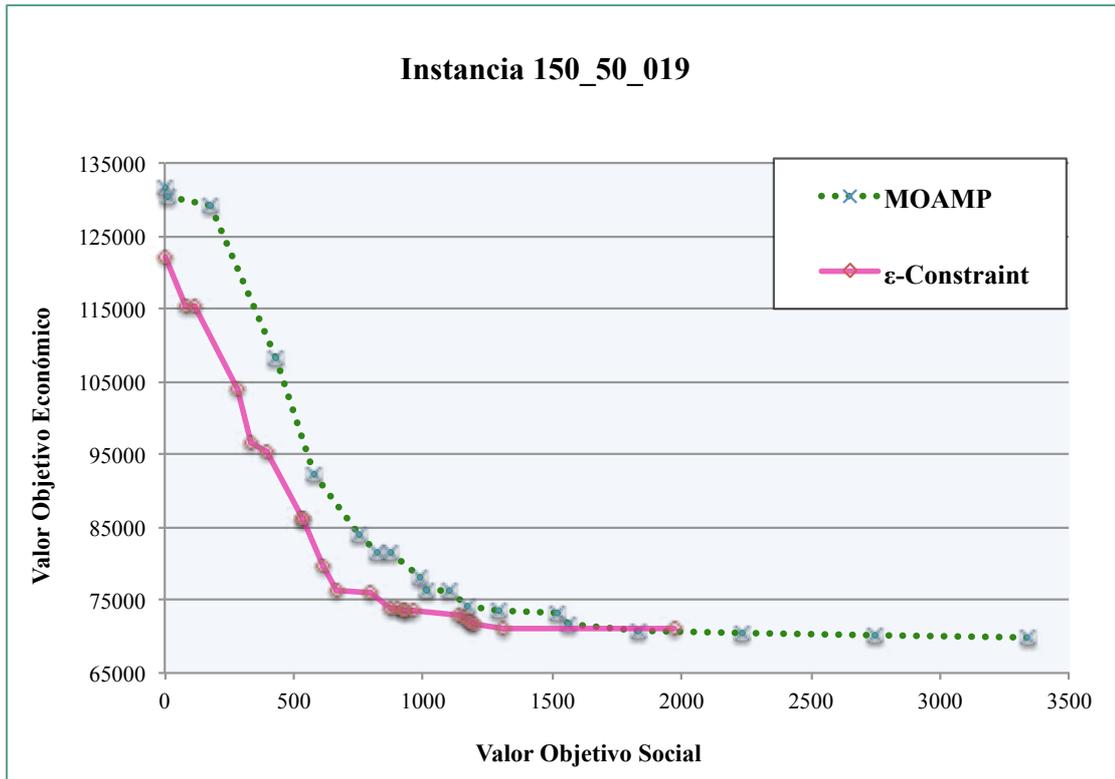










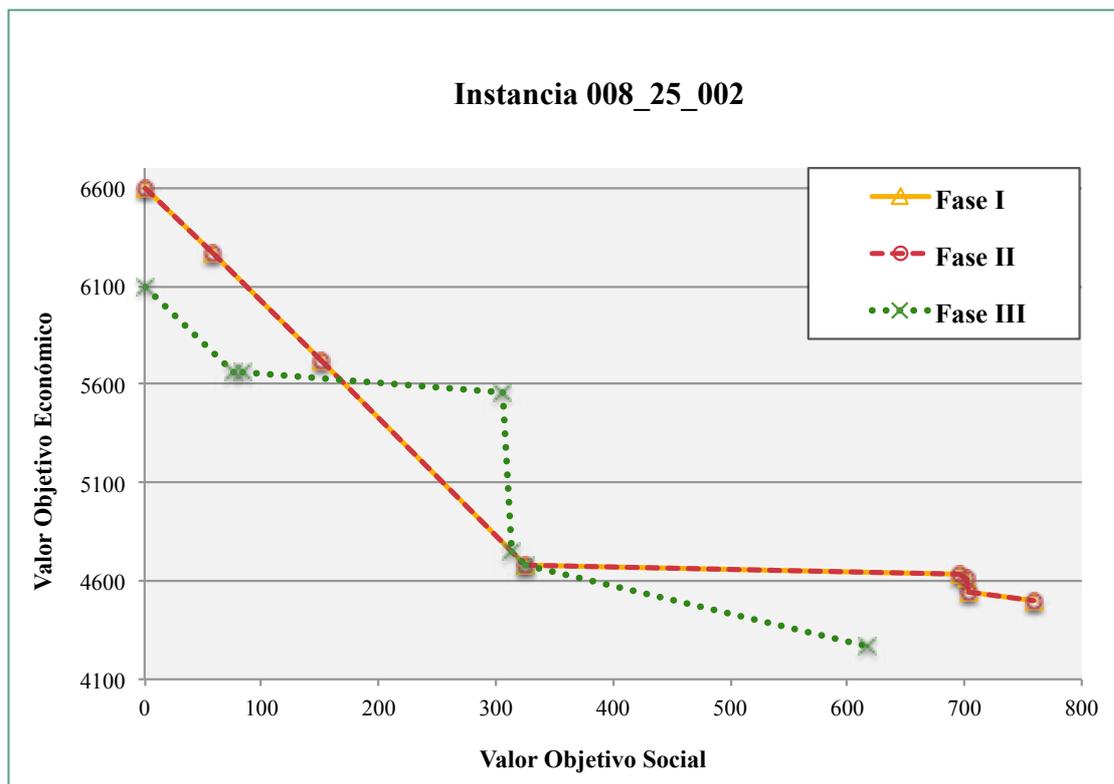


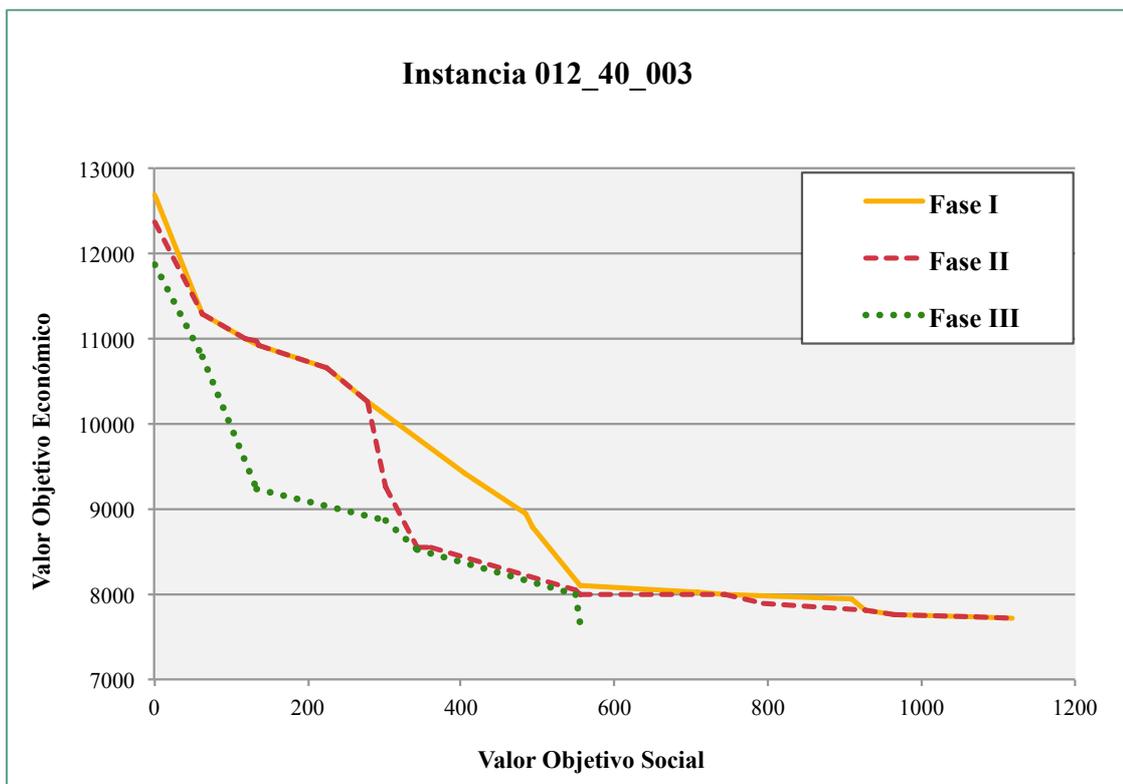
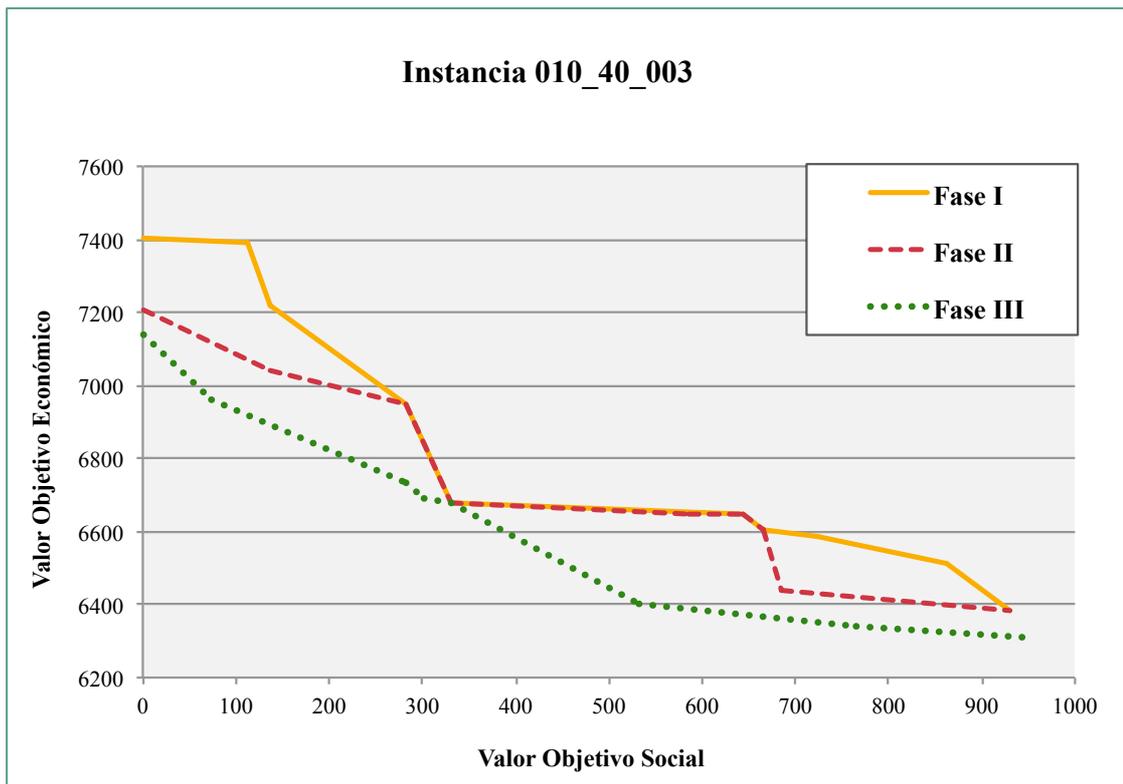
Observando las gráficas anteriores, de forma más o menos general, se pueden extraer las siguientes conclusiones:

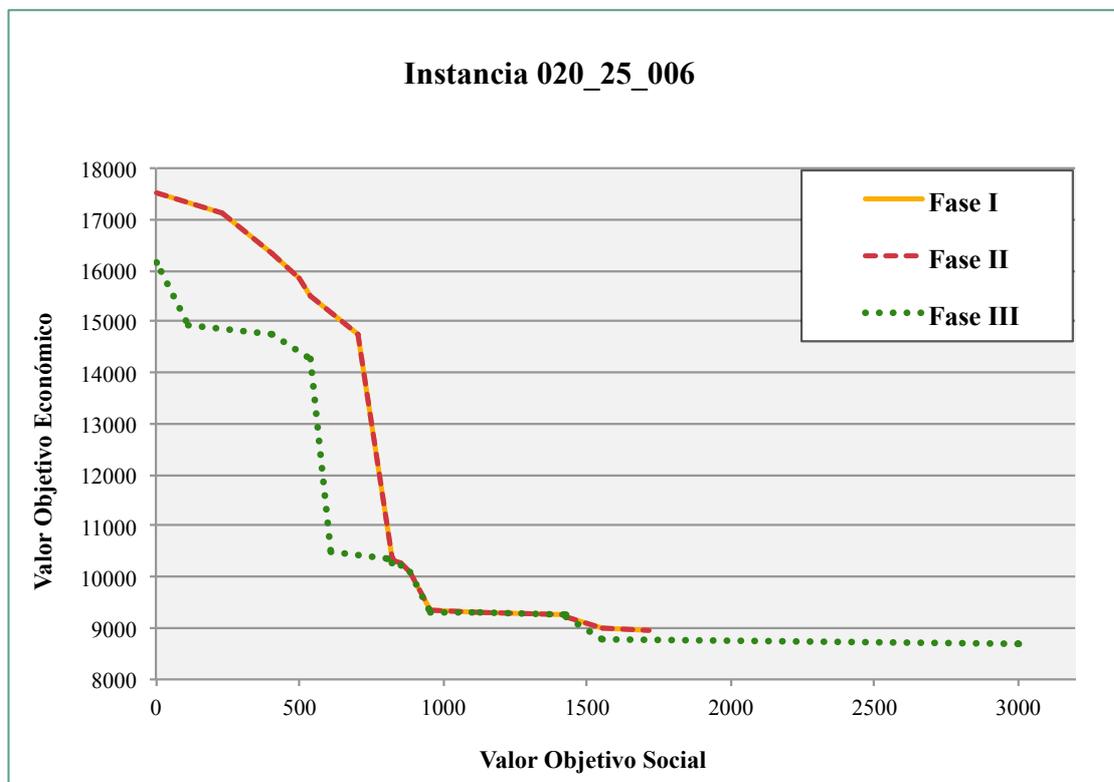
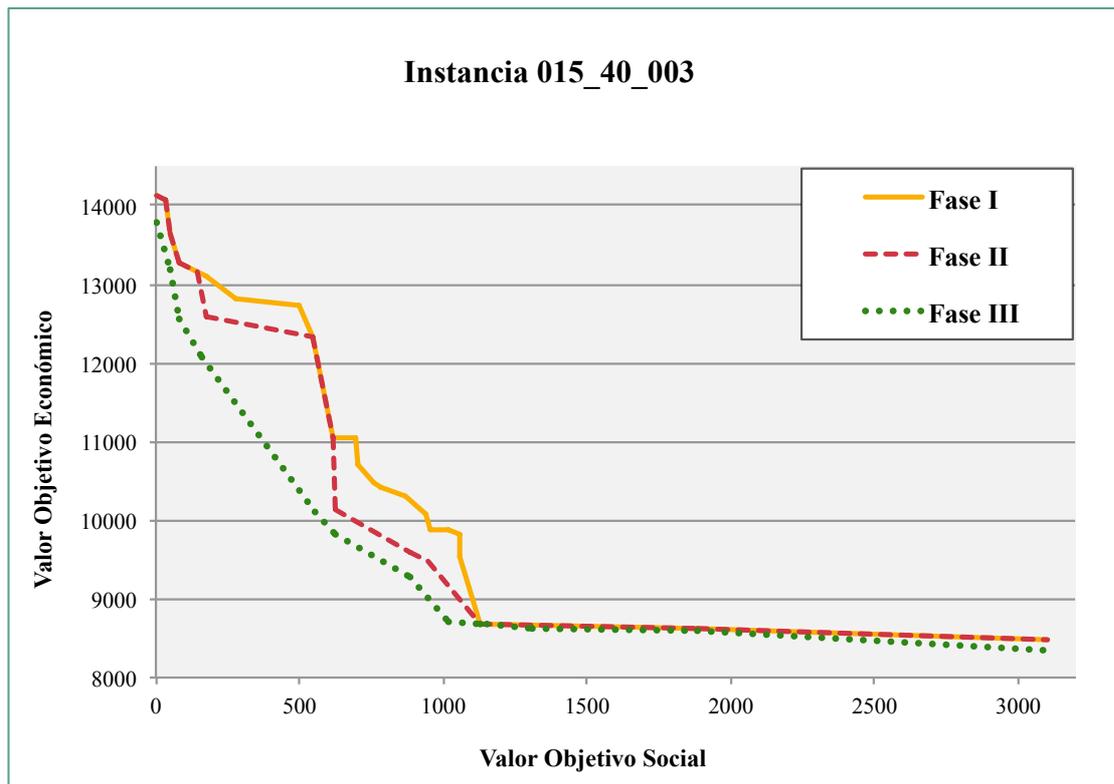
- Al igual que se ha concluido a partir de los resultados de la Tabla 17, se advierte en las gráficas como para las instancias de menor tamaño ($n < 30$) los conjuntos de soluciones no dominadas obtenidos por MOAMP son mejores que los conseguidos por el método MHeC. Análogamente, en las instancias de mayor tamaño ($n \geq 30$) es el método MHeC el que logra mejores resultados.
- Ambos métodos obtienen soluciones similares en la “esquina izquierda”. Los dos obtienen el óptimo social ya que en este objetivo social alcanzan 0, valor que obviamente no puede ser mejorado. La diferencia se presenta en el valor de la distancia total recorrida (f_2) de estos óptimos sociales. En algunos casos este valor es menor en el óptimo social obtenido por MOAMP ($n = 8, 10, 12, 15, 20$ y 30), y en otros el valor es menor en el óptimo social obtenido por MHeC ($n = 50, 70, 100$ y 150).
- En cuanto a la “esquina derecha” se observa que, en todas las figuras el método MOAMP obtiene claramente mejores soluciones con respecto a la distancia total recorrida. En algunos casos las diferencias entre las soluciones obtenidas por ambos métodos son muy relevantes, como en el caso de $n = 8, 10, 12, 15, 20$ y 50 .
- Finalmente, también parece claro que, por lo general, en los puntos intermedios la curva obtenida por MHeC obtiene mejores soluciones que dominan claramente a las obtenidas por MOAMP. Tan sólo, en la cercanía al extremo derecho, que tal y como se ha comentado es donde MOAMP se comporta mejor, existen algunas soluciones obtenidas por el método

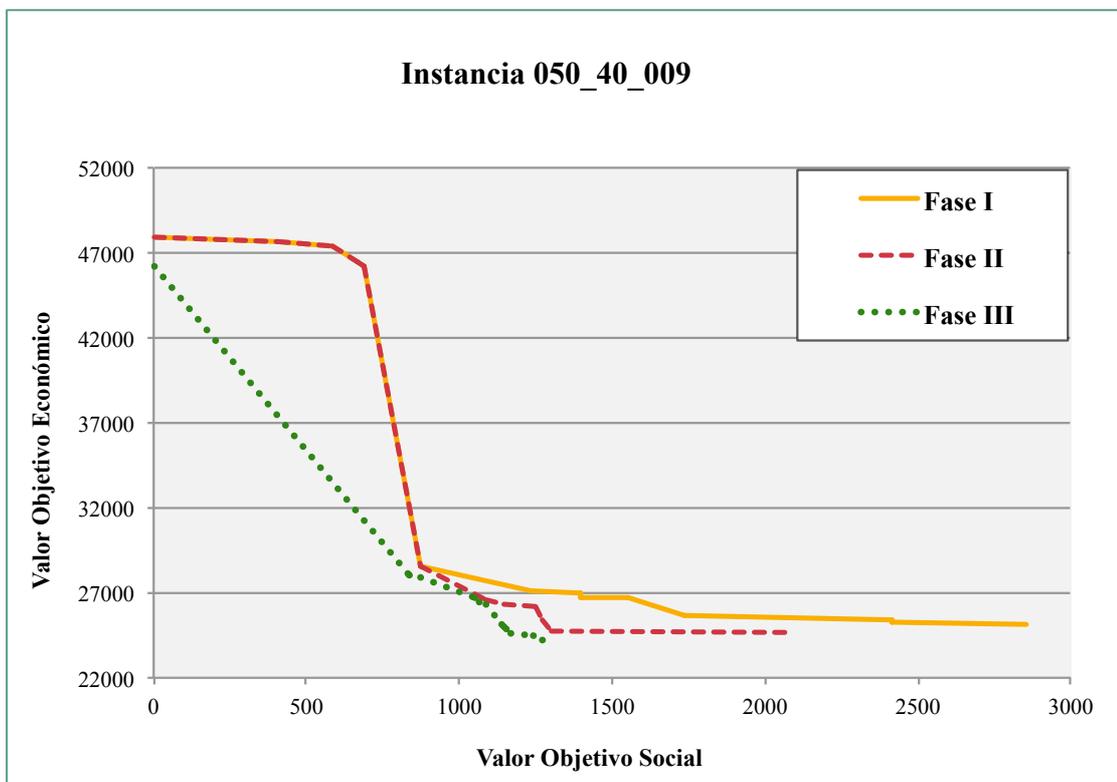
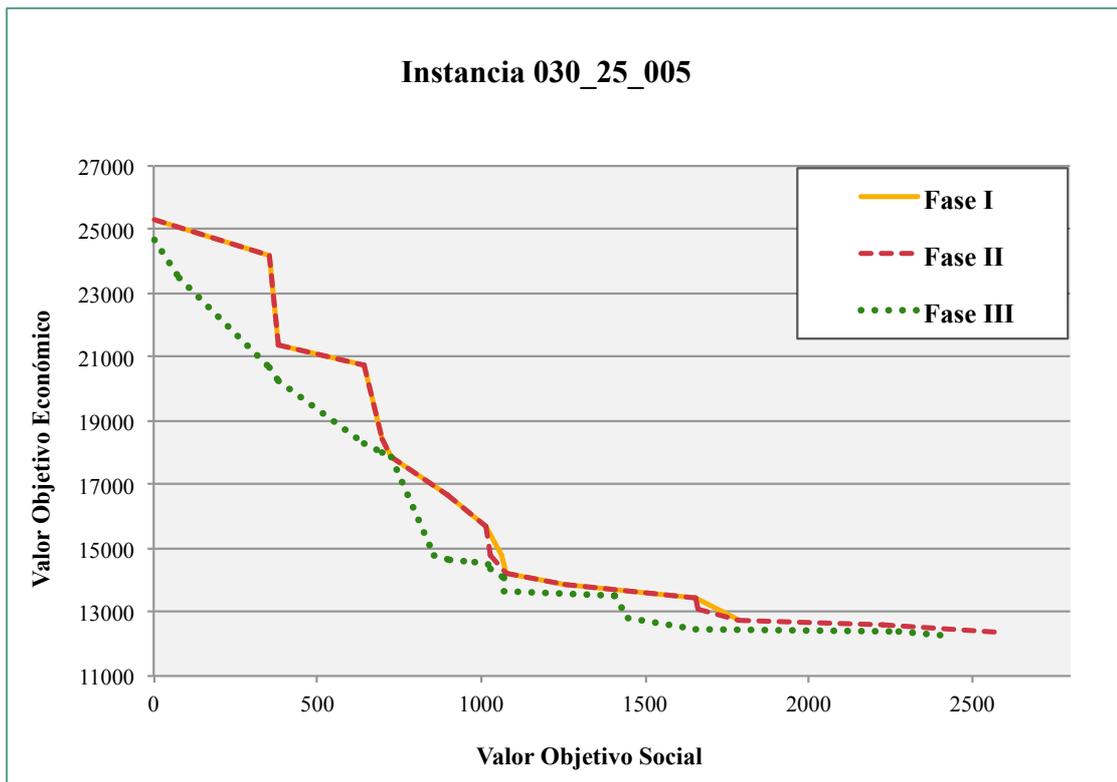
MOAMP que no son dominadas por las conseguidas por el método MHeC. También ocurre esto en algún caso aislado ($n = 8, 10, 12$ y 15) en la cercanía de la esquina izquierda. Precisamente, éstos son cuatro de los seis casos donde el óptimo social obtenido por MOAMP domina al óptimo obtenido por el método MHeC.

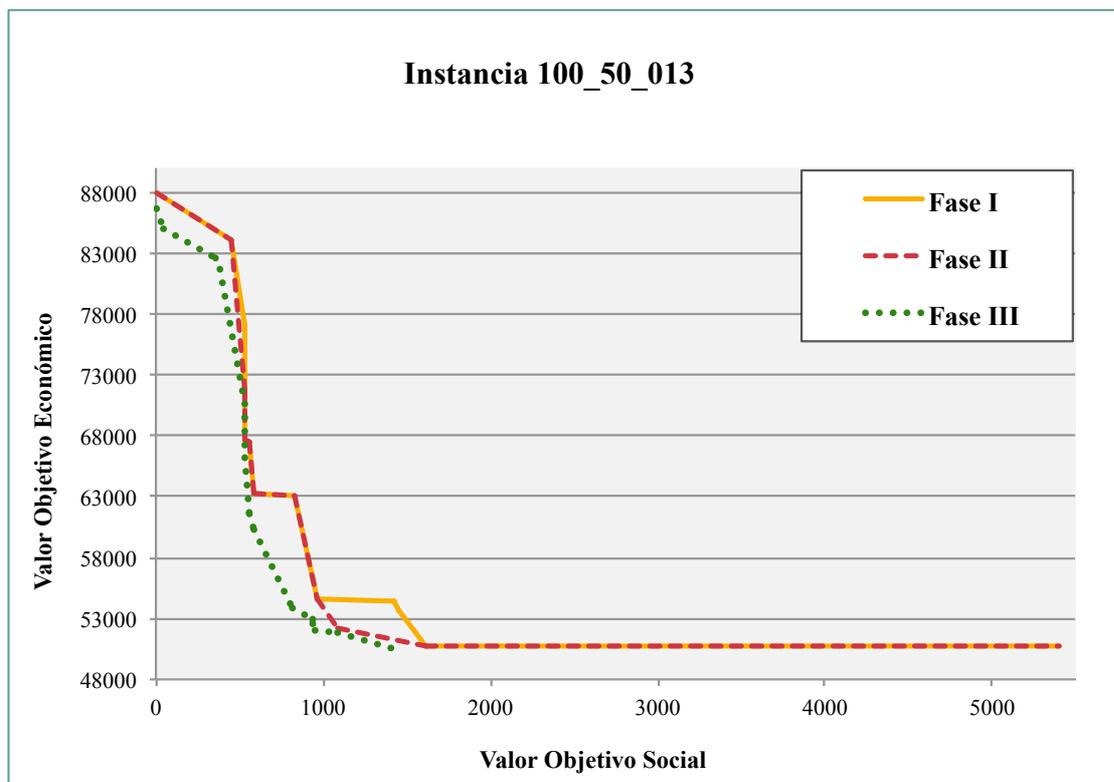
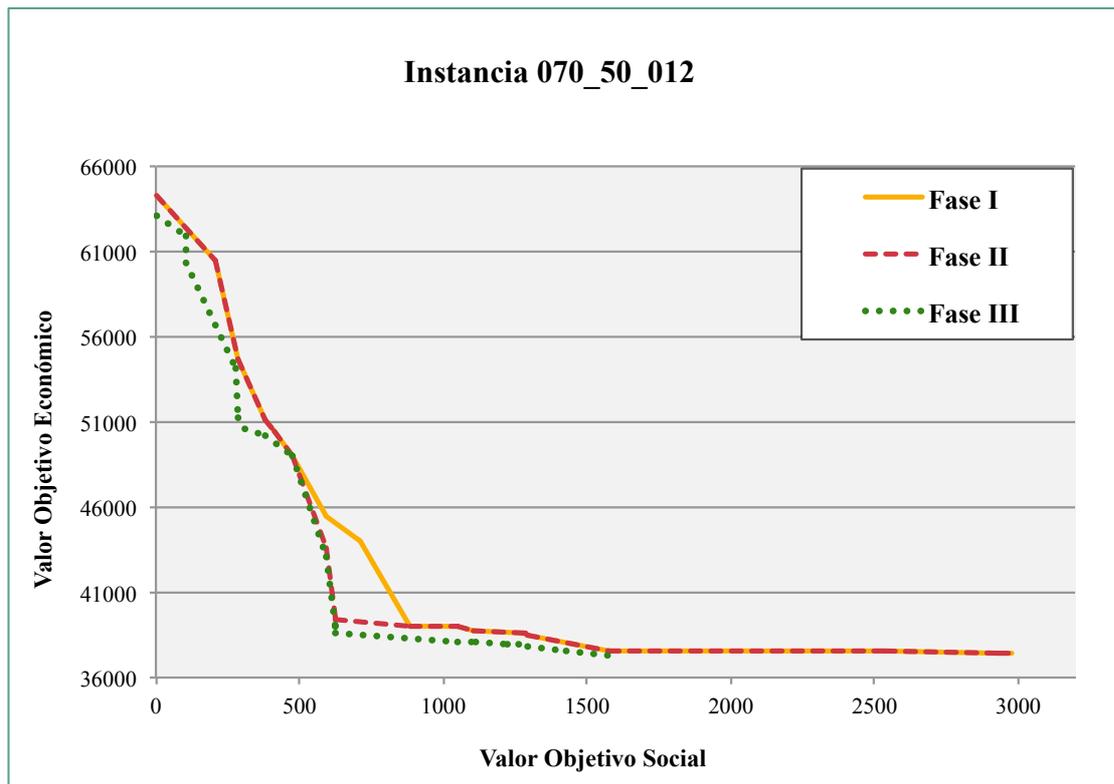
Finalmente, para cada una de las nueve instancias mostradas en las gráficas anteriores, se presentan las correspondientes gráficas detalladas, que incorporan el conjunto de soluciones no dominadas obtenido en cada una de las tres fases del método MOAMP.

