

Experimentos y análisis de resultados tras la aplicación de metaheurísticas al problema de la selección de la solución deseada

E. Yeguas¹, M.V. Luzón², E. Barreiro³, R. Joan-Arinyo⁴

¹Escuela Politécnica Superior
Universidad de Córdoba
Campus Rabanales. Marie Curie (C3 - Anexo). 14071 Córdoba
eyeguas@uco.es

²E.T.S. Ingenierías Informática y Telecomunicación
Universidad de Granada
C/ Periodista Daniel Saucedo Aranda s/n. 18071 Granada
luzon@ugr.es

³Escuela Superior de Ingeniería Informática
Universidad de Vigo
Av. As Lagoas s/n, E-32004 Ourense
enrique@uvigo.es

⁴Grup d'Informàtica a l'Enginyeria
E.T.S. d'Enginyeria Industrial de Barcelona
Universitat Politècnica de Catalunya
Av Diagonal 647, 8a. 08028 Barcelona
robert@lsi.upc.edu

Resumen

En este trabajo se realizará un estudio experimental de los algoritmos desarrollados en [YLB06]. Se describen los experimentos que han sido diseñados, cómo han sido diseñados y la transformación efectuada sobre los algoritmos para poder realizarlos a partir de una integración práctica con un sistema de resolución de restricciones geométricas. Se indican los parámetros que rigen los experimentos. Finalmente, se muestran los resultados en forma de tablas y gráficas y se comenta qué implican éstos para cada algoritmo. Por último, se comparan los distintos algoritmos y las distintas filosofías entre sí desde el punto de vista de su eficacia, eficiencia y potencialidad en general prácticas para resolver el problema de la selección de la solución deseada.

1. Introducción

El problema de la selección de la solución deseada dentro del campo de la Resolución de Restricciones Geométricas se puede abordar desde diferentes puntos de vista. Entre ellas, una

de las más recientes es la que permite abordarlo como un problema de optimización combinatoria.

En base a tal propuesta, se analizaron las diferentes tendencias en algoritmos de optimización combinatoria que podrían proporcionar teóricamente buenas soluciones a nuestro problema: soluciones cercanas a las óptimas en un corto tiempo. Todos estos algoritmos quedan englobados como sabemos dentro del término de Metaheurísticas.

Dispuesto el problema y conocida la forma de resolverlo a partir de la cual se abordará, hemos de conectar ambas, esbozándose la implementación de distintos algoritmos correspondientes a las diferentes Metaheurísticas para resolver nuestro problema.

Llega por tanto la hora de ver cómo funcionan estos algoritmos, implementados en trabajos anteriores, [YLB06]. Para ello se realizarán una serie de experimentos y se analizarán los resultados de los mismos para sacar conclusiones acerca de tal funcionamiento de Metaheurísticas sobre el problema de la selección de la solución deseada.

Los experimentos consistirán en partir de una serie de casos significativos del problema y lanzar los distintos algoritmos implementados sobre los mismos para comprobar sus resultados en un corto espacio de tiempo, pues nuestro objetivo es verificar las posibilidades de obtener buenos resultados para una posible aplicación práctica.

Este trabajo presentará los distintos experimentos a realizar, cómo se han diseñado y realizado, cómo ha sido aplicado cada algoritmo: qué parámetros y ejecuciones han regido tal aplicación, los resultados obtenidos y un análisis y conclusiones tanto individuales como en conjunto de los mismos para los diferentes niveles de filosofía de Metaheurísticas implementados.

2. Diseño de los experimentos

Como sabemos, nuestro objetivo es el de realizar un estudio del comportamiento de las diferentes *Metaheurísticas* para la resolución del problema de la selección de la solución deseada dentro de un sistema de resolución de restricciones geométricas (planteado como problema de optimización combinatoria). Para ello se ha seleccionado un conjunto bastante amplio y representativo de los diferentes tipos de *Metaheurísticas* y lo hemos aplicado sobre diferentes casos significativos y representativos del problema, observando los resultados obtenidos.

Para realizar tales experimentos hemos partido de los algoritmos implementados y adaptados al problema a resolver, y de una interfaz sobre la que construir casos del problema o a la que proporcionarlos y que los transformase en entradas que pudieran hacer evolucionar a los algoritmos.

Esa interfaz corresponde a un solver que se conectará con los distintos algoritmos a la hora de la resolución a partir de optimización combinatoria de cada caso que se le presente. Así, nuestro trabajo una vez elaborados los algoritmos, ha sido el de conectarlos con el solver para realizar los experimentos.

Una vez conectados los algoritmos, ha bastado realizar los distintos experimentos con los casos y obtener los resultados que posteriormente serán objeto de estudio detallado.

Veamos a continuación cómo hemos realizado la puesta a punto de nuestros programas una vez implementados a la hora de realizar los experimentos, en qué consiste concretamente la interfaz (solver), cómo se ha realizado la integración y qué supone la realización de cada experimento o experimentación sobre un caso concreto. Posteriormente comentaremos los casos

utilizados para el estudio, los experimentos que se han realizado con tales casos y los distintos parámetros que han guiado tales experimentos.

2.1. Interfaz para la realización de los experimentos: el solver SolBCN

La resolución de un problema de restricciones geométricas exige la existencia de un solver sobre el que se plasme una figura, sus restricciones y la instanciación de los parámetros y que proporcione la solución al problema: aquella figura que más se ajusta a lo que el usuario desea, cumpliendo todas las restricciones establecidas.

En aquel momento se dispuso un ejemplo del mismo, un prototipo de sistema de satisfacción de restricciones geométricas que implementa los métodos constructivos, construido en Prolog y que presenta una interfaz que simplifica la edición de problemas, la resolución y la selección de la solución deseada. Este solver maneja configuraciones geométricas bidimensionales creadas a partir de elementos geométricos tales como puntos, segmentos, arcos y círculos. Las restricciones que pueden definirse sobre dichos elementos incluyen distancia entre dos puntos, distancia perpendicular de un punto a un segmento, ángulo entre dos segmentos, tangencia, etc.

Este solver ha sido precisamente el que hemos utilizado para la realización de los experimentos. A continuación pasamos a describirlo desde un punto de vista experimental y práctico.

El solver SolBCN es un sencillo programa que consta de varias ventanas de trabajo y que trabaja bajo el sistema operativo Linux. La ventana principal es la que alberga el croquis (sketcher).

En ella podemos diseñar, guardar, abrir un croquis realizado y albergado en un fichero (extensión .bcn5) y realizar tareas de configuración.

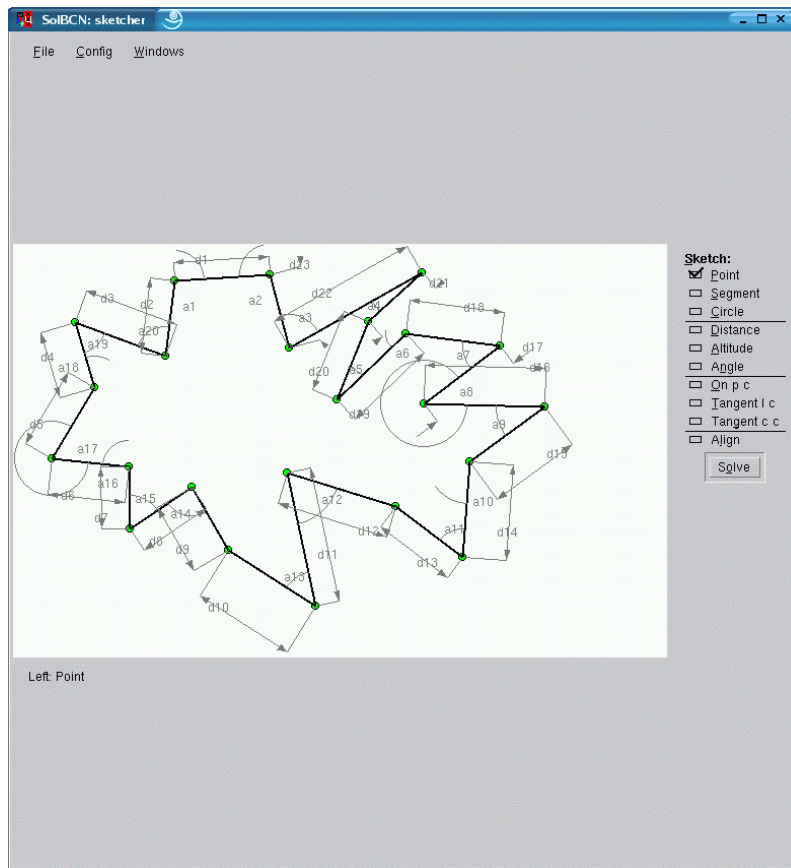


Figura 1.: Ventana principal del solver: diseño del croquis (sketcher)

Junto a esta ventana principal se disponen tres ventanas secundarias que corresponden a tres modos distintos de resolver el problema: tres modos distintos de seleccionar un índice (ya sabemos que un índice representa a una solución).

De ellas sólo nos interesa aquella que permite resolver el problema reformulándolo como un problema de optimización combinatoria: predicates-based index selector (el selector de índice basado en predicados).

En esta ventana definimos los predicados adicionales sobre el croquis y nos proporciona la solución a partir de un algoritmo de optimización combinatoria: en este caso a partir de un algoritmo genético básico.

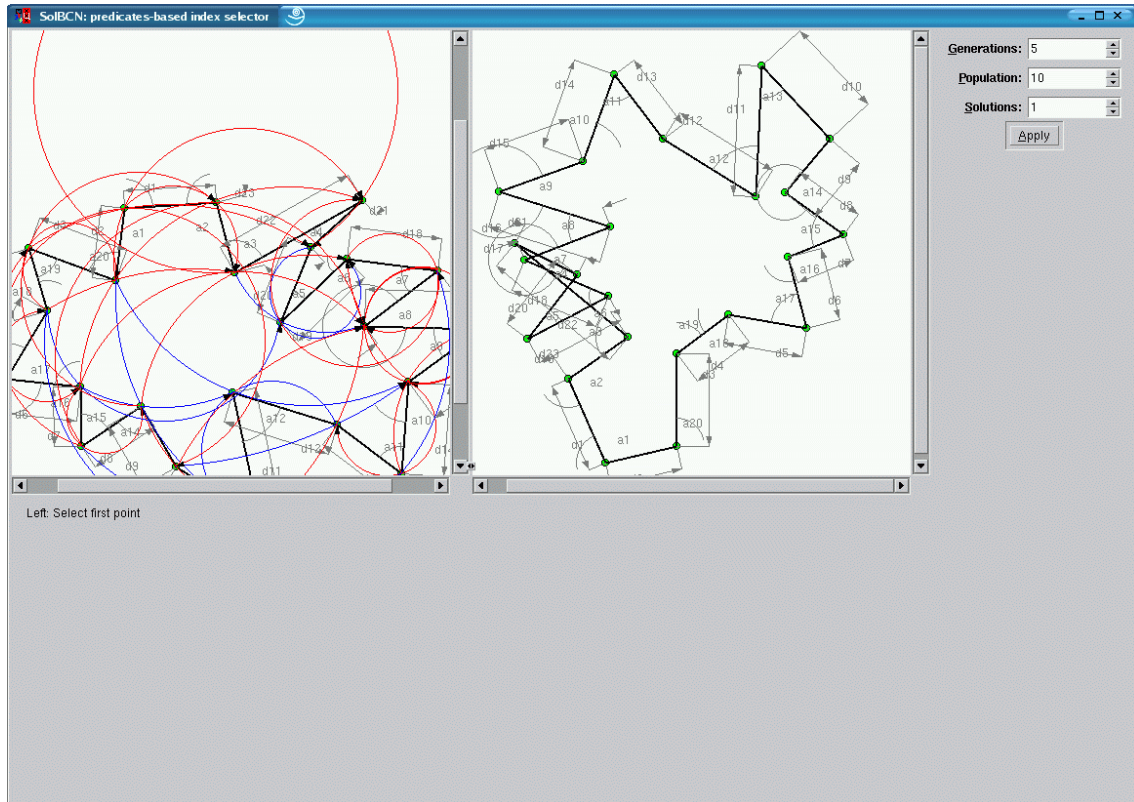


Figura 2.: Ventana del solver de selección de solución deseada basada en predicados adicionales.

El proceso para resolver un caso mediante optimización combinatoria es el siguiente. En primer lugar hemos de cargar el caso desde disco o bien diseñarlo en la ventana del croquis (sketcher). Una vez que lo hemos definido en la misma, debemos pulsar *Solve* para pasar a la resolución del mismo, después sobre la ventana de selección de índice basada en predicados hemos de definir las restricciones y finalmente una vez establecidos los parámetros del algoritmo de optimización combinatoria, pulsaremos *Apply* en esa misma ventana y se presentará la solución en el panel de la derecha. El panel de la izquierda es el que se utiliza para definir restricciones.

Para poder realizar los experimentos, nuestro objetivo es hacer que al pulsar *Apply* se ejecute el algoritmo correspondiente a la *Metaheurística* que se desee. Para ello, debemos profundizar en el código del solver y verificar la posibilidad de introducir el algoritmo deseado donde se encuentra el algoritmo genético que ejecuta por defecto.

El solver está programado en Prolog, pero nuestros algoritmos interesa que corran lo más rápido posible y de ahí que el lenguaje elegido sea C. Debemos integrar nuestro algoritmo en C dentro de Prolog, lo que se hace a partir de la librería *SWI-Prolog*. La comunicación con el solver queda establecida a partir de esta librería, tanto para recibir la entrada al algoritmo como para proporcionar la salida y así mismo, para evaluar cada una de las soluciones generadas durante cada algoritmo.

Sin embargo, podemos conectar en cada caso con un sólo programa foráneo, en principio, con un solo algoritmo.

2.2. Implementación e integración de los algoritmos para la experimentación

Tras implementar los algoritmos correspondientes a las *Metaheurísticas* seleccionadas, como ya se ha indicado, en lenguaje C, tenemos un conjunto de algoritmos de optimización sobre codificación binaria que funcionan por sí solos. Sin embargo, para poderlos aplicar a la resolución del problema de la solución deseada, como hemos visto, hemos de conectarlos con el solver, tenemos que disponerlos como parte del mismo a través de una serie de directivas y de funciones correspondientes a la librería *SWI-Prolog*. Los algoritmos carecen de independencia tras la adaptación, de tal forma que sólo funcionan a través del solver.

Puesto que en cada momento un sólo algoritmo puede funcionar con el solver, cada vez que deseemos utilizar uno distinto hemos de compilar nuevamente la parte del solver correspondiente al algoritmo y volver a ejecutarlo.

Para probar inicialmente los algoritmos, sin necesidad de depender del solver, lo que se hizo fue disponer sobre un fichero el espacio de búsqueda correspondiente a un caso sencillo del problema. Una vez que los algoritmos funcionaron bien se realizó la integración correspondiente con el solver.

De cara a la experimentación sencilla, cada algoritmo recibe un fichero de configuración en el que se le proporcionan todos los parámetros y proporciona dos ficheros de resultados: uno en el que establece la solución encontrada y otro en el que dispone el proceso de convergencia hacia tal solución. Todo ello además de proporcionar la correspondiente salida a la interfaz del solver. En la implementación original cada algoritmo recibía un parámetro que correspondía al fichero de configuración. En la correspondiente a la integración con el solver, dicho paso de parámetros se ha simulado estableciendo un nombre fijo para el fichero de configuración para cada algoritmo.

2.3. Casos experimentales

Partimos de un total de 8 casos de experimentación. Estos casos vienen proporcionados en forma de ficheros .bcn5. En ellos se establecen las figuras geométricas, las restricciones, la instanciación de parámetros y las restricciones adicionales. Las restricciones adicionales han sido definidas sobre la ventana del solver correspondiente al selector de índice basado en predicados, concretamente en su panel izquierdo.

Solamente se han definido restricciones correspondientes a punto a la izquierda de un segmento o punto a la derecha de un segmento. Éstas se definen seleccionando tres puntos sobre la figura en ese panel izquierdo una vez cargada y pulsado *Solve* en la ventana principal. El segundo punto se establece que estará a la izquierda o la derecha del segmento que establecen el primero y el tercero: ello se indica a partir del color (rojo significa izquierda y azul significa derecha).

Los casos revisten cierta complejidad. Son de distintos tamaños, todos ellos grandes y tienen definidos distinto número de predicados adicionales. Se presentan dos casos de cada tamaño (en total tendremos cuatro tamaños distintos). La Tabla 1 muestra las características de los casos que hemos utilizado para el estudio: tamaño, número de predicados adicionales que tiene definidos y el número que hemos asignado a cada uno para identificarlo del resto.

Nº Caso	Tamaño (n)	Número de Restricciones Adicionales Definidas
1	135	25
2	142	24
3	149	27
4	156	28

5	135	26
6	143	29
7	149	30
8	156	35

Tabla 1.: Casos de experimentación.

La solución óptima a cada caso debe cumplir todas las restricciones definidas, o sea, la tercera columna representa la solución óptima a alcanzar por los distintos algoritmos a la hora de resolver cada caso.

En las páginas sucesivas se muestra las figuras que corresponden a cada caso, con y sin las restricciones adicionales definidas.

CASO 1

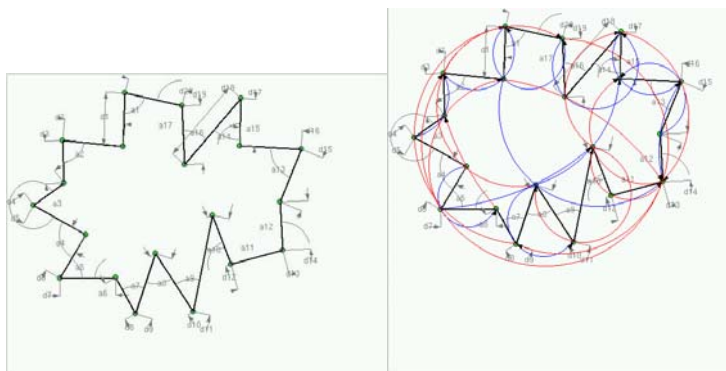


Figura 3.: Figuras correspondientes al caso 1: izqda sin restricciones definidas y dcha con las restricciones definidas por el usuario.

CASO 2

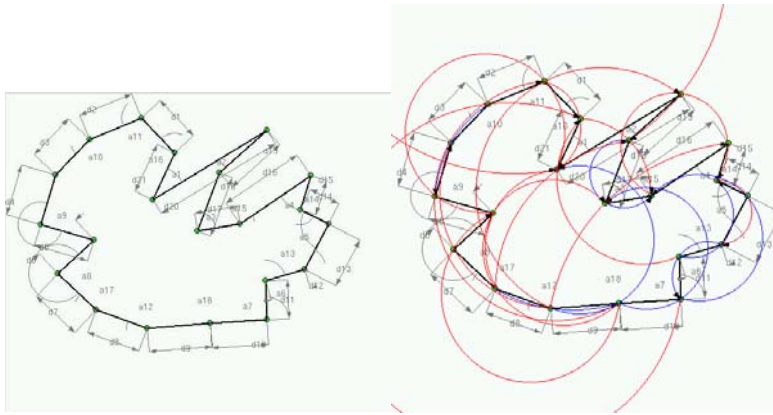


Figura 4.: Figuras correspondientes al caso 2: izqda sin restricciones definidas y dcha con las restricciones definidas por el usuario.

CASO 3

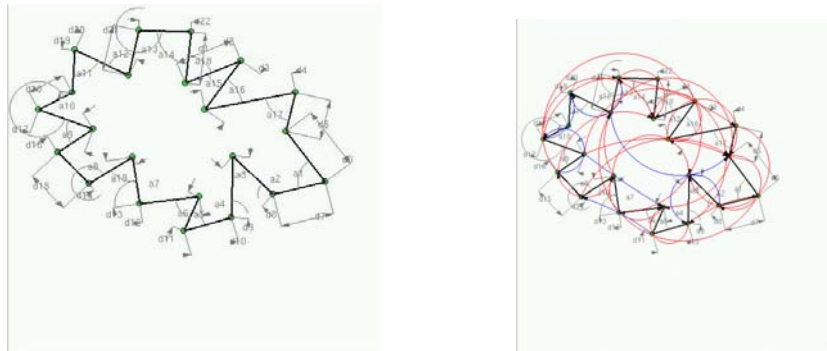


Figura 5.: Figuras correspondientes al caso 3: izqda sin restricciones definidas y dcha con las restricciones definidas por el usuario.

CASO 4

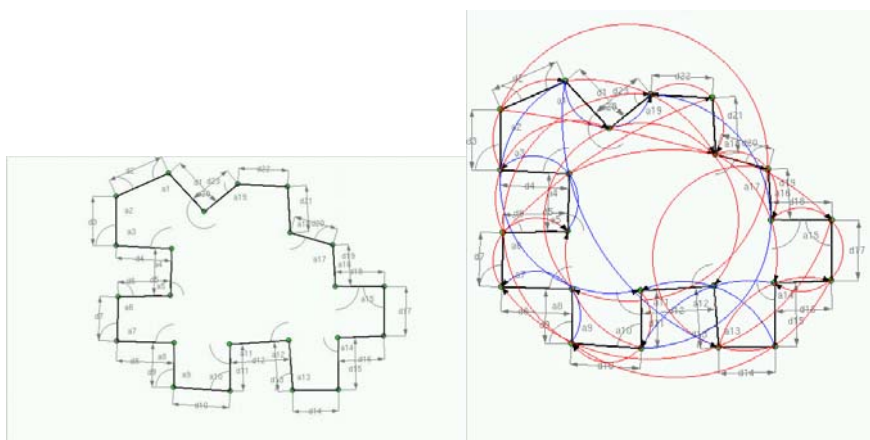


Figura 6.: Figuras correspondientes al caso 4: izqda sin restricciones definidas y dcha con las restricciones definidas por el usuario.

CASO 5

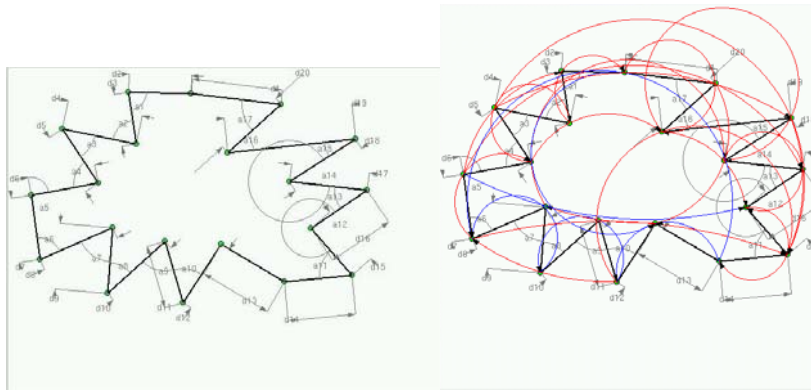


Figura 7.: Figuras correspondientes al caso 5: izqda sin restricciones definidas y dcha con las restricciones definidas por el usuario.