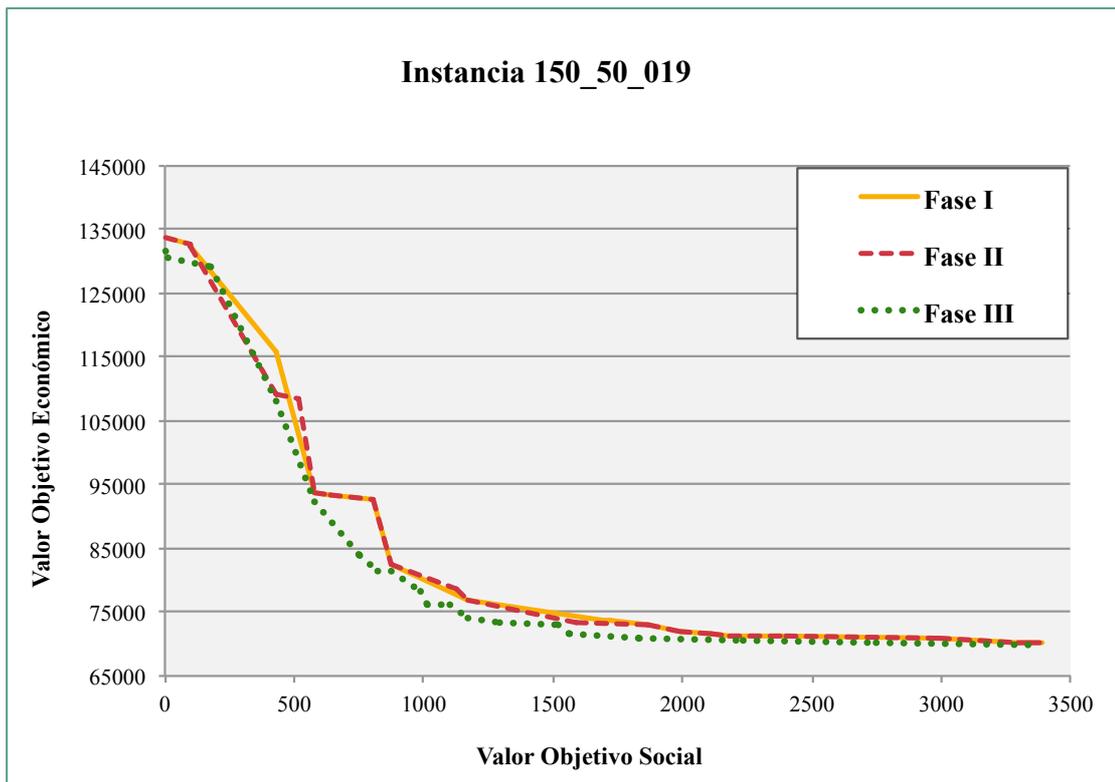












En general, de las gráficas anteriores se observa que la FASE II obtiene soluciones ligeramente mejores que la FASE I, mientras que la FASE III consigue mejorar de forma relevante a las obtenidas por las otras dos. Un aspecto interesante en este sentido es que, también la FASE III mejora los puntos extremos originales, los obtenidos en la FASE I. Por ejemplo, al fijarse en el “extremo izquierdo”, ya en la FASE I se obtiene el óptimo social, con valor 0 en f_1 , sin embargo, este extremo es mejorado en las otras fases, especialmente durante la FASE III, con soluciones que presentan un mejor valor en f_2 . Algo parecido ocurre con el extremo derecho, donde se observan las siguientes situaciones: a) las mejores soluciones de la FASE I con respecto a f_2 son mejoradas a veces en f_2 en las siguientes fases, b) o bien en las siguientes fases se obtienen soluciones con similar valor de f_2 pero mejor valor en f_1 .

En este sentido una mejora metodológica sería intentar, en la FASE I, reforzar la dominancia de los puntos extremos iniciales. Es decir, si estamos hallando el óptimo con respecto a f_1 habría que tener en cuenta el valor de f_2 como un segundo criterio. Es decir, elegir siempre la mejor solución con respecto a f_1 y, en caso de igualdad, en este valor elegir la mejor solución con respecto a f_2 . Y cuando se esté hallando el óptimo con respecto a f_2 actuar de forma análoga.

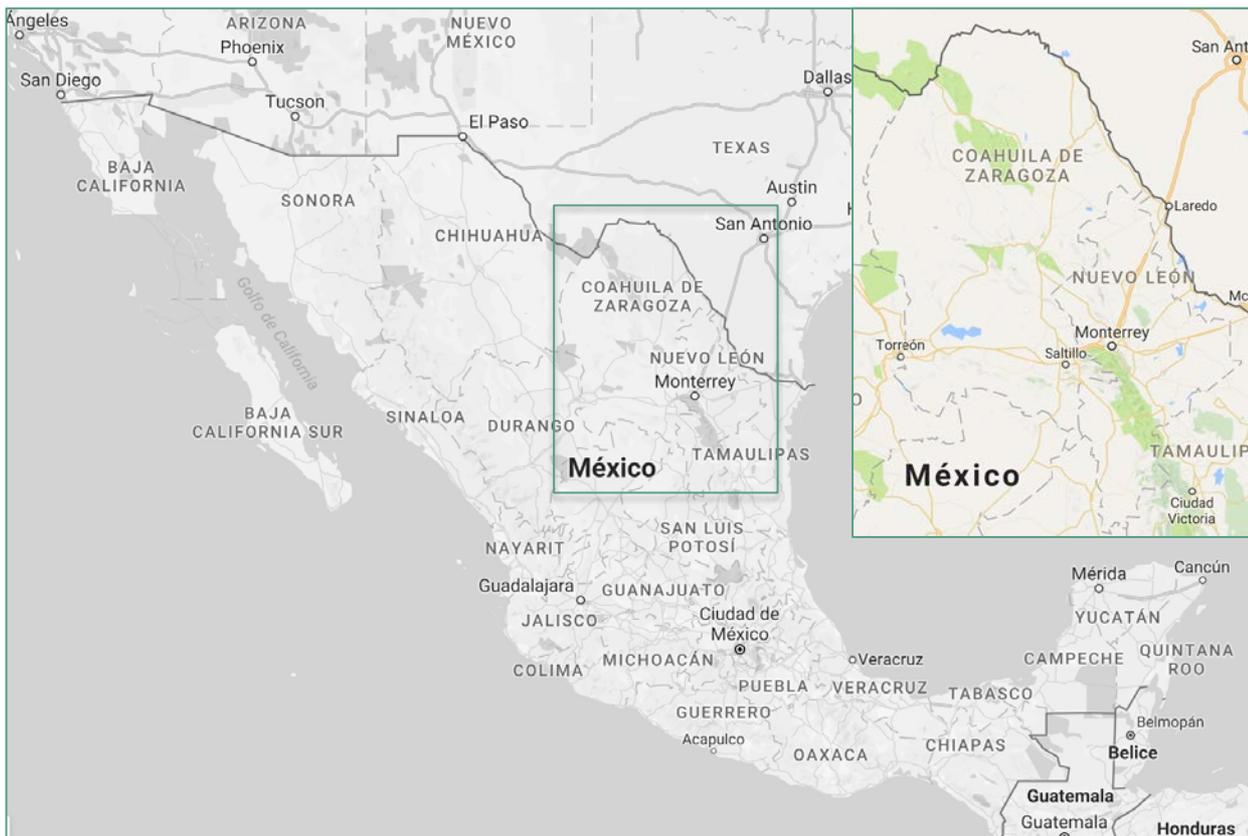
10.7.- LA APLICACIÓN A UN CASO REAL

Por último, en esta sección, se presenta la aplicación del modelo y de la metodología propuesta al caso real del traslado de pacientes para recibir tratamientos especializados entre dos ciudades mexicanas.

10.7.1.- El contexto

La ciudad de Saltillo se encuentra en el norte de México y es la capital del estado de Coahuila de Zaragoza. En ella, se dispone de hospitales de medicina general y de especialización pertenecientes al Instituto Mexicano de Seguridad Social (IMSS) conocidos como hospitales de segundo nivel, pero no cuenta con hospitales catalogados como de tercer nivel. Es en los hospitales de tercer nivel donde se llevan a cabo algunos tratamientos especializados, por ejemplo, tratamientos oncológicos, de pediatría y cardiología especializados, entre otros.

Su proximidad con la ciudad de Monterrey, capital del estado de Nuevo León, propicia que los pacientes que requieren tratamientos especializados sean canalizados a los hospitales de esta ciudad para su atención médica. De acuerdo a los datos proporcionados por el IMSS, son alrededor de 300 personas las que se trasladan, diariamente, de Saltillo a Monterrey con el fin de recibir algún tratamiento médico especializado.



Los pacientes provienen de distintos barrios de la ciudad de Saltillo y se dirigen principalmente a cinco hospitales en Monterrey, ubicados en distintos puntos geográficos de la ciudad. La distancia por carretera desde la ciudad de Saltillo a Monterrey es de 80 km. En concreto, de los datos del IMSS consultados se estima que la distribución de los pacientes trasladados entre los cinco hospitales es la siguiente:

- A la Clínica 25 alrededor del 56% del total de los pacientes. Éste es el hospital con tratamientos oncológicos como quimioterapias, radioterapias, neurocirugías.
- A la Clínica 34, donde se ofrecen los tratamientos de cardiología, acuden un 20% de los pacientes.
- A la Clínica 21, con tratamientos de traumatología, alrededor del 11% de los pacientes.
- A la Clínica 33, de ginecología, alrededor de un 10%.
- A la Clínica 22, especializada en tratamientos de psiquiatría, un 3% del total de pacientes.

Es necesario indicar que, los pacientes son citados en diferentes horarios por la mañana o por la tarde y los tratamientos tienen diferente duración. El problema, consiguientemente, consiste en diseñar adecuadamente el traslado de los pacientes, para que acudan a su cita a tiempo y reciban su tratamiento.

En la actualidad el IMSS no cuenta con la infraestructura en vehículos necesaria para atender el traslado, entre Saltillo y Monterrey, de todos los pacientes. Es por ello que, algunos pacientes se trasladan en vehículos particulares, otros en transporte público con bonos de autobús pagados por el IMSS y otros lo hacen en vehículos o ambulancias del IMSS. Estos últimos son los pacientes que tienen alguna discapacidad o que por su condición médica no pueden viajar en transporte convencional de manera independiente.

Se pretende evaluar la factibilidad de alcanzar un escenario ideal en el que el IMSS pueda programar los viajes de los pacientes para que sean trasladados a recibir su tratamiento y regresen a su hogar en vehículos adecuados y adaptados para el transporte de personas que van o salen de un hospital, esto es, que sean trasladados en ambulancias o vehículos con personal que pueda brindar los primeros auxilios o asistencia especializada en caso de ser necesario.

Como se ha explicado en capítulos anteriores, al diseñar rutas para el traslado de pacientes en vehículos compartidos, es muy importante tratar de reducir el tiempo excedente de viaje de cada uno de ellos, esto es, si consideramos el viaje directo de su hogar al hospital y el tiempo de viaje al tener que compartir el vehículo y tener que ir a recoger a otras personas a sus hogares antes de ir al hospital. El segundo aspecto importante, también a considerar, es que el IMSS deseará minimizar los costos de los traslados, al ser un servicio brindado por este organismo, es decir, tratará de reducir la distancia total que recorrerán sus vehículos. Por todo lo anterior, en este caso real se aplica el modelo y la metodología para la resolución de un problema biobjetivo propuestos en este trabajo de investigación, en el que se busca minimizar tanto los costos, medidos por la distancia recorrida, como el tiempo excedente de viaje de los pacientes.

10.7.2.- Las instancias reales y pseudoreales

Una vez testado el algoritmo, en las secciones anteriores del presente capítulo, y para proceder a su ejecución sobre el problema real arriba contextualizado, es necesario definir las instancias a analizar mediante toda la información necesaria que deben contener para que el algoritmo se pueda ejecutar.

Los tipos de vehículos con los que cuenta el IMSS, que son los que se van a considerar en el problema, permiten trasladar a varios pacientes simultáneamente y están adaptados para el transporte de pasajeros con sillas de ruedas. A continuación se describen y representan las siete instancias analizadas en esta sección para la optimización de traslados de pacientes desde varios lugares de Saltillo a los diferentes hospitales en Monterrey.

La localización de los diversos nodos considerados en las instancias se ha realizado empleando coordenadas UTM representadas en la plataforma NEVA - Sistema para la generación de instancias y representación de soluciones en modelos de rutas de transporte y logística, registrada por Dr. Joaquín Pacheco Bonrostro en 2015. Para la obtención de la matriz de distancias entre cada par de nodos i, j se han determinado las distancias mínimas entre ellos haciendo uso del algoritmo de Dijkstra. Una vez determinada la matriz de distancias, se ha obtenido la matriz de tiempos como una función lineal de la distancia y de las características de la vía.

Antes de detallar cada una de las siete instancias, se requiere precisar que, en todas ellas, se ha supuesto un paciente recogido en cada nodo, solicitudes de un único pasajero, y se ha considerado la existencia de dos depósitos, uno en Saltillo, lugar de donde salen los vehículos, y otro en Monterrey, donde se estacionan para esperar el regreso.

I) INSTANCIA REAL

Se trata de una instancia de tamaño $n = 5$, con 5 nodos de recogida en Saltillo y 5 hospitales de descarga en Monterrey. Para realizar los traslados, se cuenta con 2 vehículos, $m = 2$, con capacidad para 3 y 2 pacientes respectivamente, $Q_{max} = 3$. Y, en este caso, $L_{max} = 150$ minutos.

Se han calculado las matrices de distancias y tiempos, que en este caso son de 12×12 , para los nodos que se enumeran y representan a continuación:

- Orígenes (Saltillo): **Nodo 1:** Clínica 2 (depósito de donde salen los vehículos) /
 Nodo 2: Centro / **Nodo 3:** Mirasierra / **Nodo 4:** Saltillo2000 /
 Nodo 5: Guayulera / **Nodo 6:** Lourdes



Destinos (Monterrey): **Nodo 7:** Clínica 25 / **Nodo 8:** Clínica 34 / **Nodo 9:** Clínica 21 /
Nodo 10: Clínica 22 / **Nodo 11:** Clínica 33 /
Nodo 12: Final (depósito donde descansan los vehículos)

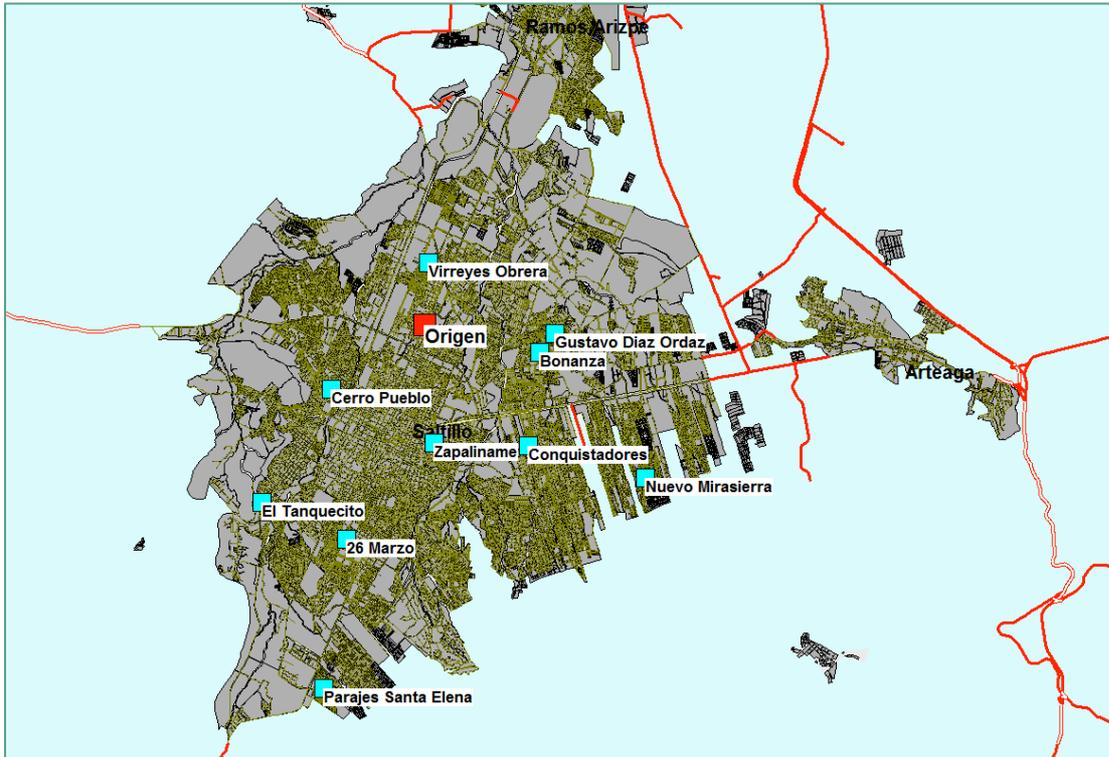


II) INSTANCIA PSEUDOREAL 1 [*Inst010_05_003*]

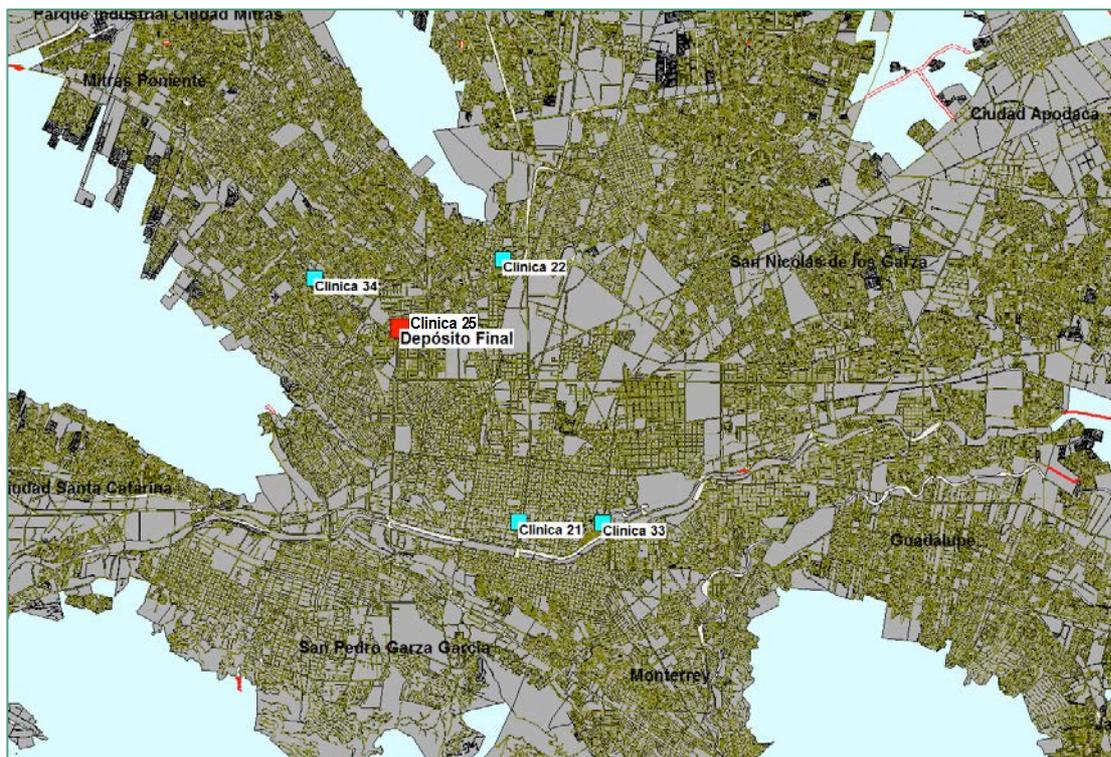
Como se ha explicado anteriormente, las instancias se denotan por *InstXXX_YY_ZZZ*, donde *XXX* es el valor de n , *YY* el valor de Q_{max} y *ZZZ* el valor de m . En este caso, se trata de una instancia de tamaño $n = 10$, con 10 nodos de recogida en Saltillo generados para esta instancia y 5 hospitales de descarga en Monterrey, idénticos en todas las instancias consideradas. Para realizar los traslados, se cuenta con 3 vehículos, $m = 3$, con capacidad para 5 pacientes, $Q_{max} = 5$. Y, en este caso, $L_{max} = 150$ minutos.

Se han calculado las matrices de distancias y tiempos, que en este caso son de 17×17 , para los nodos que se enumeran y representan a continuación:

Orígenes (Saltillo): **Nodos 1 a 11** que se corresponden con los puntos representados en el gráfico siguiente.



Destinos (Monterrey): **Nodo 12:** Clínica 25 / **Nodo 13:** Clínica 34 / **Nodo 14:** Clínica 21 / **Nodo 15:** Clínica 22 / **Nodo 16:** Clínica 33 / **Nodo 17:** Final (depósito donde descansan los vehículos)

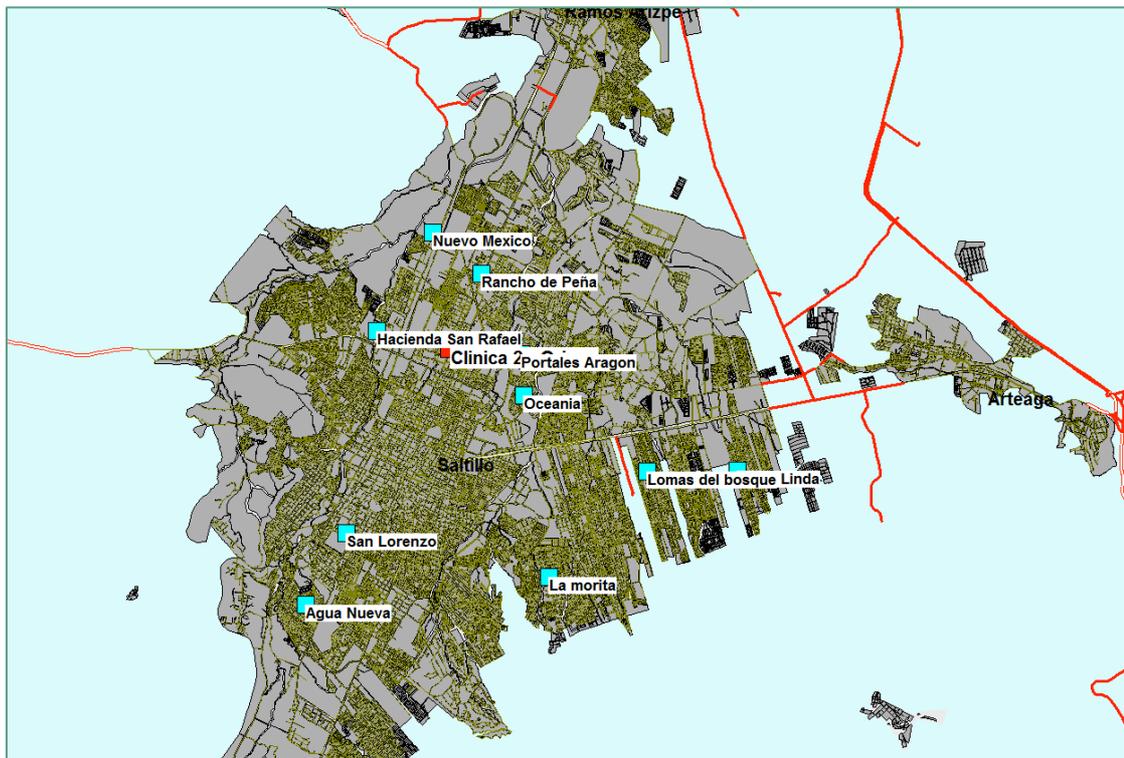


III) INSTANCIA PSEUDOREAL 2 [Inst010_05_003]

Se trata de otra instancia de tamaño $n = 10$, con 10 nodos de recogida en Saltillo generados para esta instancia concreta y los 5 hospitales habituales de descarga en Monterrey. Para realizar los traslados, se cuenta con 3 vehículos, $m = 3$, con capacidad para 5 pacientes, $Q_{max} = 5$. Y, en este caso, $L_{max} = 150$ minutos.

Se han calculado las matrices de distancias y tiempos, que en este caso son de 17×17 , para los nodos que se enumeran y representan a continuación:

Orígenes (Saltillo): **Nodos 1 a 11** que se corresponden con los puntos representados en el gráfico siguiente.



Destinos (Monterrey): **Nodo 12:** Clínica 25 / **Nodo 13:** Clínica 34 / **Nodo 14:** Clínica 21 /
Nodo 15: Clínica 22 / **Nodo 16:** Clínica 33 /
Nodo 17: Final (depósito donde descansan los vehículos)
 (análogos a los representados para las instancias real y pseudoreal 1)

IV) INSTANCIA PSEUDOREAL 3 [Inst010_05_003]

Se ha generado una tercera instancia de tamaño $n = 10$, con 10 nodos diferentes de recogida en Saltillo y los 5 hospitales de descarga en Monterrey. Para realizar los traslados, se cuenta de nuevo con 3 vehículos, $m = 3$, con capacidad para 5 pacientes, $Q_{max} = 5$. Y, también en este caso, $L_{max} = 150$ minutos.

Se han calculado las matrices de distancias y tiempos, que son de 17×17 , para los nodos que se enumeran y representan a continuación:

Orígenes (Saltillo): **Nodos 1 a 11** que se corresponden con los puntos representados en el gráfico siguiente.



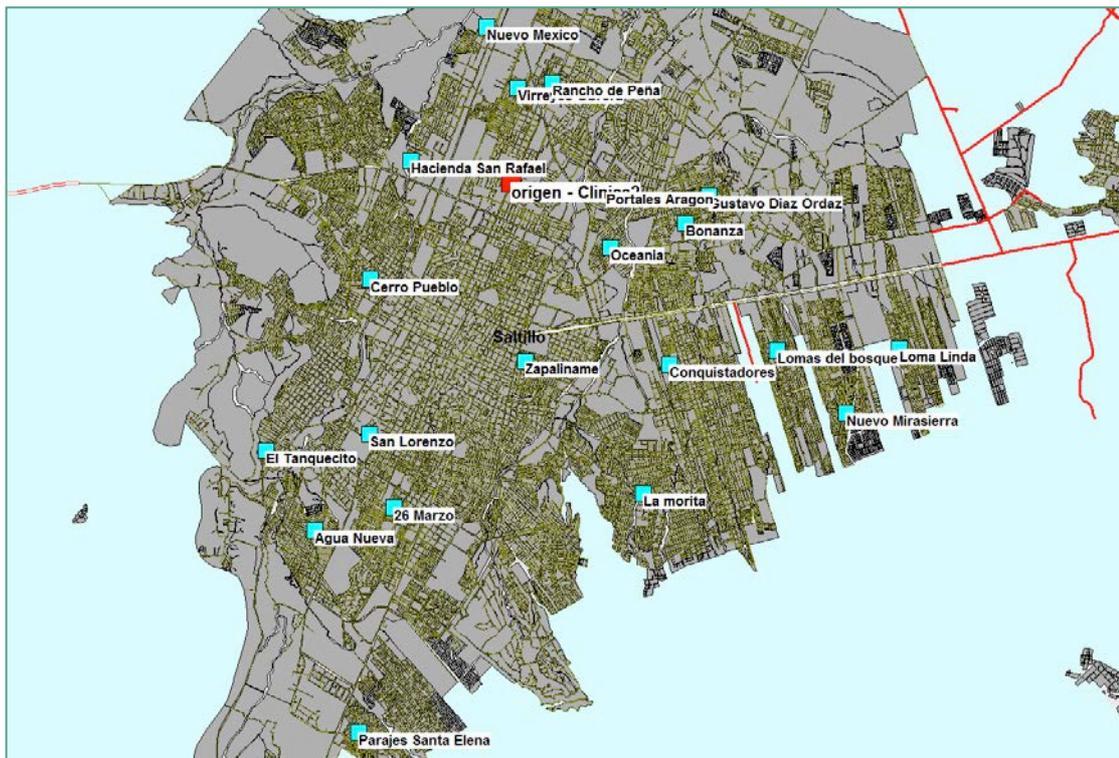
Destinos (Monterrey): **Nodo 12:** Clínica 25 / **Nodo 13:** Clínica 34 / **Nodo 14:** Clínica 21 /
Nodo 15: Clínica 22 / **Nodo 16:** Clínica 33 /
Nodo 17: Final (depósito donde descansan los vehículos)
(análogos a los representados para las instancias real y pseudoreal 1)

V) INSTANCIA PSEUDOREAL 4 [Inst020_05_006]

Se trata de una instancia de tamaño $n = 20$, con 20 nodos de recogida en Saltillo generados específicamente para esta instancia y 5 hospitales de descarga en Monterrey, idénticos en todas las instancias consideradas. Para realizar los traslados, se cuenta con 6 vehículos, $m = 6$, con capacidad para 5 pacientes, $Q_{max} = 5$. Y, en este caso, $L_{max} = 150$ minutos.

Se han calculado las matrices de distancias y tiempos, que en este caso son de 27×27 , para los nodos que se enumeran y representan a continuación:

Orígenes (Saltillo): **Nodos 1 a 21** que se corresponden con los puntos representados en el gráfico siguiente.



Destinos (Monterrey): **Nodo 22:** Clínica 25 / **Nodo 23:** Clínica 34 / **Nodo 24:** Clínica 21 /
Nodo 25: Clínica 22 / **Nodo 26:** Clínica 33 /
Nodo 27: Final (depósito donde descansan los vehículos)
 (análogos a los representados para las instancias real y pseudoreal 1)

VI) INSTANCIA PSEUDOREAL 5 [Inst020_05_006]

Se trata de otra instancia de tamaño $n = 20$, con 20 nodos de recogida en Saltillo generados para esta instancia en concreto y los 5 hospitales habituales de descarga en Monterrey. Para realizar los traslados, se cuenta con 6 vehículos, $m = 6$, con capacidad para 5 pacientes, $Q_{max} = 5$. Y, en este caso, $L_{max} = 150$ minutos.

Se han calculado las matrices de distancias y tiempos, que en este caso también son de 27×27 , para los nodos que se enumeran y representan a continuación:

Orígenes (Saltillo): **Nodos 1 a 21** que se corresponden con los puntos representados en el gráfico siguiente.



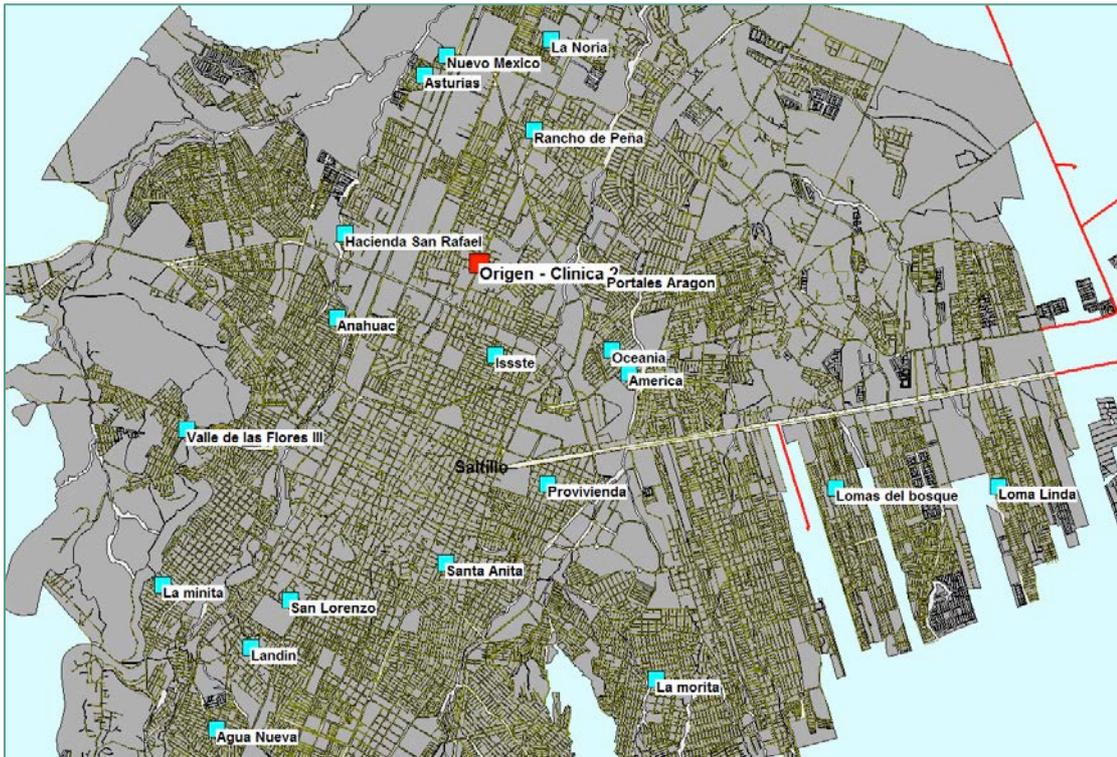
Destinos (Monterrey): **Nodo 22:** Clínica 25 / **Nodo 23:** Clínica 34 / **Nodo 24:** Clínica 21 /
Nodo 25: Clínica 22 / **Nodo 26:** Clínica 33 /
Nodo 27: Final (depósito donde descansan los vehículos)
(análogos a los representados para las instancias real y pseudoreal 1)

VII) INSTANCIA PSEUDOREAL 6 [Inst020_05_006]

Se ha generado una tercera instancia de tamaño $n = 20$, con 20 nodos diferentes de recogida en Saltillo y los 5 hospitales de descarga en Monterrey. Para realizar los traslados, se cuenta de nuevo con 6 vehículos, $m = 6$, con capacidad para 5 pacientes, $Q_{max} = 5$. Y, también en este caso, $L_{max} = 150$ minutos.

Se han calculado las matrices de distancias y tiempos, que son de 27×27 , para los nodos que se enumeran y representan a continuación:

Orígenes (Saltillo): **Nodos 1 a 21** que se corresponden con los puntos representados en el gráfico siguiente.



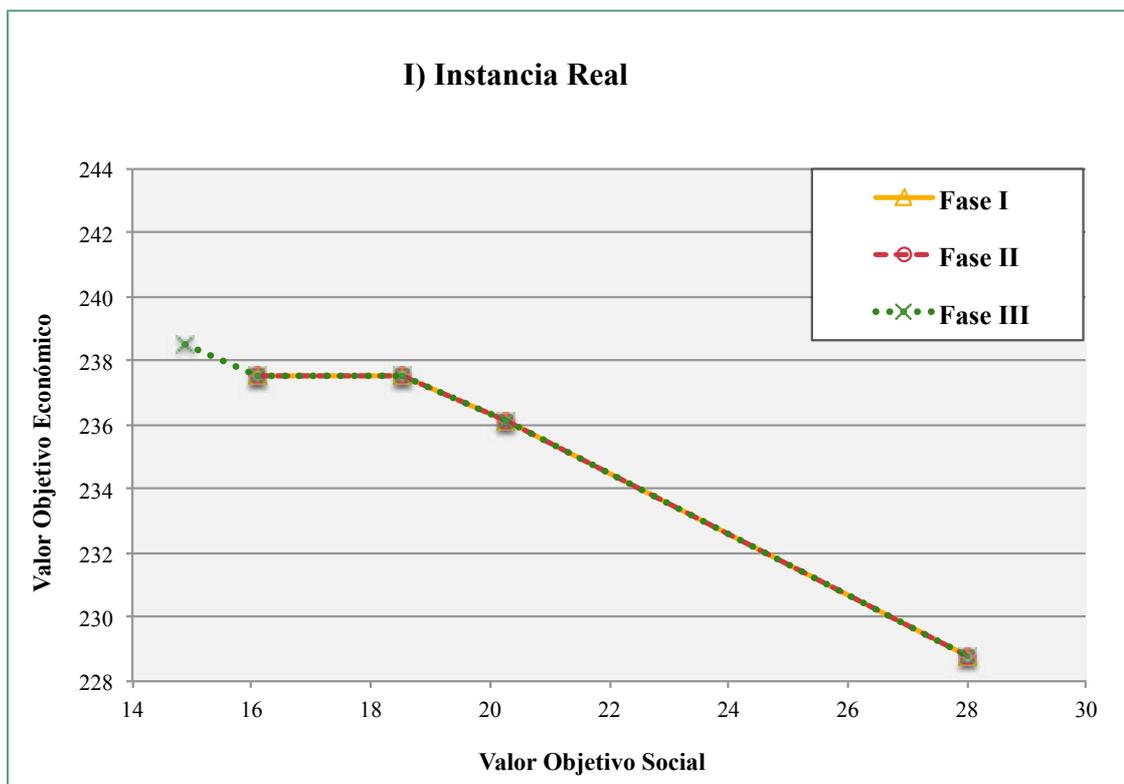
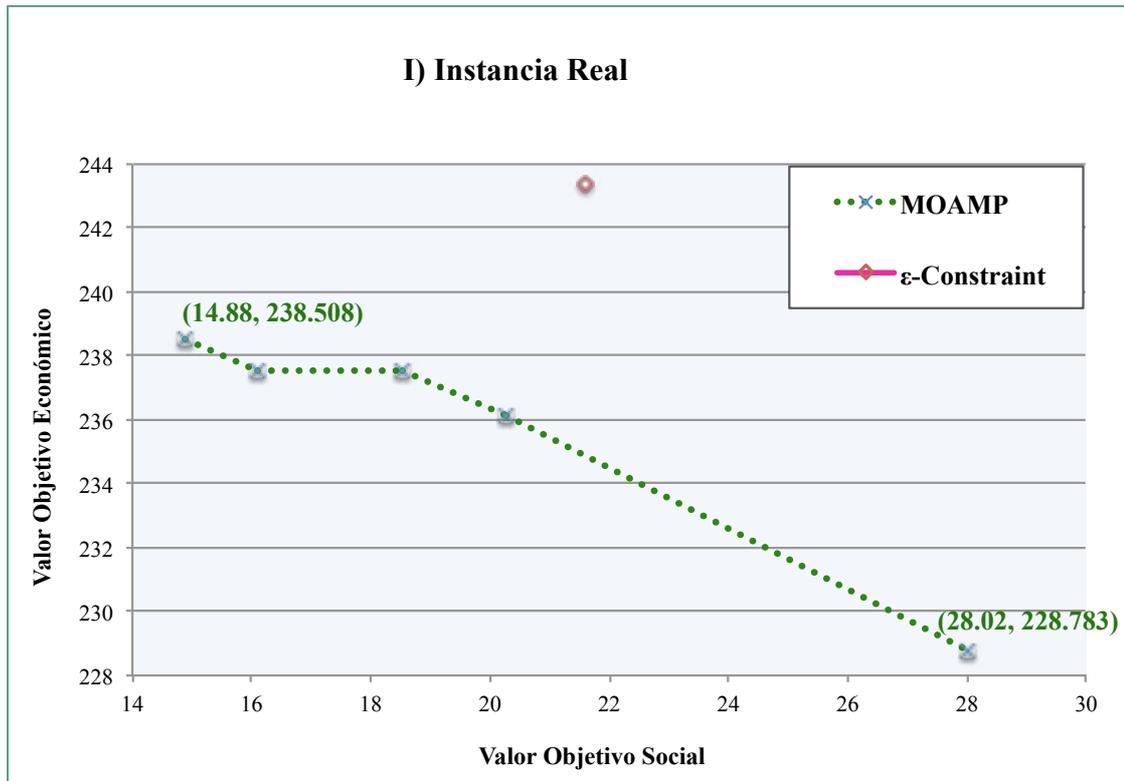
Destinos (Monterrey): **Nodo 22:** Clínica 25 / **Nodo 23:** Clínica 34 / **Nodo 24:** Clínica 21 /
Nodo 25: Clínica 22 / **Nodo 26:** Clínica 33 /
Nodo 27: Final (depósito donde descansan los vehículos)
 (análogos a los representados para las instancias real y pseudoreal 1)

10.7.3.- Los resultados

Las siete instancias, real y pseudoreales, anteriormente definidas se van a resolver empleando MOAMP y MHeC. La resolución, pasa por obtener el conjunto de soluciones no dominadas con ambas estrategias, que se reflejarán sobre un mismo gráfico a fin de observar y comparar las características de los mismos. Así mismo, se presentan las correspondientes gráficas detalladas que incorporan el conjunto de soluciones no dominadas obtenido en cada una de las tres fases del método MOAMP para cada instancia.

Por último, se muestran las rutas de la solución social y de la solución económica representadas en la plataforma NEVA para cada instancia resuelta.

I) INSTANCIA REAL



Para la instancia real analizada MOAMP obtiene la solución social, (14.88 ,238.508), empleando los dos vehículos disponibles y diseñando las dos rutas que se presentan seguidamente.



[RUTA 1] **Nodo 1:** Clínica 2 (depósito de donde salen los vehículos) -- **Nodo 5:** Guayulera -- **Nodo 2:** Centro -- **Nodo 3:** Mirasierra -- **Nodo 8:** Clínica 34 -- **Nodo 7:** Clínica 25 -- **Nodo 10:** Clínica 22 -- **Nodo 12:** Final (depósito donde descansan los vehículos)



[RUTA 2] **Nodo 1:** Origen -- **Nodo 6:** Lourdes -- **Nodo 4:** Saltillo2000 -- **Nodo 11:** Clínica 33 -- **Nodo 9:** Clínica 21 -- **Nodo 12:** Final

La solución económica de la instancia real, (28.02 , 228.783), también es obtenida por MOAMP haciendo uso de los dos vehículos y diseñando las dos rutas que siguen.

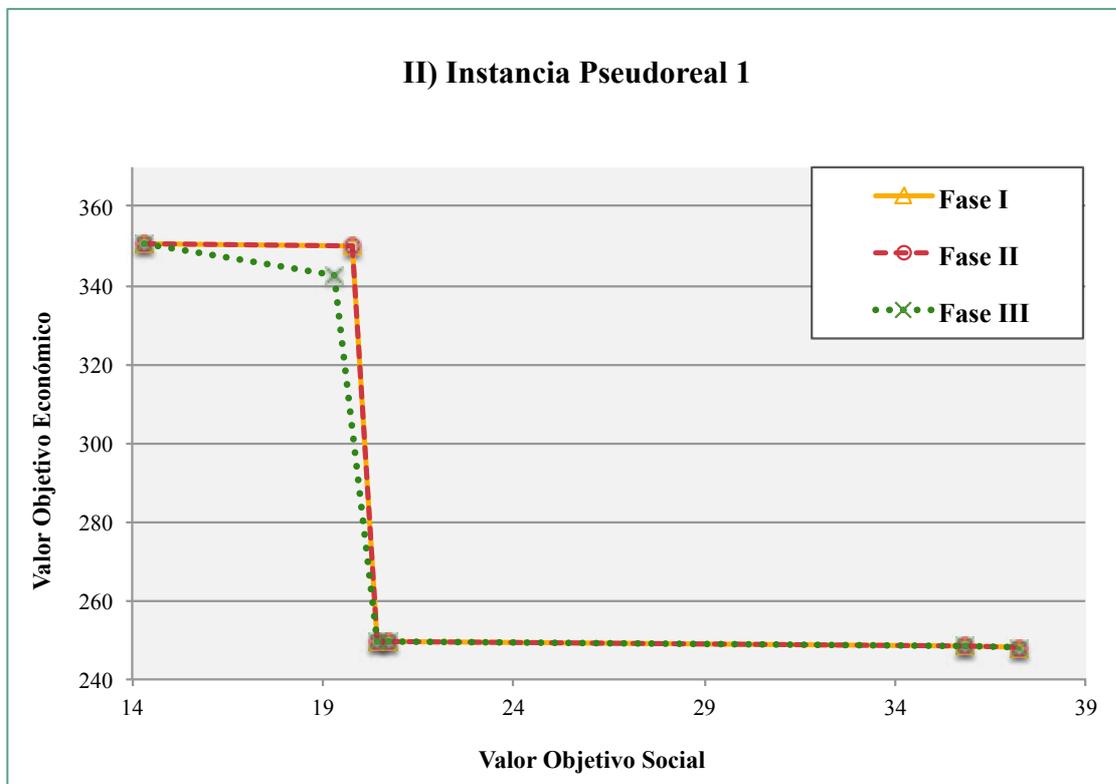
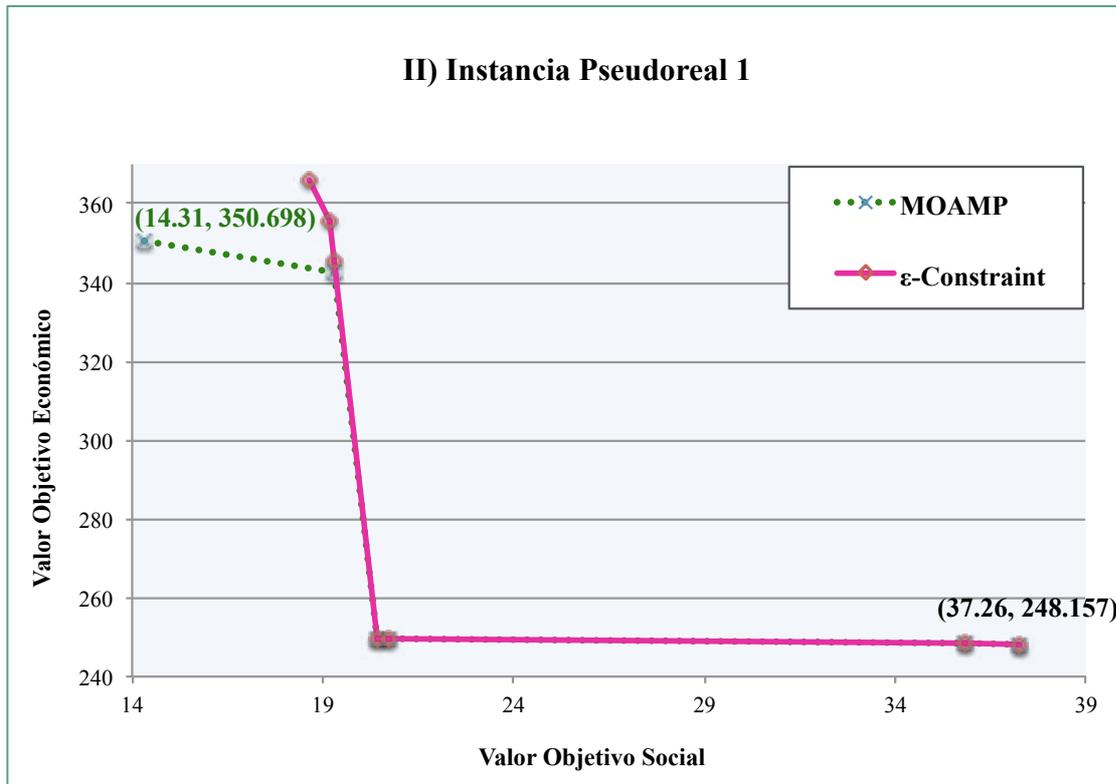


[RUTA 1] Nodo 1: Origen -- **Nodo 2:** Centro -- **Nodo 3:** Mirasierra -- **Nodo 8:** Clínica 34 -- **Nodo 7:** Clínica 25 -- **Nodo 12:** Final

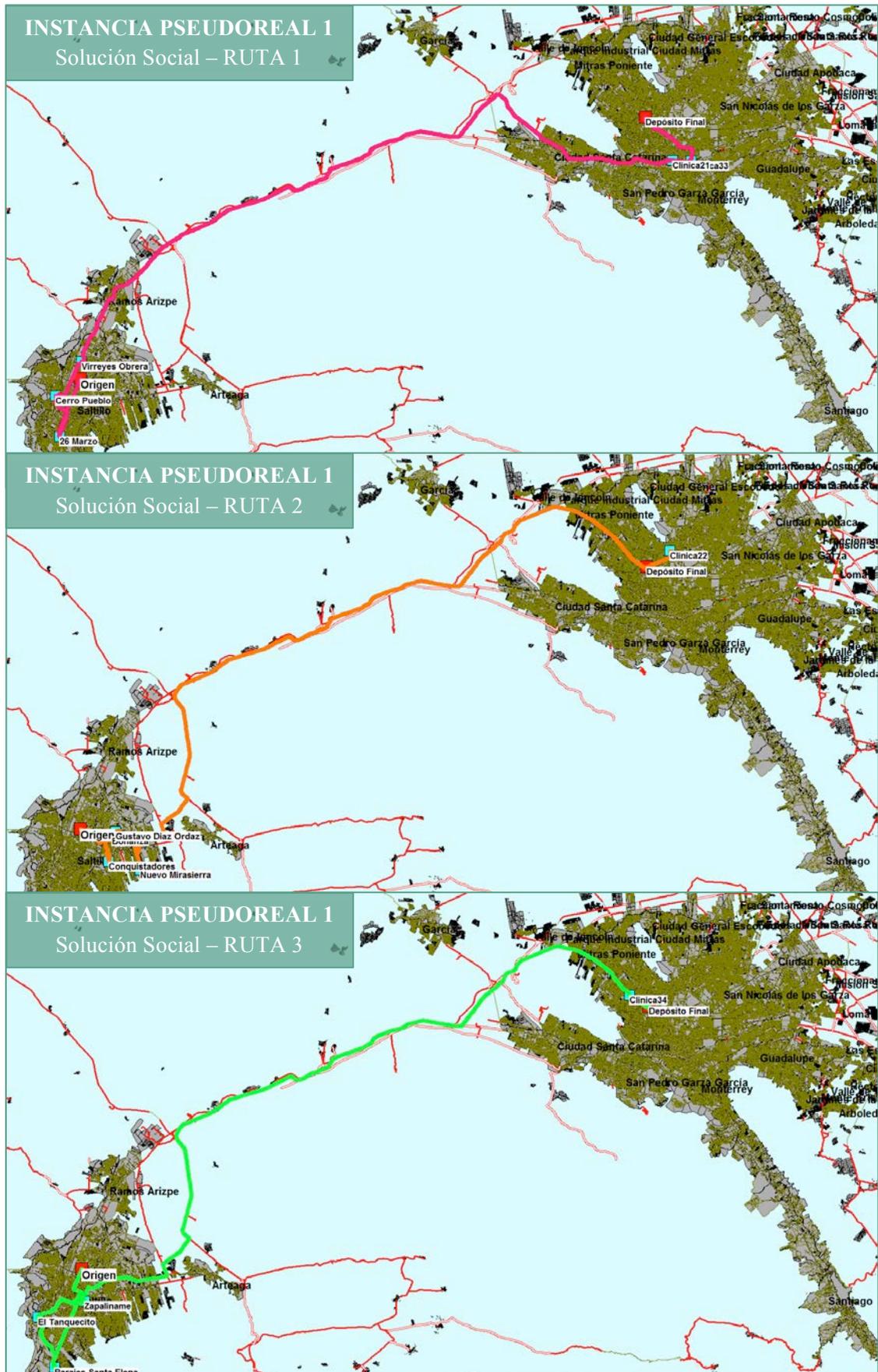


[RUTA 2] **Nodo 1:** Origen -- **Nodo 6:** Lourdes -- **Nodo 5:** Guayulera -- **Nodo 4:** Saltillo2000 -- **Nodo 9:** Clínica 21 -- **Nodo 11:** Clínica 33 -- **Nodo 10:** Clínica 22 -- **Nodo 12:** Final

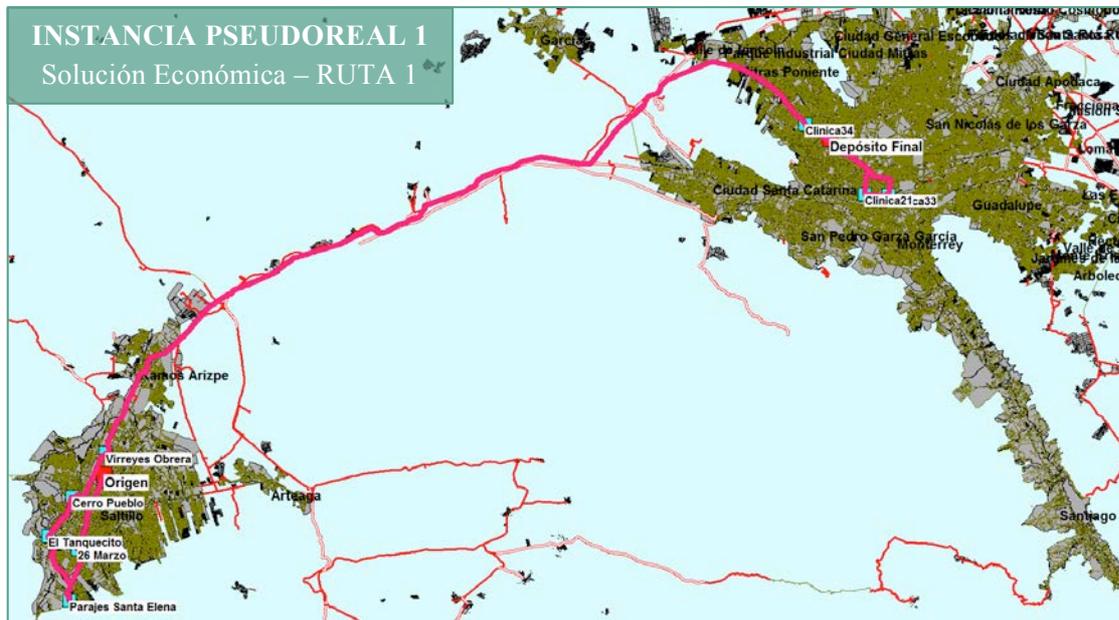
II) INSTANCIA PSEUDOREAL 1



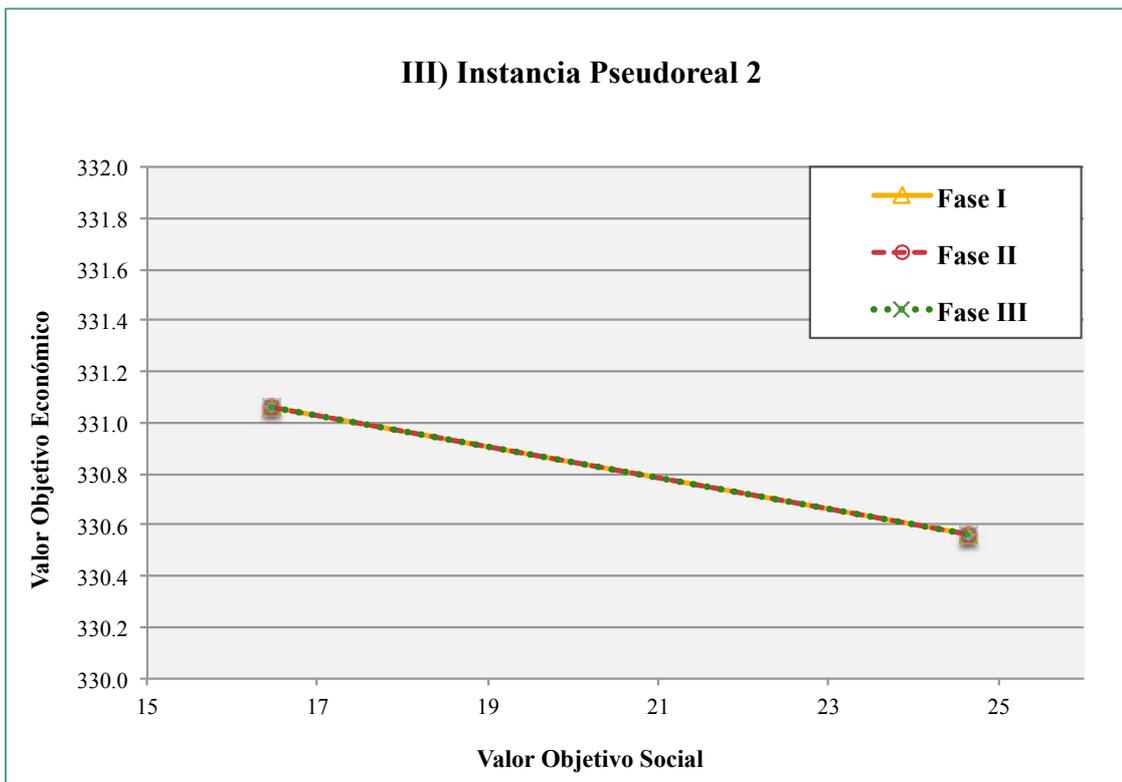
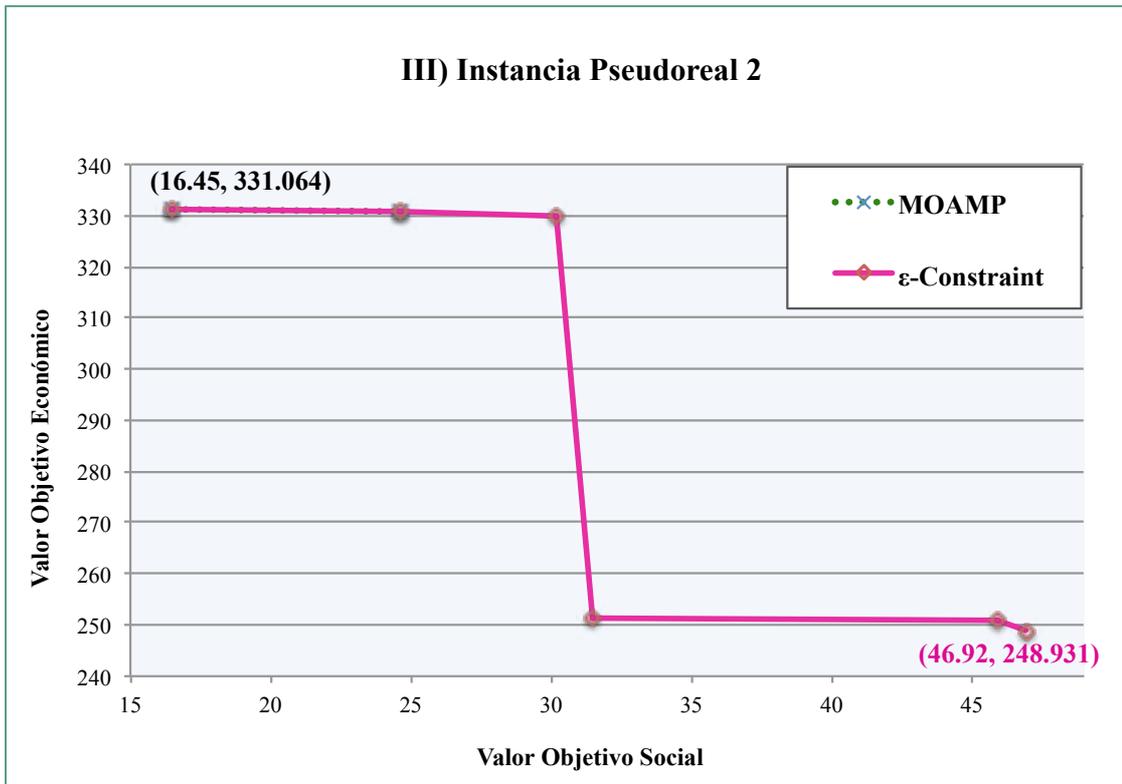
Para la instancia pseudoreal 1, MOAMP obtiene la solución social, (14.31 , 350.698), empleando los tres vehículos disponibles y diseñando las tres rutas que siguen:



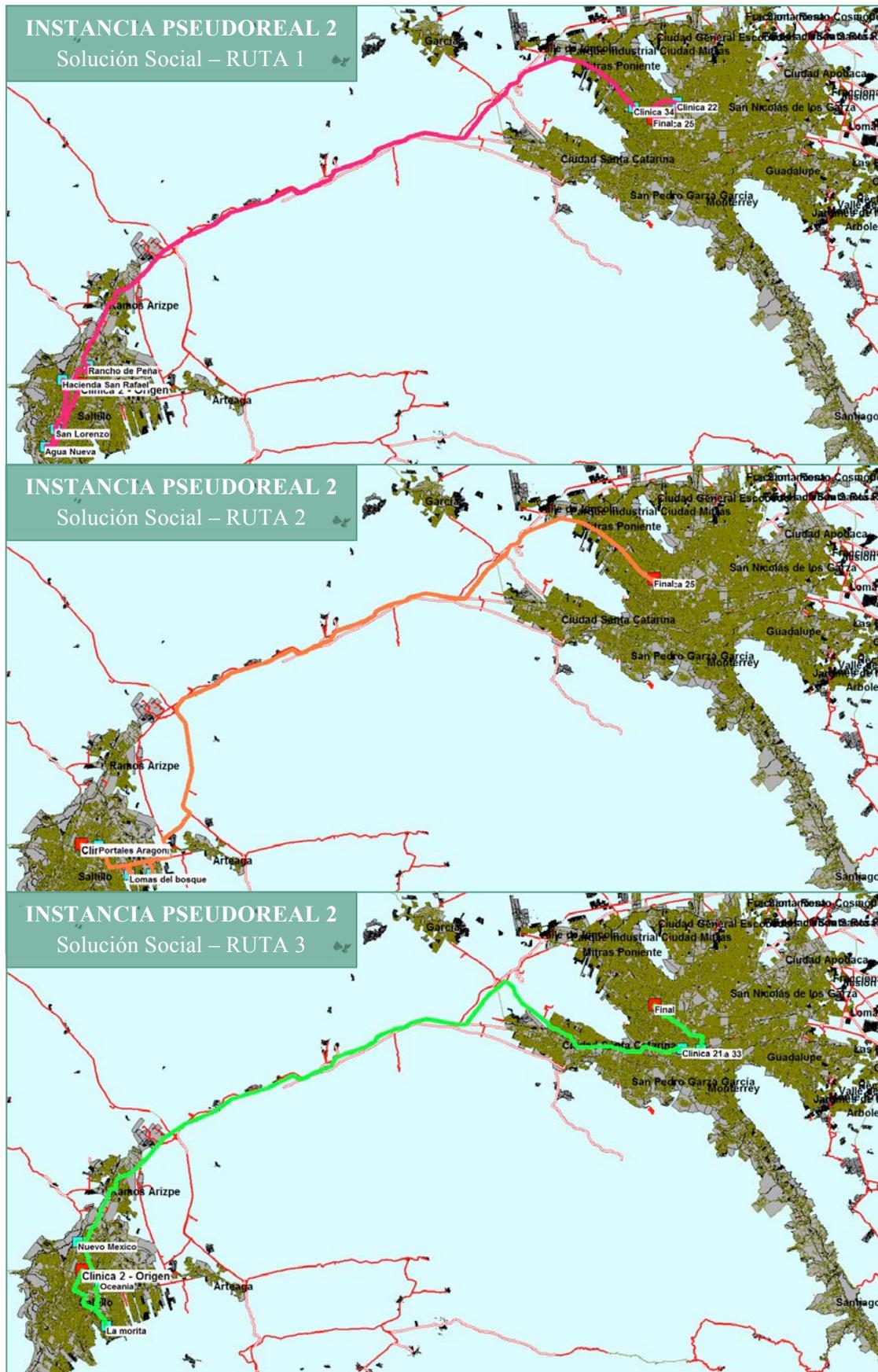
La solución económica para la instancia pseudoreal 1, (37.26, 248.157), es obtenida por ambos métodos, MOAMP y MHeC, haciendo uso de dos vehículos y diseñando las dos rutas que siguen.



III) INSTANCIA PSEUDOREAL 2



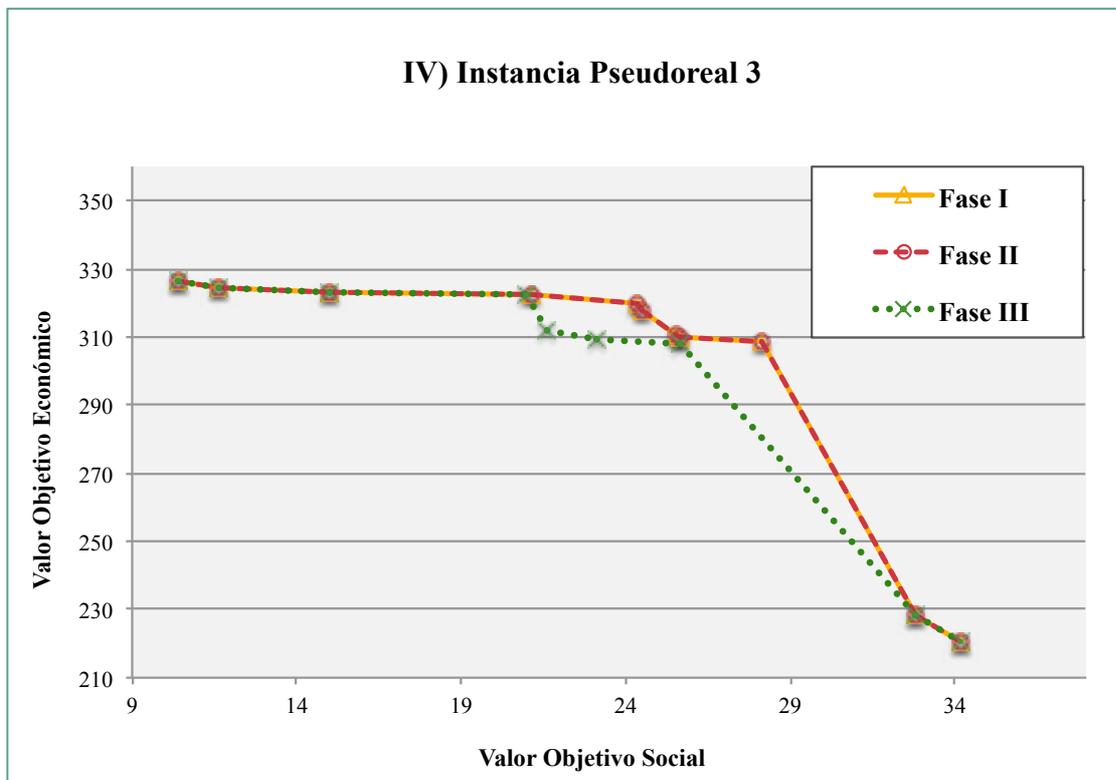
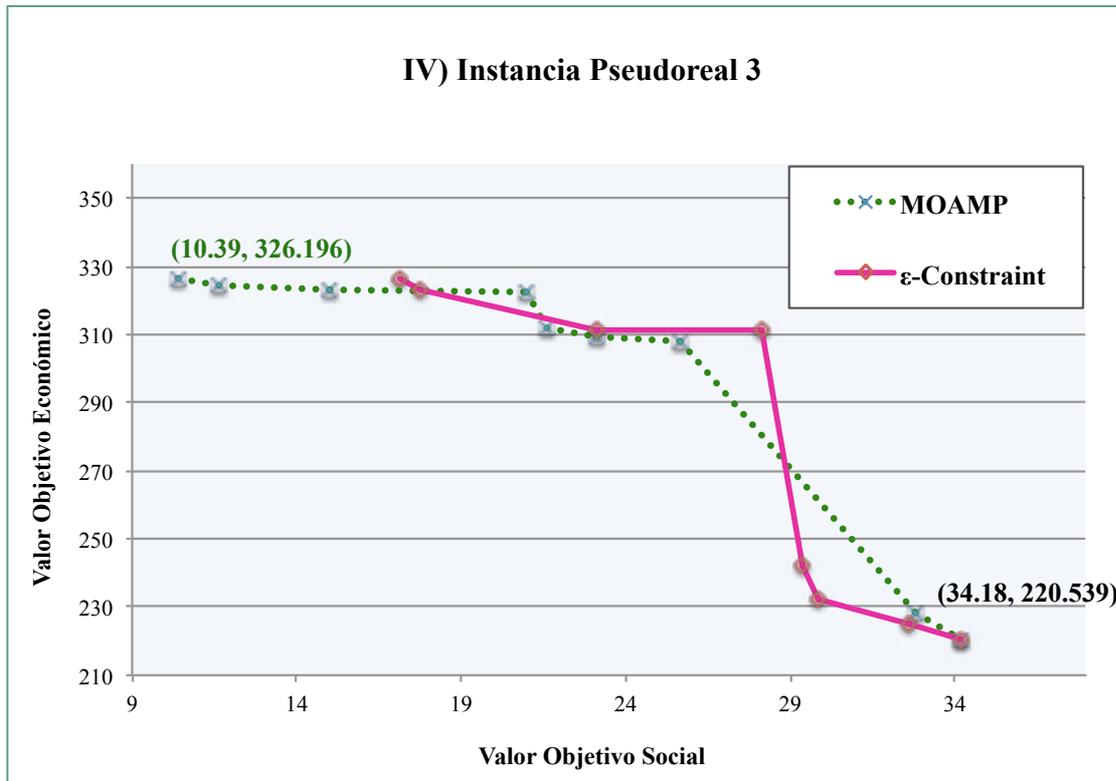
Para la instancia pseudoreal 2, tanto MOAMP como MHeC, obtienen la solución social, (16.45 , 331.064), empleando los tres vehículos disponibles y diseñando las rutas que siguen.



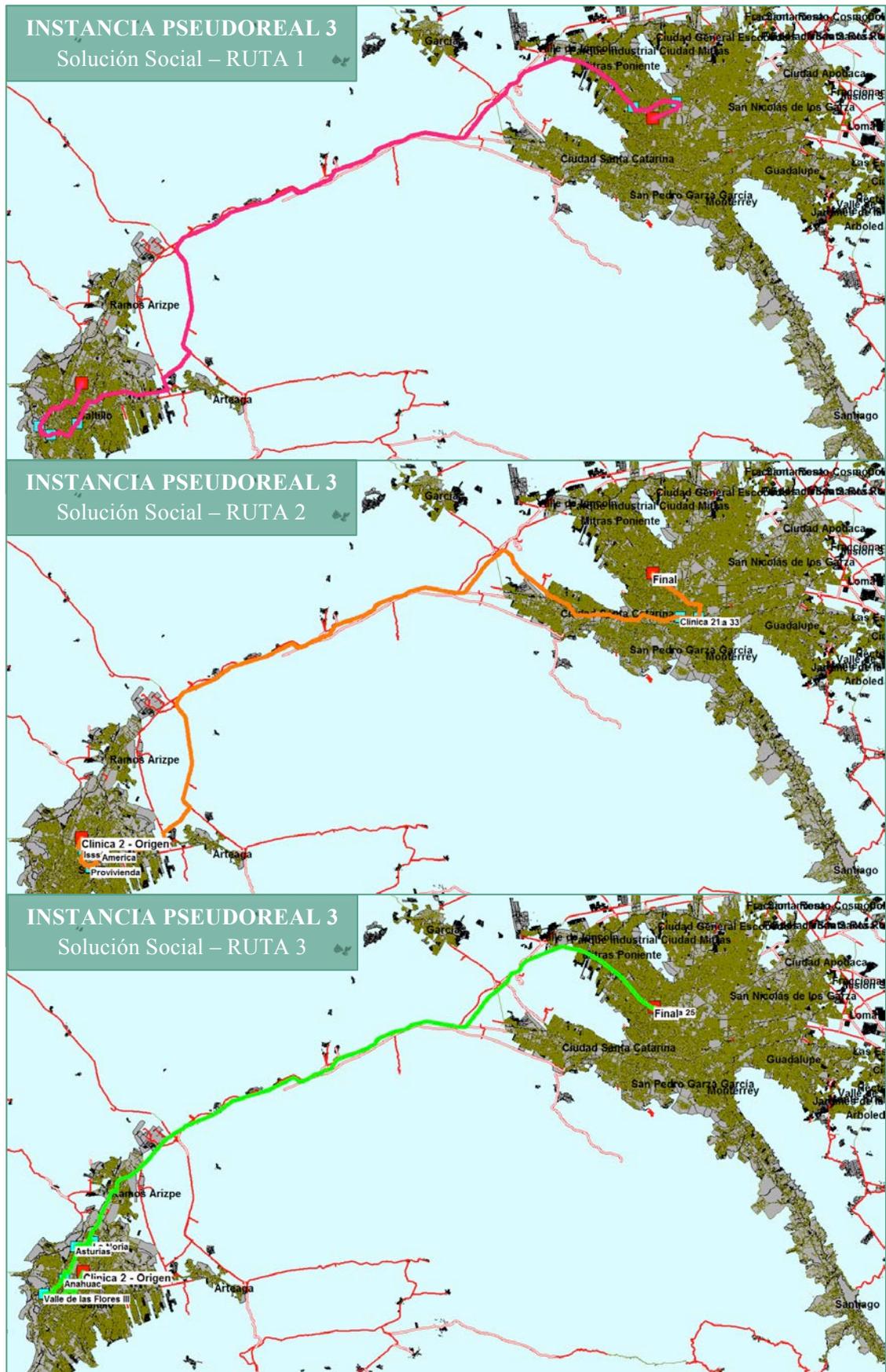
La solución económica para la instancia pseudoreal 2, (46.92 , 248.931), es obtenida por el método MHeC, haciendo uso únicamente de dos vehículos y diseñando las dos rutas representadas a continuación.



IV) INSTANCIA PSEUDOREAL 3



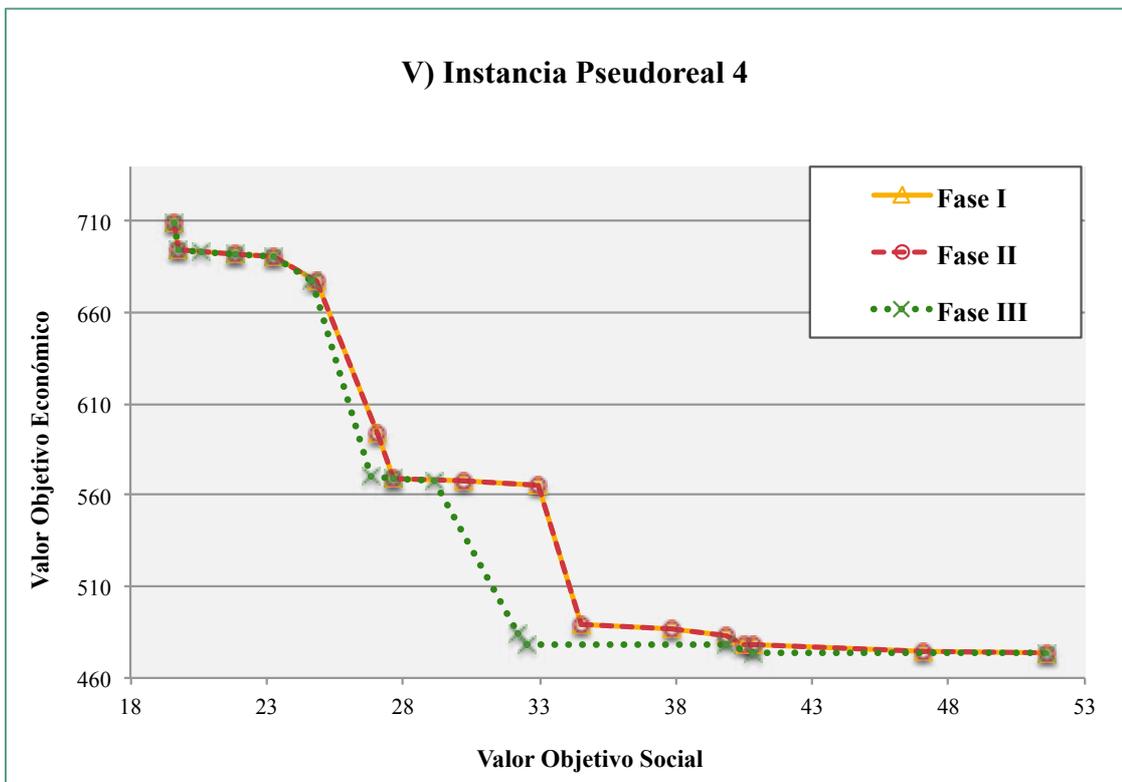
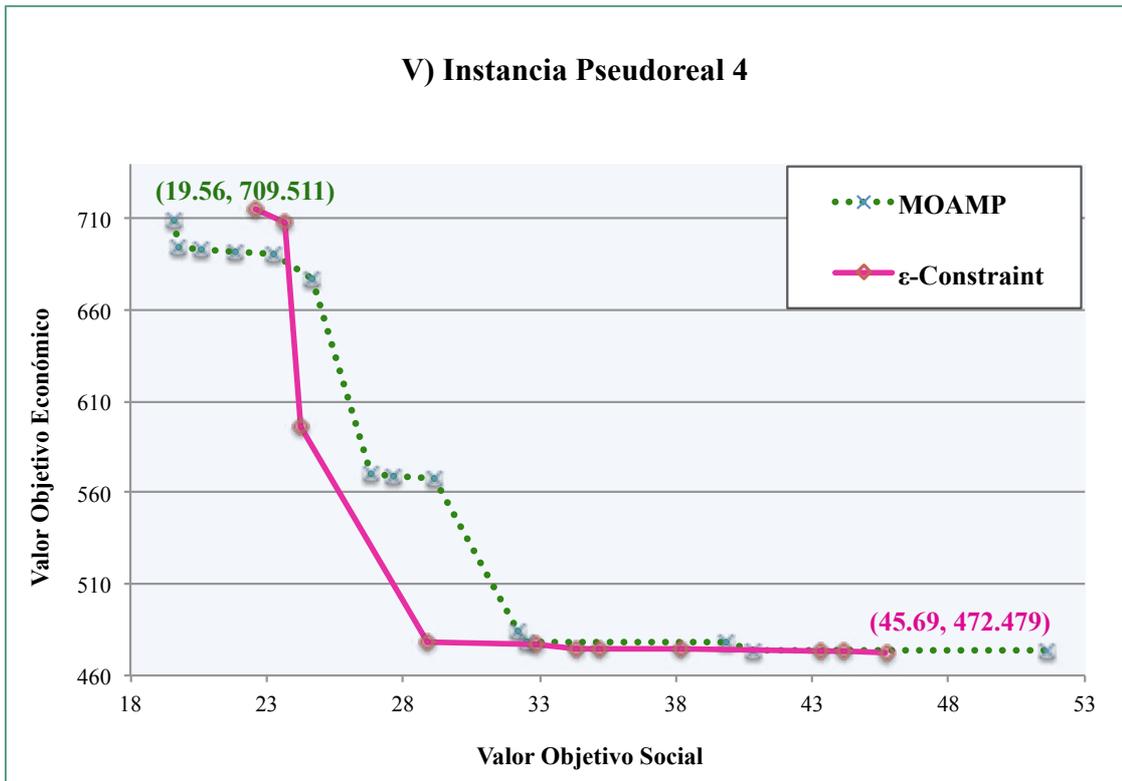
En el caso de la instancia pseudoreal 3, la solución social es obtenida por MOAMP, (10.39 , 326.196), empleando los tres vehículos disponibles y diseñando las rutas que siguen.



La solución económica para la instancia pseudoreal 3, (34.18, 220.539), es obtenida por ambos métodos, MOAMP y MHeC, haciendo uso únicamente de dos vehículos y diseñando las dos rutas representadas a continuación.

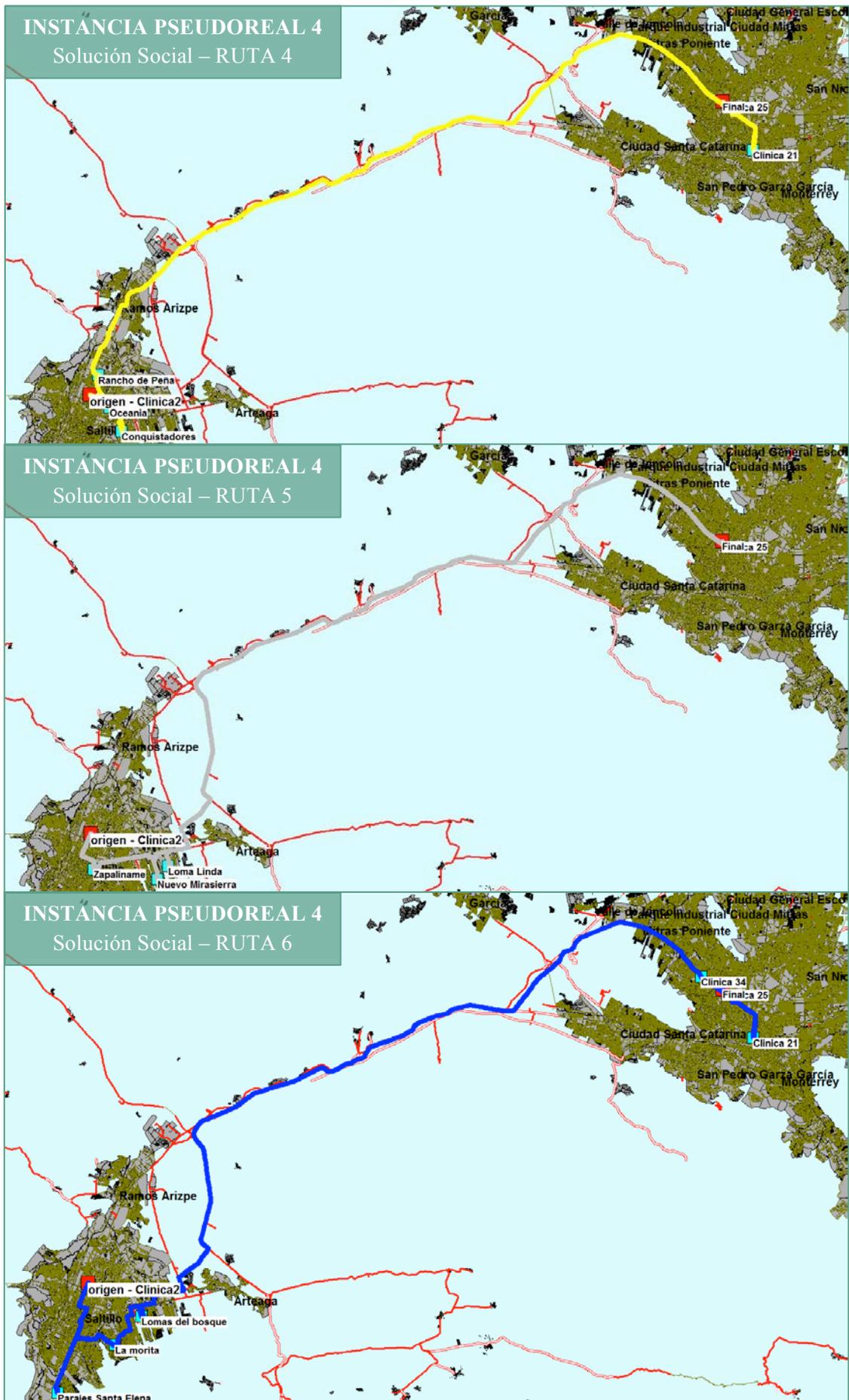


V) INSTANCIA PSEUDOREAL 4

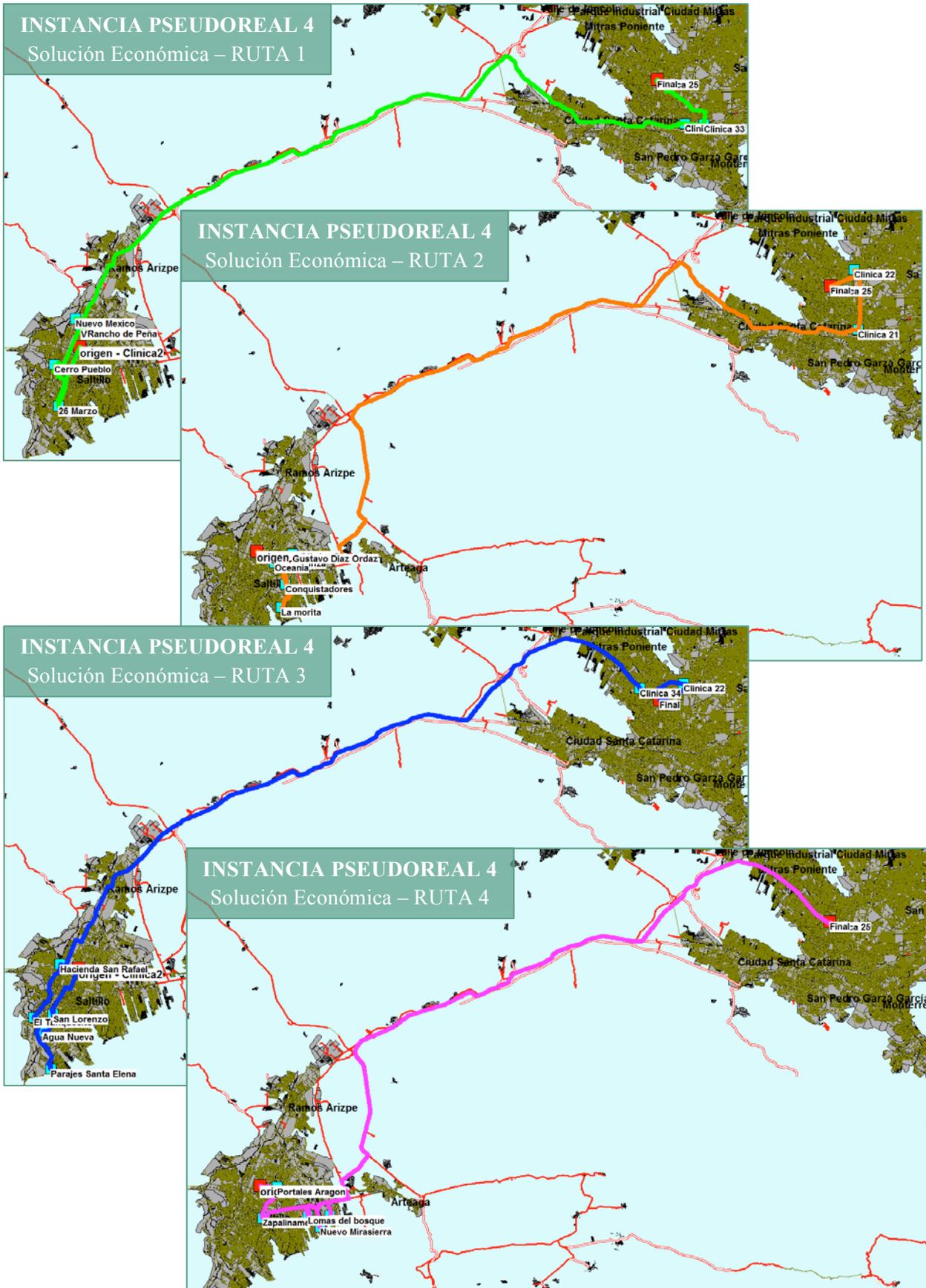


En el caso de la instancia pseudoreal 4, la solución social es de nuevo obtenida por MOAMP, (19.56 , 709.511), empleando los seis vehículos disponibles y diseñando las rutas que siguen.