CASO 6

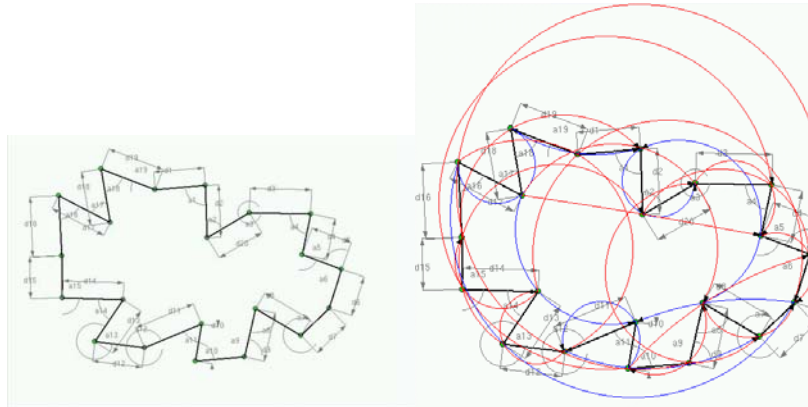


Figura 8.: Figuras correspondientes al caso 6: izqda sin restricciones definidas y dcha con las restricciones definidas por el usuario.

CASO 7

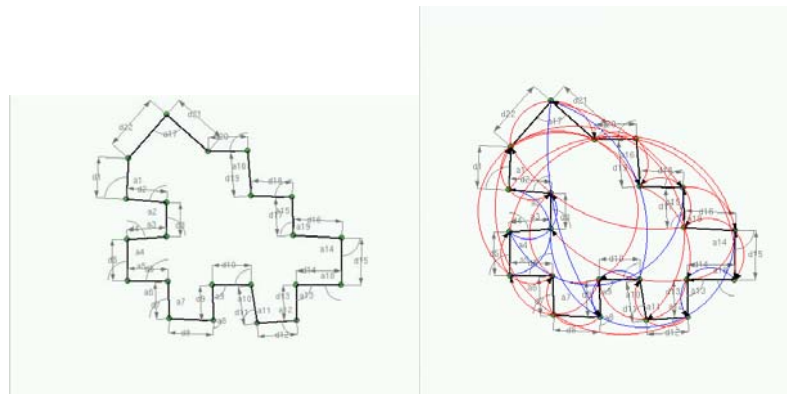


Figura 9.: Figuras correspondientes al caso 7: izqda sin restricciones definidas y dcha con las restricciones definidas por el usuario.

CASO 8

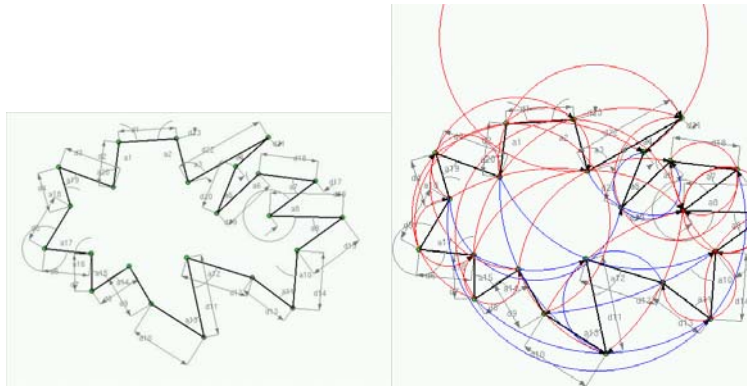


Figura 10.: Figuras correspondientes al caso 8: izqda sin restricciones definidas y dcha con las restricciones definidas por el usuario.

2.4. Parámetros utilizados

A continuación pasamos a detallar cada uno de los parámetros que hemos utilizado en cada uno de los algoritmos.

2.4.1. Parámetros comunes a todos los algoritmos

Los siguientes parámetros son comunes a todos y cada uno de los algoritmos implementados.

Parámetro	Valor
<i>Tiempo de ejecución (en minutos)</i>	5
<i>Semilla</i>	58, 294294, 10057

Tabla 2.: Parámetros comunes a todos los algoritmos implementados

2.4.2. Parámetros específicos para cada algoritmo

Métodos de Trayectoria

Búsqueda Local

Ascensión de Colinas

Hay suficiente con los parámetros comunes a todos los algoritmos.

Búsqueda Local Múltiple

Hay suficiente con los parámetros comunes a todos los algoritmos.

Enfriamiento Simulado

Enfriamiento Simulado Secuencial

Parámetro	Valor
<i>Nº de enfriamientos</i>	40
<i>Nº de vecinos generados por enfriamiento</i>	20

Tabla 3.: Parámetros específicos del algoritmo de Enfriamiento Simulado Secuencial

Enfriamiento Simulado Paralelo (parámetros comunes)

Parámetro	Valor
<i>Nº de ejecuciones</i>	5
<i>Nº de enfriamientos</i>	20
<i>Nº de vecinos generados por enfriamiento</i>	8

Tabla 4.: Parámetros específicos para todos los Algoritmos de Enfriamiento Simulado Paralelo.

Enfriamiento Simulado Paralelo (varias ejecuciones secuenciales)

No necesita definir ningún parámetro más.

Enfriamiento Simulado Paralelo (estrategia de división)

Parámetro	Valor
<i>Nº de enfriamientos a transcurrir para comunicación entre las ejecuciones</i>	4

Tabla 5.: Parámetros específicos del algoritmo de Enfriamiento Simulado Paralelo (estrategia de división).

Búsqueda Tabú

Parámetro	Valor
<i>Nº de vecinos generados por iteración</i>	10
<i>Nº de reinicializaciones en la búsqueda</i>	4

Tabla 6.: Parámetros específicos del algoritmo de Búsqueda Tabú.

Búsqueda Multiarranque

Búsqueda Local Iterativa

Búsqueda Local Iterativa (básica)

Parámetro	Valor
<i>Probabilidad de mutación</i>	0.3

Tabla 7.: Parámetros específicos del algoritmo ILS básico.

Búsqueda Local Iterativa (basada en Poblaciones)

Parámetro	Valor
<i>Nº de soluciones de la población</i>	5
<i>Probabilidad de mutación</i>	0.2
<i>Nº de iteraciones a transcurrir para el reemplazamiento del peor</i>	3

Tabla 8.: Parámetros específicos del algoritmo ILS basado en Poblaciones.

Búsqueda basada en Vecindarios Cambiantes

Parámetro	Valor
Probabilidad inicial de vecindario	0.1
Tamaño máximo de vecindario	5

Tabla 9.: Parámetros específicos del algoritmo VNS.

Métodos basados en Poblaciones

Algoritmos Genéticos

CHC

Parámetro	Valor
Nº individuos de la población	20
Nº individuos mejores para la reinicialización	3
Porcentaje de bits de variación para la reinicialización (tanto por 1)	0.35

Tabla 10.: Parámetros específicos del algoritmo CHC.

Resto de Algoritmos Genéticos (parámetros comunes)

Parámetro	Valor
Nº individuos de la población	21
Probabilidad de cruce	0.6
Probabilidad de mutación	0.2

Tabla 11.: Parámetros específicos para todos los Algoritmos Genéticos excepto CHC.

Algoritmo Genético básico (modelo generacional)

No es necesario definir más parámetros de los ya establecidos.

Algoritmos Genéticos Multimodales

Clearing

Parámetro	Valor
Radio del nicho	7
Nº de individuos por nicho	7

Tabla 12.: Parámetros específicos del algoritmo AG clearing.

Sharing

Parámetro	Valor
Radio del nicho	7
Regulador sharing	2.0

Tabla 13.: Parámetros específicos del algoritmo AG sharing.

Continuously updated fitness sharing

Parámetro	Valor
<i>Radio del nicho</i>	3
<i>Tamaño máximo del nicho</i>	3
<i>Regulador sharing</i>	2.0

Tabla 14.: Parámetros específicos del algoritmo AG Continuously Updated Sharing.

Nichos elitista con sharing

Parámetro	Valor
<i>Radio del nicho</i>	7
<i>Regulador sharing</i>	2.0

Tabla 15.: Parámetros específicos del algoritmo AG nichos elitista con sharing.

Multitud con multinichos (multiniche-crowding)

Parámetro	Valor
<i>Número de elementos del grupo usados en la selección (Cs)</i>	6
<i>Número de grupos para el reemplazo (Cf)</i>	3
<i>Número de elementos de cada uno de los Cf grupos (s)</i>	3

Tabla 16.: Parámetros específicos del algoritmo AG Multitud con Multinichos (multiniche-crowding).

PBIL

PBIL básico (parámetros comunes)

Parámetro	Valor
<i>Nº de muestras a generar por iteración</i>	15
<i>Probabilidad de mutación</i>	0.2
<i>Desplazamiento de mutación</i>	0.05
<i>Ratio de aprendizaje</i>	0.1

Tabla 17.: Parámetros específicos de todos los algoritmos PBIL básicos.

PBIL básico (actualización a partir del mejor)

No se necesitan definir más parámetros de los ya establecidos.

PBIL básico (actualización a partir del mejor y del peor)

Parámetro	Valor
<i>Ratio de aprendizaje negativo</i>	0.075

Tabla 18.: Parámetros específicos de algoritmo PBIL básico (actualización a partir del mejor y del peor).

PBIL básico (actualización a partir de los M mejores)

Parámetro	Valor
<i>M mejores</i>	3

Tabla 19.: Parámetros específicos de todos los algoritmos PBIL básico (actualización a partir de los M mejores).

CGA

CGA básico

Parámetro	Valor
<i>Nº total individuos población de referencia</i>	80

Tabla 20.: Parámetros específicos del algoritmo Compact-GA básico.

CGA extendido

Parámetro	Valor
<i>Nº total individuos población de referencia</i>	800
<i>Nº de muestras por iteración</i>	10

Tabla 21.: Parámetros específicos del algoritmo Compact-GA extendido.

Optimización a partir de Colonias de Hormigas

Parámetros comunes

Parámetro	Valor
<i>Nº de hormigas</i>	10
<i>Parámetro α regla transición</i>	1.0
<i>Parámetro β regla transición</i>	1.0
<i>Parámetro ρ regla actualización feromona</i>	0.5

Tabla 22.: Parámetros específicos de todos los algoritmos ACO.

Sistema de Hormigas y Sistema de Hormigas con Búsqueda Local

Parámetro	Valor
<i>Porcentaje de reinicialización de hormigas (tanto por uno)</i>	0.5

Tabla 23.: Parámetros específicos de los algoritmos del Sistema de Hormigas y el Sistema de Hormigas con Búsqueda Local.

Sistema de Hormigas de la Mejor y la Peor Hormiga

Parámetro	Valor
Porcentaje de reinicialización de hormigas (tanto por uno)	0.5
Probabilidad de mutación	0.4
Parámetro σ de desplazamiento de mutación	5.0

Tabla 24.: Parámetros específicos del algoritmo del Sistema de Hormigas de la Mejor y la Peor Hormiga.

Sistema de Hormigas basado en Rangos

Parámetro	Valor
Porcentaje de reinicialización de hormigas (tanto por uno)	0.5
M hormigas mejores a tener en cuenta para la actualización de feromona	3

Tabla 25.: Parámetros específicos del algoritmo del Sistema de Hormigas basado en Rangos.

Sistema de Hormigas MAX-MIN

Parámetro	Valor
Valor mínimo de feromona permitido	0.01
Valor máximo de feromona permitido	0.99
Longitud de la lista de soluciones elite	3

Tabla 26.: Parámetros específicos del Sistema de Hormigas MAX-MIN.

2.4.3. Número de experimentos

A continuación se indicará el número de experimentos llevados a cabo para cada uno de los algoritmos.

Tenemos, como sabemos, un total de 27 algoritmos. Cada uno de esos algoritmos se aplicará sobre los 8 casos del problema dispuestos para la experimentación. La aplicación de cada algoritmo a cada caso del problema supondrá la realización de tres ejecuciones, una para cada semilla de las establecidas en los parámetros. De este modo, para cada algoritmo se realizarán un total de 24 ejecuciones (8 casos y 3 ejecuciones por caso). Cada ejecución supondrá un tiempo de 5 minutos.

En total se han realizado 648 ejecuciones (54 horas de ejecución).

3. Resultados experimentales

Todos los experimentos han sido realizados sobre un equipo con procesador Pentium IV a 2.4 GHz con 256MB de memoria DDR RAM y bajo el sistema operativo Linux SuSe 8.1.

A continuación presentamos los resultados obtenidos por cada uno de los algoritmos correspondientes a cada una de las distintas Metaheurísticas utilizadas. Los resultados serán agrupados en base a la clasificación de Metaheurísticas ya conocida. Se proporcionarán las tablas de los resultados de las ejecuciones para cada algoritmo, tablas resumen para cada filosofía concreta y finalmente una tabla resumen final para todas ellas.

Debemos tener en cuenta las siguientes abreviaturas para simbolizar cada algoritmo...

- **Métodos de trayectoria**
 - Búsqueda Local.
 - **Básica (BL).**
 - **Múltiple (MLS).**
 - Enfriamiento Simulado.
 - **Enfriamiento Simulado Simple (SA).**
 - Enfriamiento Simulado Paralelo.
 - **Sólo múltiples ejecuciones (SAPI).**
 - **Estrategia de división (SAP2).**
 - **Búsqueda Tabú (BT).**
 - Búsqueda Multiarranque.
 - Búsqueda Local Iterativa.
 - **Simple (ILS).**
 - **Basada en Poblaciones (ILSP).**
 - **Búsqueda basada en Vecindarios Cambiantes (VNS).**
- **Métodos basados en Poblaciones**
 - Algoritmos Genéticos.
 - **Básico (AGb).**
 - **CHC (CHC).**
 - Multimodales.
 - **Sharing (Sharing).**
 - **Continuously updated sharing (CUShar).**
 - **Clearing (Clearing).**
 - **Nichos elitista con sharing (NEShar).**
 - **Multitud con multinichos (MCrowd).**
 - Metaheurísticas basadas en Modelos Probabilísticos.
 - Evolución de Poblaciones mediante Aprendizaje Incremental (PBIL).
 - Básico
 - **Actualización basada en el mejor (PBIL1).**
 - **Actualización basada en el mejor y el peor (PBIL2).**
 - Actualización basada en las M mejores soluciones.
 - **Actualización con las M (PBIL3).**
 - **Actualización a partir de solución de consenso (PBIL4).**
 - Compact-GA.
 - **Básico (CGA).**
 - **Extendido (CGA_E).**
 - Sistema de Hormigas.
 - **Básico (SH).**
 - **Con Búsqueda Local (SHBL).**
 - **Sistema de Hormigas basado en Rangos (SHRB).**
 - **Sistema de Hormigas MAX-MIN (SHMM).**
 - **Sistema de Hormigas de la Mejor y la Peor Hormiga (SHBW).**

3.1. Métodos de trayectoria

3.1.1. Búsqueda Local

Básica (Ascensión de Colinas)

BL	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	18	19	-1	25
<i>CASO 2 (n=142)</i>	21	22	22	24
<i>CASO 3 (n=149)</i>	23	21	20	27
<i>CASO 4 (n=156)</i>	21	23	17	28
<i>CASO 5 (n=135)</i>	20	22	23	26
<i>CASO 6 (n=143)</i>	21	23	20	29
<i>CASO 7 (n=149)</i>	23	22	25	30
<i>CASO 8 (n=156)</i>	28	-1	27	35

Tabla 27.: Resultados experimentales para el algoritmo de Ascensión de Colinas

BL	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	-1	12	19	25
<i>CASO 2 (n=142)</i>	21	21.66	22	24
<i>CASO 3 (n=149)</i>	20	21.33	23	27
<i>CASO 4 (n=156)</i>	17	20.33	23	28
<i>CASO 5 (n=135)</i>	20	21.66	23	26
<i>CASO 6 (n=143)</i>	20	21.33	23	29
<i>CASO 7 (n=149)</i>	22	23.33	25	30
<i>CASO 8 (n=156)</i>	-1	18	28	35

Tabla 28.: Resumen de los resultados experimentales para el algoritmo de Ascensión de Colinas.

Múltiple (MLS)

MLS	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	21	22	20	25
<i>CASO 2 (n=142)</i>	19	18	20	24
<i>CASO 3 (n=149)</i>	20	18	22	27
<i>CASO 4 (n=156)</i>	22	22	21	28
<i>CASO 5 (n=135)</i>	21	22	19	26
<i>CASO 6 (n=143)</i>	20	18	19	29
<i>CASO 7 (n=149)</i>	23	27	23	30
<i>CASO 8 (n=156)</i>	25	27	29	35

Tabla 29.: Resultados experimentales para el algoritmo de Búsqueda Local Múltiple.

MLS	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	20	21	22	25
<i>CASO 2 (n=142)</i>	18	19	20	24
<i>CASO 3 (n=149)</i>	18	20	22	27
<i>CASO 4 (n=156)</i>	21	21.66	22	28
<i>CASO 5 (n=135)</i>	19	20.66	22	26

CASO 6 (n=143)	18	19	20	29
CASO 7 (n=149)	23	24.33	27	30
CASO 8 (n=156)	25	27	29	35

Tabla 30.: Resumen de los resultados de los experimentos para el algoritmo de Búsqueda Local Múltiple.

Resumen Algoritmos de Búsqueda Local

BL	CASO 1 (25) N=135			CASO 2 (24) n=142			CASO 3 (27) n=149			CASO 4 (28) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
BL(LS)	-1	12	19	21	21.66	22	20	21.33	23	17	20.33	23
MLS	20	21	22	18	19	20	18	20	22	21	21.66	22
BL	CASO 5 (26) N=135			CASO 6 (29) n=143			CASO 7 (30) n=149			CASO 8 (35) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
BL(LS)	20	21.66	23	20	21.33	23	22	23.33	25	-1	18	28
MLS	19	20.66	22	18	19	20	23	24.33	27	25	27	29

Tabla 31.: Resultados de los experimentos para los algoritmos de Búsqueda Local.

Búsqueda Local (Básica y Múltiple)	Peor		Media	Mejor		Óptimo
CASO 1 (n=135)	-1	BL	16.5	22	MLS	25
CASO 2 (n=142)	18	MLS	20.33	22	BL	24
CASO 3 (n=149)	18	MLS	20.66	23	BL	27
CASO 4 (n=156)	17	BL	21	23	BL	28
CASO 5 (n=135)	19	MLS	21.16	23	BL	26
CASO 6 (n=143)	18	MLS	20.16	23	BL	29
CASO 7 (n=149)	22	BL	23.83	27	MLS	30
CASO 8 (n=156)	-1	BL	22.5	29	MLS	35

Tabla 32.: Resumen de los resultados de los experimentos para los algoritmos de Búsqueda Local.

3.1.2. Enfriamiento Simulado

Enfriamiento Simulado Secuencial (SA)

SA	S1	S2	S3	Óptimo
CASO 1 (n=135)	19	10	-1	25
CASO 2 (n=142)	14	15	19	24
CASO 3 (n=149)	16	18	9	27
CASO 4 (n=156)	20	14	11	28
CASO 5 (n=135)	14	13	15	26
CASO 6 (n=143)	17	18	16	29
CASO 7 (n=149)	22	-1	-1	30
CASO 8 (n=156)	25	21	16	35

Tabla 33.: Resultados de los experimentos para el algoritmo de Enfriamiento Simulado Secuencial.

SA	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
CASO 1 (n=135)	-1	9.33	19	25
CASO 2 (n=142)	14	16	19	24
CASO 3 (n=149)	9	14.33	18	27
CASO 4 (n=156)	11	15	20	28
CASO 5 (n=135)	13	14	15	26
CASO 6 (n=143)	16	17	18	29
CASO 7 (n=149)	-1	6.66	22	30
CASO 8 (n=156)	16	20.66	25	35

Tabla 34.: Resumen de los resultados de los experimentos para el algoritmo de Enfriamiento Simulado Secuencial.

Enfriamiento Simulado Paralelo (SAP)

Varias ejecuciones secuenciales (SAP1)

SAP1	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
CASO 1 (n=135)	17	19	18	25
CASO 2 (n=142)	17	15	19	24
CASO 3 (n=149)	15	15	14	27
CASO 4 (n=156)	20	18	18	28
CASO 5 (n=135)	14	19	12	26
CASO 6 (n=143)	19	20	20	29
CASO 7 (n=149)	23	17	16	30
CASO 8 (n=156)	25	20	17	35

Tabla 35.: Resultados de los experimentos para el algoritmo de Enfriamiento Simulado Paralelo (varias ejecuciones secuenciales).

SAP1	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
CASO 1 (n=135)	17	18	19	25
CASO 2 (n=142)	15	17	19	24
CASO 3 (n=149)	14	14.66	15	27
CASO 4 (n=156)	18	18.66	20	28
CASO 5 (n=135)	12	15	19	26
CASO 6 (n=143)	19	19.66	20	29
CASO 7 (n=149)	16	18.66	23	30
CASO 8 (n=156)	17	20.66	25	35

Tabla 36.: Resumen de los resultados de los experimentos para el algoritmo de Enfriamiento Simulado Paralelo (varias ejecuciones secuenciales).

Enfriamiento Simulado Paralelo (SAP)

Estrategia de División (SAP2)

SAP2	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
CASO 1 (n=135)	14	17	18	25
CASO 2 (n=142)	16	16	21	24
CASO 3 (n=149)	20	19	21	27
CASO 4 (n=156)	16	17	17	28
CASO 5 (n=135)	20	14	19	26

<i>CASO 6 (n=143)</i>	19	19	19	29
<i>CASO 7 (n=149)</i>	23	25	20	30
<i>CASO 8 (n=156)</i>	23	22	23	35

Tabla 37.: Resultados de los experimentos para el algoritmo de Enfriamiento Simulado Paralelo (estrategia de división).

SAP2	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	14	16.33	18	25
<i>CASO 2 (n=142)</i>	16	17.66	21	24
<i>CASO 3 (n=149)</i>	19	20	21	27
<i>CASO 4 (n=156)</i>	16	16.66	17	28
<i>CASO 5 (n=135)</i>	14	17.66	20	26
<i>CASO 6 (n=143)</i>	19	19	19	29
<i>CASO 7 (n=149)</i>	20	22.66	25	30
<i>CASO 8 (n=156)</i>	22	22.66	23	35

Tabla 38.: Resumen de los resultados de los experimentos para el algoritmo de Enfriamiento Simulado Paralelo (estrategia de división).

Resumen Algoritmos de Enfriamiento Simulado

SA	CASO 1 (25) <i>N=135</i>			CASO 2 (24) <i>n=142</i>			CASO 3 (27) <i>n=149</i>			CASO 4 (28) <i>n=156</i>		
	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>
SA	-1	9.33	19	14	16	19	9	14.33	18	11	15	20
SAP1	17	18	19	15	17	19	14	14.66	15	18	18.66	20
SAP2	14	16.33	18	16	17.66	21	19	20	21	16	16.66	17
SA	CASO 5 (26) <i>N=135</i>			CASO 6 (29) <i>n=143</i>			CASO 7 (30) <i>n=149</i>			CASO 8 (35) <i>n=156</i>		
	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>
SA	13	14	15	16	17	18	-1	6.66	22	16	20.66	25
SAP1	12	15	19	19	19.66	20	16	18.66	23	17	20.66	25
SAP2	14	17.66	20	19	19	19	20	22.66	25	22	22.66	23

Tabla 39.: Resultados de los experimentos para los algoritmos de Enfriamiento Simulado.