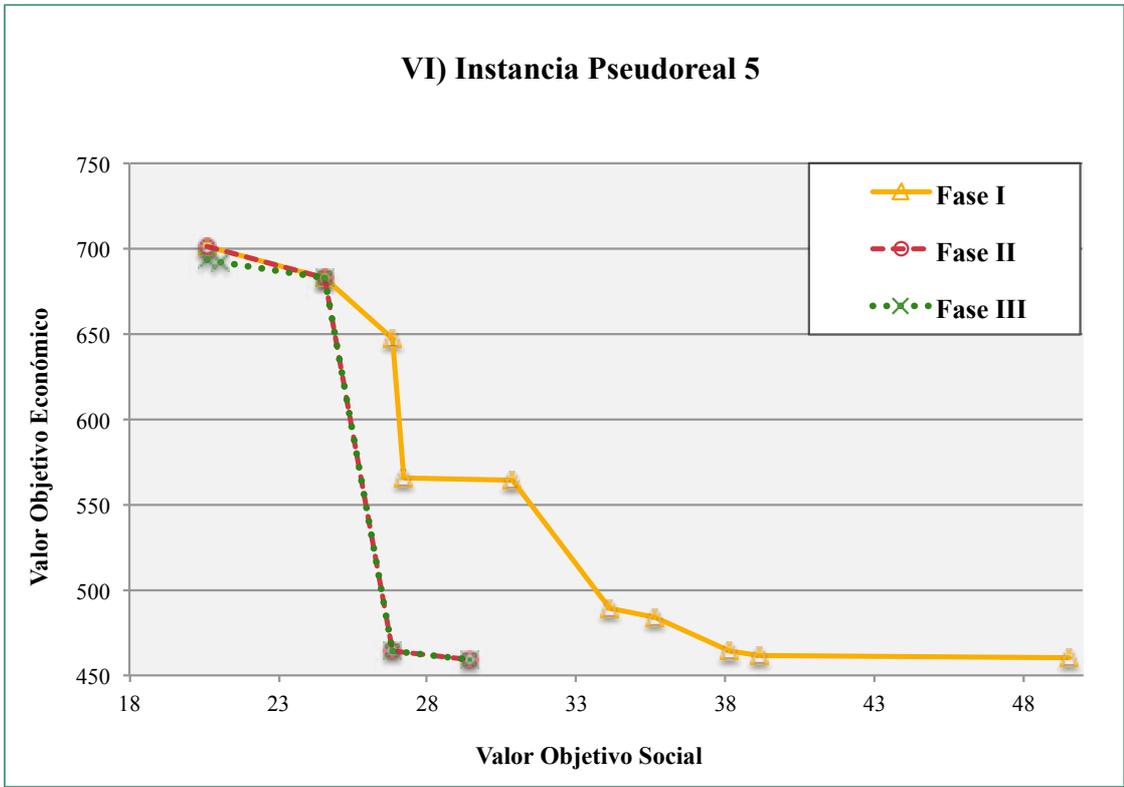
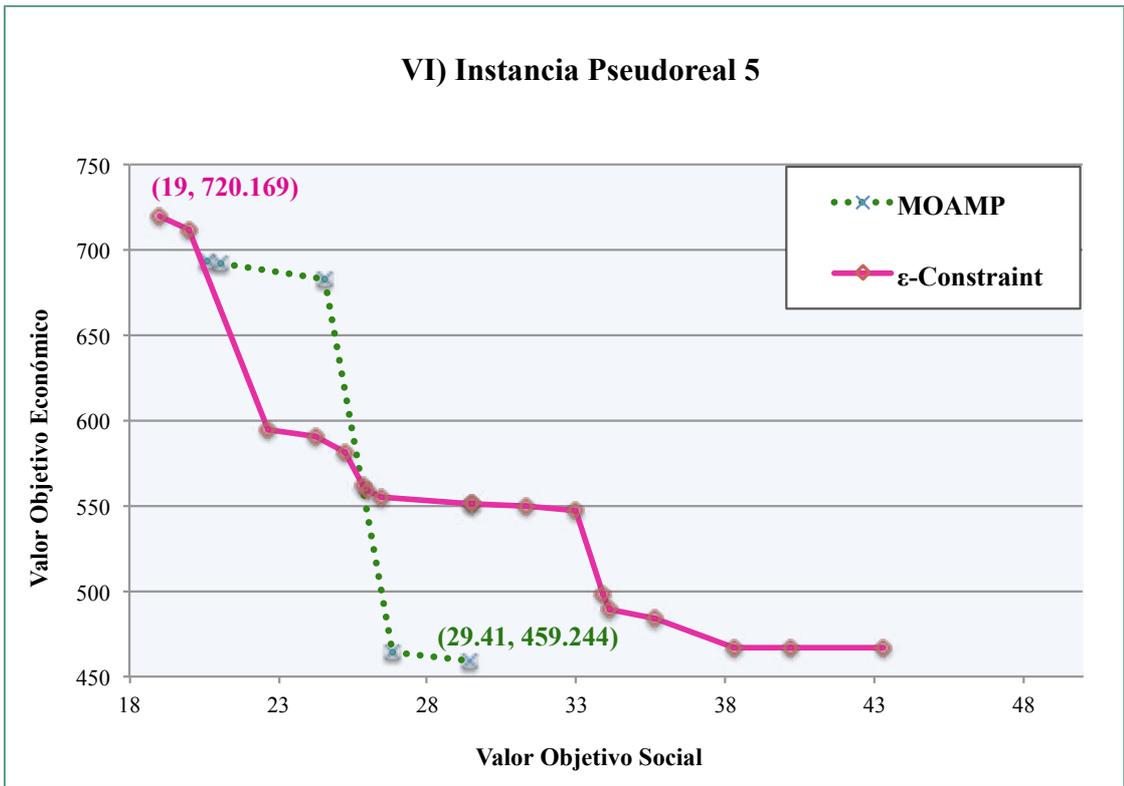




La solución económica para la instancia pseudoreal 4, (45.69, 472.479), es obtenida por el método MHeC, haciendo uso de cuatro vehículos y diseñando las rutas que siguen.

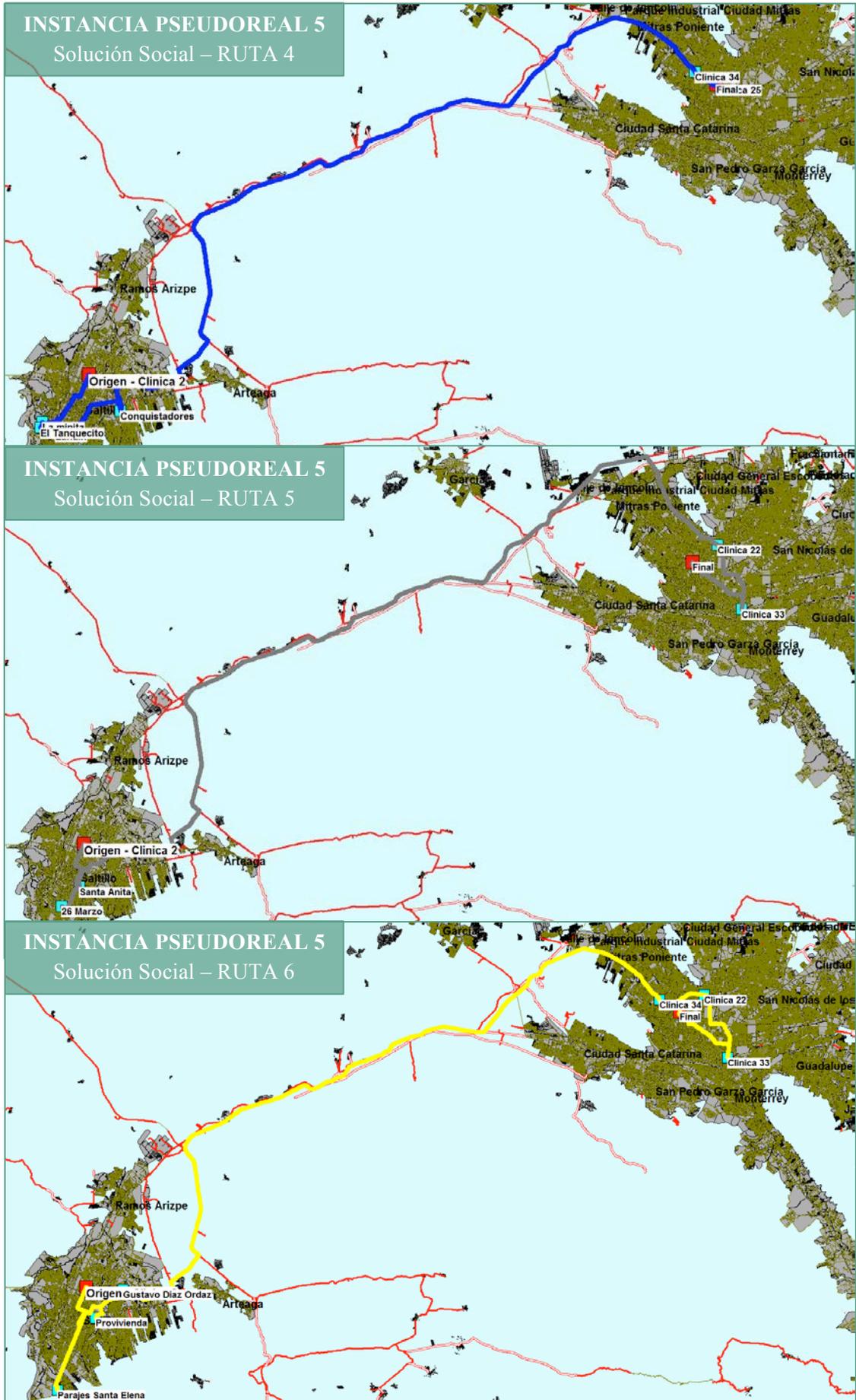


VI) INSTANCIA PSEUDOREAL 5

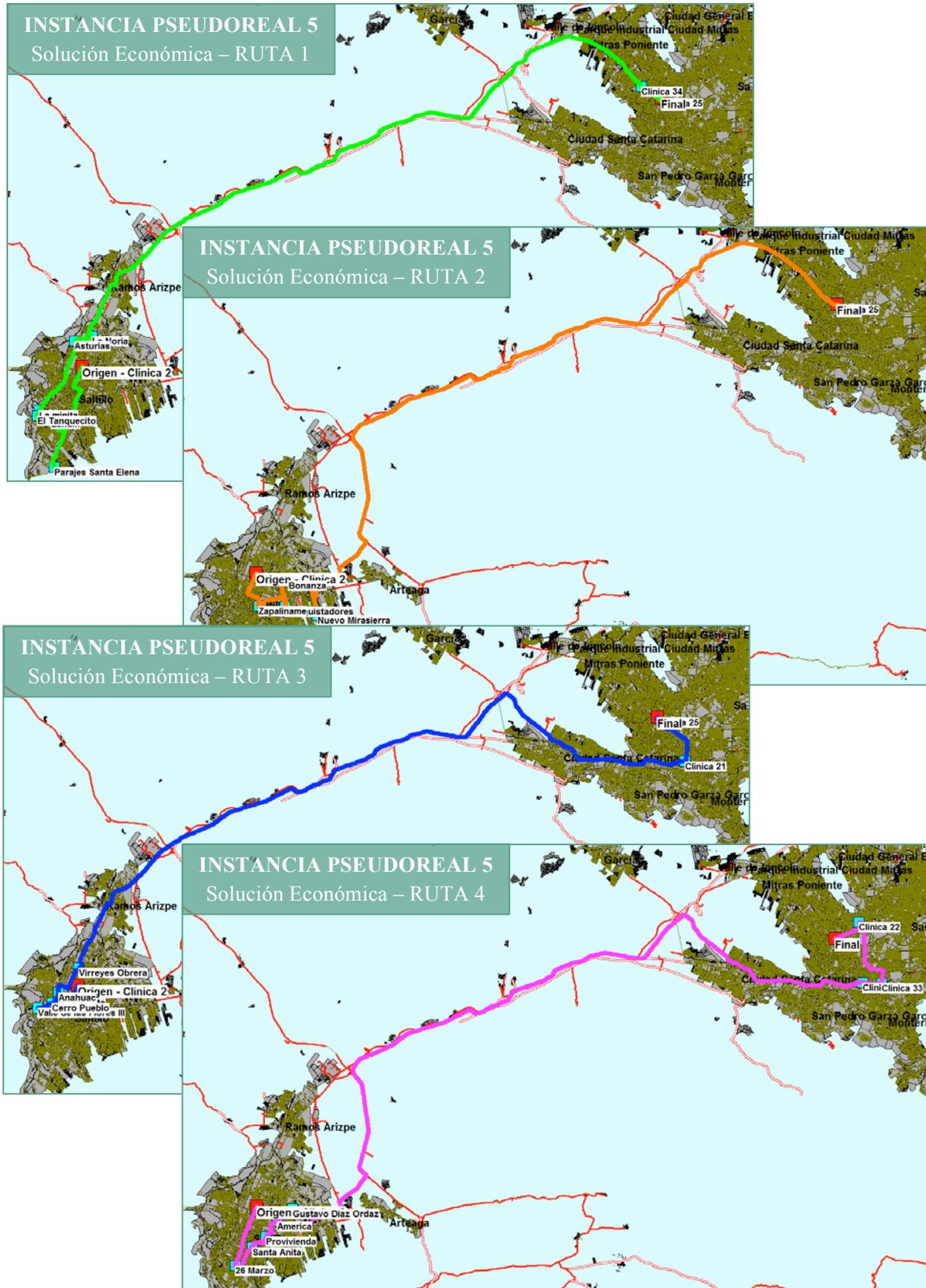


En el caso de la instancia pseudoreal 5, la solución social es obtenida por MHeC, (19.00 , 720.169), empleando los seis vehículos disponibles y diseñando las rutas que siguen.

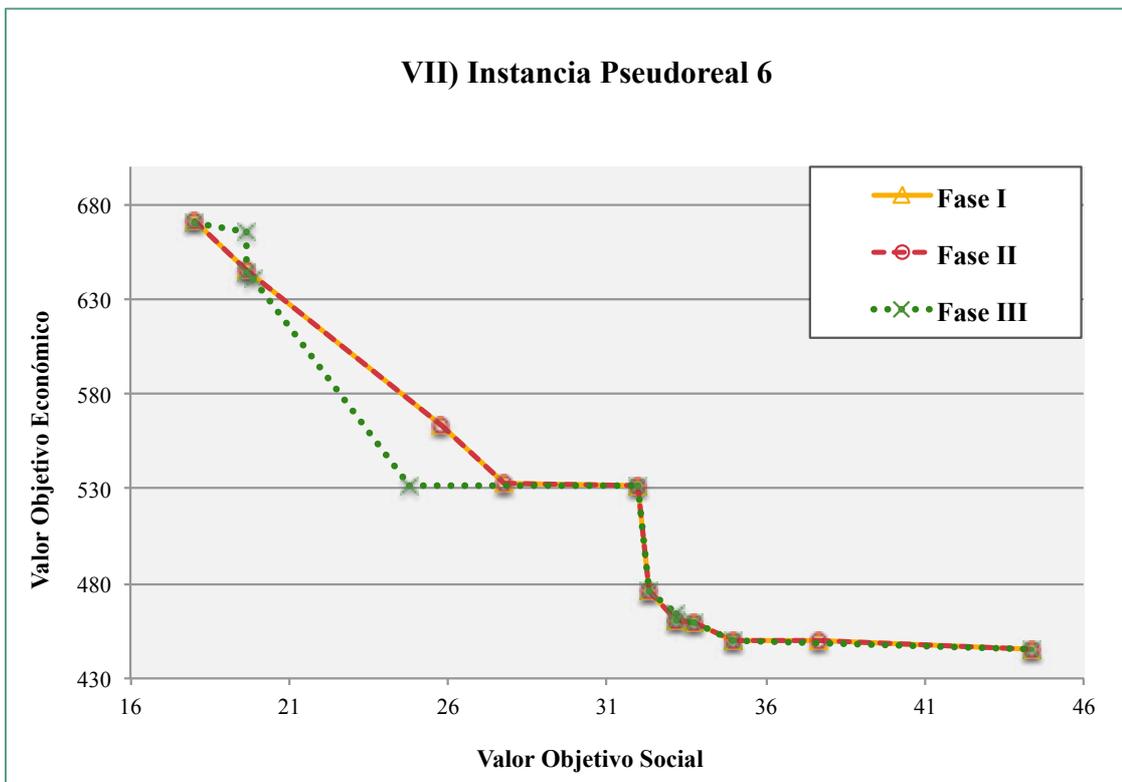
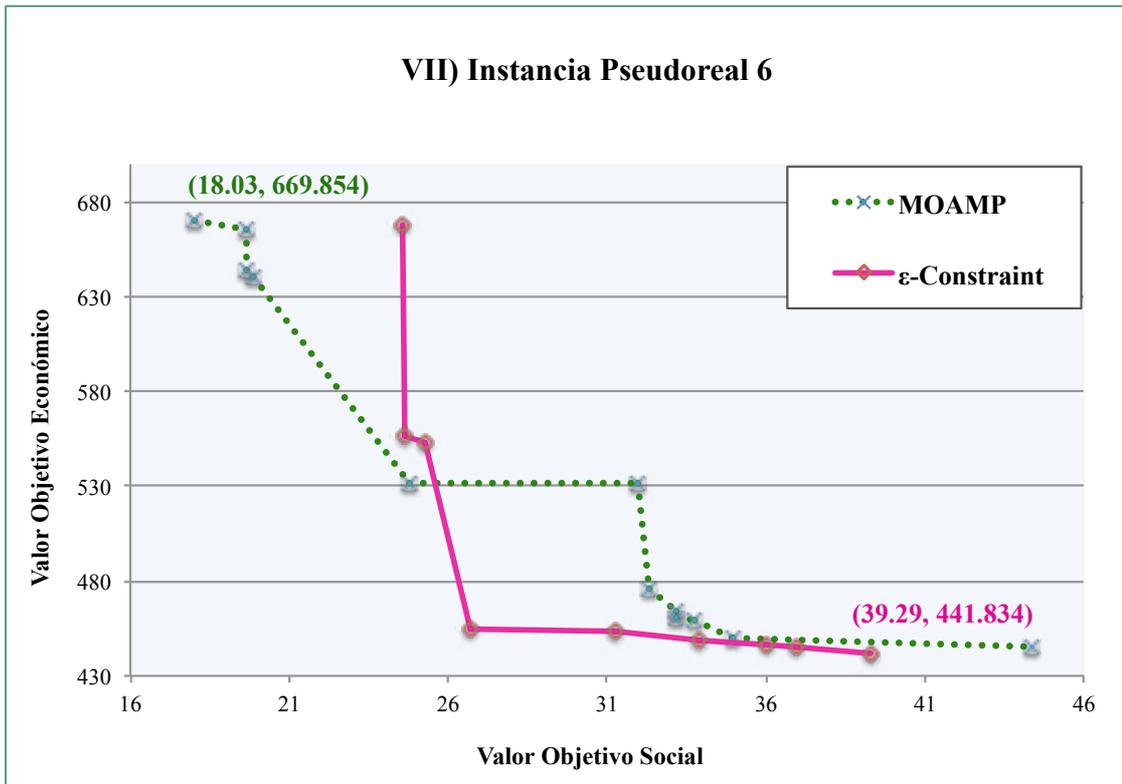




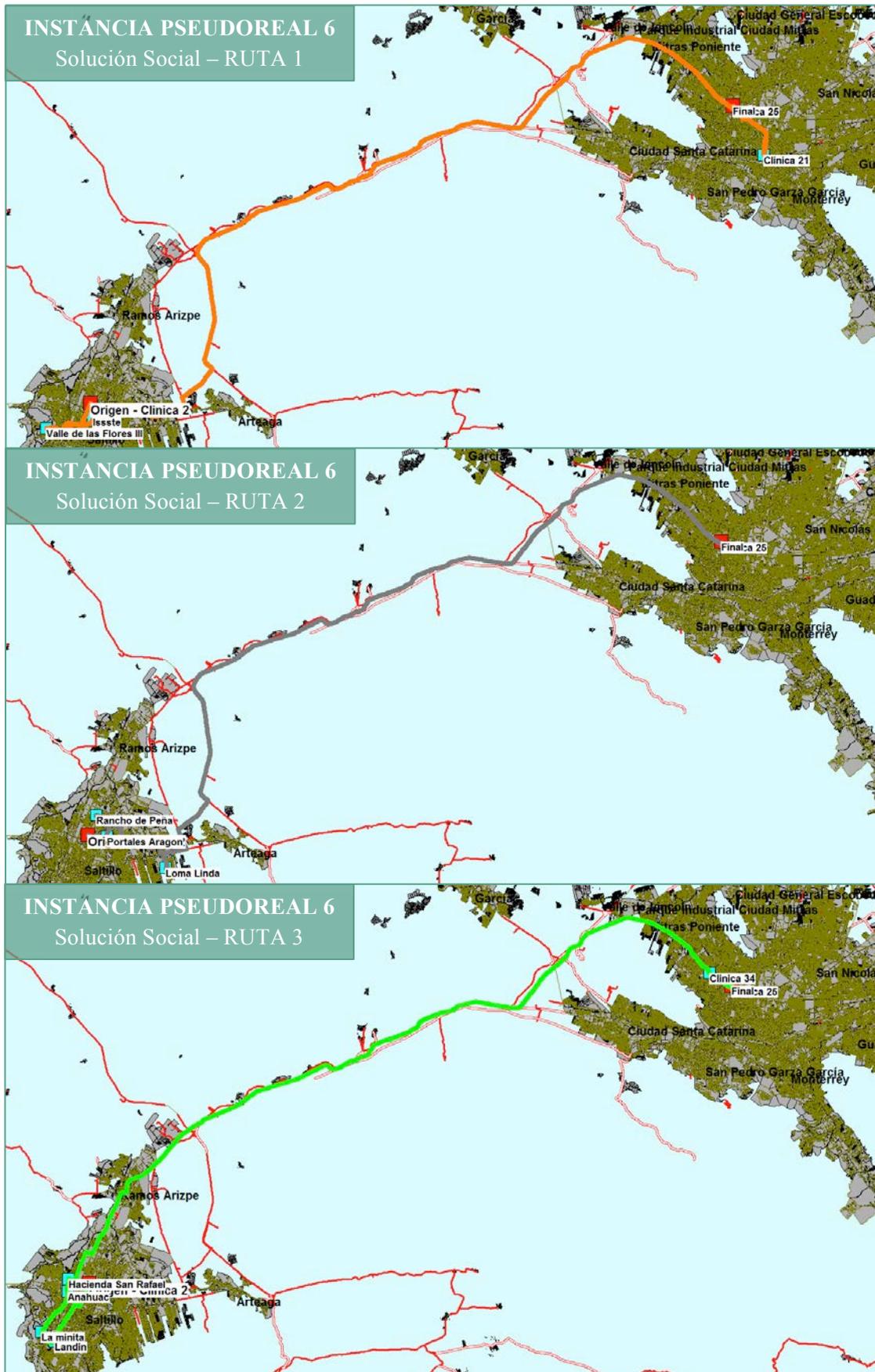
La solución económica en el caso de la instancia pseudoreal 5, (29.41 ,459.244), es obtenida por el método MOAMP, haciendo uso de cuatro vehículos y diseñando las rutas que siguen.