Enfriamiento Simulado	<i>Peor</i>		<i>Media</i>	<i>Mejor</i>		<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	-1	SA	14.55	19	SA, SAP1	25
<i>CASO 2 (n=142)</i>	14	SA	16.88	21	SAP2	24
<i>CASO 3 (n=149)</i>	9	SA	16.33	21	SAP2	27
<i>CASO 4 (n=156)</i>	11	SA	16.77	20	SA, SAP1	28
<i>CASO 5 (n=135)</i>	12	SAP1	15.55	20	SAP2	26
<i>CASO 6 (n=143)</i>	16	SA	18.55	20	SAP1	29
<i>CASO 7 (n=149)</i>	-1	SA	16	25	SAP2	30
<i>CASO 8 (n=156)</i>	16	SA	21.33	25	SA, SAP1	35

Tabla 40.: Resumen de los resultados de los experimentos para los algoritmos de Enfriamiento Simulado.

3.1.3. Búsqueda Tabú

BT	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	23	20	18	25
<i>CASO 2 (n=142)</i>	21	21	21	24
<i>CASO 3 (n=149)</i>	20	21	21	27
<i>CASO 4 (n=156)</i>	21	19	22	28
<i>CASO 5 (n=135)</i>	20	20	19	26
<i>CASO 6 (n=143)</i>	21	20	20	29
<i>CASO 7 (n=149)</i>	20	21	25	30
<i>CASO 8 (n=156)</i>	30	24	25	35

Tabla 41.: Resultados de los experimentos para el algoritmo de Búsqueda Tabú.

BT	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	18	20.33	23	25
<i>CASO 2 (n=142)</i>	21	21	21	24
<i>CASO 3 (n=149)</i>	20	20.66	21	27
<i>CASO 4 (n=156)</i>	19	20.66	22	28
<i>CASO 5 (n=135)</i>	19	19.66	20	26
<i>CASO 6 (n=143)</i>	20	20.33	21	29
<i>CASO 7 (n=149)</i>	20	22	25	30
<i>CASO 8 (n=156)</i>	24	26.33	30	35

Tabla 42.: Resumen de los resultados de los experimentos para el algoritmo de Búsqueda Tabú.

3.1.4. Búsqueda Multiarranque

Búsqueda Local Iterativa

Básica (ILS)

ILS	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	19	21	19	25
<i>CASO 2 (n=142)</i>	19	19	19	24
<i>CASO 3 (n=149)</i>	21	19	21	27
<i>CASO 4 (n=156)</i>	24	20	20	28
<i>CASO 5 (n=135)</i>	19	19	19	26
<i>CASO 6 (n=143)</i>	21	19	21	29
<i>CASO 7 (n=149)</i>	22	23	24	30
<i>CASO 8 (n=156)</i>	26	28	26	35

Tabla 43.: Resultados de los experimentos para el algoritmo de Búsqueda Local Iterativa.

ILS	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	19	19.66	21	25
<i>CASO 2 (n=142)</i>	19	19	19	24
<i>CASO 3 (n=149)</i>	19	20.33	21	27
<i>CASO 4 (n=156)</i>	20	21.33	24	28
<i>CASO 5 (n=135)</i>	19	19	19	26
<i>CASO 6 (n=143)</i>	19	20.33	21	29

<i>CASO 7 (n=149)</i>	22	23	24	30
<i>CASO 8 (n=156)</i>	26	26.66	28	35

Tabla 44.: Resumen de los resultados de los experimentos para el algoritmo de Búsqueda Local Iterativa.

Búsqueda Local Iterativa

Basada en Poblaciones

ILSP	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	24	20	24	25
<i>CASO 2 (n=142)</i>	20	20	22	24
<i>CASO 3 (n=149)</i>	22	23	23	27
<i>CASO 4 (n=156)</i>	22	21	19	28
<i>CASO 5 (n=135)</i>	21	21	23	26
<i>CASO 6 (n=143)</i>	20	25	21	29
<i>CASO 7 (n=149)</i>	25	23	26	30
<i>CASO 8 (n=156)</i>	32	28	26	35

Tabla 45.: Resultados de los experimentos para el algoritmo de Búsqueda Local Iterativa basada en Poblaciones.

ILSP	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	20	22.66	24	25
<i>CASO 2 (n=142)</i>	20	20.66	22	24
<i>CASO 3 (n=149)</i>	22	22.66	23	27
<i>CASO 4 (n=156)</i>	19	20.66	22	28
<i>CASO 5 (n=135)</i>	21	21.66	23	26
<i>CASO 6 (n=143)</i>	20	22	25	29
<i>CASO 7 (n=149)</i>	23	24.66	26	30
<i>CASO 8 (n=156)</i>	26	28.66	32	35

Tabla 46.: Resumen de los resultados de los experimentos para el algoritmo de Búsqueda Local Iterativa basada en Poblaciones.

Búsqueda basada en Vecindarios Cambiantes (VNS)

VNS	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	21	20	21	25
<i>CASO 2 (n=142)</i>	21	22	21	24
<i>CASO 3 (n=149)</i>	24	21	24	27
<i>CASO 4 (n=156)</i>	21	23	23	28
<i>CASO 5 (n=135)</i>	19	26	20	26
<i>CASO 6 (n=143)</i>	26	20	22	29
<i>CASO 7 (n=149)</i>	26	27	22	30
<i>CASO 8 (n=156)</i>	27	29	24	35

Tabla 47.: Resultados de los experimentos para el algoritmo de Búsqueda basada en Vecindarios Cambiantes.

VNS	Peor	Media	Mejor	Óptimo
CASO 1 (n=135)	20	20.66	21	25
CASO 2 (n=142)	21	21.66	22	24
CASO 3 (n=149)	21	23	24	27
CASO 4 (n=156)	21	22.33	23	28
CASO 5 (n=135)	19	21.66	26	26
CASO 6 (n=143)	20	22.66	26	29
CASO 7 (n=149)	22	25	27	30
CASO 8 (n=156)	24	26.66	29	35

Tabla 48.: Resumen de los resultados de los experimentos para el algoritmo de Búsqueda basada en Vecindarios Cambiantes.

Resumen Algoritmos de Búsqueda Multiarranque

BM	CASO 1 (25) N=135			CASO 2 (24) n=142			CASO 3 (27) n=149			CASO 4 (28) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
ILS	19	19.66	21	19	19	19	19	20.33	21	20	21.33	24
ILSP	20	22.66	24	20	20.66	22	22	22.66	23	19	20.66	22
VNS	20	20.66	21	21	21.66	22	21	23	24	21	22.33	23
BM	CASO 5 (26) N=135			CASO 6 (29) n=143			CASO 7 (30) n=149			CASO 8 (35) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
ILS	19	19	19	19	20.33	21	22	23	24	26	26.66	28
ILSP	21	21.66	23	20	22	25	23	24.66	26	26	28.66	32
VNS	19	21.66	26	20	22.66	26	22	25	27	24	26.66	29

Tabla 49.: Resultados de los experimentos para los algoritmos de Búsqueda Multiarranque.

Búsqueda Multiarranque	Peor		Media	Mejor		Óptimo
CASO 1 (n=135)	19	ILS	21	24	ILSP	25
CASO 2 (n=142)	19	ILS	20.45	22	ILSP, VNS	24
CASO 3 (n=149)	19	ILS	22	24	VNS	27
CASO 4 (n=156)	19	ILSP	21.45	24	ILS	28
CASO 5 (n=135)	19	ILS, VNS	20.77	26	VNS	26
CASO 6 (n=143)	19	ILS	21.66	26	VNS	29
CASO 7 (n=149)	22	ILS, VNS	24.22	27	VNS	30
CASO 8 (n=156)	24	VNS	27.33	32	ILSP	35

Tabla 50.: Resumen de los resultados de los experimentos para los algoritmos de Búsqueda Multiarranque.

3.1.5. Comparación de Métodos de trayectoria

MT	CASO 1 (25) N=135			CASO 2 (24) n=142			CASO 3 (27) n=149			CASO 4 (28) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
BL	-1	16.5	22	18	20.33	22	18	20.66	23	17	21	23
BL(LS)	-1	12	19	21	21.66	22	20	21.33	23	17	20.33	23

<i>MLS</i>	20	21	22	18	19	20	18	20	22	21	21.66	22
<i>SA</i>	-1	14.55	19	14	16.88	21	9	16.33	21	11	16.77	20
<i>SA</i>	-1	9.33	19	14	16	19	9	14.33	18	11	15	20
<i>SAPI</i>	17	18	19	15	17	19	14	14.66	15	18	18.66	20
<i>SAP2</i>	14	16.33	18	16	17.66	21	19	20	21	16	16.66	17
<i>BT</i>	18	20.33	23	21	21	21	20	20.66	21	19	20.66	22
<i>BM</i>	19	21	24	19	20.45	22	19	22	24	19	21.45	24
<i>ILS</i>	19	19.66	21	19	19	19	19	20.33	21	20	21.33	24
<i>ILSP</i>	20	22.66	24	20	20.66	22	22	22.66	23	19	20.66	22
<i>VNS</i>	20	20.66	21	21	21.66	22	21	23	24	21	22.33	23
MT	CASO 5 (26) <i>N=135</i>			CASO 6 (29) <i>n=143</i>			CASO 7 (30) <i>n=149</i>			CASO 8 (35) <i>n=156</i>		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
<i>BL</i>	19	21.16	23	18	20.16	23	22	23.83	27	-1	22.5	29
<i>BL(LS)</i>	20	21.66	23	20	21.33	23	22	23.33	25	-1	18	28
<i>MLS</i>	19	20.66	22	18	19	20	23	24.33	27	25	27	29
<i>SA</i>	12	15.55	20	16	18.55	20	-1	16	25	16	21.33	25
<i>SA</i>	13	14	15	16	17	18	-1	6.66	22	16	20.66	25
<i>SAPI</i>	12	15	19	19	19.66	20	16	18.66	23	17	20.66	25
<i>SAP2</i>	14	17.66	20	19	19	19	20	22.66	25	22	22.66	23
<i>BT</i>	19	19.66	20	20	20.33	21	20	22	25	24	26.33	30
<i>BM</i>	19	20.77	26	19	21.66	26	22	24.22	27	24	27.33	32
<i>ILS</i>	19	19	19	19	20.33	21	22	23	24	26	26.66	28
<i>ILSP</i>	21	21.66	23	20	22	25	23	24.66	26	26	28.66	32
<i>VNS</i>	19	21.66	26	20	22.66	26	22	25	27	24	26.66	29

Tabla 51.: Resultados de los experimentos para los algoritmos correspondientes a los Métodos de Trayectoria.

Métodos Trayectoria	<i>Peor</i>		<i>Mejor</i>		<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	-1	BL, SA	24	ILSP	25
<i>CASO 2 (n=142)</i>	14	SA	22	BL, ILSP, VNS	24
<i>CASO 3 (n=149)</i>	9	SA	24	VNS	27
<i>CASO 4 (n=156)</i>	11	SA	24	ILS	28
<i>CASO 5 (n=135)</i>	12	SAPI	26	VNS	26
<i>CASO 6 (n=143)</i>	16	SA	26	VNS	29
<i>CASO 7 (n=149)</i>	-1	SA	27	VNS, MLS	30
<i>CASO 8 (n=156)</i>	-1	BL	32	ILSP	35

Tabla 52.: Resumen de los resultados de los experimentos para los algoritmos correspondientes a los Métodos de Trayectoria (a).

Métodos Trayectoria	<i>Peor Media</i>		<i>Mejor Media</i>		<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	9.33	SA	22.66	ILSP	25
<i>CASO 2 (n=142)</i>	16	SA	22.66	BL, VNS	24
<i>CASO 3 (n=149)</i>	14.33	SA	23	VNS	27
<i>CASO 4 (n=156)</i>	15	SA	22.33	VNS	28

CASO 5 (n=135)	14	SA	21.66	BL, ILSP, VNS	26
CASO 6 (n=143)	17	SA	22.66	VNS	29
CASO 7 (n=149)	6.66	SA	25	VNS	30
CASO 8 (n=156)	18	BL	28.66	ILSP	35

Tabla 53.: Resumen de los resultados de los experimentos para los algoritmos correspondientes a los Métodos de Trayectoria (b).

3.2. Métodos basados en población

3.2.1. Algoritmos Genéticos

AG Básico (modelo generacional)

AGb	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
CASO 1 (n=135)	23	21	21	25
CASO 2 (n=142)	23	20	21	24
CASO 3 (n=149)	27	22	26	27
CASO 4 (n=156)	23	25	23	28
CASO 5 (n=135)	22	22	20	26
CASO 6 (n=143)	24	23	22	29
CASO 7 (n=149)	25	27	24	30
CASO 8 (n=156)	31	30	27	35

Tabla 54.: Resultados de los experimentos para el algoritmo Genético Básico (modelo generacional).

AGb	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
CASO 1 (n=135)	21	21.66	23	25
CASO 2 (n=142)	20	21.33	23	24
CASO 3 (n=149)	22	25	27	27
CASO 4 (n=156)	23	23.66	25	28
CASO 5 (n=135)	20	21.33	22	26
CASO 6 (n=143)	22	23	24	29
CASO 7 (n=149)	24	25.33	27	30
CASO 8 (n=156)	27	29.33	31	35

Tabla 55.: Resumen de los resultados de los experimentos para el algoritmo Genético Básico (modelo generacional).

CHC

CHC	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
CASO 1 (n=135)	21	20	23	25
CASO 2 (n=142)	19	23	21	24
CASO 3 (n=149)	25	25	22	27
CASO 4 (n=156)	25	22	22	28
CASO 5 (n=135)	22	21	22	26
CASO 6 (n=143)	25	23	23	29
CASO 7 (n=149)	30	30	29	30
CASO 8 (n=156)	31	29	32	35

Tabla 56.: Resultados de los experimentos para el algoritmo CHC.

CHC	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
CASO 1 (n=135)	20	21.33	23	25
CASO 2 (n=142)	19	21	23	24
CASO 3 (n=149)	22	24	25	27
CASO 4 (n=156)	22	23	25	28
CASO 5 (n=135)	21	21.66	22	26
CASO 6 (n=143)	23	23.66	25	29
CASO 7 (n=149)	29	29.66	30	30
CASO 8 (n=156)	29	30.66	32	35

Tabla 57.: Resumen de los resultados de los experimentos para el algoritmo CHC.

AG Multimodales (AG clearing)

Clearing	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
CASO 1 (n=135)	20	22	21	25
CASO 2 (n=142)	22	23	22	24
CASO 3 (n=149)	25	23	25	27
CASO 4 (n=156)	25	24	25	28
CASO 5 (n=135)	22	23	20	26
CASO 6 (n=143)	22	24	26	29
CASO 7 (n=149)	27	29	26	30
CASO 8 (n=156)	32	31	30	35

Tabla 58.: Resultados de los experimentos para el algoritmo AG clearing.

Clearing	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
CASO 1 (n=135)	20	21	22	25
CASO 2 (n=142)	22	22.33	23	24
CASO 3 (n=149)	23	24.33	25	27
CASO 4 (n=156)	24	24.66	25	28
CASO 5 (n=135)	20	21.66	23	26
CASO 6 (n=143)	22	24	26	29
CASO 7 (n=149)	26	27.33	29	30
CASO 8 (n=156)	30	31	32	35

Tabla 59.: Resumen de los resultados de los experimentos para el algoritmo AG clearing.

AG Multimodales (AG sharing)

Sharing	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
CASO 1 (n=135)	22	21	21	25
CASO 2 (n=142)	23	21	22	24
CASO 3 (n=149)	24	22	24	27
CASO 4 (n=156)	23	25	25	28
CASO 5 (n=135)	22	23	22	26
CASO 6 (n=143)	24	25	24	29
CASO 7 (n=149)	27	26	23	30
CASO 8 (n=156)	29	31	28	35

Tabla 60.: Resultados de los experimentos para el algoritmo AG sharing.

Sharing	Peor	Media	Mejor	Óptimo
CASO 1 (n=135)	21	21.33	22	25
CASO 2 (n=142)	21	22	23	24
CASO 3 (n=149)	22	23.33	24	27
CASO 4 (n=156)	23	24.33	25	28
CASO 5 (n=135)	22	22.33	23	26
CASO 6 (n=143)	24	24.33	25	29
CASO 7 (n=149)	23	25.33	27	30
CASO 8 (n=156)	28	29.33	31	35

Tabla 61.: Resumen de los resultados experimentales para el AG sharing.

AG Multimodales (AG continuously updated sharing)

CUSharing	S1	S2	S3	Óptimo
CASO 1 (n=135)	20	19	21	25
CASO 2 (n=142)	21	20	17	24
CASO 3 (n=149)	22	21	24	27
CASO 4 (n=156)	21	22	21	28
CASO 5 (n=135)	21	19	22	26
CASO 6 (n=143)	22	20	21	29
CASO 7 (n=149)	23	23	23	30
CASO 8 (n=156)	26	30	28	35

Tabla 62.: Resultados de los experimentos para el algoritmo AG continuously updated sharing.

CUSharing	Peor	Media	Mejor	Óptimo
CASO 1 (n=135)	19	20	21	25
CASO 2 (n=142)	17	19.33	21	24
CASO 3 (n=149)	21	22.33	24	27
CASO 4 (n=156)	21	21.33	22	28
CASO 5 (n=135)	19	20.66	22	26
CASO 6 (n=143)	20	21	22	29
CASO 7 (n=149)	23	23	23	30
CASO 8 (n=156)	26	28	30	35

Tabla 63.: Resumen de los resultados de los experimentos para el algoritmo AG continuously updated sharing.

AG Multimodales (AG nichos elitista con sharing)

NESharing	S1	S2	S3	Óptimo
CASO 1 (n=135)	22	25	24	25
CASO 2 (n=142)	23	22	22	24
CASO 3 (n=149)	24	24	24	27
CASO 4 (n=156)	25	25	21	28
CASO 5 (n=135)	20	23	21	26
CASO 6 (n=143)	23	25	23	29
CASO 7 (n=149)	28	29	26	30
CASO 8 (n=156)	29	32	31	35

Tabla 64.: Resultados de los experimentos para el algoritmo AG nichos elitista con sharing.

NESharing	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
CASO 1 (n=135)	22	23.66	25	25
CASO 2 (n=142)	22	22.33	23	24
CASO 3 (n=149)	24	24	24	27
CASO 4 (n=156)	21	23.66	25	28
CASO 5 (n=135)	20	21.33	23	26
CASO 6 (n=143)	23	23.66	25	29
CASO 7 (n=149)	26	27.66	29	30
CASO 8 (n=156)	29	30.66	32	35

Tabla 65.: Resumen de los resultados de los experimentos para el algoritmo AG nichos elitista con sharing.

AG Multimodales (AG multiniche-crowding)

MCrowding	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
CASO 1 (n=135)	22	25	22	25
CASO 2 (n=142)	22	21	23	24
CASO 3 (n=149)	23	23	24	27
CASO 4 (n=156)	24	25	24	28
CASO 5 (n=135)	22	23	21	26
CASO 6 (n=143)	24	23	22	29
CASO 7 (n=149)	26	27	28	30
CASO 8 (n=156)	31	30	31	35

Tabla 66.: Resultados de los experimentos para el algoritmo AG multitud con multinichos (multiniche-crowding).

MCrowding	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
CASO 1 (n=135)	22	23	25	25
CASO 2 (n=142)	21	22	23	24
CASO 3 (n=149)	23	23.33	24	27
CASO 4 (n=156)	24	24.33	25	28
CASO 5 (n=135)	21	22	23	26
CASO 6 (n=143)	22	23	24	29
CASO 7 (n=149)	26	27	28	30
CASO 8 (n=156)	30	30.66	31	35

Tabla 67.: Resumen de los resultados de los experimentos para el algoritmo AG multitud con multinichos (multiniche-crowding).

Resumen Algoritmos Genéticos

AG	CASO 1 (25) N=135			CASO 2 (24) n=142			CASO 3 (27) n=149			CASO 4 (28) n=156		
	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>
AGb	21	21.66	23	20	21.33	23	22	25	27	23	23.66	25
CHC	20	21.33	23	19	21	23	22	24	25	22	23	25
Clearing	20	21	22	22	22.33	23	23	24.33	25	24	24.66	25
Sharing	21	21.33	22	21	22	23	22	23.33	24	23	24.33	25
CUShar	19	20	21	17	19.33	21	21	22.33	24	21	21.33	22
NEShar	22	23.66	25	22	22.33	23	24	24	24	21	23.66	25
MCrowd	22	23	25	21	22	23	23	23.33	24	24	24.33	25

AG	CASO 5 (26) N=135			CASO 6 (29) n=143			CASO 7 (30) n=149			CASO 8 (35) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
AGb	20	21.33	22	22	23	24	24	25.33	27	27	29.33	31
CHC	21	21.66	22	23	23.66	25	29	29.66	30	29	30.66	32
Clearing	20	21.66	23	22	24	26	26	27.33	29	30	31	32
Sharing	22	22.33	23	24	24.33	25	23	25.33	27	28	29.33	31
CUShar	19	20.66	22	20	21	22	23	23	23	26	28	30
NEShar	20	21.33	23	23	23.66	25	26	27.66	29	29	30.66	32
MCrowd	21	22	23	22	23	24	26	27	28	30	30.66	31

Tabla 68.: Resultados de los experimentos para los Algoritmos Genéticos.

AG	Peor		Media	Mejor		Óptimo
CASO 1 (n=135)	19	CUShar	21.71	25	NEShar, Mcrowd	25
CASO 2 (n=142)	17	CUShar	21.47	23	Todos excepto CUShar	24
CASO 3 (n=149)	21	CUShar	23.76	27	AGb	27
CASO 4 (n=156)	21	CUShar, NEShar	23.56	25	Todos excepto CUShar	28
CASO 5 (n=135)	19	CUShar	21.56	23	Clearing Sharing NEShar Mcrowd	26
CASO 6 (n=143)	20	CUShar	23.23	26	Clearing	29
CASO 7 (n=149)	23	Sharing, CUShar	26.47	30	CHC	30
CASO 8 (n=156)	26	CUShar	29.95	32	CHC, Clearing NEShar	35

Tabla 69.: Resumen de los resultados de los experimentos para los Algoritmos Genéticos.

3.2.2. Evolución de poblaciones mediante Aprendizaje Incremental: PBIL.

PBIL Básico (Actualización a partir del mejor)

PBIL1	S1	S2	S3	Óptimo
CASO 1 (n=135)	22	24	25	25
CASO 2 (n=142)	20	19	21	24
CASO 3 (n=149)	24	24	23	27
CASO 4 (n=156)	23	22	24	28
CASO 5 (n=135)	21	22	22	26
CASO 6 (n=143)	25	24	25	29
CASO 7 (n=149)	25	25	24	30
CASO 8 (n=156)	30	29	31	35

Tabla 70.: Resultados de los experimentos para el algoritmo del PBIL Básico.

PBIL1	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	22	23.66	25	25
<i>CASO 2 (n=142)</i>	19	20	21	24
<i>CASO 3 (n=149)</i>	23	23.66	24	27
<i>CASO 4 (n=156)</i>	22	23	24	28
<i>CASO 5 (n=135)</i>	21	21.66	22	26
<i>CASO 6 (n=143)</i>	24	24.66	25	29
<i>CASO 7 (n=149)</i>	24	24.66	25	30
<i>CASO 8 (n=156)</i>	29	30	31	35

Tabla 71.: Resumen de los resultados experimentales para PBIL Básico.

PBIL Básico (Actualización a partir del mejor y del peor)

PBIL2	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	25	21	25	25
<i>CASO 2 (n=142)</i>	22	20	21	24
<i>CASO 3 (n=149)</i>	25	25	23	27
<i>CASO 4 (n=156)</i>	23	23	23	28
<i>CASO 5 (n=135)</i>	25	23	22	26
<i>CASO 6 (n=143)</i>	25	25	25	29
<i>CASO 7 (n=149)</i>	27	28	26	30
<i>CASO 8 (n=156)</i>	32	28	33	35

Tabla 72.: Resultados de los experimentos para el algoritmo del PBIL Básico.

PBIL2	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	21	23.66	25	25
<i>CASO 2 (n=142)</i>	20	21	22	24
<i>CASO 3 (n=149)</i>	23	24.33	25	27
<i>CASO 4 (n=156)</i>	23	23	23	28
<i>CASO 5 (n=135)</i>	22	23.33	25	26
<i>CASO 6 (n=143)</i>	25	25	25	29
<i>CASO 7 (n=149)</i>	26	27	28	30
<i>CASO 8 (n=156)</i>	28	31	33	35

Tabla 73.: Resumen de los resultados de los experimentos para el algoritmo PBIL Básico.

PBIL Básico (Actualización con las M mejores soluciones de manera equitativa)

PBIL3	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	23	23	23	25
<i>CASO 2 (n=142)</i>	22	22	22	24
<i>CASO 3 (n=149)</i>	26	27	23	27
<i>CASO 4 (n=156)</i>	24	21	22	28
<i>CASO 5 (n=135)</i>	23	22	22	26
<i>CASO 6 (n=143)</i>	25	25	25	29
<i>CASO 7 (n=149)</i>	28	27	24	30
<i>CASO 8 (n=156)</i>	32	33	27	35

Tabla 74.: Resultados de los experimentos para el algoritmo del PBIL Básico.

PBIL3	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	23	23	23	25
<i>CASO 2 (n=142)</i>	22	22	22	24
<i>CASO 3 (n=149)</i>	23	25.33	27	27
<i>CASO 4 (n=156)</i>	21	22.33	24	28
<i>CASO 5 (n=135)</i>	22	22.33	23	26
<i>CASO 6 (n=143)</i>	25	25	25	29
<i>CASO 7 (n=149)</i>	24	26.33	28	30
<i>CASO 8 (n=156)</i>	27	30.66	33	35

Tabla 75.: Resumen de los resultados de los experimentos para el algoritmo PBIL Básico

PBIL Básico (Actualización con las M mejores soluciones a partir de solución de consenso)

PBIL4	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	20	24	21	25
<i>CASO 2 (n=142)</i>	19	24	22	24
<i>CASO 3 (n=149)</i>	23	25	22	27
<i>CASO 4 (n=156)</i>	24	22	24	28
<i>CASO 5 (n=135)</i>	22	22	22	26
<i>CASO 6 (n=143)</i>	24	24	25	29
<i>CASO 7 (n=149)</i>	25	27	26	30
<i>CASO 8 (n=156)</i>	30	30	26	35

Tabla 76.: Resultados de los experimentos para el algoritmo del PBIL Básico.

PBIL4	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	20	21.66	24	25
<i>CASO 2 (n=142)</i>	19	21.66	24	24
<i>CASO 3 (n=149)</i>	22	23.33	25	27
<i>CASO 4 (n=156)</i>	22	23.33	24	28
<i>CASO 5 (n=135)</i>	22	22	22	26
<i>CASO 6 (n=143)</i>	24	24.33	25	29
<i>CASO 7 (n=149)</i>	25	26	27	30
<i>CASO 8 (n=156)</i>	26	28.66	30	35

Tabla 77.: Resumen de los resultados experimentales para PBIL Básico.

Compact-GA básico

CGA	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	21	22	20	25
<i>CASO 2 (n=142)</i>	20	20	20	24
<i>CASO 3 (n=149)</i>	23	23	23	27
<i>CASO 4 (n=156)</i>	22	21	21	28
<i>CASO 5 (n=135)</i>	21	23	21	26
<i>CASO 6 (n=143)</i>	22	21	23	29
<i>CASO 7 (n=149)</i>	23	24	24	30
<i>CASO 8 (n=156)</i>	29	27	29	35

Tabla 78.: Resultados de los experimentos para el algoritmo Compact-GA básico.

CGA	Peor	Media	Mejor	Óptimo
CASO 1 (n=135)	20	21	22	25
CASO 2 (n=142)	20	20	20	24
CASO 3 (n=149)	23	23	23	27
CASO 4 (n=156)	21	21.33	22	28
CASO 5 (n=135)	21	21.66	23	26
CASO 6 (n=143)	21	22	23	29
CASO 7 (n=149)	23	23.66	24	30
CASO 8 (n=156)	27	28.33	29	35

Tabla 79.: Resumen de los resultados de los experimentos para el algoritmo Compact-GA básico.

Compact-GA Extendido

CGA_E	S1	S2	S3	Óptimo
CASO 1 (n=135)	22	22	22	25
CASO 2 (n=142)	20	19	22	24
CASO 3 (n=149)	22	22	23	27
CASO 4 (n=156)	21	22	22	28
CASO 5 (n=135)	20	21	21	26
CASO 6 (n=143)	21	21	21	29
CASO 7 (n=149)	28	25	26	30
CASO 8 (n=156)	27	27	27	35

Tabla 80.: Resultados de los experimentos para el algoritmo Compact-GA extendido.

CGA_E	Peor	Media	Mejor	Óptimo
CASO 1 (n=135)	22	22	22	25
CASO 2 (n=142)	19	20.33	22	24
CASO 3 (n=149)	22	22.33	23	27
CASO 4 (n=156)	21	21.66	22	28
CASO 5 (n=135)	20	20.66	21	26
CASO 6 (n=143)	21	21	21	29
CASO 7 (n=149)	25	26.33	28	30
CASO 8 (n=156)	27	27	27	35

Tabla 81.: Resumen de los resultados de los experimentos para el algoritmo Compact-GA extendido.

Resumen Algoritmos de Evolución de Poblaciones mediante Aprendizaje Incremental

PBIL	CASO 1 (25) N=135			CASO 2 (24) n=142			CASO 3 (27) n=149			CASO 4 (28) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
PBIL1	22	23.66	25	19	20	21	23	23.66	24	22	23	24
PBIL2	21	23.66	25	20	21	22	23	24.33	25	23	23	23
PBIL3	23	23	23	22	22	22	23	25.33	27	21	22.33	24
PBIL4	20	21.66	24	19	21.66	24	22	23.33	25	22	23.33	24
CGA	20	21	22	20	20	20	23	23	23	21	21.33	22
CGA_E	22	22	22	19	20.33	22	22	22.33	23	21	21.66	22

PBIL	CASO 5 (26) N=135			CASO 6 (29) n=143			CASO 7 (30) n=149			CASO 8 (35) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
PBIL1	21	21.66	22	24	24.66	25	24	24.66	25	29	30	31
PBIL2	22	23.33	25	25	25	25	26	27	28	28	31	33
PBIL3	22	22.33	23	25	25	25	24	26.33	28	27	30.66	33
PBIL4	22	22	22	24	24.33	25	25	26	27	26	28.66	30
CGA	21	21.66	23	21	22	23	23	23.66	24	27	28.33	29
CGA_E	20	20.66	21	21	21	21	25	26.33	28	27	27	27

Tabla 82.: Resultados de los experimentos para los algoritmos de Evolución de Poblaciones mediante Aprendizaje Incremental (PBIL).

PBIL	Peor	Media	Mejor	Óptimo		
CASO 1 (n=135)	20	PBIL4, CGA	22.5	25	PBIL1, PBIL2	25
CASO 2 (n=142)	19	PBIL1, PBIL4, CGA_E	20.83	24	PBIL4	24
CASO 3 (n=149)	22	PBIL4, CGA_E	23.66	27	PBIL3	27
CASO 4 (n=156)	21	PBIL3, CGA, CGA_E	22.44	24	PBIL1, PBIL3, PBIL4	28
CASO 5 (n=135)	20	CGA_E	21.94	25	PBIL2	26
CASO 6 (n=143)	21	CGA, CGA_E	23.66	25	PBIL1, PBIL2, PBIL3, PBIL4	29
CASO 7 (n=149)	23	CGA	25.66	28	PBIL2, PBIL3, CGA_E	30
CASO 8 (n=156)	26	PBIL4	29.28	33	PBIL2, PBIL3	35

Tabla 83.: Resumen de los resultados de los experimentos para los algoritmos de Evolución de Poblaciones mediante Aprendizaje Incremental (PBIL).

3.2.3. Optimización a partir de Colonias de Hormigas

Sistema de Hormigas (SH)

SH	S1	S2	S3	Óptimo
CASO 1 (n=135)	21	21	20	25
CASO 2 (n=142)	19	19	19	24
CASO 3 (n=149)	21	23	22	27
CASO 4 (n=156)	21	19	22	28
CASO 5 (n=135)	16	19	17	26
CASO 6 (n=143)	22	24	23	29
CASO 7 (n=149)	25	24	24	30
CASO 8 (n=156)	26	27	31	35

Tabla 84.: Resultados de los experimentos para el algoritmo del Sistema de Hormigas.

SH	Peor	Media	Mejor	Óptimo
CASO 1 (n=135)	20	20.66	21	25
CASO 2 (n=142)	19	19	19	24
CASO 3 (n=149)	21	22	23	27
CASO 4 (n=156)	19	20.66	22	28
CASO 5 (n=135)	16	17.33	19	26
CASO 6 (n=143)	22	23	24	29
CASO 7 (n=149)	24	24.33	25	30
CASO 8 (n=156)	26	28	31	35

Tabla 85.: Resumen de resultados experimentales para el Sistema de Hormigas.

Sistema de Hormigas con Búsqueda Local (SHBL)

SHBL	S1	S2	S3	Óptimo
CASO 1 (n=135)	19	21	19	25
CASO 2 (n=142)	20	19	19	24
CASO 3 (n=149)	20	20	22	27
CASO 4 (n=156)	21	21	19	28
CASO 5 (n=135)	20	19	21	26
CASO 6 (n=143)	22	22	24	29
CASO 7 (n=149)	23	22	22	30
CASO 8 (n=156)	29	27	30	35

Tabla 86.: Resultados de los experimentos para el algoritmo del Sistema de Hormigas con Búsqueda Local.

SHBL	Peor	Media	Mejor	Óptimo
CASO 1 (n=135)	19	19.66	21	25
CASO 2 (n=142)	19	19.33	20	24
CASO 3 (n=149)	20	20.66	22	27
CASO 4 (n=156)	19	20.33	21	28
CASO 5 (n=135)	19	20	21	26
CASO 6 (n=143)	22	22.66	24	29
CASO 7 (n=149)	22	22.33	23	30
CASO 8 (n=156)	27	28.66	30	35

Tabla 87.: Resumen de los resultados de los experimentos para el algoritmo del Sistema de Hormigas con Búsqueda Local.

Sistema de Hormigas de la Mejor y la Peor Hormiga (SHBW)

SHBW	S1	S2	S3	Óptimo
CASO 1 (n=135)	21	20	22	25
CASO 2 (n=142)	21	21	22	24
CASO 3 (n=149)	22	22	23	27
CASO 4 (n=156)	23	22	23	28
CASO 5 (n=135)	21	21	19	26
CASO 6 (n=143)	21	24	24	29
CASO 7 (n=149)	27	24	27	30
CASO 8 (n=156)	31	32	25	35

Tabla 88.: Resultados de los experimentos para el algoritmo del Sistema de Hormigas de la Mejor y la Peor Hormiga.

SHBW	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	20	21	22	25
<i>CASO 2 (n=142)</i>	21	21.33	22	24
<i>CASO 3 (n=149)</i>	22	22.33	23	27
<i>CASO 4 (n=156)</i>	22	22.66	23	28
<i>CASO 5 (n=135)</i>	19	20.33	21	26
<i>CASO 6 (n=143)</i>	21	23	24	29
<i>CASO 7 (n=149)</i>	24	26	27	30
<i>CASO 8 (n=156)</i>	25	29.33	32	35

Tabla 89.: Resumen de los resultados de los experimentos para el algoritmo del Sistema de Hormigas de la Mejor y la Peor Hormiga.

Sistema de Hormigas Basado en Rangos (SHRB)

SHRB	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	25	21	20	25
<i>CASO 2 (n=142)</i>	20	22	22	24
<i>CASO 3 (n=149)</i>	22	22	23	27
<i>CASO 4 (n=156)</i>	20	22	24	28
<i>CASO 5 (n=135)</i>	22	21	24	26
<i>CASO 6 (n=143)</i>	21	22	24	29
<i>CASO 7 (n=149)</i>	27	26	24	30
<i>CASO 8 (n=156)</i>	30	30	26	35

Tabla 90.: Resultados de los experimentos para el algoritmo del Sistema de Hormigas basado en Rangos.

SHRB	<i>Peor</i>	<i>Media</i>	<i>Mejor</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	20	22	25	25
<i>CASO 2 (n=142)</i>	20	21.33	22	24
<i>CASO 3 (n=149)</i>	22	22.33	23	27
<i>CASO 4 (n=156)</i>	20	22	24	28
<i>CASO 5 (n=135)</i>	21	22.33	24	26
<i>CASO 6 (n=143)</i>	21	22.33	24	29
<i>CASO 7 (n=149)</i>	24	25.66	27	30
<i>CASO 8 (n=156)</i>	28	28.66	30	35

Tabla 91.: Resultados para el Sistema de Hormigas basado en Rangos.

Sistema de Hormigas MAX-MIN (SHMM)

SHMM	<i>S1</i>	<i>S2</i>	<i>S3</i>	<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	21	20	19	25
<i>CASO 2 (n=142)</i>	20	20	20	24
<i>CASO 3 (n=149)</i>	22	23	23	27
<i>CASO 4 (n=156)</i>	19	22	21	28
<i>CASO 5 (n=135)</i>	21	21	19	26
<i>CASO 6 (n=143)</i>	25	23	24	29
<i>CASO 7 (n=149)</i>	27	26	24	30
<i>CASO 8 (n=156)</i>	29	29	30	35

Tabla 92.: Resultados de los experimentos para el algoritmo del Sistema de Hormigas MAX-MIN.

SHMM	Peor	Media	Mejor	Óptimo
CASO 1 (n=135)	19	20	21	25
CASO 2 (n=142)	20	20	20	24
CASO 3 (n=149)	22	22.66	23	27
CASO 4 (n=156)	19	20.66	22	28
CASO 5 (n=135)	19	20.33	21	26
CASO 6 (n=143)	23	24	25	29
CASO 7 (n=149)	24	25.66	27	30
CASO 8 (n=156)	29	29.33	30	35

Tabla 93.: Resumen de los resultados de los experimentos para el algoritmo del Sistema de Hormigas MAX-MIN.

Resumen Algoritmos de Optimización a partir de Colonias de Hormigas (ACO)

ACO	CASO 1 (25) n=135			CASO 2 (24) N=142			CASO 3 (27) n=149			CASO 4 (28) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
SH	20	20.66	21	19	19	19	21	22	23	19	20.66	22
SHBL	19	19.66	21	19	19.33	20	20	20.66	22	19	20.33	21
SHBW	20	21	22	21	21.33	22	22	22.33	23	22	22.66	23
SHMM	19	20	21	20	20	20	22	22.66	23	19	20.66	22
SHRB	20	22	25	20	21.33	22	22	22.33	23	20	22	24
ACO	CASO 5 (26) n=135			CASO 6 (29) N=143			CASO 7 (30) n=149			CASO 8 (35) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
SH	16	17.33	19	22	23	24	24	24.33	25	26	28	31
SHBL	19	20	21	22	22.66	24	22	22.33	23	27	28.66	30
SHBW	19	20.33	21	21	23	24	24	26	27	25	29.33	32
SHMM	19	20.33	21	23	24	25	24	25.66	27	29	29.33	30
SHRB	21	22.33	24	21	22.33	24	24	25.66	27	28	28.66	30

Tabla 94.: Resultados de los experimentos para los algoritmos de Optimización a partir de Colonias de Hormigas.

ACO (SH, SHBL, SHRB, SHMM, SHBW)	Peor	Media	Mejor	Óptimo		
CASO 1 (n=135)	19	SHBL, SHMM	20.66	25	SHRB	25
CASO 2 (n=142)	19	SH, SHBL	20.2	22	SHBW, SHRB	24
CASO 3 (n=149)	20	SHBL	22	23	SH, SHBW, SHMM SHRB	27
CASO 4 (n=156)	19	SH, SHBL, SHMM	21.26	24	SHRB	28
CASO 5 (n=135)	16	SH	20.06	24	SHRB	26
CASO 6	21	SHBW,	23	25	SHMM	29

(n=143)		SHRB				
CASO 7 (n=149)	22	SHBL	24.8	27	SHBW, SHMM SHRB	30
CASO 8 (n=156)	25	SHBW	28.8	32	SHBW	35

Tabla 95.: Resumen de los resultados de los experimentos para los algoritmos de Optimización a partir de Colonias de Hormigas.

3.2.4. Comparación Métodos basados en población

MP	CASO 1 (25) N=135			CASO 2 (24) n=142			CASO 3 (27) n=149			CASO 4 (28) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
AG	19	21.71	25	17	21.47	23	21	23.76	27	21	23.56	25
AGb	21	21.66	23	20	21.33	23	22	25	27	23	23.66	25
CHC	20	21.33	23	19	21	23	22	24	25	22	23	25
Clearing	20	21	22	22	22.33	23	23	24.33	25	24	24.66	25
Sharing	21	21.33	22	21	22	23	22	23.33	24	23	24.33	25
CUShar	19	20	21	17	19.33	21	21	22.33	24	21	21.33	22
NEShar	22	23.66	25	22	22.33	23	24	24	24	21	23.66	25
MCrowd	22	23	25	21	22	23	23	23.33	24	24	24.33	25
PBIL	20	22.5	25	19	20.83	24	22	23.66	27	21	22.44	24
PBIL1	22	23.66	25	19	20	21	23	23.66	24	22	23	24
PBIL2	21	23.66	25	20	21	22	23	24.33	25	23	23	23
PBIL3	23	23	23	22	22	22	23	25.33	27	21	22.33	24
PBIL4	20	21.66	24	19	21.66	24	22	23.33	25	22	23.33	24
CGA	20	21	22	20	20	20	23	23	23	21	21.33	22
CGA_E	22	22	22	19	20.33	22	22	22.33	23	21	21.66	22
ACO	19	29.66	25	19	20.2	22	20	22	23	19	21.26	24
SH	20	20.66	21	19	19	19	21	22	23	19	20.66	22
SHBL	19	19.66	21	19	19.33	20	20	20.66	22	19	20.33	21
SHBW	20	21	22	21	21.33	22	22	22.33	23	22	22.66	23
SHMM	19	20	21	20	20	20	22	22.66	23	19	20.66	22
SHRB	20	22	25	20	21.33	22	22	22.33	23	20	22	24

Tabla 96.: Resultados de los experimentos para los algoritmos correspondientes a los Métodos basados en Población (a).

MP	CASO 5 (26) n=135			CASO 6 (29) n=143			CASO 7 (30) n=149			CASO 8 (35) n=156		
	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor	Peor	Media	Mejor
AG	19	21.56	23	20	23.23	26	23	26.47	30	26	29.95	32
AGb	20	21.33	22	22	23	24	24	25.33	27	27	29.33	31
CHC	21	21.66	22	23	23.66	25	29	29.66	30	29	30.66	32
Clearing	20	21.66	23	22	24	26	26	27.33	29	30	31	32
Sharing	22	22.33	23	24	24.33	25	23	25.33	27	28	29.33	31
CUShar	19	20.66	22	20	21	22	23	23	23	26	28	30
NEShar	20	21.33	23	23	23.66	25	26	27.66	29	29	30.66	32
MCrowd	21	22	23	22	23	24	26	27	28	30	30.66	31

PBIL	20	21.94	25	21	23.66	25	23	25.66	28	26	29.28	33
PBIL1	21	21.66	22	24	24.66	25	24	24.66	25	29	30	31
PBIL2	22	23.33	25	25	25	25	26	27	28	28	31	33
PBIL3	22	22.33	23	25	25	25	24	26.33	28	27	30.66	33
PBIL4	22	22	22	24	24.33	25	25	26	27	26	28.66	30
CGA	21	21.66	23	21	22	23	23	23.66	24	27	28.33	29
CGA_E	20	20.66	21	21	21	21	25	26.33	28	27	27	27
ACO	16	20.06	24	21	23	25	22	24.8	27	25	28.8	32
SH	16	17.33	19	22	23	24	24	24.33	25	26	28	31
SHBL	19	20	21	22	22.66	24	22	22.33	23	27	28.66	30
SHBW	19	20.33	21	21	23	24	24	26	27	25	29.33	32
SHMM	19	20.33	21	23	24	25	24	25.66	27	29	29.33	30
SHRB	21	22.33	24	21	22.33	24	24	25.66	27	28	28.66	30

Tabla 97.: Resultados de los experimentos para los algoritmos correspondientes a los Métodos basados en Población (b).

Métodos Población	Peor		Mejor		Óptimo
CASO 1 (n=135)	19	CUShar, SHBL, SHMM	25	NEShar, MCrowd, PBIL1, PBIL2, SHRB	25
CASO 2 (n=142)	17	CUShar	24	PBIL4	24
CASO 3 (n=149)	20	SHBL	27	AGb, PBIL3	27
CASO 4 (n=156)	19	SH, SHBL, SHMM	25	AGb, CHC, Clearing, Sharing, NEShar, Mcrowd	28
CASO 5 (n=135)	16	SH	25	PBIL2	26
CASO 6 (n=143)	20	CUShar	26	Clearing	29
CASO 7 (n=149)	22	SHBL	30	CHC	30
CASO 8 (n=156)	25	SHBW	33	PBIL2, PBIL3	35

Tabla 98.: Resumen de los resultados de los experimentos para los algoritmos correspondientes a los Métodos basados en Población (a).

Métodos Población	Peor Media		Mejor Media		Óptimo
CASO 1 (n=135)	19.66	SHBL	23.66	NEShar, PBIL1, PBIL2	25
CASO 2 (n=142)	19	SH	22.33	Clearing, NEShar	24
CASO 3 (n=149)	20.66	SHBL	25.33	PBIL3	27
CASO 4 (n=156)	20.33	SHBL	24.66	Clearing	28
CASO 5 (n=135)	17.33	SH	23.33	PBIL2	26
CASO 6 (n=143)	21	CUShar, CGA_E	25	PBIL2, PBIL3	29
CASO 7 (n=149)	22.33	SHBL	29.66	CHC	30
CASO 8 (n=156)	27	CGA_E	31	Clearing, PBIL2	35

Tabla 99.: Resumen de los resultados de los experimentos para los algoritmos correspondientes a los Métodos basados en Población (b).

3.3. Comparación entre todos los algoritmos (resumen de todos los resultados obtenidos)

A continuación mostraremos un conjunto de tablas que resumirán los resultados para todos los algoritmos con objeto de permitir su comparación.

En la primera de las tablas estableceremos, para cada caso del problema, el orden obtenido por cada algoritmo desde el punto de vista de la media de resultados para sus tres ejecuciones. Para cada algoritmo y caso del problema se indicará un número que representará el orden, pudiendo coincidir dicho número con varios algoritmos si sus resultados también coinciden. Si el número de orden es mayor supone peor clasificación.