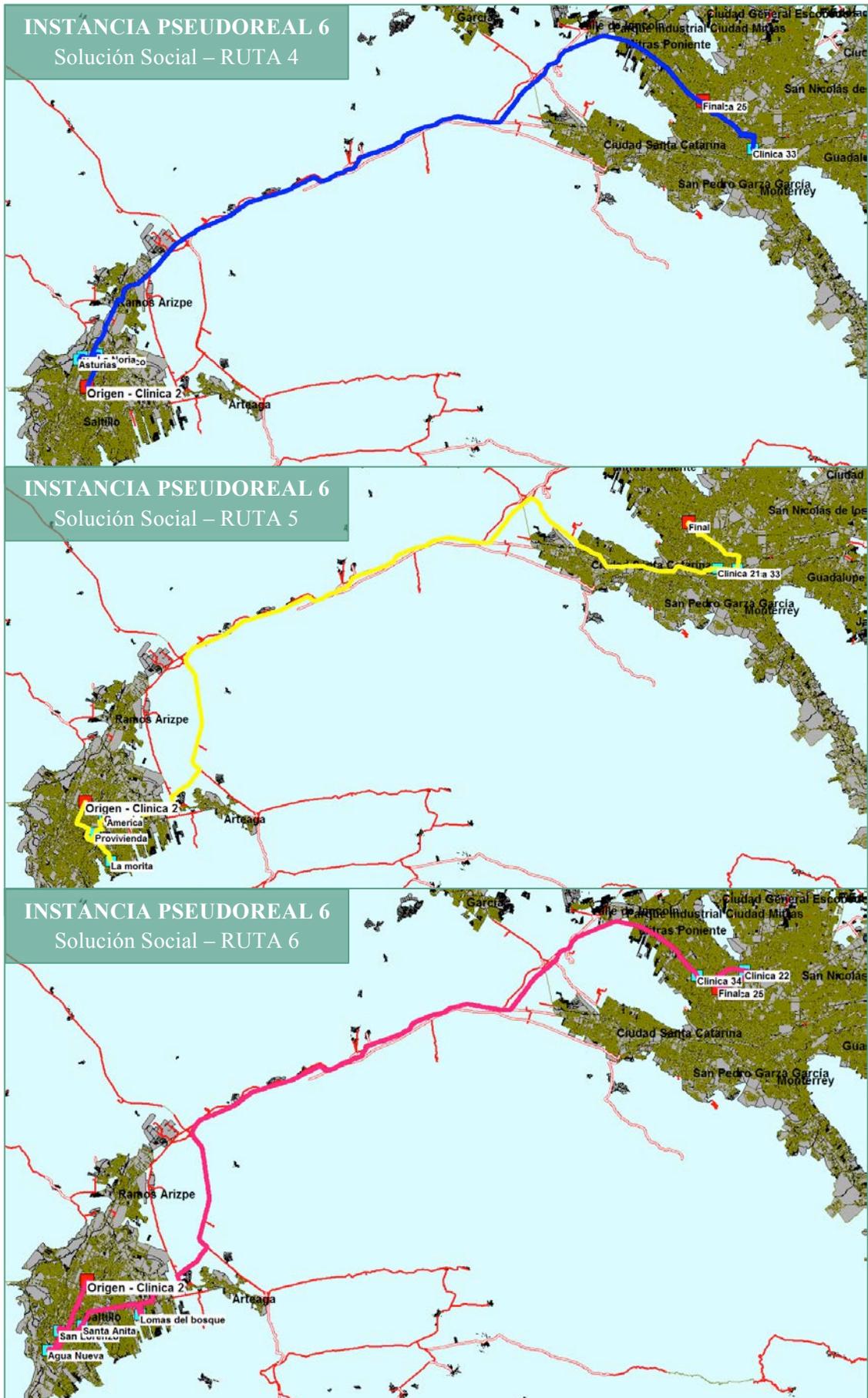


VII) INSTANCIA PSEUDOREAL 6

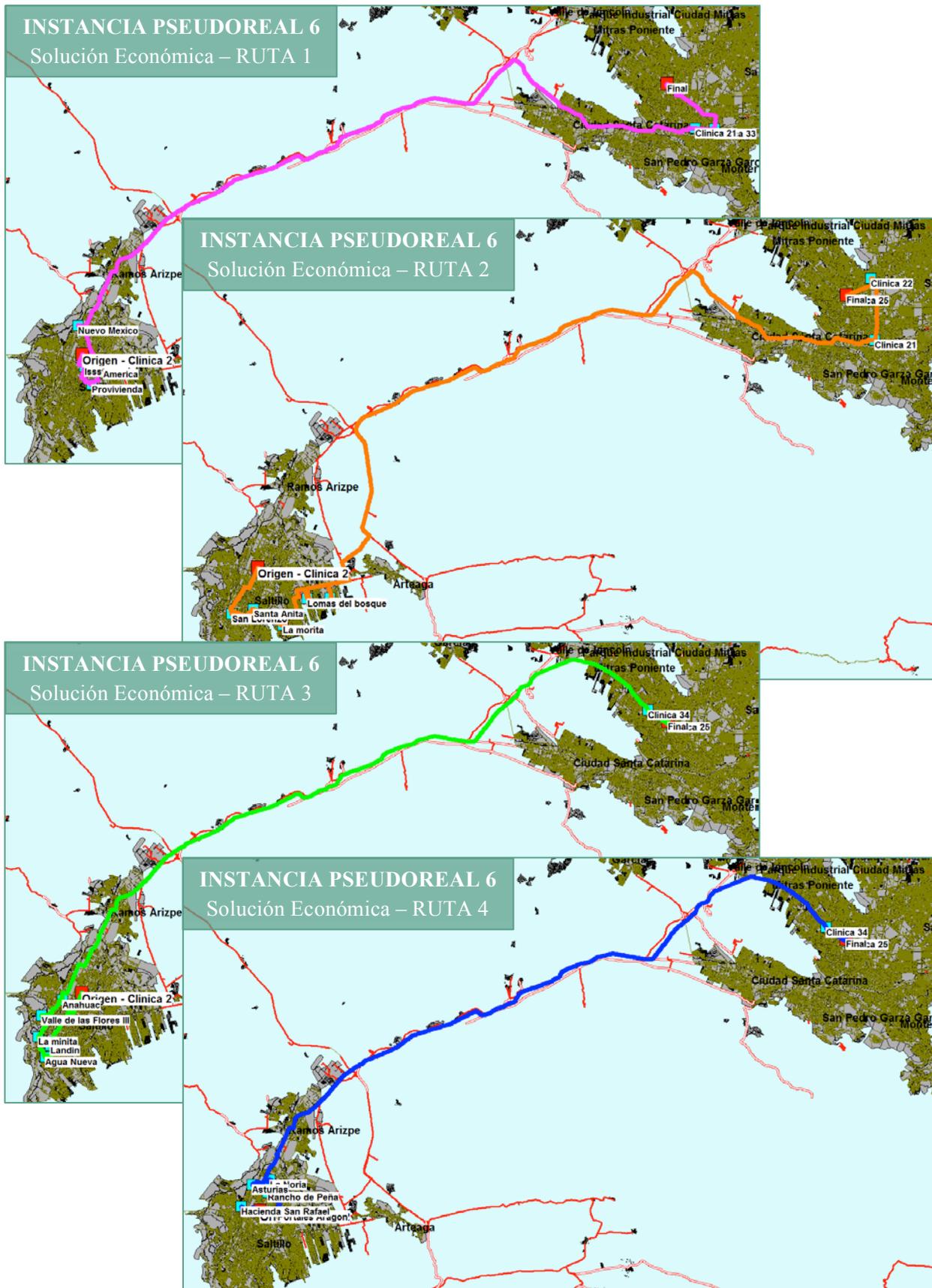


En el caso de la instancia pseudoreal 6, la solución social es obtenida por MOAMP, (18.03 , 669.854), empleando los seis vehículos disponibles y diseñando las rutas que siguen.





La solución económica en el caso de la instancia pseudoreal 6, (39.29 ,441.834), es obtenida por el método MHeC, haciendo uso únicamente de cuatro vehículos y diseñando las rutas que siguen.



Se aprecia, al igual que en las instancias ficticias, como para las instancias de menor tamaño los conjuntos de soluciones no dominadas obtenidos por MOAMP son mejores que los conseguidos por el método MHeC.

En seis de las siete instancias reales y pseudoreales analizadas la mejor solución social es obtenida por MOAMP, es decir, presenta un mejor comportamiento en la “esquina izquierda”.

En cuanto a la “esquina derecha”, se observan resultados diversos; en tres de las siete instancias el método MHeC obtiene claramente mejores soluciones con respecto a la distancia total recorrida, para dos instancias pseudoreales, de tamaño $n = 10$, ambos métodos consiguen obtener la misma mejor solución económica y en otros dos casos es MOAMP el que marca el extremo en dicha zona.

Finalmente, también parece repetirse que, por lo general, en los puntos intermedios la curva obtenida por MHeC obtiene mejores soluciones que dominan claramente a las obtenidas por MOAMP en las instancias mayores, $n = 20$. Pero en las instancias menores, que tal y como se ha comentado es donde MOAMP se comporta mejor, existen algunas soluciones obtenidas por el método MOAMP que no son dominadas por las conseguidas por el método MHeC.

**11.- CONCLUSIONES, APORTACIONES, REFLEXIONES Y
FUTURAS LÍNEAS DE INVESTIGACIÓN**

11.- CONCLUSIONES, APORTACIONES, REFLEXIONES Y FUTURAS LÍNEAS DE INVESTIGACIÓN

11.1.- CONCLUSIONES

La “independencia” para desplazarse de una persona que, bien por discapacidad o por envejecimiento natural, tenga su movilidad reducida debe ser un objetivo social actual. El hecho de plantear el fomento y la mejora del transporte de este sector de la población como un objetivo social nos hace dirigir la aplicabilidad de los resultados de la investigación al sector público local y regional.

En la actualidad las administraciones necesitan racionalizar sus recursos sin que el nivel de servicio ofrecido a los ciudadanos sufra menoscabo. Es ahora cuando se requieren sistemas de decisión y de gestión de recursos más eficientes que los tradicionalmente empleados.

En este contexto surge este innovador proyecto de investigación que ha desarrollado una herramienta metodológica para resolver el problema de diseño de rutas para trasladar a personas de movilidad reducida, personas de avanzada edad o con algún tipo de discapacidad, que impida su completa independencia a la hora de trasladarse.

Existe una demanda creciente en el transporte de personas de movilidad reducida que hay que satisfacer a través de servicios de transporte público colectivo con la mayor calidad posible. La investigación se ha centrado en optimizar los viajes diarios que necesitan realizar por motivo salud o educación, hacia centros de tratamiento, centros de día e incluso instalaciones hospitalarias y educativas.

El diseño de las rutas de transporte se ha propuesto a través de una variante del DARP adaptada para este problema, en el que todas las solicitudes de traslados se conocen con anticipación, disponiendo en el depósito de una flota fija de vehículos adaptados para su óptima asignación a las posibles rutas.

Las decisiones que se deben tomar a la hora de resolver este tipo de problemas de transporte, para que sean realmente eficaces, deben recoger diferentes restricciones y considerar varios objetivos, que en algunos casos entran en conflicto.

El estudio se ha planteado con enfoque multiobjetivo. Uno de los objetivos del problema es de carácter económico, en el que se ha buscado minimizar el total de la distancia recorrida. El otro objetivo de carácter social, para mejorar la calidad del servicio minimizando el máximo tiempo excedente de viaje del usuario, medido como la diferencia entre: el tiempo de viaje al compartir el vehículo con otros usuarios menos el tiempo al realizar el recorrido de forma directa, es decir, sin paradas intermedias.

Una vez revisada la bibliografía, se han encontrado pocos trabajos que aborden el problema del DARP bajo el enfoque de optimización multiobjetivo y, al menos desde nuestro conocimiento, no existe ninguna referencia en la literatura que verse sobre el DARP con estos dos objetivos concretos. Al plantearse un modelo biobjetivo no va a existir una solución óptima única, sino que, se persigue encontrar el conjunto de soluciones eficientes, frentes de Pareto, o una aproximación a dicho conjunto, que sirva de apoyo para la adecuada toma de decisiones respecto al traslado de los usuarios.

Las metodologías de resolución metaheurísticas, y sus aplicaciones a problemas multiobjetivo, resultan ser un campo muy estudiado en los últimos años, que han demostrando llegar a mejores resultados que los tradicionales heurísticos. y, por ello, se han seleccionado para ser adaptadas y aplicadas en esta investigación.

Inicialmente, se ha propuesto una formulación del problema como un modelo de programación lineal entera-mixta. Se ha empleado Cplex para resolver instancias del problema de forma exacta.

Posteriormente, se han propuesto y comparado dos metodologías heurísticas de resolución. Una primera basada en la estrategia MOAMP y una segunda consistente en una adaptación heurística del método ϵ -constraint, MHeC.

En ambos métodos, se han introducido estrategias especialmente diseñadas para acelerar el chequeo de la factibilidad y el cálculo de las funciones objetivos cuando se producen inserciones y movimientos en las rutas. Éstos van a asociados a los dos procedimientos; constructivo y de búsqueda tabú y a las diferentes variantes de ambos empleadas.

A partir de las pruebas computacionales ejecutadas se ha concluido que esta variante biobjetivo del DARP, al tratarse de un *NP-Hard*, sólo se puede resolver de forma exacta en un tiempo razonable para instancias de tamaño pequeño, hasta 6 solicitudes. Para hallar aproximaciones a la curva de eficiencia en instancias de mayor tamaño se hace imprescindible el uso de los métodos de solución metaheurísticos propuestos. Resultando muy conveniente la incorporación de las estrategias de aceleración, que presentan una gran reducción de los tiempos de computación para instancias de gran tamaño, sobretodo, aplicadas al procedimiento de búsqueda tabú **TabuSearch**.

El método heurístico MOAMP es adecuado para la resolución de instancias de tamaño intermedio, hasta 20 solicitudes, con una ligera diferencia en el tiempo computacional. Es para las instancias de tamaño mayor para las que, a pesar del gran tiempo de cálculo requerido, se ha observado que el MHeC se señala como el más recomendable en cuanto a los conjuntos de soluciones no dominadas obtenidos.

A través de las metodologías empleadas en la resolución del modelo diseñado para el problema, se obtienen buenos frentes de Pareto que van a proporcionar diversas soluciones al tomador de decisiones. Así, según la situación existente en el momento de la toma de decisión, se puede optar y adoptar la más adecuada.

En la planificación del transporte de personas con movilidad reducida la toma de decisiones adecuadas puede aportar a una entidad, de ámbito público o privado, grandes beneficios: reducción de costes, mejora en el nivel de servicio al usuario, racionalización de turnos de trabajo del personal y de los recursos materiales, etc., que indirectamente pueden conllevar otras ventajas como mejora de la imagen, ayuda al medio ambiente y sostenibilidad, por la reducción del consumo de combustibles y, por consiguiente, la reducción de los efectos contaminantes derivados de la actividad del transporte (contaminación acústica, ambiental, ...).

La originalidad de la investigación radica en la temática central de los problemas de transporte que se analizan, ya que el diseño y optimización multiobjetivo del TRANSPORTE DE PERSONAS DISCAPACITADAS es un problema que, como ha quedado patente, no ha sido muy estudiado hasta la fecha y que da lugar a modelos muy interesantes y de gran repercusión social.

11.2.- APORTACIONES

El problema tratado es innovador no sólo por la temática del mismo, transporte de personas de movilidad reducida, sino también por la modelización y planteamiento de los objetivos que se pretenden optimizar.

Debido a la dimensión social del servicio de transporte analizado y con el fin de planificar eficientemente las rutas, no sólo se ha buscado la minimización de los costes de las operaciones, sino también el minimizar los inconvenientes generados a los usuarios trasladados.

La mayoría de los trabajos científicos revisados que tratan el DARP con varios objetivos, los abordan como sumas ponderadas entre los objetivos o consideran alguno de los objetivos como una restricción en el modelo. Son pocos los trabajos que abordan el problema del DARP bajo el enfoque de optimización multiobjetivo, tratando de obtener frentes de Pareto para una adecuada toma de decisiones respecto al traslado de los usuarios. E incluso entre los DARP mono-objetivo revisados, no se han encontrado ejemplos que incluyan los objetivos estudiados en esta investigación.

En la misma línea, se desea destacar el enfoque novedoso conseguido a través de la búsqueda de una gran robustez en el análisis de la calidad del servicio del transporte mediante la definición del objetivo social.

Algunos autores, en problemas de rutas y, en concreto, en aplicaciones del DARP consideran que el tiempo de viaje del usuario puede ser un objetivo a minimizar para así mejorar la calidad del servicio de transporte ofertado. Dicho planteamiento se ha descartado en la presente investigación, con el fin de evitar que usuarios con un tiempo elevado de ir directamente del punto de recogida al punto de destino, denominado como “*tiempo ideal*”, impida obtener mejores soluciones para el conjunto de solicitudes.

En consecuencia, se ha fijado el incremento sobre el “*tiempo ideal*” como una medida de “no-calidad” para el usuario. Esto es, la diferencia entre el tiempo de viaje final, pudiendo o no incluir paradas intermedias, y el tiempo directo si no se realizase ninguna parada.

Analizando el tratamiento, que para la optimización del incremento, han realizado otros autores, se han encontrado publicaciones en las que el objetivo persigue minimizar la media de los incrementos sobre “*tiempo ideal*” en el conjunto de todos los usuarios. De la misma manera, éste enfoque no se ha considerado y se ha intentado evitar, así, el riesgo de que existan algunos clientes nada satisfechos con el servicio, con grandes tiempos de viaje sobre su tiempo ideal, que queden ocultos tras medias buenas que dejen al resto satisfechos.

Por todo lo anterior, se ha programado como objetivo social la minimización del máximo tiempo excedente de viaje, medido como la diferencia entre; el tiempo de viaje al compartir el vehículo con otros usuarios y el tiempo al realizar el recorrido de forma directa, es decir, sin paradas intermedias.

El objetivo social modelizado, es robusto en el marco general del DARP pero, se vuelve aún más relevante en la aplicación concreta del transporte de personas de movilidad reducida. Ya que, se pretende diseñar una herramienta para ofrecer un servicio de transporte colectivo adaptado de calidad, sin incorporar incomodidades extras, como puede ser el aumentar excesivamente sus tiempos de viaje, a una tarea que, por sí misma, ya conlleva una gran dificultad.

De manera destacable, señalar la aportación metodológica presentada en este trabajo de investigación por el desarrollo e incorporación de estrategias de aceleración adaptadas a los procedimientos del método de resolución propuesto. El empleo de dichas estrategias ha tenido por objeto acelerar de manera significativa los procedimientos **Constructive** y **TabuSearch**, agilizando los chequeos de la factibilidad, así como, el cálculo de las funciones objetivo del problema, especialmente la función objetivo social debido a su complejidad.

Las estrategias de aceleración desarrolladas para chequear la factibilidad de las inserciones del procedimiento **Constructive** y de los movimientos del procedimiento **TabuSearch** han perseguido permitir comprobar ágilmente que en cada iteración se cumpla: La restricción de capacidad máxima de los vehículos; La restricción de tiempo máximo de duración de cada ruta; Que los puntos de carga y descarga de una misma solicitud se encuentren en la misma ruta y Las restricciones de precedencia, que el punto de carga se encuentre antes que el punto de descarga.

Una vez realizado el chequeo de la factibilidad, se tiene que determinar qué solicitudes van a estar afectadas por dicha inserción o movimiento, modificando su valor de las funciones objetivo, y en qué cuantía va a cambiar dicho valor. Por medio de las estrategias de aceleración propuestas se ha intentado reducir el excesivo tiempo de resolución que dicho cómputo conlleva.

Todas las estrategias, definen y emplean una serie de variables auxiliares con las que se puede chequear la factibilidad y el cálculo de las funciones objetivo en un número reducido de operaciones. En muchos casos este número de operaciones no depende del tamaño del problema.

A pesar de que las operaciones realizadas con estas variables globales, suponen un desarrollo de código adicional, se ha comprobado, por los resultados de los experimentos computacionales, que las estrategias de aceleración implementadas agilizan considerablemente el proceso de resolución. Sobretudo, se obtienen grandes beneficios al aplicarlas en el procedimiento **TabuSearch** y a la hora de resolver problemas de gran tamaño, que son precisamente los más exigentes con respecto a tiempos de cálculo. Así por ejemplo, en instancias de tamaño 100 y 150 la incorporación de estas estrategias supone emplear entre el 1.5% y el 2% del tiempo de cálculo empleado cuando éstas no se aplican. Se entiende, por ello, que las estrategias de aceleración diseñadas son una de las principales aportaciones de este trabajo de investigación.

En definitiva, las aportaciones de esta investigación se pueden extraer como:

- Problema social novedoso.
- Objetivo social innovador y, socialmente, muy robusto.
- Modelo formulado para la resolución de instancias pequeñas de forma exacta con Cplex.
- Estrategias heurísticas diseñadas para resolver el problema en tamaños grandes.
- Estrategias de aceleración específicas.
- Posibilidad de extensión a otro tipo de problemas.

11.3.- REFLEXIONES

En este trabajo se ha desarrollado un método para resolver el problema de diseño de rutas para trasladar a personas de movilidad reducida, personas de avanzada edad o con algún tipo de discapacidad que impida su completa independencia a la hora de trasladarse.

La idea del problema surge al concentrar la atención en aquellos traslados que el colectivo de personas de movilidad reducida realiza a centros de tratamiento, centros de día e incluso instalaciones hospitalarias y educativas. Se ha buscado, por tanto, optimizar los viajes diarios que, las personas pertenecientes a dicho colectivo, necesitan realizar por motivo salud o educación, asistidos en el marco de un servicio de transporte público colectivo.

Se ha estudiado, en consecuencia, escenarios en que, desde un punto de vista geográfico, se transportan personas desde diferentes orígenes a distintos destinos. Pero es importante resaltar que, todos estos puntos, por el carácter social que tiene este tipo de servicio de transporte, se van a localizar en ámbitos locales, provinciales o, a lo sumo, dentro del contexto de una misma Comunidad Autónoma.

La motivación de los trayectos para los que se oferta el transporte, hace que los destinos sean sedes, instituciones o centros que dan servicio a una comunidad de personas concreta. Es por ello, que se ha estudiado el problema en un medio en el que la ordenación del territorio asume una distribución geográfica con unas isócronas de acceso que permiten dar servicios completos a dicha comunidad.

Consiguientemente, en el planteamiento del problema se han tratado situaciones en las que los tiempos de traslado directo entre el origen y el destino de las personas pueden ser diferentes, presentando una ligera heterogeneidad, pero que no van a alcanzar en ningún caso valores muy elevados, ya que el estudio se ha planteado en contextos con una relativamente buena y equitativa distribución de dotaciones y servicios sociales, sanitarios y educativos.

Las características que presentan los traslados objeto de estudio y, más concretamente, el hecho de que los destinos de los usuarios se ubiquen en entidades sociales, centros sanitarios o, incluso, instalaciones educativas, que en muchos casos presentan una gestión de carácter público o concertado, ha motivado el empleo de un DARP sin ventanas de tiempo al llevar aparejados horarios similares de entrada en los destinos.

Y es más, tras el análisis de datos asociados a varios casos reales en los que el problema podría implementarse, se ha visto que la ventanas de tiempo en el destino eran muy homogéneas, esto es que todos indicaban poder llegar a su destino a la hora correspondiente con los horarios de entrada o admisión en los centros, siendo estos horarios iguales o muy similares.

Dado que el tipo de desplazamientos analizados en esta investigación presentan, en general, heterogeneidad de tiempos de recogida en el origen pero bastante homogeneidad en cuanto a los tiempos de llegada a los destinos, con el fin de controlar mejor esta singularidad del problema, el DARP ha incluido la restricción temporal de tiempo máximo de cada ruta, prescindiendo en este caso de un análisis con ventanas de tiempo.

Y para ello, en la formulación matemática del modelo, se ha incluido L_{max} que hace referencia al tiempo máximo que transcurre entre la primera recogida y la última descarga para cada ruta. El valor de L_{max} se fija como una restricción específica para cada instancia que se intente solucionar.

El problema que se ha abordado en este trabajo persigue el diseño de un sistema de transporte colectivo puerta a puerta para el traslado de personas de movilidad reducida. Concretamente, se trata del diseño de rutas para transportar a estas personas desde un punto de recogida, normalmente situado cerca del hogar o residencia, a un destino, que puede ser un centro de día, un hospital, un colegio, etc. Tanto los puntos de recogida como los puntos de destino pueden ser diferentes para los distintos usuarios, es decir, por ejemplo partir de distintas residencias hacia distintos centros de cuidados u hospitales. Así, los usuarios se agrupan en solicitudes, donde cada solicitud viene definida por: un punto origen, un punto de destino y un conjunto determinado de usuarios que deben ser llevados desde ese origen a ese destino. Para la realización de este servicio de transporte se ha considerado una flota homogénea de vehículos.

Y precisamente, en lo que a los vehículos se refiere, indicar que el diseño de las rutas se ha propuesto a través de un DARP estático sin ventanas de tiempo, en el que todas las solicitudes de traslados se conocen con anticipación, disponiendo en el depósito de una flota fija de vehículos adaptados.

Hay que hacer notar que aunque en este problema se considera una flota de tamaño fijo m , sin coste por vehículo, los métodos que se desarrollan podrían ser adaptados a problemas con un número variable de vehículos y con costes por vehículo usado. La estrategia sería la siguiente: Valdría con considerar diferentes valores de m en un rango que se considere razonable para cada instancia concreta, dependiendo por ejemplo de la disponibilidad, y resolver la instancia para cada uno de los valores de m . A continuación, se unirían todos los conjuntos de soluciones no dominadas en un gran conjunto. En cada una de las soluciones obtenidas en este gran conjunto se agregaría en la función económica el coste por vehículo y por distancia recorrida. Se podría distinguir incluso el coste por vehículo de flota propia o ajena. De este gran conjunto se eliminarían finalmente las soluciones dominadas considerando la función social y la nueva función económica agregada.

Por otra parte, derivadas de la observación de los resultados obtenidos por los dos métodos heurísticos, propuestos para la resolución del problema, se presentan las siguientes reflexiones:

- Ambos métodos permiten resolver de forma aproximada instancias de gran tamaño. Instancias compuestas por hasta 150 solicitudes han sido resueltas en un tiempo computacional aceptable: unos 2.000 segundos (35 minutos) MOAMP y unos 6.000 segundos (100 minutos) MHeC. No obstante, la evolución de los tiempos da a entender que para tamaños mayores, de 200 o más solicitudes, los tiempos de cálculo pueden no ser tan razonables.
- No hay un método que domine claramente al otro en la obtención de la solución “esquina izquierda”. En casi el total de los casos analizados los dos métodos alcanzan el valor de 0 en el objetivo social en esa solución “esquina izquierda”, que obviamente es óptimo en ese objetivo, pero se suelen diferenciar en el valor del objetivo económico, siendo a veces mejor la “esquina izquierda” obtenida por MOAMP y a veces mejor la “esquina izquierda” obtenida por el método MHeC. Por tanto, las soluciones finales que ambos métodos obtienen para la “esquina izquierda”, o lo que es lo mismo las mejores soluciones sociales de los conjuntos obtenidos, no suelen coincidir, a pesar de que usan el mismo generador (**GeneradorSocial**) para construir dichas soluciones. Es decir, la solución obtenida por el procedimiento **GeneradorSocial** es mejorada en pasos o fases posteriores por ambos métodos. Este hecho se aprecia de forma clara

- en las gráficas que comparan las 3 fases de MOAMP. También se observa en dichas gráficas que, en general, el procedimiento **GeneradorSocial** es capaz de obtener el valor de 0 en el objetivo social, y por tanto el óptimo en este objetivo, pero no consigue alcanzar el extremo izquierdo de la curva de eficiencia.
- En lo que respecta a la solución “esquina derecha”, queda manifiesto que el método MOAMP obtiene mejores resultados. Parece evidente, en este caso, que el uso de un generador específico para obtener esta solución esquina derecha (**GeneradorEconomico**) es más eficaz que la estrategia empleado por el método MHeC que es resolver el problema *MonoSocial(Dmx)* con valores bajos de *Dmx*. Hay que recordar que el problema *MonoSocial(Dmx)* es resuelto por el procedimiento **GeneradorSocial** con la restricción adicional de que la distancia total recorrida sea menor que *Dmx*. Por tanto, aunque se use para valores bajos de *Dmx*, el procedimiento **GeneradorSocial** no está diseñado para la obtención de buenas soluciones económicas y, de ahí, los “malos” resultados obtenidos por el método MHeC.
 - Asimismo, se observa que, en algunas instancias, la solución “esquina derecha” obtenida en la primera fase de MOAMP es mejorada en las otras fases. No obstante estas mejoras son más sutiles que las que se producen en la solución “esquina izquierda”.
 - En cuanto a las soluciones intermedias, en general y de forma clara, el mejor comportamiento lo demuestra el método MHeC, principalmente a medida que el tamaño de las instancias crece. Sólo en las “cercanías de la esquina derecha”, debido a lo comentado en los párrafos anteriores, las soluciones de MOAMP parecen ser mejores. Hay que tener en cuenta que, una vez determinados los extremos o esquinas izquierda y derecha, el método MOAMP obtiene el resto de soluciones no dominadas mediante movimientos vecinales que guían de una solución a otra. Por su parte, el método MHeC genera cada una de sus soluciones una a una con el procedimiento **GeneradorSocial**, añadiendo la restricción adicional de que la distancia total sea menor que *Dmx*. Este generador funciona bien hasta que los valores de *Dmx* empiezan a ser bajos. Se ha observado que, en estos casos, durante la ejecución del procedimiento **GeneradorSocial**, el procedimiento **Constructivo** no es capaz de hallar soluciones factibles, no se consigue que la distancia total sea menor que *Dmx*, y, por consiguiente, tampoco el procedimiento **GeneradorSocial** lo logra. Esto se debe a que, aunque se tenga en cuenta esta restricción, el **Constructivo** esta guiado por el objetivo social, al ejecutarse dentro del **GeneradorSocial**. Finalmente, como conclusión de este párrafo, se intuye que construir y generar soluciones no dominadas, una a una en cada zona del espacio de búsqueda con procedimientos adecuados para cada una de estas zonas, obtiene mejores resultados que el uso de movimientos vecinales para obtener un solución no dominada partiendo de otra no dominada mediante dichos movimientos.
 - Por último, a partir de todas las observaciones anteriores se pueden obtener las siguientes mejoras metodológicas:
 - La primera mejora se refiere a modificar los generadores para conseguir los puntos extremos. Como se ha comentado, el procedimiento **GeneradorSocial**, sin restricciones de distancia máxima, es capaz de obtener el óptimo social, normalmente, con valor 0 en las instancias analizadas. Sin embargo, no tiene en cuenta el objetivo económico y cuando alcanza un óptimo social, el generador no cambia luego la solución, aunque encuentre

otros óptimos sociales con menor distancia recorrida. Habría entonces que modificar el procedimiento de búsqueda tabú, cuando se ejecuta en el procedimiento **GeneradorSocial**. Este cambio consistiría en modificar el criterio que usa para comparar soluciones, en la selección de mejores movimientos, en la actualización de la mejor solución encontrada y en la aplicación del criterio de aspiración, de la siguiente forma:

Una solución S' es mejor que otra S si y sólo si

$$(f_1(S') < f_1(S)) \text{ o bien } (f_1(S') = f_1(S) \text{ y } f_2(S') < f_2(S))$$

Hay que tener en cuenta que, la función social f_1 es una función escalonada, como lo son todas las funciones “max”. En general, los objetivos min-max son de gran dificultad tanto cuando se intenta hallar el óptimo con métodos exactos, como cuando se halla una aproximación de forma heurística. Esta dificultad se deriva de la existencia de muchas soluciones con el mismo valor en este objetivo social y, por tanto, hay que buscar otros criterios, además del valor de la función, para discriminar una de otras. Precisamente, es mediante la modificación propuesta que se va a conseguir aportar un segundo criterio para discriminar entre soluciones. Además, se incorpora también el concepto de dominancia, asegurando, así, que la mejor solución social encontrada domina a todas las soluciones visitadas con el mismo valor.