Para esta tabla se constituirá una tabla resumen adjunta que realizará una clasificación global de los algoritmos según la media. En esta tabla quedarán ordenados los algoritmos por el orden medio para todos los casos del problema, quedando una clasificación de regularidad global de los algoritmos en cuanto a la media.

En la segunda de las tablas indicaremos el orden obtenido por cada algoritmo para cada caso del problema atendiendo a la mejor solución obtenida. Para cada algoritmo y caso del problema se indicará un número que representará el orden, pudiendo coincidir dicho número con varios algoritmos si sus resultados también coinciden. Si el número de orden es mayor supone peor clasificación.

Para esta tabla se constituirá una tabla resumen adjunta que realizará una clasificación global de los algoritmos según la mejor solución obtenida. En esta tabla quedarán ordenados los algoritmos por el orden medio para todos los casos del problema, quedando una clasificación de regularidad global de los algoritmos en cuanto al mejor resultado obtenido.

Y, finalmente, en la última tabla se mostrará la cuantía del mejor resultado global obtenido para cada caso y el/los algoritmo/s que lo obtuvieron, en contraste con el óptimo global de cada uno de los mencionados casos.

Clasificación de los algoritmos asociados a las distintas Metaheurísticas según la media de resultados obtenidos

MH	CASO 1	CASO 2	CASO 3	CASO 4	CASO 5	CASO 6	CASO 7	CASO 8
MT								
BL								
BL(LS)	14	3	11	12	4	10	13	13
MLS	7	10	14	9	6	14	11	8
SA								
SA	15	13	16	15	14	15	19	12
SAPI	12	12	15	13	13	13	18	12
SAP2	13	11	14	14	11	14	15	11
BT	9	5	12	11	9	12	17	10
BM								
ILS	11	10	13	10	10	12	14	9
ILSP	3	6	8	11	4	9	10	5
VNS	8	3	7	7	4	7	9	9
MP								
AG								
AGb	5	4	2	3	5	6	8	4
CHC	6	5	4	5	4	5	1	2
Clearing	7	1	3	1	4	4	3	1
Sharing	6	2	6	2	2	3	8	4
CUShar	10	9	9	10	6	11	14	7

<i>NEShar</i>	1	1	4	3	5	5	2	2
<i>MCrowd</i>	2	2	6	2	3	6	4	2
PBIL								
<i>PBIL1</i>	1	8	5	5	4	2	10	3
<i>PBIL2</i>	1	5	3	5	1	1	4	1
<i>PBIL3</i>	2	2	1	7	2	1	5	2
<i>PBIL4</i>	5	3	6	4	3	3	6	5
<i>CGA</i>	7	8	7	10	4	9	12	6
<i>CGA_E</i>	4	7	9	9	6	11	5	8
ACO								
<i>SH</i>	8	10	10	11	12	6	11	7
<i>SHBL</i>	11	9	12	12	8	7	16	5
<i>SHBW</i>	7	4	9	6	7	6	6	4
<i>SHMM</i>	10	8	8	11	7	4	7	4
<i>SHRB</i>	4	4	9	8	2	8	7	5

Tabla 100.: Orden para cada caso del problema en que queda cada uno de los algoritmos con los que hemos experimentado de acuerdo a su media de resultados.

MH	Algoritmo	Regularidad
1°	PBIL2	2.625
2°	PBIL3	2.75
3°	NEShar	2.875
4°	Clearing	3
5°	MCrowd	3.375
6°	CHC	4
7°	Sharing	4.125
8°	PBIL4	4.375
9°	AGb	4.625
10°	PBIL1	4.75
11°	SHRB	5.875
12°	SHBW	6.125
13°	VNS	6.75
14°	ILSP	7
15°	SHMM	7.375
-	CGA_E	7.375
17°	CGA	7.875
18°	SH	9.375
19°	CUShar	9.5
20°	MLS	9.875
21°	BL	10
-	SHBL	10
23°	BT	10.625
24°	ILS	11.125
25°	SAP2	12.875
26°	SAP1	13.5
27°	SA	14.875

Tabla 101.: Clasificación de los algoritmos con los que hemos experimentado de acuerdo a la regularidad de su media de resultados.

Clasificación de los algoritmos asociados a las distintas Metaheurísticas según los mejores resultados obtenidos

MH	CASO 1	CASO 2	CASO 3	CASO 4	CASO 5	CASO 6	CASO 7	CASO 8
MT								
BL								
<i>BL(LS)</i>	6	3	4	3	4	4	6	6
<i>MLS</i>	4	5	5	4	5	7	4	5
SA								
<i>SA</i>	6	6	7	6	9	9	9	8
<i>SAP1</i>	6	6	8	6	8	7	8	8
<i>SAP2</i>	7	4	6	7	7	8	6	9
BT	3	4	6	4	7	6	6	4
BM								
<i>ILS</i>	5	6	6	2	8	6	7	6
<i>ILSP</i>	2	3	4	4	4	2	5	2
<i>VNS</i>	5	3	3	3	1	1	4	5
MP								
AG								
<i>AGb</i>	3	2	1	1	5	3	4	3
<i>CHC</i>	3	2	2	1	5	2	1	2
<i>Clearing</i>	4	2	2	1	4	1	2	2
<i>Sharing</i>	4	2	3	1	4	2	4	3
<i>CUShar</i>	5	4	3	4	5	5	8	4
<i>NEShar</i>	1	2	3	1	4	2	2	2
<i>MCrowd</i>	1	2	3	1	4	3	3	3
PBIL								
<i>PBIL1</i>	1	4	3	2	5	2	6	3
<i>PBIL2</i>	1	3	2	3	2	2	3	1
<i>PBIL3</i>	3	3	1	2	4	2	3	1
<i>PBIL4</i>	2	1	2	2	5	2	4	4
<i>CGA</i>	4	5	4	4	4	4	7	5
<i>CGA_E</i>	4	3	4	4	6	6	3	7
ACO								
<i>SH</i>	5	6	4	4	8	3	6	3
<i>SHBL</i>	5	5	5	5	6	3	8	4
<i>SHBW</i>	4	3	4	3	6	3	4	2
<i>SHMM</i>	5	5	4	4	6	2	4	4
<i>SHRB</i>	1	3	4	2	3	3	4	4

Tabla 102.: Orden para cada caso del problema en que queda cada uno de los algoritmos con los que hemos experimentado de acuerdo a los mejores resultados.

MH	Algoritmo	Regularidad
1°	NEShar	2.125
-	PBIL2	2.125
3°	CHC	2.25
-	Clearing	2.25
5°	PBIL3	2.375

6°	MCrowd	2.5
7°	AGb	2.75
-	PBIL4	2.75
9°	Sharing	2.875
10°	SHRB	3
11°	VNS	3.125
12°	ILSP	3.25
-	PBIL1	3.25
14°	SHBW	3.625
15°	SHMM	4.25
16°	BL	4.5
17°	CGA	4.625
-	CGA_E	4.625
19°	CUShar	4.75
20°	MLS	4.875
-	SH	4.875
22°	BT	5
23°	SHBL	5.125
24°	ILS	5.75
25°	SAP2	6.75
26°	SAP1	7.125
27°	SA	7.5

Tabla 103.: Clasificación de los algoritmos con los que hemos experimentado de acuerdo a la regularidad de sus mejores resultados.

Mejores resultados obtenidos para cada caso del problema

Métodos Población	<i>Mejor</i>		<i>Óptimo</i>
<i>CASO 1 (n=135)</i>	25	NEShar, Mcrowd, PBIL1, PBIL2, SHRB	25
<i>CASO 2 (n=142)</i>	24	PBIL4	24
<i>CASO 3 (n=149)</i>	27	AGb, PBIL3	27
<i>CASO 4 (n=156)</i>	25	AGb, CHC, Clearing, Sharing, NEShar, Mcrowd	28
<i>CASO 5 (n=135)</i>	26	VNS	26
<i>CASO 6 (n=143)</i>	26	Clearing, VNS	29
<i>CASO 7 (n=149)</i>	30	CHC	30
<i>CASO 8 (n=156)</i>	33	PBIL2, PBIL3	35

Tabla 103.: Mejores resultados obtenidos por todos los algoritmos para cada caso del problema.

4. Análisis de resultados

A continuación vamos a realizar un análisis de los resultados obtenidos en las tablas atendiendo a cada algoritmo en concreto. El objetivo es ir ascendiendo en el análisis efectuando la comparación entre las distintas filosofías y los distintos tipos de *Metaheurísticas* gracias a las tablas resumen que se confeccionaron y se presentan en la sección anterior, para finalmente realizar la comparación entre la totalidad de *Metaheurísticas* utilizadas e implementadas.

4.1. Métodos de trayectoria

4.1.1. Búsqueda Local

Búsqueda Local Básica (Ascensión de Colinas)

Resultados

Como puede comprobarse en las tablas 27 y 28, la *Ascensión de Colinas* produce unos resultados bastante homogéneos en los que no existen variaciones muy significativas de unas ejecuciones a otras para un mismo caso, aunque existen ligeras excepciones. Dichas excepciones quedan representadas de manera más expresiva por la aparición como resultado de -1 para dos de las ejecuciones (existen otras ejecuciones en las que existe variación pero no tan pronunciada). Como sabemos, los valores de la función de fitness van desde 0 hasta el número de predicados adicionales definidos por el usuario. El valor -1 es el fitness asociado a una solución no factible.

Teniendo en cuenta esto, puesto que la ascensión de colinas tiene como filosofía general la ascensión sin límites hasta llegar a la primera cima del espacio de búsqueda, si dicha primera cima corresponde a un valor de poca calidad, el algoritmo es incapaz de recuperarse y se mantiene ahí, finalizando. De esta forma, se corre el riesgo de un punto de partida localizado en una región del espacio de búsqueda de poca calidad, lo que nos lleva directamente a un mal resultado. Así, aunque el resto de resultados se mantienen guardando las distancias al óptimo pero con relativa cercanía, esas dos pequeñas puntualizaciones suponen el no poder enfrentarnos con garantías a un caso de nuestro problema en el que la forma del espacio de búsqueda nos podría jugar una mala pasada.

Los resultados establecen que la *Búsqueda Local* puede producir resultados aceptables, pero que por sí misma puede resultar insuficiente debido a su posible capacidad de sucumbir ante la no recuperación por una mala salida. Puede tratarse por tanto de una aportación adecuada en combinación con otros algoritmos pero con una filosofía exterior que evite estos estancamientos prematuros.

Se puede observar así mismo que, a medida que asciende el número de predicados, asciende así mismo la diferencia entre el óptimo y el resultado proporcionado por el algoritmo, a excepción del primer caso. La definición de más predicados supone la complicación del espacio de búsqueda ya que se reduce el número de soluciones al problema como ya sabemos. El que para un caso aislado ocurra lo contrario puede venir definido por la propia forma del espacio de búsqueda de partida, aunque sin más resultados, podría también venir dado por la poca confianza que inspira en su actuación el algoritmo de *Búsqueda Local* en base a su punto de partida.

Por otra parte, el tamaño del caso del problema tiene una influencia conjunta con el número de predicados definidos, ya que cuanto mayor es el tamaño del problema, más predicados son necesarios definir por parte del usuario. Tenemos una ligera excepción en el caso 2, pero como vemos, es aquel cuyo óptimo es más alcanzable por parte del algoritmo en sus ejecuciones, dado que su espacio de búsqueda presentará menos obstáculos.

Hay que indicar así mismo que, la *Búsqueda Local* no invierte la totalidad del tiempo de ejecución en ninguno de los casos, estancándose alrededor de los 4 minutos de ejecución. Desperdicia gracias a su filosofía e incapacidad, un tiempo valiosísimo de ejecución. La solución está, como ya se adelantó, en un nuevo algoritmo que subsane la principal dificultad y riesgo a la vez de la *Búsqueda Local* con una filosofía exterior. El siguiente algoritmo es uno de los más básicos de los que tienen tal objetivo.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto.

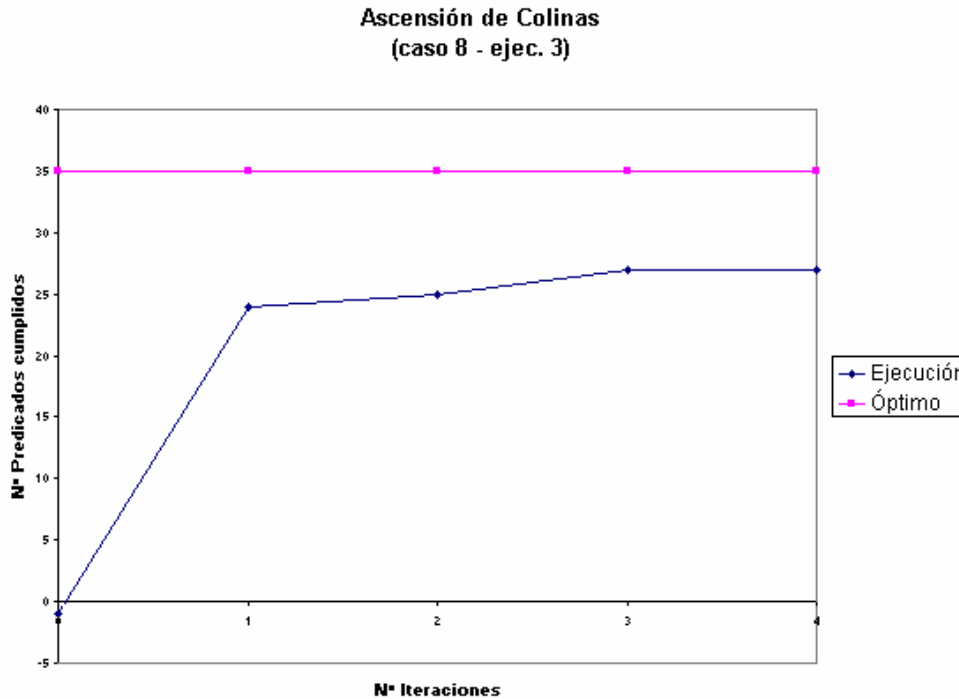


Figura 11.: Convergencia para una ejecución típica del algoritmo de Ascensión de Colinas para un caso del problema.

En la Figura 11 podemos comprobar el comportamiento ya adelantado para el algoritmo de *Ascensión de Colinas* en el comentario de los resultados. Debido a la filosofía del algoritmo éste acaba cuando de una iteración a otra no mejora la solución.

En este caso hay suerte en el punto de partida, ya que nos conduce a una región que trata de profundizar hacia la calidad en la cercanía del óptimo. En otros casos nos quedamos en el mismo punto de partida dado que en la segunda iteración no hemos sido capaces de mejorar los resultados de la primera. Pero, lo que está claro es que la gráfica de convergencia de la *Ascensión de colinas* es una gráfica ascendente (mejora de una iteración a otra) hasta una cima que normalmente no se encuentra muy lejana: es muy difícil mejorar la solución siempre de una iteración a otra.

Para esta gráfica concreta la línea ascendente se detiene en la cuarta iteración, pero proporcionándose una solución ciertamente buena, algo que no puede asegurarse dada la naturaleza del algoritmo. Así, para otra semilla distinta puede que ascendamos un poco más o puede que lo hagamos menos, pero está claro que los recursos para el ascenso son muy pocos y que permiten pocas esperanzas de llegar a regiones de calidad.

Búsqueda Local Múltiple (MLS)

Resultados

Para la *Búsqueda Local Múltiple* los resultados son casi completamente homogéneos. No existe más que una excepción, para el caso 7, y que viene dada por la aleatoriedad que se lleva asociada en el viaje a lo largo del espacio de búsqueda. Con respecto a la bondad de los

resultados, nuevamente nos mantenemos guardando las distancias con respecto al óptimo para cada caso, sin profundizar excesivamente en zonas de soluciones de alta calidad.

En efecto, al igual que para el caso anterior, también comprobamos que a medida que descende el número de predicados definidos y así mismo el tamaño del problema, más se acerca el algoritmo, en sus ejecuciones, al óptimo. Existen ciertas discrepancias a esta regla pero prácticamente de escasa relevancia.

Tenemos ante nosotros a un remedio a la principal dificultad de la *Búsqueda Local*, dado que iniciamos una *Búsqueda Local* nueva cada vez que nos quedamos estancados en una cima. Así, si no hay suerte la primera vez, podemos intentarlo otra vez en otra región distinta del espacio de búsqueda. Completamos así el tiempo desaprovechado por la *Búsqueda Local* simple de una forma útil.

Pero, hay que tener en cuenta que las búsquedas locales simples incluidas dentro de la *Búsqueda Local Múltiple* no son búsquedas exhaustivas, ya que hacer cada *Búsqueda Local* simple exhaustiva supone emplear una gran cantidad de tiempo (4 min. aprox.) y solamente partimos de 5 min. de ejecución. Por tanto, las búsquedas locales simples empleadas dentro de la *Búsqueda Local Múltiple* no generarán todo el vecindario de una solución, sino que generarán parte del vecindario de cada una (el 5%). Este cambio ha sido forzado en el algoritmo por la experimentación, ya que en principio la *Búsqueda Local* simple era exhaustiva para todos los algoritmos. Sin embargo, teniendo en cuenta los resultados prácticos, todos los algoritmos que incorporan *Búsqueda Local* simple van a insertar la *Búsqueda Local* restringida a la generación en cada iteración del 5% del vecindario.

En base a lo anterior, la *Búsqueda Local Múltiple* no es simplemente una ampliación a varias ejecuciones de la *Ascensión de Colinas*, sino que establece una menor exploración del vecindario de cada región en la que se lanza, y por consiguiente una posibilidad teórica de no ahondar suficientemente para obtener soluciones de buena calidad. En efecto, las soluciones obtenidas no son de muy buena calidad, sino que se mantienen en la línea establecida por la *Ascensión de Colinas* empeorando incluso los resultados como veremos en la comparación que a continuación realizaremos.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 12).

La figura demuestra el cambio tan rotundo que supone en la convergencia la multiplicación de búsquedas locales (no exhaustivas en nuestro caso) a la hora de constituir el algoritmo. Desaparece ahora el peligro de finalización del algoritmo con la única limitación del tiempo de ejecución. Así, la disposición de búsquedas locales encadenadas supone ir explorando someramente distintas regiones del espacio de búsqueda y cosechar de vez en cuando una solución que mejora todas las anteriores.

Se dan por tanto períodos de estancamiento que obedecen a la no superación de las distintas búsquedas locales del mayor encontrado por una búsqueda local hasta el momento. Pero, sin embargo, llega un momento en el que la exploración de una determinada región del espacio supone encontrar una solución cuyo fitness supera al de la mejor encontrada hasta ese momento.

Puesto que el algoritmo va tanteando el espacio de búsqueda y saltando bruscamente de unas zonas a otras, puede salir de los estancamientos. Pero, por otra parte, dentro de una buena región es incapaz de profundizar, pues la vecindad que comprueba es muy pequeña como ya se

indicó. De ahí que se salga de los estancamientos, pero que se sea incapaz de llegar a proporcionar soluciones de calidad.

El algoritmo no tiene intenciones manifiestas de detenerse, dado que la ejecución se ha realizado durante un periodo de tiempo muy corto. Sin embargo se puede comprobar que la subida es cada vez menor y que el estancamiento cada vez será mayor. Además, el punto de partida suele tener grandes probabilidades de ser una solución de calidad dudosa (en este caso una solución no factible).

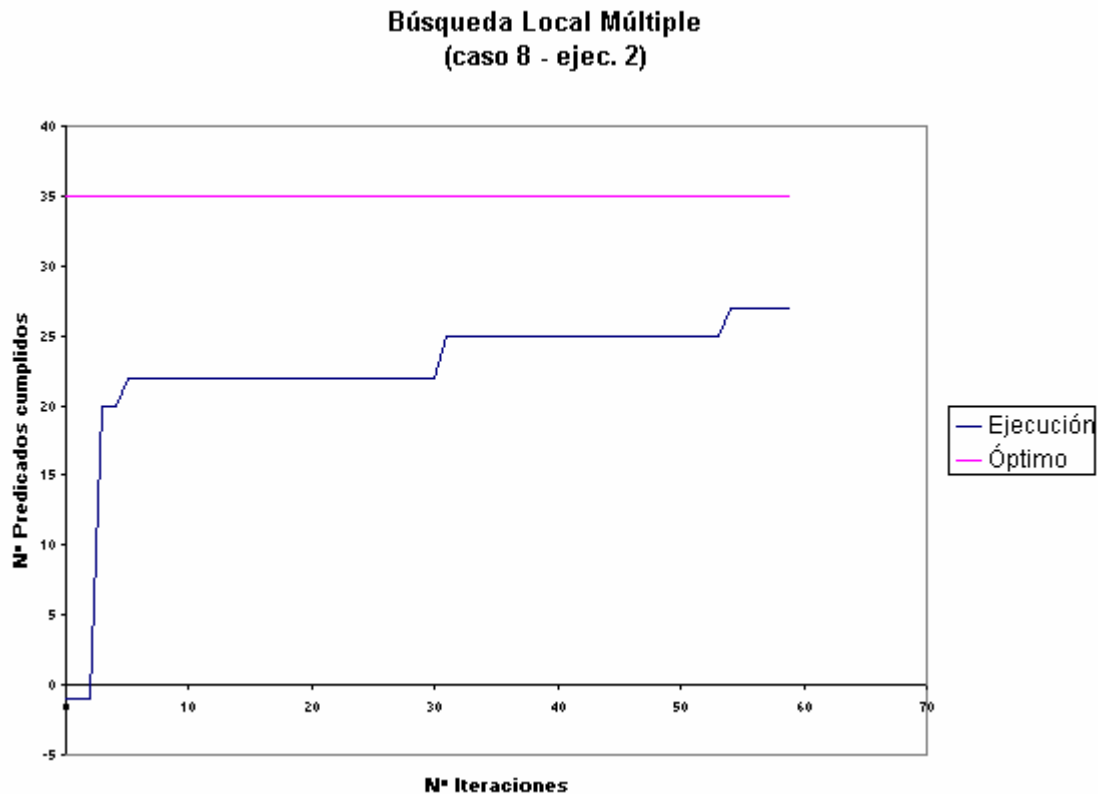


Figura 12.: Convergencia para una ejecución típica del algoritmo de Búsqueda Local Múltiple (MLS) para un caso del problema.

Hay que tener en cuenta que, si se le hubiese presentado esta salida a la Ascensión de Colinas, su resultado hubiera sido una solución no factible. He aquí claramente la diferencia entre algoritmos.

Comparación de algoritmos de Búsqueda Local

Como ya se adelantó, no tratamos con la *Ascensión de Colinas* y con un simple encadenamiento de *Ascensiones de Colinas*, donde la simple tendría todas las de perder, sino que, debido a las limitaciones temporales de la experimentación, tenemos una *Búsqueda Local Múltiple* donde cada unidad de *Búsqueda Local Simple* genera una pequeña parte del vecindario (el 5% por iteración). Así, ambos algoritmos tienen su parte de desventaja y ventaja: la *Búsqueda Local* simple permite una exploración exhaustiva, pero no es más que eso, una *Búsqueda Local* simple, mientras que la *Búsqueda Local múltiple*, aunque tiene múltiples posibilidades ante el estancamiento, no tiene posibilidad de generar todo el vecindario en cada iteración, realizando una exploración insuficiente.

Comprobando los resultados tenemos un reparto prácticamente equitativo de triunfos y derrotas. Así, para aquellos casos en los que tiene la mejor solución uno de los algoritmos, es el

otro el que tiene la peor y viceversa, salvo para el caso 4, en el que el de resultados menos homogéneos de ambos, copa los dos lugares (*Ascensión de colinas*).

Con respecto a los mejores resultados tenemos a la *Ascensión de colinas* proporcionándolos en cinco ocasiones y a la *Búsqueda Local Múltiple* en el resto (tres casos). La *Ascensión de colinas* en una sola iteración profundiza más que la *Búsqueda Local Múltiple* en el resto de iteraciones debido a su exploración exhaustiva y de ahí esa ligera ventaja en los mejores resultados. Sin embargo, debido a su reiteración, la *Búsqueda Local Múltiple* es capaz de alcanzar el mejor resultado en tres casos estableciéndose una ventaja reducida.

En lo que se refiere a los peores resultados, ambos algoritmos obtienen en cuatro de los casos el peor resultado. En este caso se mantiene el equilibrio. No influye el número de predicados adicionales ni el tamaño correspondientes al caso del problema. Sin embargo, sí que aparecen resultados significativos dentro de lo negativo correspondientes a la *Ascensión de Colinas* que ya fueron analizados en su momento, mientras que los asociados a la *Búsqueda Local Múltiple* entran dentro de la normalidad o la homogeneidad de resultados.

Es importante la necesidad de complementarse que tienen ambos algoritmos a la hora de constituir la mejor posibilidad a extraer de su filosofía: un algoritmo de búsqueda local múltiple con capacidades de búsqueda local simple cercana a la exhaustividad en cada iteración. Sin el complemento que cada uno necesita del otro, ambos quedan obstaculizados a la hora de obtener soluciones de más calidad.

El valor de la media que puede verse en la tabla 32 no es útil para comparar en este caso los dos algoritmos, pero puede ser de utilidad a la hora de comparar los algoritmos de *Búsqueda Local* en general con algoritmos de otra filosofía.

Tenemos, en resumen, un cierto equilibrio en los resultados de ambos algoritmos donde se mantienen unos resultados más homogéneos favorables a la *Búsqueda Local Múltiple*, pero que no acaban de convertirse en los mejores resultados, que forman parte en su mayoría de la *Ascensión de Colinas*, a pesar de que con este último algoritmo tenemos la posibilidad del estancamiento prematuro sin salida.

4.1.2. Enfriamiento Simulado

Enfriamiento Simulado Secuencial (SA)

Resultados

Los resultados obtenidos para el *Enfriamiento Simulado Secuencial* son como vemos carentes en su práctica totalidad de homogeneidad. Parece existir una dependencia del resultado bastante vinculada al punto inicial de búsqueda: prueba de ello son los resultados tan diferentes para un mismo caso. Al algoritmo le es difícil abandonar en ciertos casos las zonas de estancamiento como lo prueban los resultados para diferentes ejecuciones del valor -1 (la solución encontrada no es factible). La semilla tiene la última palabra en la evolución, así como la forma del espacio de búsqueda en el que se caiga en el inicio: si este espacio de búsqueda es accidentado tiene lugar un cierto avance y movimiento en el algoritmo, si por el contrario las áreas del espacio de búsqueda son llanuras le es muy difícil escapar.

Los mejores resultados que se obtienen distan mucho del óptimo. El algoritmo tiene poca capacidad de profundización y avance quedando pronto estancado sin posibilidad de evolución. Se mantienen amplias distancias con respecto al óptimo. Las regiones de calidad quedan lejos de ser exploradas. Y, si las mejores soluciones dejan mucho que desear, las peores entran dentro de la falta total de calidad con resultados incluso no factibles como ya hemos indicado. Tenemos resultados muy desequilibrados, que suponen la no explotación ni

exploración del espacio de búsqueda, con poco recorrido y dirección en la búsqueda y mucho estancamiento prematuro.

Tenemos resultados prácticamente similares en distancia al óptimo para todos los casos, excepto para los dos primeros en los que nos acercamos un poco, pero siempre dentro de la lejanía. Salvo esas dos excepciones, el aumento o reducción del tamaño de los casos del problema no influye para que mejoren o empeoren los resultados siempre en una misma línea. También ocurre lo mismo para el número de predicados adicionales. Tenemos pues que, los resultados son similares para todos los casos sin seguir con sus diferencias una relación manifiesta directa o inversa con respecto a número de predicados adicionales o tamaño de los casos.

Estamos ante un algoritmo indefenso en todos los sentidos para la explotación y exploración de los espacios de búsqueda correspondientes a los casos del problema. Se puede necesitar de más tiempo para poder ver progresos o bien cambios hacia la modificación de la explotación y la exploración para que en ese poco tiempo se puedan realizar más avances. Esto último es lo que tratan de hacer los algoritmos de *Enfriamiento Simulado Paralelos*.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 13).

La convergencia para el algoritmo de *Enfriamiento Simulado* puede comprobarse ser completamente nula. El algoritmo evoluciona durante las primeras 10-15 iteraciones, quedándose estancado completa y definitivamente hasta el final de la ejecución. La capacidad de avance del algoritmo de *Enfriamiento Simulado* queda anulada completamente en regiones llanas o poco accidentadas del espacio de búsqueda. Así mismo no reacciona en ningún caso para trasladarse a regiones más prometedoras donde sea posible escapar del estancamiento.

Prácticamente el algoritmo se limita a evolucionar en la región de partida y cuando encuentra su trofeo, no es capaz de moverse hacia otras áreas o regiones del espacio de búsqueda, cayendo en el estancamiento, que como puede comprobarse es muy prematuro.

El algoritmo es incapaz por tanto de producir soluciones de calidad, pues ha de caer en dichas regiones inicialmente para ello, algo que tiene poca probabilidad. No existe movilidad y nos quedamos siempre donde estábamos. Se converge rápidamente al mejor pico del entorno y no se pasa de ahí.

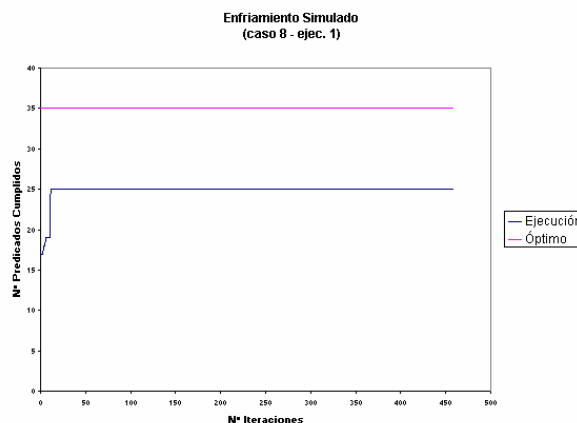


Figura 13.: Convergencia para una ejecución típica del algoritmo de Enfriamiento Simulado Secuencial para un caso del problema.

Enfriamiento Simulado Paralelo (SAP)

Varias ejecuciones secuenciales

Resultados

Para el *Enfriamiento Simulado Paralelo* se recupera en los resultados cierta homogeneidad, existiendo todavía excepciones como ocurre en los casos 5 y 8. Sabemos que este tipo de *Enfriamiento Simulado Paralelo* simplemente consiste en varias ejecuciones del *Enfriamiento Simulado Secuencial*. Con esas varias ejecuciones exploramos como mínimo una región por ejecución, de tal forma que, si una ejecución tiene malos resultados, es subsanada por el resto. Tenemos menos tiempo para la búsqueda para cada ejecución, pero podemos explorar más y existen más oportunidades de encontrar regiones de calidad en el espacio de búsqueda.

Con respecto a los mejores resultados obtenidos, hemos de decir que nos seguimos manteniendo en regiones carentes de calidad. Se mantienen mucho las distancias con respecto a los distintos óptimos para cada caso. No conseguimos profundizar en las regiones ni incluso quedarnos a las puertas de las regiones de calidad. Se sigue la línea del *Enfriamiento Simulado Secuencial*, pues seguimos estancándonos de forma prematura en todas y cada una de las ejecuciones secuenciales dentro del *Enfriamiento Simulado Paralelo*. Sin embargo, en el *Enfriamiento Simulado Paralelo* de varias ejecuciones secuenciales tenemos varias oportunidades de exploración o disposición de un nuevo punto de partida, lo que hace que no se mejoren los resultados debido al comportamiento del algoritmo, pero sí que se mejoren los peores resultados.

Suben para este algoritmo los peores resultados y la media, pues como ya se ha indicado, al existir un mayor número de oportunidades, si una de ellas no tiene buenos resultados, puede ser compensada por las otras. Pero, puesto que no hay comunicación en la búsqueda, solamente podemos tener esa ventaja y no podemos enderezar el rumbo de una ejecución nefasta desde el inicio hasta el fin. Se ha mejorado la homogeneidad de resultados sin mayor acercamiento al óptimo y permaneciendo prácticamente en la misma isla que antes. Caben esperanzas en la comunicación entre las distintas ejecuciones y de ahí el uso del siguiente algoritmo.

Nuevamente tenemos buenos resultados para los casos 1 y 2 (dos de los pequeños), pero ahí nos quedamos manteniendo el resto de distancias al óptimo, sin existir en ningún caso dependencia manifiesta con el tamaño del problema o el número de predicados adicionales definidos.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 14).

El comportamiento asociado al algoritmo de *Enfriamiento Simulado Paralelo* que utiliza varias ejecuciones secuenciales sin conexión, es idéntico al del *Enfriamiento Simulado Secuencial*. La repetición del algoritmo sin criterio y con la falta de comunicación entre las ejecuciones supone el caer nuevamente en las redes. La falta de salida y potencia para escapar de la región de partida, que normalmente no suele albergar soluciones de calidad, supone proporcionar el mismo resultado prácticamente con una que con varias ejecuciones del algoritmo.

Poco avance tenemos en la mayoría de los casos como puede verse en la gráfica. Solamente la existencia de más intentos por el mismo precio, pues todas las ejecuciones

secuenciales se estancan de manera prematura, permite tener más esperanzas de obtener mejores resultados. Y, hablamos siempre de resultados dentro de la distancia a la calidad.

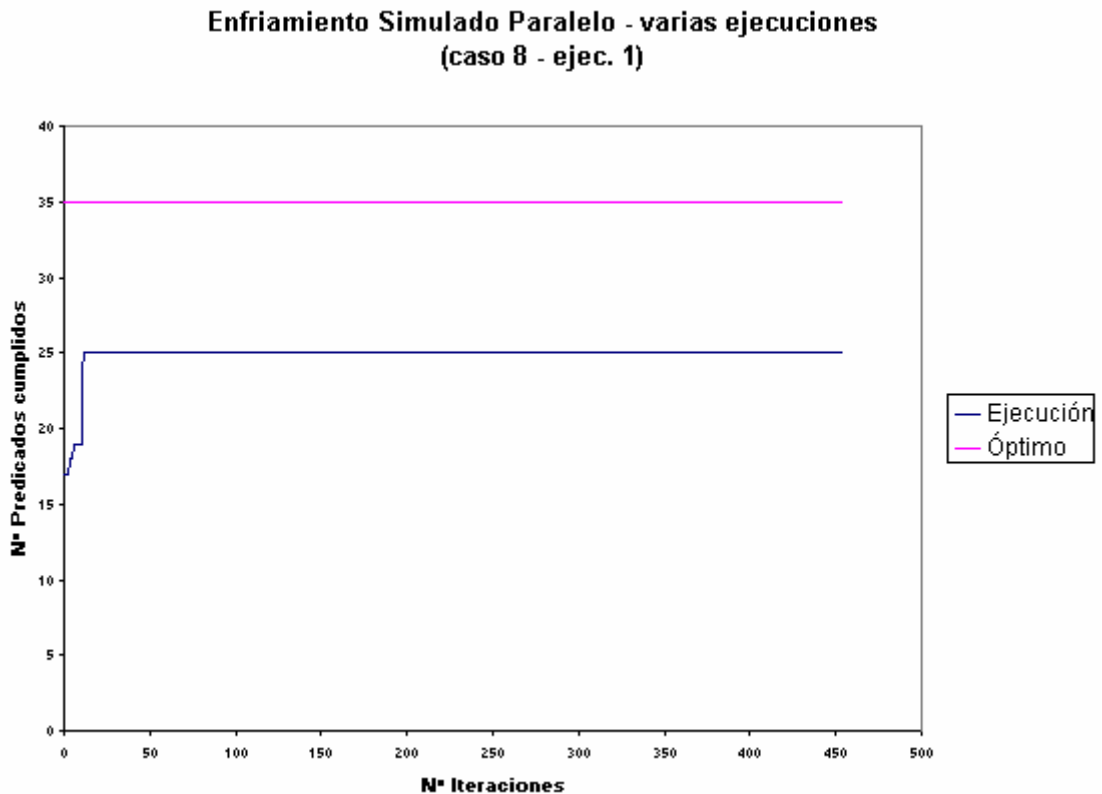


Figura 14.: Convergencia para una ejecución típica del algoritmo de Enfriamiento Simulado Paralelo (varias ejecuciones secuenciales) para un caso del problema.

Enfriamiento Simulado Paralelo

Estrategia de División

Resultados

Seguimos con los resultados obtenidos para el algoritmo de *Enfriamiento Simulado Paralelo con Estrategia de División* en busca de la homogeneidad, pero sin llegarla a conseguir y manteniéndose todavía ciertos casos en los que apenas puede percibirse. Al incorporar comunicación en las ejecuciones dentro del *Enfriamiento Simulado Paralelo* poco hemos conseguido, prácticamente hemos dado un pequeño paso, pero siempre en la misma región carente de calidad, tanto para mejores como para peores soluciones.

Las mejores soluciones siguen en la distancia, aunque para algunos casos ha tenido lugar un pequeño acercamiento al óptimo, pero siempre necesitado de intensificación. Sin embargo, la comunicación no ha resultado rentable para otros casos, dado el alto grado de estancamiento existente. Así, la solución esperanzadora al principio de la búsqueda que empuja a todas las ejecuciones dentro del algoritmo, resulta ser la trampa para el estancamiento definitivo y el no progreso. Las peores soluciones siguen la misma tónica que la dada para las mejores soluciones, y así mismo la media. El avance se produce por tanto dependiendo de la forma del espacio de búsqueda para cada caso concreto, avance que en algunos casos se convierte en retroceso.

Seguimos también sin la dependencia manifiesta de los resultados del número de predicados adicionales definidos ni del tamaño del caso del problema. Incluso en este caso no existen soluciones buenas para ningún caso del problema, sino que en los casos que se avanza lo hacemos en una instancia muy corta.

Tenemos exploración, pero existe mucha falta de explotación y capacidad para escapar del estancamiento que resulta irremediable en la mayoría de los casos. Sin intensificación y sin capacidad de escapar de los óptimos locales solo nos queda esperar a tener suerte en la exploración, algo que normalmente no ocurre y por eso estamos donde estamos en los resultados finales.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 15).

La convergencia aquí es completamente distinta a la de los dos algoritmos anteriores. La variación del algoritmo es significativa con respecto a los otros dos prácticamente similares (uno secuencial y otro la simple multiplicación del secuencial). Ahora existe la comunicación en la búsqueda y se anima a las diferentes ejecuciones en un momento dado a seguir el curso de aquella que se aloja en la región más prometedora en dicho momento.

Así, aunque existe estancamiento, la comunicación va ayudando a subsanarlo e ir saltando de los diferentes óptimos locales en busca de regiones más prometedoras. La intención del algoritmo en su convergencia es bastante buena. Pero, se debe tener en cuenta que cada vez se prolongan más las zonas de estancamiento y que a pesar de la buena intención, la solución se mejora poco a poco. Todavía es deseable más calidad, que posiblemente se obtendría a más largo plazo.

Sin embargo, se trata de un avance importante en la exploración y la convergencia adecuada en busca de la calidad. Los resultados son todavía mejorables, pero la comunicación ha supuesto una gran aportación en la evolución del algoritmo, antes indefenso y sin capacidad de búsqueda. Aunque seguimos partiendo de malas soluciones, ahora la capacidad de recuperación y salto entre regiones es mayor.

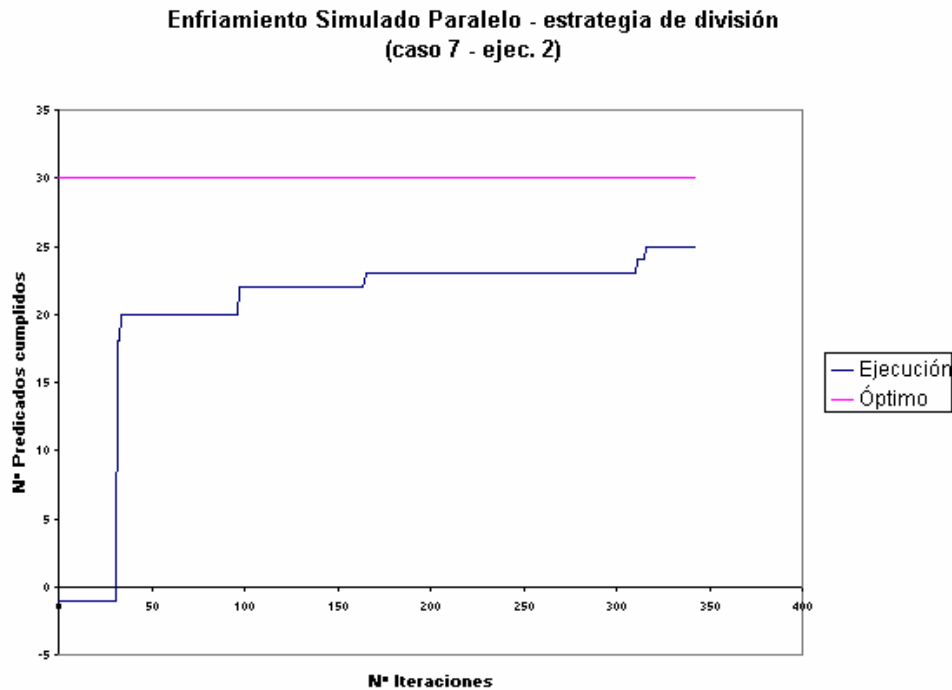


Figura 15.: Convergencia para una ejecución típica del algoritmo de Enfriamiento Simulado Paralelo (estrategia de división) para un caso del problema.

Comparación de algoritmos de Enfriamiento Simulado

Los resultados proporcionados por parte de los algoritmos de *Enfriamiento Simulado* dejan mucho que desear, manteniéndose lejos de la homogeneidad, aunque intentando alcanzarla en los algoritmos paralelos. Los resultados se mantienen en la lejanía con respecto a los óptimos, sin capacidad de mejora. No existen variaciones significativas en los resultados de los algoritmos, a excepción de la subida por parte de los algoritmos paralelos de los peores resultados obtenidos gracias a su mayor número de posibilidades de exploración.

Los mejores resultados obtenidos distan completamente de las regiones de calidad, salvo las ligeras excepciones proporcionadas por la versión secuencial y la versión paralela simple del *Enfriamiento Simulado* para los casos 1 y 2. Ninguna alternativa consigue dar esperanzas de mejores opciones. No se consigue subsanar el problema del estancamiento prematuro y, aunque el de la exploración intenta mejorar, lo que contribuye al aumento de la homogeneidad, no existen avances significativos. Las medias de resultados son bastante bajas. Ni el paralelismo ni la comunicación han conseguido dar un paso significativo hacia la calidad.

Entre los algoritmos de *Enfriamiento Simulado*, la falta de homogeneidad conduce al *Enfriamiento Simulado Secuencial* a proporcionar los peores resultados y así mismo parte de los mejores: el amplio abanico de soluciones posibles lo abarca todo. El *Enfriamiento Simulado Paralelo de varias ejecuciones secuenciales* es un algoritmo idéntico al anterior, pero que supone una mayor exploración por el mayor número de intentos sobre el espacio de búsqueda, de ahí que evite los peores resultados del *Enfriamiento Simulado Secuencial*, pero recoja la línea de los mejores. Por último, el *Enfriamiento Simulado Paralelo con Estrategia de División*, evita así mismo peores resultados, más que el otro *Enfriamiento Simulado Paralelo*, pero a la hora de proporcionar buenos resultados presenta carencias en ciertos espacios de búsqueda, precisamente en los que los otros dos algoritmos proporcionan los mejores resultados.

Ninguno de los algoritmos se encuentra a la cabeza, dependerá del espacio de búsqueda para proporcionar mejores o peores resultados, dada la gran dificultad que todos ellos tienen con

el estancamiento prematuro y la intensificación. No existe dependencia manifiesta entre los resultados y el tamaño del problema, pero de todos ellos en conjunto sí se puede percibir una dependencia más o menos indirecta con respecto al número de predicados adicionales definidos (más predicados peores resultados y viceversa).

Los algoritmos de *Enfriamiento Simulado* proporcionan por tanto unos resultados muy lejanos de lo deseable, pues presentan muchas carencias, sobre todo en lo que se refiere al abandono del estancamiento prematuro y a la intensificación.

4.1.3. Búsqueda Tabú

Resultados

Los resultados obtenidos para la *Búsqueda Tabú* se mantienen dentro de la homogeneidad, pero siempre con excepciones para ciertos casos, como ocurre para el caso 1 de manera más significativa. La bondad de estos resultados se localiza en una zona intermedia donde no se penetra lo suficientemente en las regiones atractivas del espacio de búsqueda y se guardan las distancias sin proporcionar apenas un resultado de calidad para alguno de los casos (caso 2). Tenemos componente de exploración, pero nos quedamos sin posibilidad alguna en todo caso al borde de las regiones de calidad.

Las peores soluciones gozan de cierta calidad, pero dado el número de iteraciones que supone el poco tiempo de ejecución, no somos capaces de aplicar todos los componentes del algoritmo en toda su potencialidad en busca de buenas soluciones. Vemos indicios de posibilidades de obtener calidad, pero nos quedamos ahí. La búsqueda a corto plazo realiza prácticamente su función, pero es el componente externo, la búsqueda a largo plazo, la que no puede redirigir la búsqueda lo suficiente, pues no existe tiempo suficiente para hacer discurrir a la búsqueda a corto plazo sobre el espacio de búsqueda.

El algoritmo se dispone a la misma distancia al óptimo en sus resultados para todos los casos, excepto para dos de ellos en los que se proporcionan soluciones buenas, sin establecerse en ningún momento dependencia manifiesta con respecto al tamaño del problema o el número de predicados adicionales definidos para los distintos casos del problema.

Estamos ante un algoritmo que requiere de tiempo para discurrir con todas sus capacidades. Obtenemos aquí un término medio de su desarrollo, donde no somos capaces de penetrar prácticamente en áreas de calidad del espacio de búsqueda y nos quedamos en la periferia. Se necesita esperar al estancamiento completo en la búsqueda a corto plazo para aplicar la búsqueda a largo plazo y poder entrar en nuevos frentes en el espacio de búsqueda.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 16).

La *Búsqueda Tabú* entra pronto dentro de regiones de calidad y profundiza en las mismas hasta quedarse a las puertas de las regiones optimales en el primer tercio de la ejecución del algoritmo. Una vez allí no tiene capacidad de recuperación y permanece en el estancamiento hasta que finaliza la ejecución del algoritmo.