- Una modificación análoga se puede considerar para el procedimiento **GeneradorCostes**. El cambio es recomendable aunque quizás, en este caso, sea menos relevante. La función económica es una suma, funciones normalmente más “continuas”, y, en este sentido, suele ser menos escalonada que los criterios min-max. Por ello, en general, no va a ser tan fácil encontrar soluciones diferentes con el mismo valor de la distancia. No obstante, aplicar el criterio de aspiración propuesto siempre es recomendable.
- Otra reflexión, más general, es que de una solución eficiente es difícil llegar a otro óptimo de Pareto a menos que éste esté relativamente cerca. De hecho, en Gómez (2010) y Gómez et al. (2015) ya se analizó la dificultad de llegar de un extremo, o esquina, al otro de la curva de eficiencia mediante búsqueda tabú con el uso de MOAMP. En las investigaciones posteriores de Pacheco et al. (2013), García et al. (2013) y Martínez-Puras y Pacheco (2016), ya se incorporó a la metodología de MOAMP la generación de los extremos de la curva de forma independiente con generadores adaptados a cada extremo. Sin embargo, la observación de los resultados indica que también es difícil trasladarnos entre puntos eficientes intermedios simplemente con movimientos vecinales.

La búsqueda tabú es un método que ha demostrado funcionar bien en muchos problema combinatorios pero precisa de soluciones iniciales adecuadas. El método **Constructivo** parece aportar esas buenas soluciones a la búsqueda tabú cuando ambos se ejecutan dentro del procedimiento **GeneradorSocial**, adaptado para considerar la restricción de distancia máxima D_{mx} . Este procedimiento, **GeneradorSocial**, parece funcionar bastante bien para hallar aproximaciones a puntos eficientes en regiones no muy alejadas del óptimo social, valores no muy bajos de D_{mx} . No obstante, para regiones cercanas a la esquina derecha, valores bajos de D_{mx} , ésta estrategia parece fallar, incluso, llegándose a observar la dificultad de encontrar soluciones factibles. Por tanto, hay que usar procedimientos

generadores similares al empleado, es decir que combinen los procedimientos **Constructivo** y **TabuSearch**, pero con diferentes criterios. Una idea es, prescindir de la restricción de distancia máxima y evitar generar soluciones no factibles. En su lugar, se puede usar la función F_λ como criterio en ambos procedimientos. Los valores de λ pueden variar de forma gradual según algún criterio preestablecido. Una variante de esta idea puede ser emplear la función alternativa G_λ siguiente:

$$G_\lambda(S) = \lambda \cdot \frac{f_1(S) - f_1^{min}}{f_1^{max} - f_1^{min}} + (1 - \lambda) \cdot \frac{f_2(S) - f_2^{min}}{f_2^{max} - f_2^{min}}$$

es decir, minimizar la combinación lineal en lugar del máximo de ambos componentes. La función F_λ está basada en la distancia L_∞ y, por ello, los óptimos según F_λ son puntos eficientes. Sin embargo, el carácter “max” de esta función hace que sea difícil tanto su tratamiento como la obtención de los óptimos, por lo comentado anteriormente. Aunque habría que analizarlo, a priori, la función G_λ podría tener un comportamiento más suave.

- Otra alternativa al empleo de procedimientos generadores con el procedimiento **Constructivo**, puede ser el uso de la búsqueda tabú para moverse entre puntos muy próximos. Es decir, enlazar ejecuciones del procedimiento de búsqueda tabú donde el punto final de una ejecución sea el punto inicial de la siguiente. Esto recupera la idea inicial de MOAMP, que perseguía enlazar procedimientos de búsqueda tabú, pero con la implementación consistente en que los valores de λ , en lugar de generarse aleatoriamente, se incrementen, o disminuyan dependiendo del extremo con el que se comience, más suavemente y de forma gradual.

Finalmente, es necesario mencionar que, todas las sugerencias metodológicas presentadas precisan de su futuro análisis a través de pruebas adecuadas. Simultáneamente, es posible que estas reflexiones, sugerencias o modificaciones puedan ser adaptadas con éxito a otros problemas biobjetivo de trabajos existentes y venideros.

11.4.- FUTURAS LÍNEAS DE INVESTIGACIÓN

Los resultados obtenidos en este trabajo animan a apuntar nuevas líneas de investigación de cara al futuro. Algunos de los aspectos en los que se tiene previsto trabajar son los siguientes:

- Se considera muy relevante para la investigación el tratamiento de problemas reales, localizando otros ejemplos donde poder aplicar el modelo general propuesto. Así, se pretende comparar las soluciones obtenidas por la metodología planteada para el problema real, con la solución que el organismo o empresa en cuestión esté adoptando.
- Se tiene proyectado diseñar otros métodos alternativos, como pueden ser las estrategias NSGA-II, por ser consideradas muy adecuadas para la resolución de problemas multiobjetivo. Sería interesante realizar un estudio comparativo entre las diferentes metodologías propuestas, tanto en lo que se refiere a calidad de las soluciones como a los tiempos de cálculo computacional que emplee cada una.
- Si bien en este caso se han perseguido dos objetivos vinculados con el coste y la calidad, en el futuro se pretende incorporar otros objetivos que pueden resultar ser críticos en la explotación del servicio de transporte de personas de movilidad reducida, como por ejemplo: objetivos ecológicos y medioambientales que busquen reducir el consumo de combustibles.
- Se plantean futuras extensiones considerando un número variable de vehículos, a los que se les podrá así mismo asignar costes en función de si pertenecen a flota propia o si se necesita contratar vehículos adicionales, flota ajena, para una explotación más óptima, socialmente.
- Se ha considerado, en este estudio, un colectivo muy concreto de usuarios, personas con movilidad reducida. Pero se puede ver que, por la naturaleza del problema descrito, la herramienta metodológica para resolver el problema de diseño de rutas podrá aplicarse al caso del transporte escolar con destino a varios centros educativos.

Se pretende fomentar e incentivar la investigación con la incorporación del trabajo a una plataforma virtual que el Departamento de Economía Aplicada de la Universidad de Burgos está preparando. De esta forma, se posibilitará que las personas que estén investigando en esta misma línea e, incluso, que entidades que puedan requerir de soluciones similares puedan informarse y disponer del avance. Esto, no sólo fomenta la colaboración con otros investigadores sino que, promueve la transferencia, tecnología y gestión del conocimiento en entornos abiertos.

12.- BIBLIOGRAFÍA

12.- BIBLIOGRAFÍA

- Aarts, E.H.L. y J. Lenstra (1997) *Local Search in Combinatorial Optimization*. Wiley, Chichester.
- Aarts, E.H.L. y J.H.M. Korst (1989) *Simulated Annealing and Boltzmann Machines*. Wiley, Chichester.
- Acevedo, J. y J. P. Bocarejo (2009) "Movilidad sostenible: una construcción multidisciplinaria" *Revista de Ingeniería*. Universidad de Los Andes. Núm. 29
- Affenzeller M., Wagner S., Winkler S. and Beham A. (2009) *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*, Taylor and Francis.
- Alabas-Uslu, C. (2008) "A Self-tuning Heuristic for a Multi-objective Vehicle Routing Problem," *Journal of Operations Research Society* 59, 7, pp. 988-996
- Alba, E. y C. Cotta (2006) "Evolutionary Algorithms" *Handbook of Bioinspired Algorithms and Applications*, S. Olariu, A.Y. Zomaya. (eds.), pp. 3-19, Chapman & Hall/CRC, Boca Ratón FL
- Alfa, A. S. (1986) "Scheduling of vehicles for transportation of elderly" *Transportation Planning and Tech.* 11(3):203–212
- Álvarez, A., S. Casado, J.L. González-Velarde y J. Pacheco (2010) "A Computational Tool for Optimizing the Urban Public Transport: A Real Application" *Journal of Computer and System Sciences International* 49, 2, 244–252
- Alves, M.J. y J. Climaco (2000) "An interactive method for 0-1 multiobjective problems using Simulated Annealing and Tabu Search" *Journal of Heuristics* 6(3): p. 385-403
- Angelelli, E., R. Mansini y M.G. Speranza (2010a) "Kernel search: A new heuristic framework for portfolio selection" *Computational Optimization and Applications* pp. 1–17
- Angelelli, E., R. Mansini y M.G. Speranza (2010b) "Kernel search: A general heuristic for the multi-dimensional knapsack problem" *Computers & Operations Research* 37 (11), pp. 2017-2026
- Angelelli, E., R. Mansini y M.G. Speranza (2012) "Kernel search: A new heuristic framework for portfolio selection" *Computational Optimization and Applications* 51 (1) pp. 1–17
- Bäck, T. (1995) *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*
- Battiti, R. (1996) "Reactive Search: Toward Self-Tuning Heuristics" In V.J. Rayward-Smith, editor, *Modern Heuristics Search Methods*, chapter 4, p. 61-83
- Baugh, J.W. Jr., G.K.R. Kakivaya y J.R. Stone (1998) "Intractability of the dial-a-ride problema and a multiobjective solution using simulated annealing" *Engineering Optimization* 30:91–123
- Beausoleil, R.P. (2006) "MOSS" multiobjective scatter search applied to non-linear multiple criteria optimization" *European Journal of Operational Research*, 169(2): p. 426-449
- Ben Abdelaziz, F., S.Krichen y J. Chaouachi (1999) "An Hybrid Metaheuristic for the Multiobjective Knapsack Problem" *Metaheuristics-Advances and Trends in Local Search Paradigms for Optimization*. S. Voss, et ál., Editors. Kluwer. p. 205-212
- Beume, N., C.M. Fonseca, M. López-Ibáñez, L. Paquete y J. Vahrenhold (2009) "On the complexity of computing the hypervolume indicator" *IEEE Transactions on Evolutionary Computation*, 13(5), 1075-1082
- Bigotte JF, Krass D, Antunes AP y Bermam O (2010) "Integrated modeling of urban hierarchy and transportation network planning" *Transportation Research Part A Policy and Practice* 44, 7, pp. 506-522
- Blum, C. y A. Roli (2003) "Metaheuristics in combinatorial optimization: Overview and conceptual comparison" *ACM Computing Surveys*, 35(3):268 – 308

- Boender, C.G.E. y A.H.G. Rinnooy (1983) "A Bayesian Analysis of the Number of Cells of a Multinomial Distribution" *The Statistician* 32: p. 240-248
- Boettcher, S. y A. Percus (2000) "Nature's way of optimizing" *Artificial Intelligence*, 119(1-2): p. 275-286
- Bouamama, S., Blum, y C., Boukerram, A. (2012) "A population-based iterated greedy algorithm for the minimum weight vertex cover problema" *Applied Soft Computing Journal* 12(6), 1632-1639
- Bradstreet, L. (2011) *The hypervolume indicator for multi-objective optimisation: Calculation and use*. University of Western Australia.
- Braekers, K. y A.A. Kovacs (2016) "A multi-period dial-a-ride problem with driver consistency" *Transportation Research Part B: Methodological*, 94, pp. 355-377
- Burke, E.K., G. Kendall y E. Soubeiga (2003) "A tabu-search hyperheuristic for timetabling and rostering" *Journal of Heuristics*, 9(6): p. 451-470
- Caballero, R., J. Molina y A.G. Hernández-Díaz (2007) "Metaheurísticos en Programación Multiobjetivo" *Procedimientos Metaheurísticos en Economía y Empresa*, T.L. Blanch, p. 117-138
- Caballero, R., J. Molina y M.V. Rodríguez Uría (2003) "MOAMP. Programación multiobjetivo mediante un procedimiento de Búsqueda Tabú" *Actas del II Congreso Español de Metaheurísticas y Algoritmos Evolutivos y Bioinspirados MAEB*. Gijón
- Caprara A. and M. Fischetti (1997) "Branch-and-cut algorithms". In M. Dell'Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, pages 45–64, Wiley, New York
- Casado S. y Martí R. (2007) "Principios de búsqueda dispersa" *Procedimientos metaheurísticos en economía y empresa* (Crespo E., Martí R. y Pacheco J. coord..) Monografías Rect@ 3, pp. 97-116.
- Casado, S. (2005) "Planificación de turnos en un aeropuerto: uso de simulación y metaheurísticos" *Departamento de Economía Aplicada*, Universidad de Burgos.
- CE [Comisión Europea] (2006) *Reglamento CE No. 1107/2006 del Parlamento Europeo y del Consejo, de 5 de julio de 2006, sobre los derechos de las personas con discapacidad o movilidad reducida en el transporte aéreo*. Bruselas
- CE [Comisión Europea] (2011) 2011/C 21/10 *Dictamen del Comité Económico y Social Europeo sobre la «Comunicación de la Comisión al Parlamento Europeo, al Consejo, al Comité Económico y Social Europeo y al Comité de las Regiones — Plan de Acción de Movilidad Urbana» COM(2009) 490 final*. Bruselas
- CE [Comisión Europea] (2014) *HORIZON2020 – El programa marco de investigación e innovación de la Unión Europea*. Bruselas
- Charon, I. y O. Hudry (1993) "The noising method - a new method for combinatorial optimization" *Operations Research Letters* 14(3): p. 133-137
- Chassaing, M., C. Duhamel y P. Lacomme (2016) "An ELS-based approach with dynamic probabilities management in local search for the Dial-A-Ride Problem" *Engineering Applications of Artificial Intelligence*, 48, 119-133
- Chen, S. y C. Jang (2007) "Neural network structure optimization and its application for passenger flow predicting of comprehensive transportation between cities" *Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services* 1-2, pp. 1087-1091
- Chevrier, R., A. Liefogheb, L. Jourdan y C. Dhaenensb (2012) "Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport" *Applied Soft Computing* 12, 4, pp. 1247-1258
- Christofides, N. (1994) *The bionomic algorithm*, AIRO.: Savona. Italy

- Coello, C. A., G. B. Lamont y D. A. Van Veldhuizen (2007) "Evolutionary Algorithms for Solving Multi-Objective Problems" *Genetic and Evolutionary Computation Series*, Springer
- Coello, C.A. y G.B. Lamont (2005) "Applications of multi-objective evolutionary algorithms" *Advances in Natural Computation*
- Coello, C.A. y N. Cruz-Cortés (2002) "An Approach to Solve Multiobjective Optimization Problems Based on an Artificial Immune Systems" *First International Conference on Artificial Immune Systems (ICARIS'2002)* University of Kent at Canterbury, Inglaterra
- Cohon, J. L. y D. H. Marks (1975) "A review and evaluation of multiobjective programming techniques" *Water Resources Research*, 11(2):208 – 220
- COM (1997) 276: *Proposal for a European Parliament and Council Directive relating to special provisions for vehicles used for the carriage of passengers comprising more than eight seats in addition to the driver's seat and amending Council Directive 70/156/EEC*. Bruselas
- COM (2001) 217: *Opinion of the Commission pursuant to Article 251(2) (c) of the EC Treaty, on the European Parliament's amendments to the Council's common position regarding the proposal for a Directive of the European Parliament and of the Council relating to special provisions for vehicles used for the carriage of persons comprising more than eight seats in addition to the driver's seat, and amending Directives 70/156/EEC and 97/27/EC amending the proposal of the Commission pursuant to Article 250 (2) of the EC Treaty*. Bruselas
- CONSTITUCIÓN ESPAÑOLA (1978) BOE núm. 311, 29 de diciembre de 1978
- Cook, W. y J.L. Rich (1999) "A parallel cutting plane algorithm for the vehicle routing problem with time windows" *Technical Report Computational Applied Mathematics*: Rice University, Houston TX
- Corberán, A., E. Fernández, M. Laguna y R. Martí (2002) "Heuristic Solutions to the Problem of Routing School Buses with Multiple Objectives" *Journal of the Operational Research Society* 53 (4) pp. 427-435
- Corberán, A., J. Peiró, V. Campos, F. Glover y R. Martí (2016) "Strategic oscillation for the capacitated hub location problem with modular links" *Journal of Heuristics* 22 (2) pp. 221-244
- Cordeau J. A (2006) "Branch- and cut algorithm for the dial-a-ride problem" *Operations Research* 54:573-586
- Cordeau J. F. y G. Laporte (2007) "The dial-a-ride problem: models and algorithms" *Annals of Operations Research* 153:29-46
- Cordeau, J. F. y G. Laporte (2003a) "The dial-a-ride problem (DARP): Variants, modeling issues and algorithms" *Quarterly Journal of the Belgian, French, and Italian Operations Research Societies* 1:89-101
- Cordeau, J. F. y G. Laporte (2003b) "A tabu search heuristic for the static multi-vehicle dial-a-ride problem" *Transportation Research Center Part B: Methodological*
- Cordeau, J. F., G. Laporte y A. Mercier (2000) "A unified tabu search heuristic for vehicle routing problem with time Windows" *Technical Report CRT-00-04 Centre for Research on Transportation*: Montreal, Canada
- Cotta, C. (2007) "Una Visión General de los Algoritmos Meméticos" *Procedimientos Metaheurísticos en Economía y Empresa*, ed. ASEPUMA
- Cotta, C. y E. Alba (2003) *Algoritmos Meméticos*
- Cowling, P., G. Kendall y E. Soubeiga (2002) "Hyperheuristics: A tool for rapid prototyping in scheduling and optimisation" *Applications of Evolutionary Computing, Proceedings*. p. 1-10
- Crespo E, Martí R, Pacheco J [coordinadores] (2007) *Procedimientos Metaheurísticos en Economía y Empresa*, Monografías de Rect@ 3, Tirant lo Blanch

- Culberson, J. (1998) "On the futility of blind search: An algorithmic view of "No Free Lunch"" *Evolutionary Computation*, 6: p. 109-127
- Dahl, G., K. Jörnsten y A. Lokketangen (1995) *A Tabu Search approach to the channel minimizaion problem*, ICOTA'95, Chengdu, China
- Dangond, G. C., J.-F. Jolly, A. Monteoliva y F. Rojas (2010) *Algunas reflexiones sobre la movilidad urbana en Colombia desde la perspectiva del desarrollo humano*. Grupo de Investigación Interfacultades, Bogotá, Universidad Javeriana de Bogotá, p. 5.
- Darwin, C. (1859) *The Origin of Species by Means of Natural Selection* Murray. London
- Deb, K. (2001) *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons
- Deb, K., S. Agrawal, A. Pratap y T. Meyarivan (2000) "A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA II" *KanGall Report 200001*. Indian Institute of Tecnology: Kampur, India
- Delgado, C. (2002) *Nuevas Técnicas Metaheurísticas: Aplicación al Transporte Escolar*. Departamento de Economía Aplicada, Universidad de Burgos, Burgos
- Delgado, C. y J. Pacheco (2001) "MinMax Vehicle Routing Problems: Application to School Transport in the Province of Burgos" *Lecture Notes in Economics and Mathematical Systems* 505, pp. 297-318
- Desaulniers, G., J. Desrosiers, A. Erdmann, M.M. Solomon y F. Soumis (2002) "The VRP with Pickup and Delivery. In The vehicle routing problem" *SIAM Monographs on Discrete Mathematics and Applications* 9:157–193
- Desrochers, M., J. Desrosiers y M. Solomon (1992) "A new optimization algorithm for the vehicle-routing problem with time windows" *Operations Research*, 40(2): p. 342-354
- Desrosiers, J., F. Soumis y M. Desrochers (1984) "Routing with time windows by column generation" *Networks*, 14(4): p. 545-565
- Desrosiers, J., Y. Dumas, M.M. Solomon y F. Soumis (1995) "Time constrained routing and scheduling" *Handbooks in Operations Research and Management Science, Elsevier Science*, Amsterdam, 8:35–139
- Dominguez, O., Juan, A., Nuez, I. y Ouelhadj, D. (2015) "An ILS-Biased Randomization algorithm for the Two-dimensional Loading HFVRP with Sequential Loading and Items Rotation" *Journal of the Operational Research Society* 67 (1): 37-53
- Dorigo M. y T. Stützle (2004) *Ant Colony Optimization*. The MIT Press
- Dorigo, M. (1992) "Optimization, Learning and Natural Algorithms" *PhD thesis, Dipartimento di Elettronica*, Politecnico di Milano
- Dorigo, M. y T. Stützle (2003) *Handbook of Metaheuristics, volume 57 of International Series In Operations Research and Management Science*, chapter "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances" pp. 251 – 285. Kluwer Academic Publisher
- Dorigo, M., V. Maniezzo y A. Coloni (1996) "The Ant system: Optimization by a Colony of Cooperating Agents" *IEEE Transactions on Systems, Man and Cybernetics*, Part B 26(1): p. 29-41
- Dowsland, K.A. (1993) "Some experiments with Simulated Annealing techniques for packing problems" *European Journal of Operations Research* 68(3), pp. 389-399
- Dowsland, K.A. y B. Adenso (2001) "Diseño de Heurísticas y Fundamentos del Recocido Simulado" *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial* 20: p. 34-52.
- Dragan, D., T. Kramberger y M. Lipicnik (2011) "Monte Carlo Simulation-based Approach to Optimal Bus Stops Allocation in the Municipality of Lasko" *Promet-Traffic & Transportation* 23, 4, 265-278

- Du Merle, O., D. Villeneuve, J. Desrosiers y P. Hansen (1999) "Stabilized column generation" *Discrete Mathematics*, 194(1-3): p. 229-237
- Ehrgott, M. y X. Gandibleux (2000) *A survey and annotated bibliography on Multiobjective Combinatorial Optimization*. OR Spektrum. 22: p. 425-460
- Ehrgott, M., K. Klamroth y C. Schwehm (2004) "An MCDM approach to portfolio optimization" *European Journal of Operational Research* 155(3): p. 752-770
- Erickson, M., A. Mayer y J. Horn (2001) "The Niche Pareto genetic algorithm 2 applied to the design of groundwater remediation systems" *Evolutionary Multi-Criterion Optimization, Proceedings, E. Zitzlet, et ál., Editors..* p. 681-695
- Fanjul, L. y Ruiz, R. (2010) "Iterated greedy local search methods for unrelated parallel machine scheduling" *European Journal of Operational Research* 207(1), 55-69
- Feo T.A. y Resende M.G.C. (1995) "Greedy Randomized Adaptive Search Procedures" *Journal of Global Optimization* 2, pp. 1-27
- Feo, T.A. y M.G.C. Resende (1989) "A Probabilistic Heuristic for a Computationally Difficult Set Covering Problem" *Operations Research Letters* 8: p. 67-71
- Feo, T.A. y M.G.C. Resende (1999) "Greedy randomized adaptive search procedures" *Journal of Global Optimization* 6:109 – 133
- Fourman, M.P. (1985) "Compactation of symbolyc layout using genetic algorithms" *Genetic Algorithms and their Applications: Proceeding of the First International Conference on Genetics Algorithms and Their applications*. Lawrence Erlbaum p. 141-153
- Fu, L. (2002) "Scheduling dial-a-ride paratransit under time-varying, stochastic congestion" *Transportaion Research Part B* 36:485-506
- Gandibleux, X. y A. Freville (2000) "Tabu search based procedure for solving the 0-1 multiobjective knapsack problem: The two objectives case" *Journal of Heuristics*, 6(3): p. 361-383
- Gandibleux, X., D. Vancoppenolle y D. Tuytens (1998) "A first making use of GRASP for solving MOCO problems" *Proceeding of the 14th International Conference on Multiple. Criteria Decision-Making*., p. 8-12
- Gandibleux, X., N. Mezdaoui y A. Freville (1997) "A tabú search procedure to solve multiobjective combinatorial optimization problems" *Advances in multiple objective and goal programming, Lecture Notes in Economics and Mathematical Systems*, pags 291-300, R. Caballero, F. Ruiz, y R. Steuer, Editors, Springer: Berlín
- García I., Pacheco J. y Álvarez A. (2013) "Optimizing routes and stock", *Journal of Heuristics* 19, 2, pp. 157-177
- García de la Parra, A. (2015) *Racionalización de la red de autobuses de tránsito rápido (BRT). Análisis de diferentes objetivos. Aplicación a la ciudad de México* Dpto. Economía Aplicada, Universidad de Burgos.
- García-Martínez, C., Rodríguez, F.J. y Lozano, M. (2014) "Tabu-enhanced iterated greedy algorithm: A case study in the quadratic multiple knapsack problema" *European Journal of Operational Research* 232(3), 454-463
- García-Schilardi, M.E. (2014) "Transporte público colectivo: su rol en los procesos de inclusión social" *Bitácora Urbano Territorial* 1(24)
- Garey, M.R. (1978) *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York
- Glover F. (1989) "Tabu Search: Part I" *ORSA Journal on Computing* 1, pp. 190-206
- Glover F. (1990) "Tabu Search: Part II" *ORSA Journal on Computing* 2, pp. 4-32

- Glover F. (1994) "Genetic Algorithms and Scatter Search: Unsuspected Potentials" *Statistics and Computing* 4, pp. 131-140
- Glover F. (1998) "A Template for Scatter Search and Path Relinking" *Artificial Evolution, Lecture Notes in Computer Science*, 1363, J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer and D. Snyers (Eds.) Springer pp.13-54.
- Glover, F. (1977) "Heuristics for integer programming using surrogate constraints" *Decision Sciences* 8:156 – 166
- Glover, F. (1986) "Future Paths for Integer Programming and Links to Artificial Intelligence" *Computers and Operational Research*, 13, 533–549
- Glover, F. (1993) "Tabu Thresholding: Improved Search by Nonmonotonic Trajectories" *ORSA Journal of Computing*, 1993
- Glover, F. (2000) "Multi-start and strategic oscillation methods - principles to exploit adaptive memory" *Computing Tools for modeling, Optimization and Simulation*, M. Laguna y J.L. González Velarde, Editors.. p. 1-24
- Glover, F. y G. A. Kochenberger (2003) *Handbook of Metaheuristics*. Kluwer
- Glover, F. y M. Laguna (1993) "Tabu Search" *Modern Heuristic Techniques for Combinatorial Problems (C. Reeves ed.)* Blackwell Scientific Publishing, pp. 70-141
- Glover, F. y M. Laguna (1997) *Tabu search*. Kluwer Academic Publishers, Boston
- Glover, F., M. Laguna y R. Martí (2000) "Fundamentals of Scatter Search and Path Relinking" *Control and Cybernetics*, volume 29, number 3, pp. 653-684
- Glover, F., M. Laguna y R. Martí (2003a) "Scatter Search and Path Relinking: Advances and Applications" Chapter 1: In *Handbook of Metaheuristics*. Kluwer Academics. 1-35
- Glover, F., M. Laguna y R. Martí (2003b) "Scatter Search: Diseño Básico y Estrategias Avanzadas" *Revista Iberoamericana de Inteligencia Artificial* 19 123-130
- Glover, F., M. Laguna y R. Martí (2006) "Principles of Tabu Search" *Approximation Algorithms and Metaheuristics*, Chapman & Hall/CRC
- Goldberg D.E. (1994) *Genetic and Evolutionary Algorithms Come of Age*. *Communications of the ACM*. 37, 3.
- Goldberg, D.E. (1989) "Genetic Algorithms" *Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co
- Gómez, J.R. (2010) *Diseño de un sistema de recogida de residuos urbanos: enfoque multiobjetivo y uso de metaheurísticos*. Tesis Doctoral, Universidad de Burgos
- Gómez, J.R., J. Pacheco y H. Gonzalo-Orden (2015) "A Tabu Search Method for a Bi-Objective Urban Waste Collection Problem" *Computer-Aided Civil and Infrastructure Engineering*, 30 (1), pp. 36-53
- González-R, P.L., J.M. Framinan, P. Pérez-González y R. Ruiz (2006) "Optimización en sistemas a medida mediante el algoritmo Cross-Entropy" *X Congreso de Ingeniería de Organización*, Valencia. (Spain)
- González-Velarde J.L. (2007) "GRASP: Procedimientos de búsqueda miopes, aleatorizados y adaptativos" en *Procedimientos metaheurísticos en economía y empresa* (Crespo E., Martí R. y Pacheco J. coord.) Monografías Rect@ 3, pp.75-95
- Grasas A., Juan, A.A. y Lourenço H.R. (2016) "SimILS: A Simulation-based extension of the Iterated Local Search metaheuristic for Stochastic Combinatorial" *Optimization, Journal of Simulation* 10(1), 69–77

- Grasas, A., A. Juan y H. Ramalinho (2016) "SimILS: A Simulation-based extension of the Iterated Local Search metaheuristic for Stochastic Combinatorial Optimization" *Journal of Simulation* 10(1): 69-77
- Gravel, M., W.L. Price y C. Gagne (2002) "Scheduling continuous casting of aluminum using a multiple objective ant colony optimization metaheuristic" *European Journal of Operational Research* 143(1): p. 218-229
- Grefenstette, J.J. (1986) "Optimization of control parameters for genetic algorithms" *IEEE Transactions on Systems, Man, and Cybernetics* 16(1): p. 122-128
- Guastaroba, G. y M.G. Speranza (2012a) "Kernel search: An application to the index tracking problem" *European Journal of Operational Research* 217 (1): pp. 54-68
- Guastaroba, G. y M.G. Speranza (2012b) "Kernel search for the capacitated facility location problem" *Journal of Heuristics* 18 (6): pp. 877-917
- Guastaroba, G. y M.G. Speranza (2014) "A heuristic for BILP problems: The single source capacitated facility location problem" *European Journal of Operational Research* 238 (2): pp. 438-450
- Guerriero F., F. Pezella, O. Pisacane y L. Trollini (2014) "Multi-objective optimization in dial-a-ride public transportation" *Transportation Research Procedia*, 3:299-308.
- Guntsch, M. y M. Middendorf (2003) "Solving multi-criteria optimization problems with population-based ACO" *Evolutionary Multi-Criterion Optimization, Proceedings* 2632: p. 464-478
- Gutiérrez, A. (2003) "Transporte público y exclusión social. Reflexiones para una discusión en Latinoamérica tras la década del noventa" *XIII Congreso Latinoamericano de Transporte Público y Urbano*. Lima
- Gutiérrez, A. (2008) "Geografía, transporte y movilidad". *Espacios de Crítica y Producción*. Núm. 37
- Gutiérrez, A. (2009) "De las redes de transporte al problema de la movilidad: límites físicos y analíticos de la expansión urbana en Buenos Aires". *Buenos Aires, la formación del presente*. Quito: OLAC HI.
- Hajek, B. (1988) "Cooling schedules for optimal annealing" *Mathematics of Operations Research*. 13: p. 311-329
- Hansen, M.P.(1997) "Tabu Search for Multiobjective Optimization MOTS" *13th International Conference on Multiple Criteria Decision Making*, Cape Town, South Africa
- Hansen, P. y N. Mladenovic (1997) "Variable neighborhood search for the p-median" *Location Science* 5: p. 207-226
- Hansen, P. y N. Mladenovic (2002) "A tutorial on variable neighborhood search" *Les Cahiers du GERAD*
- Hart, W.E., N. Krasnogor y J.E. Smith (2004) *Recent Advances in Memetic Algorithms* (Studies in Fuzziness and Soft Computing)
- Healy, P. y R. Moll (1995) "Theory and Methodology: A new extension of local search applied to the dial-a-ride problema" *European Journal of Operational Research* 83:83-104
- Hernández-Díaz, A.G., L.V. Santana-Quintero, C. Coello, R. Caballero y J. Molina (2006) "A new proposal for multi-objective optimization using differential evolution and rough sets theory" *Gecco 2006: Genetic and Evolutionary Computation Conference*, Vol 1 and 2, M. Keijzer, Editor. p. 675-682
- Hertz, A., B. Jaumard, C. C. Ribeiro y W. P. Formosinho (1994) "A multicriteria tabu search approach to cell formation problems in group technology with multiple objectives" *Recherche Operationelle / Operational Research* 28(3): p. 303-328
- Higgins, A.J., S. Hajkowicz y E. Bui (2008) "A multi-objective model for environmental investment decision making" *Computers and Operations Research*, 35(1): p. 253-266

- Hilliard, M.R., G. E. Liepins, M. Palmer y G. Rangarajen (1989) “The computer as a partner in algorithmic design: Automatec discovery of parameters for a multiobjective scheduling heuristic” *Impact of Recent Computer Advances on Operations Research.*, B.L.G. R.Sharda, E. Wasil, O. Balci and W. Steward, Editor. North Holland Publishing Company
- Ho, S.C. y D. Haugland (2011) “Local search heuristics for the probabilistic dial-a-ride problem” *OR Spectrum* 33, 4, pp. 961-988
- Ho, S.C. y M. Gendreau (2006) “Path relinking for the vehicle routing problema” *Journal of Heuristics*, 12(1-2): p. 55-72
- Holland J. (1975) *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor.
- Homberger, J. y H. Gehring (1999) “Two evolutionary metaheuristics for the vehicle routing problem with time Windows” *Infor*, 37(3): p. 297-318
- Horn, J. y N. Nafpliotis (1993) “Multiobjective Optimization using the Niche Pareto Genetic Algorithm” *Technical Report IlliGA1 Report 93005*, University of Illinois at Urbana-Champaign, Illinois, USA
- Hunsaker, B. Y M.W.P. Savelsbergh (2002) “Efficient feasibility testing for dial-a-ride problems” *Operations Research Letters*, 30 pp. 169–173
- Ioachim, I., J. Desrosiers, Y. Dumas, M.M. Solomon y D. Villeneuve (1995) “A Request Clustering-Algorithm for Door-to-Door Handicapped Transportation” *Transportation Science* 29, 1, pp. 63-78
- Jones, D.F., S.K. Mirrazavi y M. Tamiz (2002) “Multi-objective meta-heuristics: An overview of the current state-of-the-art” *European Journal of Operational Research*, 137(1): p. 1-9
- Juan, A., J. Faulin, J. Jorba, J. Caceres y J. Marques (2013) “Using Parallel & Distributed Computing for Solving Real-time Vehicle Routing Problems with Stochastic Demands”. *Annals of Operations Research*, 207: 43-65
- Juan, A., J. Faulin, S. Grasman, D. Riera, J. Marull y C. Mendez (2011): “Using Safety Stocks and Simulation to Solve the Vehicle Routing Problem with Stochastic Demands” *Transportation Research Part C*, 19: 751-765
- Juan, A., J. Faulin, S. Grasman, M. Rabe y G. Figueira (2015): “A review of Simheuristics: extending metaheuristics to deal with stochastic optimization problems”. *Operations Research Perspectives*, 2: 62-72
- Juan, A., Pascual, I., Guimarans, D. y Barrios, B. (2015) “Combining Biased Randomization with Iterated Local Search for solving the Multi-Depot Vehicle Routing Problem” *Int. Transactions in Operational Research*, 22 (4): 647-667
- Jünger M., Reinelt G. and Rinaldi G. (1995) [a] “The travelling salesman problem” *Handbook in Operations Research and Management Science: Network Models*. Elsevier
- Jünger M., Reinelt G. and Thienel S. (1995) [b] “Practical problem solving with cutting plane algorithms in combinatorial optimization” *Combinatorial Optimization*, pages 111–152. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Providence
- Kallehauge, B., N. Boland y O.B.G. Madsen (2007) “Path inequalities for the vehicle routing problem with time Windows” *Networks*, 49(4): p. 273-293
- Kallehauge, B.(2008) “Formulations and exact algorithms for the vehicle routing problem with time Windows” *Computers and Operations Research*, 35(7): p. 2307-2330
- Kennedy, J. (1999) “Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance” *In Proceedings of IEEE Congress on Evolutionary Computation (CEE 1999)*, pages 1931 – 1938
- Kennedy, J. y Eberhart R. (1995) “Particle swarm optimization”, in *Proc. of the IEEE Int. Conf. on Neural Networks*, Piscataway, NJ, pp. 1942–1948

- Kennedy, J. y R.C. Eberhart (2001) *Swarm Intelligence*, M.K. Publishers, Editor. California, USA.
- Kilby, P., P. Prosser y P. Shaw (1997) "Guided Local Search for the Vehicle Routing Problem" *2nd Metaheuristics International Conference (MIC 97)* Sophie Antipolis, France
- Kim, J.H. y S. Soh (2012) "Designing Hub-and-Spoke School Bus Transportation Network: A Case Study of Wonkwang University" *Promet-Traffic & Transportation* 24, 5, 389-394
- Kirpatrick, S., Gelatt Jr CD y Vecchi MP (1983) "Optimization by Simulated Annealing" *Science* 220, pp. 671-680
- Knight, K.W. y J.P. Hofer (1968) "Vehicle scheduling with timed and connected calls. A case study" *Operational Research Quarterly* 19(3): p. 299
- Knowles, J.D. y D.W. Corne (2000) "Approximating the nondominated front using the Pareto Archived Evolution Strategy" *Evolutionary computation* 8(2): p. 149-172
- Kohl, N., J. Desrosiers, O.B.G. Madsen, M.M. Solomon y F. Soumis. (1999) "2-path cuts for the vehicle routing problem with time Windows" *Transportation Science*, 33(1): p. 101-116
- Kolen, A.W.J., A. Kan y H. Trienekens (1987) "Vehicle-routing with time windows" *Operations Research*, 35(2): p. 266-273
- Koulmas C, Antony S.R. y Jaen R. (1994) "A survey of simulated annealing applications to operations-research problems" *Omega, International Journal of Operations and Production Management* 22, 1, pp.41-56.
- Laguna M. (2002) "Scatter Search", *Handbook of Applied Optimization*, P. M. Pardalos and M. G. C. Resende (Eds.), Oxford University Press, pp. 183-193.
- Laguna M. y Martí R. (2003) *Scatter Search: Methodology and Implementations* in C, Kluwer Academic Publishers: Boston.
- Laguna, M. y R. Martí (1999) "GRASP and Path Relinking for 2-layer straight line crossing minimization" *INFORMS Journal on Computing*, 11, 1, pp. 44-52
- Laguna, M. y C. Delgado (2007) "Introducción a los metaheurísticos" *Procedimientos Metaheurísticos en Economía y Empresa*, Monografías de Rect@ 3, Tirant lo Blanch, p. 3-28
- Landa, R. (2002) "Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Multiobjetivo" *Departamento De Ingeniería Eléctrica. Sección De Computación, Centro De Investigación Y De Estudios Avanzados Del Instituto Politécnico Nacional: México*
- Larrañaga, P., J.A. Lozano y H. Muhlenbein (2003) "Estimation of Distribution Algorithms Applied To Combinatorial Optimization Problems" *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial* 19: p. 149-168.
- Larrañaga, P., R. Etxeberria, J.A. Lozano y J.M. Peña (1999) "Optimization by learning and simulation of Bayesian and Gaussian networks" *Technical Report KZZA-IK-4-99*, Department of Computer Science and Artificial Intelligence, University of the Basque Country
- Li L, Zhang H, Wang X, Lu W y Mu Z (2011) "Urban Transit Coordination Using an Artificial Transportation System" *IEEE Transactions on Intelligent Transportation Systems* 12, 2, 374-383
- LIBRO VERDE (2002) *Libro Verde de la accesibilidad en España: Diagnóstico de situación y bases para elaborar un plan integral de supresión de barreras*. Instituto de Mayores y Servicios Sociales (España)
- Liu H, Liu T, Liu L, Guo HC, Yu YJ y Wang Z (2010) "Integrated Simulation and Optimization Approach for Studying Urban Transportation-Environment Systems in Beijing" *Journal of Environmental Informatics* 15, 2, pp. 99-110
- Lizárraga, C. (2006) "Movilidad urbana sostenible: un reto para las ciudades del siglo xx i". *Economía, sociedad y territorio*. Vol. 6. Núm. 22

- Los, M. y C. Lardinois (1982) "Combinatorial Programming, Statistical Optimization and the Optimal Transportation Network Problem" *Transportation Research* 2: p. 89-124
- Lourenço H.R., Martin O. y Stützle T. (2010) "Iterated Local Search: Framework and Applications" *In Handbook of Metaheuristics*, 2nd. Edition. Vol.146, pp. 363-397.
- Lourenço, H. R., O. Martin, y T. Stützle (2002) *Handbook of Metaheuristics* - chapter Iterated local search - pages 321 – 353. Kluwer Academic Publishers
- Lozano, M., Molina, D. y García-Martínez, C. (2011) "Iterated greedy for the maximum diversity problema" *European Journal of Operational Research* 214(1), 31-38
- Madavan, N.K. (2002) "Multiobjective Optimization Using a Pareto Differential Evolution Approach" *CEC2002*. p. 1145-1150
- Madsen, O.B.G. (1976) "Optimal scheduling of trucks- A routing problem with tight due times for delivery" *Optimization applied to transportation systems*, IIASA, International Institute for Applied System analysis: Laxenburgh, Austria. p. 126-136
- Madsen, O.B.G., H.F. Ravn y J.M. Rygaard (1995) "A heuristic algorithm for the dial-a-ride problem with time windows, multiple capacities, and multiple objectives" *Annals of Operations Research* 60:193–208
- Mandujano, P., R. Giesen y J.C. Ferrer (2012) "Model for Optimization of Locations of Schools and Student Transportation in Rural Areas" *Transportation Research Record* 2283, pp. 74-80
- Martí, R. (2003) "Multistart methods" *Handbook of Metaheuristics*, F. Glover y A. Kochenberger, Editors, p. 355-368
- Martí, R. y J.A. Moreno (2003) "Métodos Multiarranque. Inteligencia Artificial" *Revista Iberoamericana de Inteligencia Artificial* 19: p. 49-60
- Martí, R., A. Corberán y J. Peiró (2015) "Scatter search for an uncapacitated p-hub median problema" *Computers and Operations Research*, 58, pp. 53-66
- Martí, R., V. Campos, M. Resende y A. Duarte (2015) "Multi-objective GRASP with path-relinking" *European Journal of Operational Research* 240, 54-71
- Martínez-Puras A. y Pacheco J. (2016) "MOAMP-Tabu search and NSGA-II for a real Bi-objective scheduling-routing problem" *Knowledge-Based Systems* 112 pp. 92-104
- Mauri, G.R. y L.A. Nogueira Lorena (2006) *A Multiobjective Model and Simulated Annealing Approach for a Dial-a-Ride Problem*
- Melián, B. y F. Glover (2004) *Introducción a la búsqueda tabú*
- Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller, y E. Teller (1953) "Equation of state calculations by fast computing machines" *Journal of Chemical Physics*, 21:1087 – 1092
- Ministerio de Fomento. Encuesta de Movilidad de las personas residentes en España (MOVILIA 2006/2007).
- Mitchell, M. (1998) *An Introduction to Genetic Algorithms*, MIT Press
- Mladenovic, N. (1995) "A Variable Neighborhood Algorithm - A New Metaheuristic for Combinatorial Optimization" *Abstract of papers presented at Optimization Days*. Montreal p. 112
- Molenbruch, Y., K. Braekers, A. Caris y G. Vanden Berghe (2017) "Multi-directional local search for a bi-objective dial-a-ride problem in patient transportation" *Computers and Operations Research*, 77, pp. 58-78
- Molina, J., M. Laguna, R. Martí y R. Caballero (2005) *SSPMO: A Scatter Tabu Search Procedure for Non-Linear Multiobjective Optimization*
- Moreno, J.A. y Mladenovic, N. (2013) *Búsqueda por Entornos Variables para Planificación Logística*, Universidad de La Laguna, La Laguna, España

- Moscato, P. (1989) "On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetics Algorithms" *Technical Report Caltech Concurrent Computation Program, C3P*. California Institute of Technology, Pasadena. California USA,. Report 826
- Moura JL, Ibeas A y dell'Olio L (2010) "Optimization-Simulation Model for Planning Supply Transport to Large Infrastructure Public Works Located in Congested Urban Areas" *Networks and Spatial Economics* 10, 4, pp. 487-507
- Muelas, S., A. Latorre y J.M. Peña (2013) "A Variable Neighborhood Search Algorithm for the Optimization of a Dial-a-Ride Problem in a Large City" *Expert Systems with Applications* 40(14):5516-5531
- Mühlenbein, H. y G. Paab (1986) "From recombination of genes to the estimation of distributions I. Binary parameters" *Lecture Notes in Computer Science* 1411: Parallel Problem Solving from Nature,: p. 178-187
- Nemhauser, G. y L. Wolsey (1988) *Integer and Combinatorial Optimization*, John Wiley&Sons
- Nunes de Castro, L. y J. Timmis (2002) *An Introduction to Artificial Immune Systems: A New Computational Intelligence Paradigm*. Springer-Verlag
- Or, I. (1976) *Traveling Salesman-Type Combinatorial Problems and their Relation to the Logistics of Regional Blood Banking*, Ph.D. thesis, Northwestern University, Evanston, Illinois
- Osman, I.H. and Kelly, J.P. (eds.) (1996) *Meta-Heuristics: Theory and Applications* Ed. Kluwer Academic, Boston
- Osyczka, A. (1895) "Multicriteria optimization for engineering design" *Design Optimization*, J. S. Gero, editor,, pages 193 – 227. Academic Press
- Pacheco, J. y C. Delgado (2000) "Diseño de Metaheurísticos para Problemas de Rutas con Flota Heterogénea: Concentración Heurística" *Estudios de Economía Aplicada*, 14: p. 137-151
- Pacheco, J. y R. Martí (2006) "Tabu Search for a Multiobjective Routing Problem" *Journal of the Operational Research Society* 57, pp. 29-37
- Pacheco, J., A. Álvarez, S. Casado y J.L. González-Velarde (2009) "A tabu search approach to an urban transport problem in northern Spain" *Computers and Operations Research* 36, 3, pp. 967-979
- Pacheco, J., R. Caballero, M. Laguna y J. Molina (2013) "Bi-objective Bus Routing: An Application to School Buses in Rural Areas" *Transportation Science* 47, 3, pp. 397-411
- Padberg, M. W. y G. Rinaldi (1991) *A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems*. *SIAM Review*, 33:60–100
- Paquette J., F. Bellavance, J. Cordeau y G. Laporte (2012) "Measuring quality of service in dial-a-ride operations: the caso of a Canadian city" *Transportation* 39:539-564
- Paquette J., J. Cordeau, G. Laporte y M. Pascoal (2013) "Combining multicriteria analysis and tabu search for dial-a-ride problems" *Transportation Research Part B* 52:1-16
- Pareto, V. (1896) *Cours D'Economie Politique*, volume I and II. F. Rouge, Lausanne
- Park, J. y B.I. Kim (2010) "The school bus routing problem: A review" *European Journal of Operational Research* 202, 2, pp. 311-319
- Parragh, S.N. y V. Schmid (2013) "Hybrid column generation and large neighborhood search for the dial-a-ride problem" *Computers & Operations Research* 40, 1, pp. 490-497
- Parragh, S.N., J.F. Cordeau, K.F. Doerner y R.F. Hartl (2012) "Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints" *OR Spectrum* 34, 3, pp. 593-633
- Parragh, S.N., K.F. Doerner, R.F. Hartl y X. Gandibleux (2009) "A Heuristic Two-Phase Solution Approach for the Multi-Objective Dial-A-Ride Problem" *Networks* 54, 4, pp. 227-242

- Parsopoulos, K.E., D.K. Taoulis, N.G. Pavlidis, V.P. Plagianakos y M.N. Vrahatis (2004) "Vector evaluated differential evolution for multiobjective optimization" *Cec2004: Proceedings of the 2004 Congress on Evolutionary Computation*, Vols 1 y 2. p. 204-211
- Pelikan, M., D.E. Goldberg y F. Lobo (1999) "A survey of optimization by building and using probabilistic models" *Technical Report IlliGAL Report 99018* University of Illinois at Urbana-Champaign
- Perez, F., J. Molina, R. Caballero, C.A. Coello y A.G. Hernández-Díaz (2007) "Hibridación de métodos exactos y heurísticos para el problema multiobjetivo" *XV Jornadas de ASEPUMA y III Encuentro Internacional*. Palma de Mallorca (Spain)
- Poli R., Kennedy J. y Blackwell T. (2007) "Particle swarm optimization" *Swarm Intelligence* 1, 1, pp. 33-57
- Ponce, M.T. y M. Matos (1999) "Multicriteria distribution network planning using simulated annealing" *International Transactions in Operational Research*, 6(4): p. 377-391
- Psaraftis, H. N. (1980) "A dynamic programming solution to the single vehicle, many-to-many, immediate request dial-a-ride problema" *Transportation Science* 14(2):130-154
- Psaraftis, H. N. (1983) "An exact algorithm for the single vehicle, many-to-many dial-a-ride problem with time Windows" *Transportation Science* 17(3): 351-357
- Pullen, H.G.M. y M.H.J. Webb (1967) "A computer application to a transport scheduling problema" *Computer Journal*, 10(1): p. 10
- Real Academia Española (2014) *Diccionario de la lengua española (23.ª edición)* Madrid: Espasa. Consultado en <http://dle.rae.es/>
- Reeves, C. (2003) "Genetic Algorithms" Chapter 3 in *Handbook of Metaheuristics*. Kluwer Academics. 55-82
- Reeves, C.R. (1993) *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publishing, Oxford, UK
- Reynolds, R.G. (1994) "An Introduction to Cultural Algorithms" *Third Annual Conference on Evolutionary Programming*, A.V. Sebald y L.J. Fogel, Editors, World Scientific, River Edge, New Jersey. p. 131-139
- Ritzinger, U., J. Puchinger y R. F. Hartl (2016) "Dynamic programming based metaheuristics for the dial-a-ride problema" *Annals of Operations Research*, Springer Verlag, 236 (2)
- Rosing, K.E., C.S. Revelle y D.A. Schilling (1999) "A gamma heuristic for the p-median problema" *European Journal of Operational Research*, 117(3): p. 522-532
- Rubinstein, R.Y. (1997) "Optimization of computer simulation models with rare events" *European Journal of Operational Research* 99(1): p. 89-112
- Rubinstein, R.Y. (2001) "Combinatorial optimization, cross-entropy, ants and rare events" *Stochastic Optimization: Algorithms and Applications* 54: p. 303-363
- Ruisánchez F, dell'Olio L y Ibeas A (2012) "Design of a tabu search algorithm for assigning optimal bus sizes and frequencies in urban transport services" *Journal of Advanced Transportation* 46, 4, 366-377
- Ruiz, R. y Stützle, T. (2007) "A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problema" *European Journal of Operational Research* 177(3), 2033-2049
- Santana-Quintero, L.V., N. Ramírez-Santiago, C. A. Coello, J. Molina y A. G. Hernández-Díaz (2006) "A new proposal for multiobjective optimization using particle swarm optimization and rough sets theory" *Parallel Problem Solving from Nature - Ppsn Ix, Proceedings* 4193: p. 483-492

- Schaffer, J.D. (1985) "Multiple Objective optimization with vector evaluated genetic algorithms" *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*. pags 93-100.: Lawrence Erlbaum
- Schilde, M., K.F. Doerner y R.F. Hartl (2011) "Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports" *Computers & Operations Research* 38, 12, pp. 1719-1730
- Schmid, V. (2014) "Hybrid large neighborhood search for the bus rapid transit route design problem" *European Journal of Operational Research* 238, 2, 427-437
- Serafini, P. (1992) "Simulated Annealing for multiobjective optimization problems" *Proceedings of the 10th International Conference on Multiple Criteria Decision Making*. Taipei, Taiwan., 1: p. 87-96
- Sharma, S. y T.V. Mathew (2011) "Multiobjective network design for emission and travel-time trade-off for a sustainable large urban transportation network" *Environment and Planning B-Planning and Design* 38, 3, pp. 520-538
- Silver, E., R.V. Vidal y D. Werra (1980) "A Tutorial on Heuristic Methods" *European Journal of Operational Research* 5, 3, pp. 153-162
- Solomon, M.M. (1987) "Algorithms for the vehicle-routing and scheduling problems with time window constraints" *Operations Research*, 35(2): p. 254-265
- Soto, M., A. Ochoa, S. Acid, y L. M. de Campos (1999) "Introducing the polytree approximation of distribution algorithm" *In Second Symposium on Artificial Intelligence. Adaptive Systems*. CIMAF 99, pages 360 – 367
- Srinivas, N. y K. Deb (1994) "Multiobjective Optimization using Nondominated Sorting" *Genetic Algorithm. Evolutionary computation*. 2: p. 221-248
- Stanley, J., D. Hensher, J. Stanley, G. Currie, W. Greene y D. Vella-Brodrick (2010) *Social Exclusion and the Value of Mobility*. Sydney: University of Sydney
- Steuer, R.E. (1986) *Multiple Criteria Optimization: Theory, computation and Application*, Wiley, New York
- Storn, R. y K. Price (1997) "Differential Evolution- A Fast and Efficient Heuristic for Global Optimization over Continuous Spaces" *Journal of Global Optimization* 11: p. 341-359
- Stützle, T. (1999) "Local search algorithms for combinatorial problems analysis, algorithms and new applications". Technical report, *DISKI Dissertationen zur Künstlichen Intelligenz*. Sankt Augustin, Germany
- Taillard, E.D., L.M. Gambardella, M. Gendreau y J.-Y. Potvin (2001) "Adaptive memory programming: A unified view of metaheuristics" *European Journal of Operational Research* 135(1): p. 1-16
- Taillard, E.D., P. Badeau, M. Gendreau, F. Guertin y J.-Y. Potvin (1997) "A tabu search heuristic for the vehicle routing problem with soft time Windows" *Transportation Science* 31(2): p. 170-186
- Tang, J., Y., Kong, H. Lau, y A.W.H. Ip (2010) "A note on "Efficient feasibility testing for dial-a-ride problems"" *Operations Research Letters*, 38(5) pp. 405-407
- Teghem, J., D. Tuytens y E.L. Ulungu (2000) "An interactive heuristic method for multi-objective combinatorial optimization" *Computers & Operations Research*, 27(7-8): p. 621-634
- Toth, P. y D. Vigo (1997) "Heuristic algorithms for the handicapped persons transportation problem" *Transportation Science* 31(1):60-71
- UITP [Unión Internacional de Transporte Público] (2007) *La lucha contra la exclusión social. El papel del transporte público*. Bruselas
- UITP [Unión Internacional de Transporte Público] (2009) *Evaluación de las ventajas del transporte público*. Bruselas

- UITP [Unión Internacional de Transporte Público] (2014) *Transporte público : el motor de Europa para el periodo 2014-2019*. Bruselas
- Ulungu, E.L., J. Teghem y C. Ost (1998) "Efficiency of interactive multi-objective simulated annealing through a case study" *Journal of the Operational Research Society*, 49(10): p. 1044-1050
- Vega, P. (2006) *La accesibilidad del transporte en autobús: Diagnóstico y soluciones*. Instituto de Mayores y Servicios Sociales (España)
- Voudouris, C. y E. Tsang (1995) "Guided Local Search" *Department of Computer Science*, University of Essex: Colchester, C04 3SQ, UK
- Whittaker, J. (1990) *Graphical models in applied multivariate statistics*. John Wiley & Sons, Inc.
- Wilson, H. y H. Weissberg (1977) "Advanced dial-a-ride algorithms research project: Final Report" *Department of Civil Engineering, MIT. Technical Report R-76-20*: Cambridge
- Wilson, H. y N. Colvin (1977) "Computer control of the Rochester dial-a-ride system" *Department of Civil Engineering, MIT. Technical Report R-77-31*: Cambridge, MA
- Wilson, H., J.M. Sussman, H.K. Wang y B.T. Higonnet (1971) "Scheduling algorithms for dial-a-ride systems" *Urban system Laboratory, MIT. Technical Report USL TR-70-13*: Cambridge, MA
- Wilson, P.B. y M.D. Macleod (1993) "Low Implementations cost IIR digital filter design using genetic allgorithms" *IEE/IEEE Workshop on Natural Algorithms in Signal Processing*. Chelmsford. p. 4/1 - 4/8
- Zanakis SH and Evans JR (1981) *Heuristic Optimization: When, Why and How to Use It*, Interfaces, 11, 5, pp. 84-91
- Zhang, Z., M. Liu y A. Lim (2015) "A memetic algorithm for the patient transportation problema" *Omega (United Kingdom)*, 54, pp. 60-71
- Zidi, I., K. Mesghouni, K. Zidi y K. Ghediraa (2012) "A multi-objective simulated annealing for the multi-criteria dial a ride problem" *Engineering Applications of Artificial Intelligence* 25, 6, 1121-1131
- Zitzler, E. y L. Thiele (1999) "Multiobjective evolutionary algorithms: A comparative case study and the Strength Pareto approach" *Ieee Transactions on Evolutionary Computation*. 3(4): p. 257-271
- Zitzler, E., M. Laumanns y L. Thiele (2001) "SPEA2: Improving the Strength Pareto Evolutionary Algorithm" *Computer Engineering and Networks Laboratory (TIK)*. Swiss Federal Institute of Technology (ETH): Zurich, Switzerland