No se llega a intensificar dentro de las regiones optimales. Existe un estancamiento prematuro, pero ya dentro de la calidad que no se puede superar. Puede no explotarse lo suficientemente el mecanismo exterior del algoritmo debido al tiempo tan corto de ejecución. Podríamos estar ante otros resultados si se diese tiempo a unas reinicializaciones más espaciadas, pero aquí es él el que tiene la última palabra.

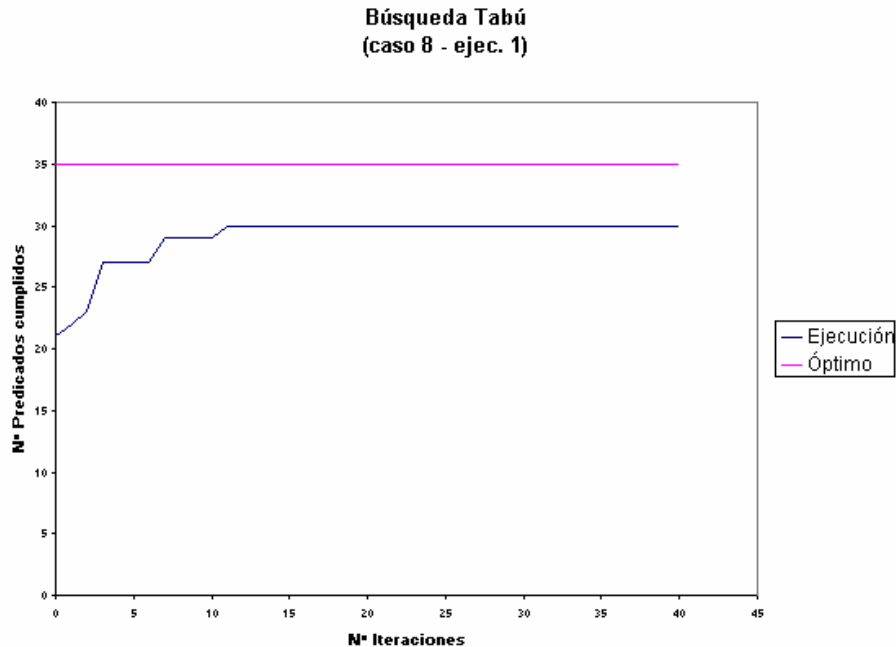


Figura 16.: Convergencia para una ejecución típica del algoritmo de Búsqueda Tabú para un caso del problema.

4.1.4. Búsqueda Multiarranque

Búsqueda Local Iterativa (ILS)

Básica

Resultados

Para la *Búsqueda Local Iterativa* podemos comprobar en los resultados de los experimentos una homogeneidad notoria: en prácticamente todos los casos excepto uno (caso 7) al menos dos ejecuciones coinciden en resultado y así mismo, las diferencias entre ejecuciones distintas de un mismo caso son de 1 a 2 predicados salvo en la primera ejecución del caso 4 (la diferencia ahí es de 4 predicados con respecto al resto de ejecuciones para dicho caso).

En lo que se refiere a la bondad de los resultados, hay que decir que los resultados proporcionados no son de muy buena calidad, manteniendo en todos los casos la distancia con respecto al óptimo. Independientemente del tamaño del caso y el número de predicados definidos, la diferencia con respecto al óptimo para el mejor resultado para cada caso se mantiene casi intacta: normalmente de 6-8 predicados, con muy pequeñas excepciones de 4-5 predicados.

Sin embargo, debemos tener en cuenta que, nuevamente, la búsqueda local simple incorporada por la *Búsqueda Local Iterativa* no es exhaustiva, sino que genera en cada iteración, como ocurría para la Búsqueda Local Múltiple y por las mismas razones, solamente el 5% del vecindario. Así, la no obtención de resultados de gran calidad, puede ser debido a que el aspecto

de profundización del algoritmo no se explota convenientemente gracias a esa Búsqueda Local simple que no intensifica lo suficiente. Recordemos que la razón de que no la utilizemos viene dada por el tiempo de ejecución de una Búsqueda Local exhaustiva o que explore en demasía el vecindario.

Ha de tenerse constancia también de que, para permitir explorar al algoritmo el espacio de búsqueda en el poco tiempo de que disponemos, el parámetro de mutación recibe un valor que no es bajo, por lo que, una vez encontrada una región satisfactoria, posiblemente la abandonemos gracias a la mutación y no tengamos en muchos casos la oportunidad de continuar por la misma región pero un poco más distanciado del punto en que nos encontrábamos antes de la mutación.

Tenemos por tanto que, la *Búsqueda Local Iterativa* proporciona resultados interesantes que necesitan de una mayor intensificación. La exploración realizada parece ser la suficiente, pero aparecen inconvenientes debido a las limitaciones de tiempo, de forma que muchas ventajas a priori de rapidez de solución, tienen sus desventajas en la práctica al no poder consumarse el asalto definitivo al óptimo.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 17).

La *Búsqueda Local Iterativa* es capaz pronto de entrar en regiones de buenas soluciones con una ascensión vertiginosa por encima de los 20 predicados. Posteriormente discurre en un avance lento pero efectivo hacia las soluciones de calidad, superando en cada caso las situaciones de estancamiento cada vez más prolongadas.

Su dirección se traduce en la continua ascensión, a pesar de los diferentes estancamientos cada vez más difíciles de superar. El tiempo acaba y queda todavía la posibilidad de la mejora en la intención del algoritmo.

El algoritmo parece tener una disposición a la proporción a largo plazo de soluciones de calidad. En lo que se refiere al tiempo correspondiente a la ejecución, hay que decir que las soluciones presentan todavía distancia con respecto a las regiones óptimas y a la calidad.

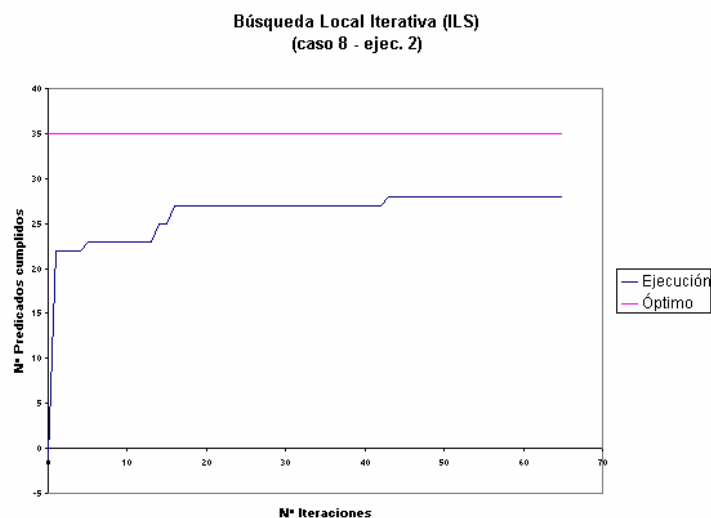


Figura 17.: Convergencia para una ejecución típica del algoritmo de Búsqueda Local Iterativa (ILS) para un caso del problema.

Búsqueda Local Iterativa

Basada en Poblaciones

Resultados

Los resultados que se obtienen a partir de las distintas ejecuciones de la *Búsqueda Local Iterativa basada en Poblaciones* entran dentro de la homogeneidad, pero con ligeros sobresaltos debidos al componente de intensificación adicional que lleva asociado el algoritmo ejecutado. Dependiendo de la forma del espacio de búsqueda dado por cada caso va a existir mayor o menor oscilación en los resultados de las tres ejecuciones. La filosofía de este algoritmo así lo preestablece.

En cuanto a la bondad de las soluciones, decir que se obtienen resultados de mucha calidad para los casos 1, 2, 5 y 8, resultados de cierta calidad para 3, 6 y 7. Tenemos por tanto una intensificación recompensada con buenos resultados. La tendencia a buenas soluciones en la búsqueda asociada a la estrategia de reemplazar el mejor y la posibilidad de exploración al tener una población de soluciones a las que se aplica *Búsqueda Local Iterativa* suponen un complemento mútuo que conduce a poder abordar cada caso del problema con ciertas garantías de éxito.

Por otra parte, el tamaño del problema y el número de predicados definidos no condicionan de manera importante las soluciones obtenidas, sino más bien la forma del espacio de búsqueda que se constituye, pues para casos de igual tamaño se obtienen resultados muy distintos, siendo uno de los ejemplos más significativos lo que ocurre para los casos 4 y 8: para el primero el resultado puede ser aceptable, mientras que para el segundo de ellos el resultado es muy bueno aproximándose al óptimo; el primero tiene un número de predicados mucho menor que el segundo y para el primero tenemos un peor resultado, cuando en teoría debería haber muchas más soluciones que cumplieran los predicados al haber definido un menor número de ellos.

En lo que se refiere a los peores resultados obtenidos para cada caso, hay que indicar que son bastante aceptables e inalcanzables incluso por los mejores resultados de otros algoritmos como hemos visto y veremos. La media por tanto queda en una línea de cierta calidad.

Hemos comprobado que, a pesar de utilizar dentro de la *Búsqueda Local Iterativa* una búsqueda local simple que no es exhaustiva, los resultados son bastante satisfactorios, gracias a la disposición de una intensificación exterior a la *Búsqueda Local Iterativa*. Disponemos por tanto de un aporte más a la *Búsqueda Local Iterativa* que tiene sus consecuencias en los resultados al establecerse dicho aporte sobre el punto débil que se comentó anteriormente para el algoritmo básico. Al tener, en este caso, mayores posibilidades de intensificación y diversificación gracias a la filosofía del algoritmo, los parámetros internos de la *Búsqueda Local Iterativa* pueden ir más dirigidos a la intensificación y dejar más a un lado la diversificación, dado que tenemos un aporte de diversificación externo.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 18).

Se puede comprobar cómo el número de iteraciones de ejecución del algoritmo es insignificante. No se puede ver el comportamiento de un algoritmo con tan poca información, aunque sí que algo podemos prever. Tenemos un total de 4 iteraciones, en las que el algoritmo se encuentra en plena explosión inicial que supone la ascensión brusca partiendo de buenas

soluciones en la primera iteración e introduciéndose en regiones de calidad para su última iteración.

El inicio vertiginoso del algoritmo es muy esperanzador en cuanto a soluciones de bastante calidad. Los resultados son excelentes y no se descartan avances posteriores si el tiempo se alargase. En poco tiempo nos hemos plantado al borde de la región optimal y todavía está por ver la actuación del algoritmo. Hay esperanzas al igual que desconocimiento de qué ocurrirá en los posibles posteriores estancamientos.

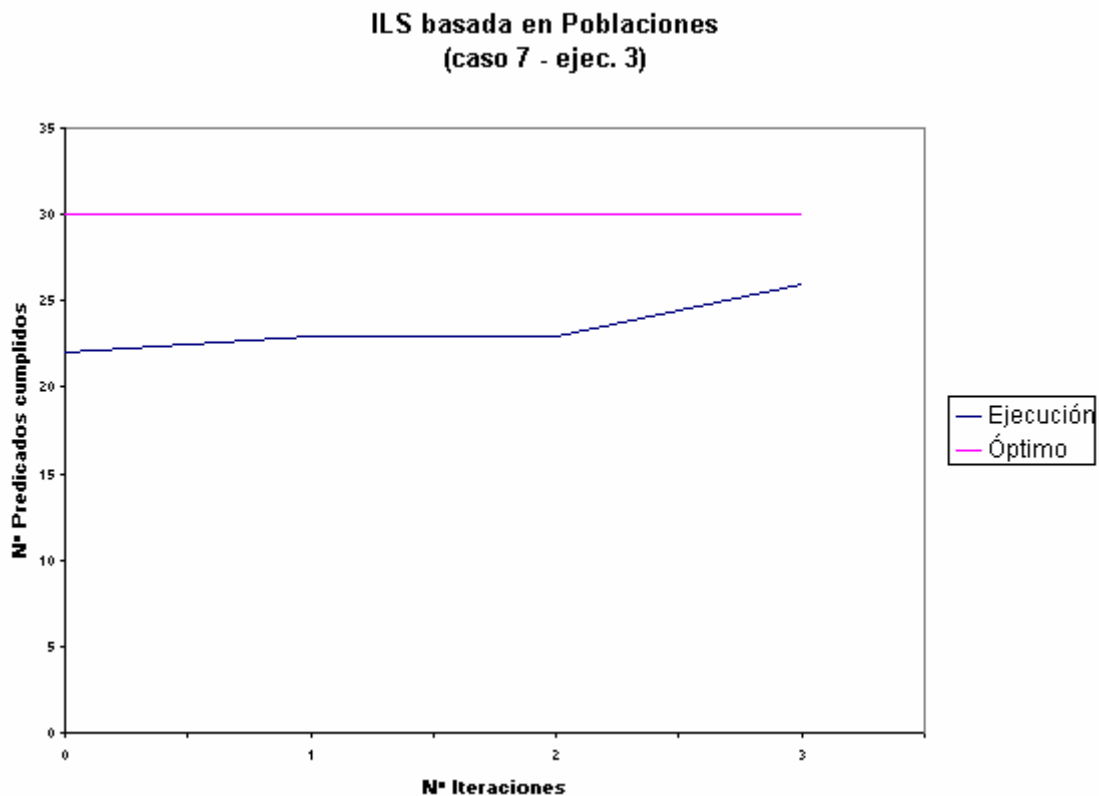


Figura 18.: Convergencia para una ejecución típica del algoritmo de Búsqueda Local Iterativa basado en Poblaciones para un caso del problema.

Búsqueda por Vecindarios Cambiantes (VNS)

Resultados

Los resultados que se obtienen de las ejecuciones correspondientes al algoritmo de *Búsqueda por Vecindarios Cambiantes* suelen mantenerse en la homogeneidad, salvo ciertas excepciones que suponen que para una de las tres ejecuciones de un caso del problema el resultado obtenido sea en sentido positivo o negativo distinto en cierta medida de las otras dos ejecuciones. Este es el caso de la tercera ejecución para el caso 7 o de la primera para el caso 6. Todo esto viene dado porque, dependiendo del espacio de búsqueda, en ciertos casos, el tamaño de vecindario máximo resulta insuficiente para salir de un óptimo local que se resiste como obstáculo.

En lo que se refiere a la calidad de los resultados tenemos que decir que estamos hablando de resultados de entidad, siendo excelentes para los casos 5 (en el que se llega a alcanzar el óptimo) y caso 2 (en el que únicamente falta el cumplimiento de dos predicados para el óptimo). El resto de resultados son buenos para 3, 6 y 7. Para los dos restantes son bastante aceptables.

La filosofía de este algoritmo que contiene un mecanismo externo para la diversificación-intensificación con la variación de vecindario, tiene la clave para abordar la mayor parte de los espacios de búsqueda correspondientes a los casos del problema con que experimentamos. No se depende aquí tanto de la no exhaustividad de la búsqueda local simple incrustada, algo que pasa prácticamente desapercibido para el discurrir y resultados del algoritmo.

Los resultados dependen del tamaño del caso del problema, siendo mejores para tamaños más pequeños y empeorando a medida que aumenta éste. Aparecen, por supuesto, excepciones, como la que proporciona el caso 1 para el que el resultado guarda un poco más las distancias con respecto al óptimo, algo que puede venir dado por la forma del espacio de búsqueda. En lo que corresponde al número de predicados adicionales definidos, hay que hablar de la no dependencia de los resultados con respecto a tal número, ya que se obtienen para un número alto de predicados definidos mejores y peores resultados y lo mismo ocurre para un número pequeño de predicados definidos.

Los peores resultados que se obtienen para cada caso y por consiguiente la media, se mantienen en una línea de cierta calidad siempre a partir de los 20 predicados cumplidos (excepto una pequeña excepción).

Disponemos con un algoritmo a tener en cuenta dentro de las Metaheurísticas para abordar el problema de la selección de la solución deseada.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 19).

La *Búsqueda por Vecindarios Cambiantes* tiene una convergencia continuamente escalonada hasta llegar a soluciones de bastante calidad dentro de las regiones óptimas. Va partiendo desde abajo y va ascendiendo poco a poco sin entorno que la detenga, saliendo de las distintas zonas de estancamiento que aparecen en la evolución del algoritmo. Acaba en un sentido ascendente y dentro de las regiones óptimas, con capacidad para decir la última palabra llegando al óptimo.

El algoritmo evoluciona con suficientes garantías para abordar los diferentes casos del problema y en particular este que tratamos. Va saliendo de los diferentes estancamientos aun siendo algunos de estos prolongados como puede comprobarse. Y por último, deja así mismo las puertas abiertas en su evolución para mejorar incluso la solución que proporciona.

Búsqueda por Vecindarios Cambiantes (VNS) (caso 7 - ejec. 2)

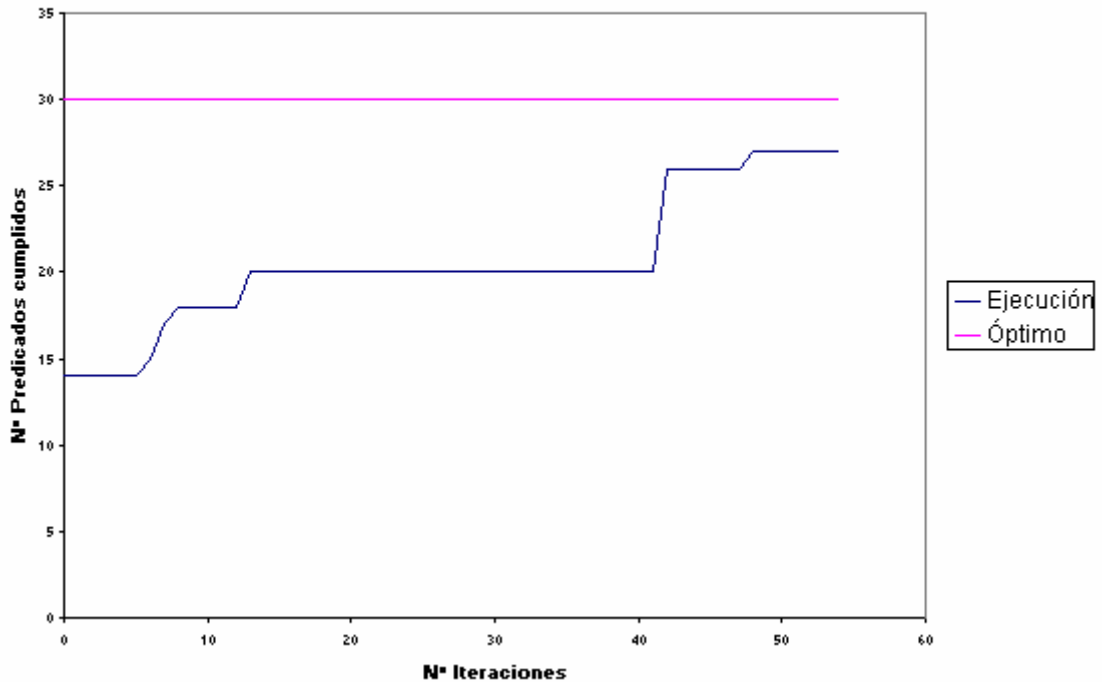


Figura 19.: Convergencia para una ejecución típica del algoritmo de Búsqueda basada en Vecindarios Cambiantes para un caso del problema.

Comparación de algoritmos de Búsqueda Multiarranque

Todos los resultados proporcionados por los algoritmos de *Búsqueda Multiarranque* se mantienen en un intervalo de cierta calidad, siempre por encima de los 19 predicados cumplidos por la figura resultante para todos y cada uno de los casos del problema. Así lo establecen los peores resultados obtenidos y así mismo las medias, que proporcionan unos resultados que profundizan dentro de los que se establecen como resultados de calidad. Los mejores resultados obtenidos son de muy buena calidad y se acercan mucho al óptimo para todos los casos con una media de dos a tres predicados, siendo el caso que más se resiste el 4, cuyo espacio de búsqueda es más difícil de abordar dados los resultados de mayor calidad y el algoritmo que los proporciona y teniendo en cuenta qué ocurre para el resto de casos del problema.

A simple vista, los resultados más bajos los proporciona la *Búsqueda Local Iterativa*, mientras que los más altos los proporciona la *Búsqueda por Vecindarios Cambiantes*. La *Búsqueda Local Iterativa basada en Poblaciones* se encuentra en un término medio entre los otros dos algoritmos. Acerca de esto, se establecen excepciones, como las correspondientes a los casos 1, 2 y 8 que indican que la *Búsqueda Local Iterativa basada en Poblaciones* también proporciona resultados de calidad para ciertos casos del problema. Así mismo, los peores resultados para los casos 5, 7 y 8 establecen la gran amplitud de resultados (menor homogeneidad) para la *Búsqueda por Vecindarios Cambiantes*, que incluso se permite para el caso 5 tener el mejor y el peor resultado. Hay que resaltar también como excepción la posibilidad de que para un caso dado, el espacio de búsqueda favorezca al algoritmo que suele proporcionar peores resultados para el resto, y siendo además uno de los de mayor tamaño: caso 4, en el que la *Búsqueda Local Iterativa* básica proporciona los mejores resultados y la *Búsqueda Local Iterativa basada en Poblaciones* el peor resultado.

Queda también claro que se obtienen mejores resultados para casos de menor tamaño y que a medida que va aumentando el tamaño empeoran o se mantienen dichos resultados. En la tabla resumen de media, mejor y peor, se puede comprobar esto, algo que no ocurría para ningún algoritmo excepto para la *Búsqueda por Vecindarios Cambiantes*, pero que la propia forma de la tabla condiciona. Sin embargo, el número de predicados definidos no interviene significativamente en los resultados, ya que para casos con un mayor número de predicados definidos se obtienen mejores resultados que para casos con menor número de predicados definidos. Influye más el tamaño que el número de predicados definidos. Para casos de igual tamaño los mejores resultados obtenidos son prácticamente similares, no así ocurre con los peores resultados y con la media, donde existen ciertas discrepancias. Sin embargo hemos de tener en cuenta que no es tanta la diferencia en el tamaño de los casos como lo es en los predicados adicionales definidos.

Se establecen, por tanto, tres escalones en lo que se refiere a los resultados obtenidos para los distintos algoritmos de *Búsqueda Multiarreglo* para los distintos casos del problema: en primer lugar tenemos a la *Búsqueda por Vecindarios Cambiantes*, tras esta se establece la *Búsqueda Local Iterativa basada en Poblaciones* y por último tenemos a la *Búsqueda Local Iterativa* básica. Hemos de tener en cuenta que en ciertas ocasiones se abre paso la *Búsqueda Local Iterativa basada en Poblaciones* por encima de las otras dos, y que, pueden existir excepciones en las que los últimos sean los primeros, como ocurre para uno de los casos (caso 4). Tenemos que, para media, mejor y peor, la regla general dada se mantiene para prácticamente todos los casos del problema, salvo las pequeñas excepciones indicadas.

4.1.5. Comparación de Métodos de trayectoria

Si verificamos los resultados correspondientes a los *Métodos de Trayectoria* podemos comprobar cómo se establecen tres grandes grupos: el primero lo constituyen los algoritmos de *Enfriamiento Simulado*, el segundo lo constituye el algoritmo de *Ascensión de Colinas*, mientras que el otro lo constituyen el resto de algoritmos.

El grupo conformado por los algoritmos de *Enfriamiento Simulado* se caracteriza por los malos resultados proporcionados para todos los casos del problema, resultados alejados en todo caso del óptimo y manteniendo una gran distancia en la búsqueda de las regiones de alta calidad. Los resultados se suelen encontrar alejados de la homogeneidad para todas las ejecuciones sobre un caso dado. Todos estos algoritmos tienen fuertes carencias de un componente intensificador y presentan un estancamiento prematuro sin capacidad de escape con mucha dependencia del punto de partida. Su peor representante es el algoritmo de *Enfriamiento Simulado Secuencial* que proporciona incluso para algunos casos soluciones no factibles y carece totalmente de homogeneidad en su rumbo y resultados. Las mejoras a este algoritmo establecidas como algoritmos paralelos consiguen subsanar la homogeneidad en las soluciones haciendo mejores las peores soluciones generadas por el anterior, pero no consiguen todavía profundizar en la búsqueda de mejores soluciones.

El tercer grupo lo conforman cinco algoritmos, que tienen como característica común la proporción de soluciones de calidad y la homogeneidad en las soluciones obtenidas para las distintas ejecuciones sobre cada caso del problema, algo que no respeta el algoritmo que compone el segundo grupo, el algoritmo de *Búsqueda Local (ascensión de colinas)* que hace honor a su filosofía dependiente del punto de partida, pudiendo proporcionar desde soluciones no factibles hasta soluciones de bastante calidad. De ahí que se haya establecido este algoritmo de *Búsqueda Local* como punto intermedio entre los algoritmos de *Enfriamiento Simulado* y el resto de algoritmos, pues el algoritmo de *Ascensión de Colinas* puede proporcionar y de hecho proporciona soluciones nefastas o de poca calidad, tal y como ocurre con los algoritmos del primer grupo, y soluciones de cierta o bastante calidad, tal y como ocurre para los algoritmos del tercer grupo.

Por último, y pasando al tercer grupo, hemos de hablar de que consiste en un grupo de algoritmos con un componente externo que dirige la búsqueda y un componente interno que profundiza sobre dicha dirección de la búsqueda. En este grupo se incluyen todos los algoritmos de *Búsqueda Multiarranque* junto con la *Búsqueda Local Múltiple* que puede considerarse como el algoritmo de *Búsqueda Multiarranque* más básico, y el algoritmo de *Búsqueda Tabú* que tiene su búsqueda a corto y largo plazo en base a lo que hemos indicado. De todos estos en cuanto a media y mejores resultados proporcionados para la mayor parte de los casos del problema destaca el algoritmo de *Búsqueda por Vecindarios Cambiantes*. Junto a este destaca también en un pequeño escalón más bajo el algoritmo de *Búsqueda Local Iterativa basada en Poblaciones*, tanto por su media como por sus resultados globales sobre todo para tres casos del problema. Los otros tres algoritmos se hacen notorios para ciertos casos y ciertas ejecuciones, pero se mantienen siempre en una misma línea de buenos y homogéneos resultados, en los que la *Búsqueda Tabú* suele guardar más las distancias que los otros dos que incluso destacan de manera importante para algún que otro caso.

No tenemos ningún algoritmo infalible, sino que para cada espacio de búsqueda existe un algoritmo que sabe mejor explotar sus capacidades para abordarlo. Parece que en media ciertos algoritmos son mejores, pero ocurre que el algoritmo que resulta mejor para un determinado caso, luego puede no ser tan bueno para otro. Pero, lo que sí está claro es que el dominio corresponde a los algoritmos con una componente exterior de dirección del algoritmo: intensificación o diversificación, y con una interior que siga los diseños dados por la exterior. Entre ellos el que resulta con más ventaja en cuanto a resultados es el de *Búsqueda por Vecindarios Cambiantes*.

4.2. Métodos basados en población

4.2.1. Algoritmos Genéticos

Algoritmo Genético básico (modelo generacional)

Resultados

Los resultados que proporciona el *Algoritmo Genético básico* entran dentro de una cierta homogeneidad. Se establecen ciertas excepciones más significativas como la segunda ejecución del caso 3 o bien la tercera ejecución para el caso 8. Con unos peores resultados siempre a partir de 20 predicados cumplidos y con una media de distancia al óptimo por parte de dichos resultados de 5-6 predicados, tenemos una tarjeta de presentación que invita a una garantía manifiesta para la obtención de buenas soluciones en cualquier ejecución y sobre cualquier caso del problema.

En lo que se refiere a los mejores resultados obtenidos debemos estar satisfechos. Se obtienen excelentes resultados para los casos 1, 2 y 3. Para el primero de ellos nos quedamos a una distancia de dos predicados, para el segundo a un predicado y para el último alcanzamos el óptimo. Para los casos 4 y 7 nos quedamos a tres predicados siendo muy buenas soluciones, y para el resto a 4 y 5 predicados, a las puertas de la región optimal. Alcanzamos resultados muy buenos para todos los casos del problema, ejerciendo un poco de más resistencia ciertos espacios de búsqueda. Siempre existe para cada caso una ejecución al menos que despunta.

Estamos ante un algoritmo elaborado, cuyos mecanismos internos son rápidos y que en poco tiempo es capaz de proporcionar soluciones satisfactorias. Es posible que los espacios de búsqueda que se resisten tuvieran su asalto si se dispusiese de más tiempo. Está claro que la influencia del tamaño del problema y del número de predicados adicionales definidos no es muy manifiesta, y sí que lo es la forma del espacio de búsqueda, que supone que para un mismo tamaño se alcancen soluciones excelentes y por otra parte no tan buenas quedándonos a las puertas de la región optimal.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 20).

El algoritmo sigue en su convergencia una ascensión escalonada continua sin permanencia excesiva en las zonas de no mejora de solución y sin llegar en ningún momento a un estancamiento manifiesto. El algoritmo parte de buenas soluciones y tiene la actitud del ascenso desde el inicio con pequeños y grandes ascensos intercalados, con una disposición final hacia la mejora y el no estancamiento. A través de un pequeño número de iteraciones se despliega prácticamente todo el potencial tanto exterior como interior en busca del óptimo, al que no se llega finalmente pero al que sí se tiene intenciones de llegar quedándonos dentro de la región óptimal.

Los resultados óptimos parecen tener su disposición a aparecer en un momento posterior al de la finalización al algoritmo. La ascensión completa tendrá lugar a un plazo más largo, pero la actitud de exploración e intensificación sobre el espacio es muy buena y de ahí las rápidas conexiones con las regiones prometedoras del espacio.

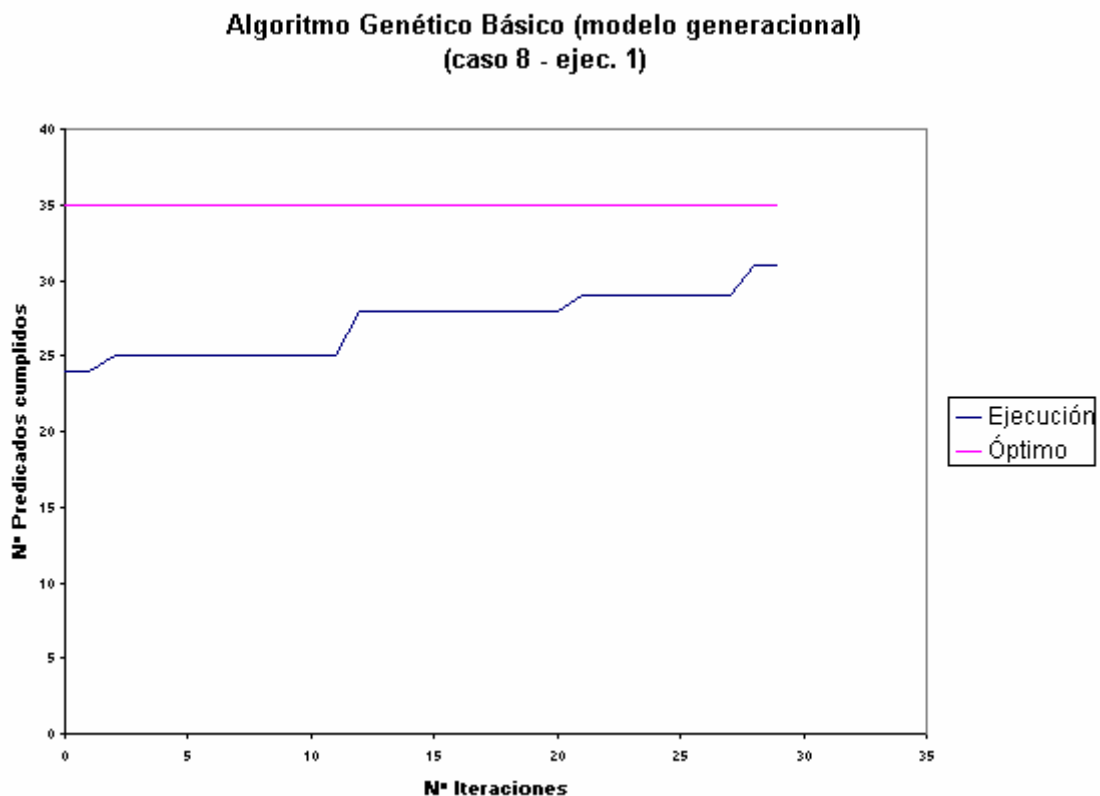


Figura 20.: Convergencia para una ejecución típica del Algoritmo Genético Básico (modelo generacional).

CHC

Resultados

El algoritmo evolutivo *CHC* tiene como sabemos el objetivo de aunar y establecer el equilibrio entre intensificación y diversificación en un *Algoritmo Genético*, a través de una serie de ideas novedosas: selección elitista, cruce uniforme HUX, prevención de incesto y reinicialización. Además tiene su aplicación más natural sobre codificación binaria, que es

nuestro caso. Es en principio una muy buena tarjeta de presentación que en realidad tendrá éxito como a continuación comentaremos.

Los resultados obtenidos para el algoritmo *CHC* entran dentro de la homogeneidad, salvo ligeras excepciones. Los peores resultados se establecen como soluciones de calidad en unos casos y como soluciones excelentes en otros. Las garantías de obtener en una ejecución una buena solución son plenas. Así, tenemos una media de distancia al óptimo por parte de las peores soluciones de 5-6 predicados, ya al borde de la región optimal y dentro de la región de calidad. E incluso aparecen grandes sorpresas como ocurre para el caso 7 en el que la peor solución se encuentra a un predicado del óptimo.

Teniendo en cuenta los peores resultados que pierden todo ese sentido, los mejores resultados entran dentro del marco optimal siendo excelentes para los casos 1, 2, 3 y 7. Para los casos 1 y 3 el algoritmo queda a tan solo dos predicados del óptimo, para el caso 2 a un predicado y para el 7 se alcanza el óptimo. Para los casos 4 y 8 se obtienen muy buenos resultados, quedando a 3 predicados del óptimo y para los dos restantes solo a 4 predicados del óptimo. Siempre se queda prácticamente dentro de la región optimal para cada caso del problema. Además, de tres ejecuciones siempre tenemos una que avanza ese mínimo para penetrar dentro de la regiones más prometedora del espacio de búsqueda.

Un buen punto de partida teórico ha conducido a su consumación con éxito debido a la rapidez de la ejecución de las distintas componentes del algoritmo para hacer evolucionar a la población en cada iteración. Suelen resistirse ciertos espacios de búsqueda que presentan una mayor dificultad de abordaje, pero dicha resistencia se recorta a lo máximo. El tamaño y el número de predicados siguen sin ser significativos en base a las diferencias para los casos sobre los que se experimenta.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 21).

Como podemos observar, tenemos muy pocas iteraciones para el algoritmo, pero muy concentradas, dados los distintos componentes que el algoritmo incluye y las grandes posibilidades que brinda cada uno. Cada iteración aglutina un completo tratamiento y generación de soluciones de tal forma que el avance que supone es espectacular dentro de la convergencia del algoritmo.

Partiendo de una solución de cierta calidad, el aprovechamiento de cada iteración supone la redistribución hacia el continuo ascenso hacia el óptimo. No se puede hablar en ningún momento de estancamiento sino de preparación para la ascensión, más manifiesta de las iteraciones 3 a la 8. Tras una ascensión vertiginosa dentro de las regiones de calidad y por último la región optimal se alcanza el óptimo global del espacio de búsqueda, sobrando incluso tiempo del establecido para la ejecución del algoritmo.

CHC (caso 7 - ejec. 2)

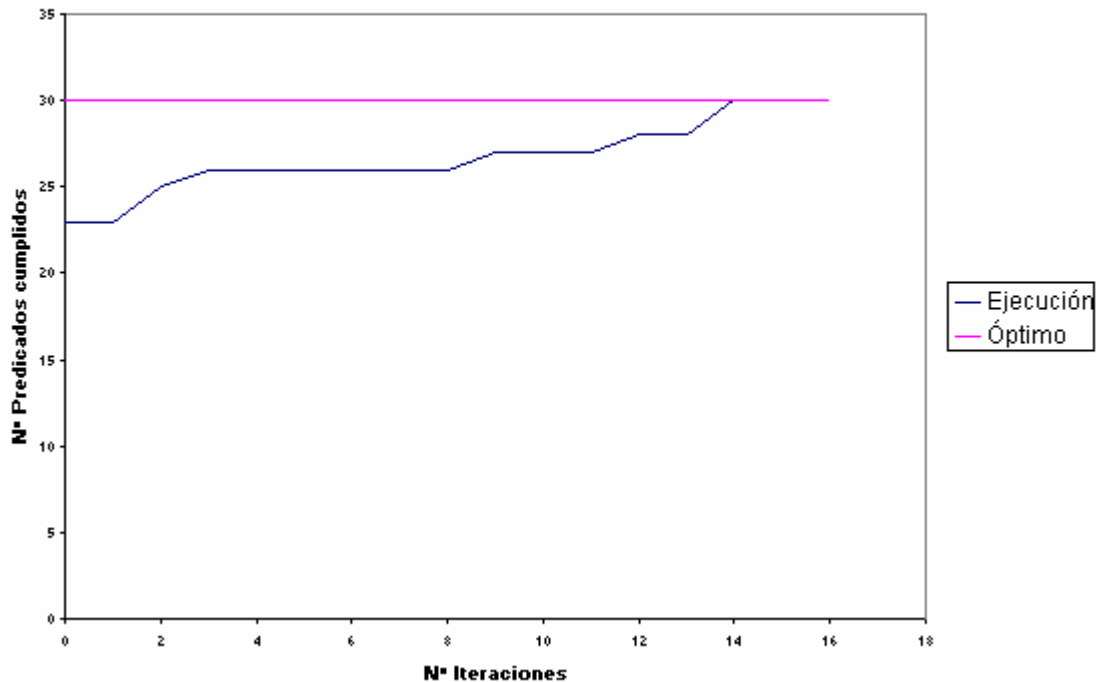


Figura 21.: Convergencia para una ejecución típica del algoritmo CHC.

Los resultados del algoritmo son excelentes y la actitud de convergencia es la ideal abordando de manera directa el espacio de búsqueda y sin ningún tipo de rodeos. La ejecución fulmina los obstáculos del espacio de búsqueda disponiéndose en la región óptima tras la simple inicialización del algoritmo. Una vez allí, hace lo más difícil: preparar el terreno para el asalto a la misma, algo que se consume rápidamente en las iteraciones posteriores a la preparación.

AG Multimodales (AG clearing)

Resultados

Los resultados obtenidos para el *Algoritmo Genético Multimodal de Aclarado* se mantienen dentro de una contrastada homogeneidad. Tratamos con mejores y peores resultados obtenidos para las ejecuciones sobre cada caso del problema muy parecidos. Se establece así la garantía de disponer para cada caso del problema y para cualquier ejecución de una solución de calidad satisfactoria.

La media de diferencia de las peores soluciones con respecto al óptimo para cada caso del problema es de 4-5 predicados. Los peores resultados están a partir de los 20 predicados y la mayoría a partir de los 22. De esos peores resultados se pueden extraer muy buenos, como ocurre para el caso 2 y no tan buenos como ocurre para el caso 6. En línea general se dan grandes esperanzas para cualquier ejecución del algoritmo sobre un caso concreto del problema.

En lo que se refiere a los mejores resultados debemos hablar de excelentes aproximaciones para los casos 2 y 7 donde nos quedamos a tan solo un predicado del óptimo, y 3, donde nos quedamos a dos predicados. Junto a estos excelentes resultados debemos hablar de una media de muy buena calidad de aproximación del resto de casos (todos ellos quedan a tres predicados del óptimo global). Independientemente del caso, de su tamaño y número de

predicados adicionales que tiene definidos, el algoritmo proporciona una estupenda solución. No se hacen en este caso distinciones de ningún tipo y siempre en pleno interior de la región optimal. Estamos ante un algoritmo muy regular capaz de abordar con éxito cualquier espacio de búsqueda.

Podemos estar ante uno de los algoritmos más prometedores en cuanto a lo que supone abordar cualquier caso del problema que tratamos. El comportamiento es plano y regulado siempre en su esencia por la búsqueda y hallazgo de la calidad.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 22).

Podemos observar en la convergencia del *Algoritmo Genético Multimodal de Aclarado* que se parten de soluciones excelentes y que directamente nos quedamos en una primera ascensión a las puertas de la región optimal. Es aquí donde parece que el algoritmo se estanca, algo erróneo dada la reacción que el algoritmo presenta al final de su ejecución, reacción que da esperanzas de la conquista del óptimo en una posteridad.

Tras un inicio fulminante donde nos plantamos directamente en las soluciones de gran calidad del espacio de búsqueda, se hace necesario reorganizar el abordaje de la región optimal a partir de la política asociada a la división del espacio en nichos, para, sirviéndonos de tal disposición ser capaces de penetrar en la región optimal.

Mantenemos la distancia con respecto al óptimo, pero la intención del algoritmo es la de recortar de manera efectiva dicha distancia y da garantías de ello al final de su ejecución. Todo ello requiere una reorganización de la búsqueda y una planificación de la misma, que es lo que internamente ocurre en el período de estancamiento.

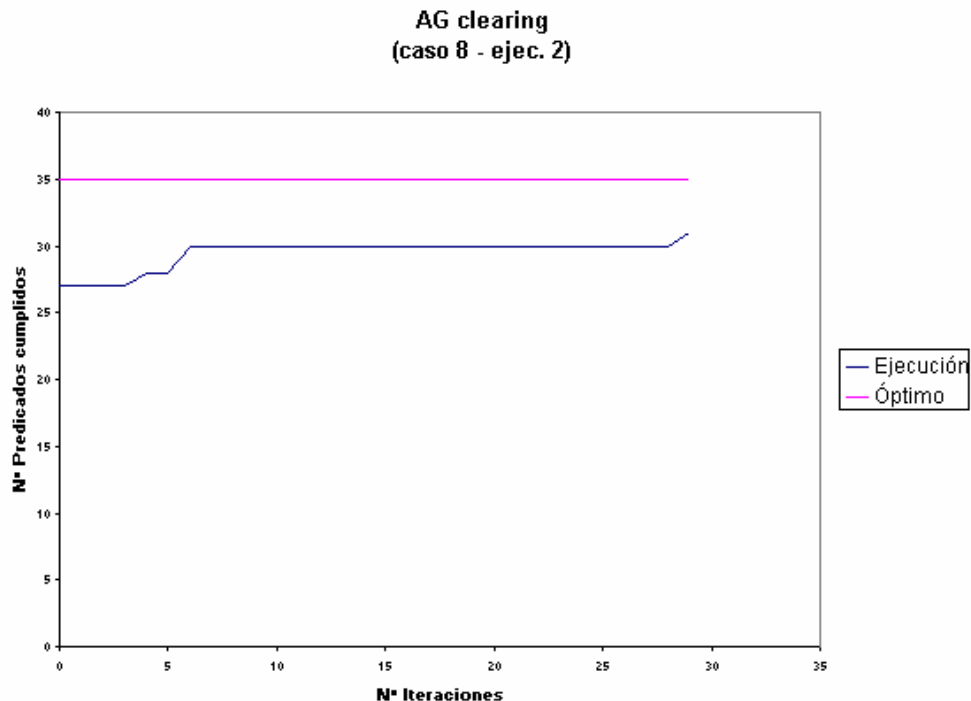


Figura 22.: Convergencia para una ejecución típica del Algoritmo Genético Multimodal de Aclarado (Clearing).

AG Multimodales (AG sharing)

Resultados

Aparece para este segundo *Algoritmo Genético Multimodal* también una gran homogeneidad en los resultados, incluso mayor que para el anterior y muy acentuada para los seis primeros casos, donde coinciden siempre dos de las tres ejecuciones en sus resultados. Los dos últimos casos constituyen una pequeña excepción que se hace explícita gracias a una de sus ejecuciones.

Los peores resultados se establecen esta vez a partir de 21 predicados cumplidos. La media de distancia al óptimo por parte de tales resultados se sigue manteniendo en torno a los 5-6 predicados, ascendiendo un poco más con respecto al algoritmo anterior, pero estableciéndose todavía en muy buena situación. Tenemos por supuesto casos en los que la distancia del peor al óptimo es de 7 predicados (casos 7 y 8) y otros en los que se reduce a 3 predicados (caso 2). Seguimos por tanto con garantías de obtener una solución de calidad para cualquier ejecución sobre cualquier caso, aunque la calidad se va reduciendo para ciertos casos del problema.

Los mejores resultados mantienen la tónica de los peores resultados aumentando así mismo la distancia a la que permanecen con respecto al óptimo global para cada caso. De esta forma tenemos un único resultado que se puede calificar de excelente: corresponde al caso 2 en el que nos quedamos a un solo predicado del óptimo. El resto de resultados pueden calificarse de buenos manteniendo todos ellos los 3 predicados de distancia a excepción de dos que quedan ahora a 4 predicados. La media de distancia al óptimo por parte de las mejores soluciones desciende aquí, pero la regularidad queda casi intacta. Otra vez se manifiesta de forma significativa.

Independientemente del tamaño del problema y del número de predicados definidos se proporcionan buenas soluciones y en una misma disposición de distancia al óptimo. El espacio de búsqueda de los casos rompe la regularidad del algoritmo pero con muy poco empuje.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 23).

Partimos en este caso también de soluciones muy buenas, pero nos mantenemos en el estancamiento al igual que ocurrió con el algoritmo anterior, pues nos encontramos a las puertas de las regiones de calidad, y muy cercanos a la región optimal. El algoritmo ha de redistribuir a las soluciones para abordar los distintos picos del espacio de búsqueda. Dicho proceso supone consumo de tiempo y he ahí el mencionado estancamiento. Una vez dispuestas las soluciones hacia los distintos picos de importancia del espacio de búsqueda se produce el asalto definitivo hacia la región optimal. En la convergencia se observa aquí una ascensión vertiginosa hacia el óptimo con apenas detenciones entre las iteraciones y que no acaba cuando el algoritmo establece su fin debido al tiempo de ejecución transcurrido.

Otra vez tenemos esperanzas de mejora al acabar la ejecución del algoritmo. La intención hacia el ascenso es palpable en el comportamiento del algoritmo, engañoso en un principio. El tiempo corta la evolución del algoritmo una vez que hemos entrado dentro de la región optimal. El comportamiento es muy parecido al del algoritmo anterior, aunque el estancamiento queda establecido desde el inicio del algoritmo y no en mitad de la evolución.

Hemos de tener en cuenta la dificultad del avance en las regiones de calidad y sobre todo en la región optimal y la necesidad de dejar discurrir a la política de intensificación-diversificación del algoritmo.

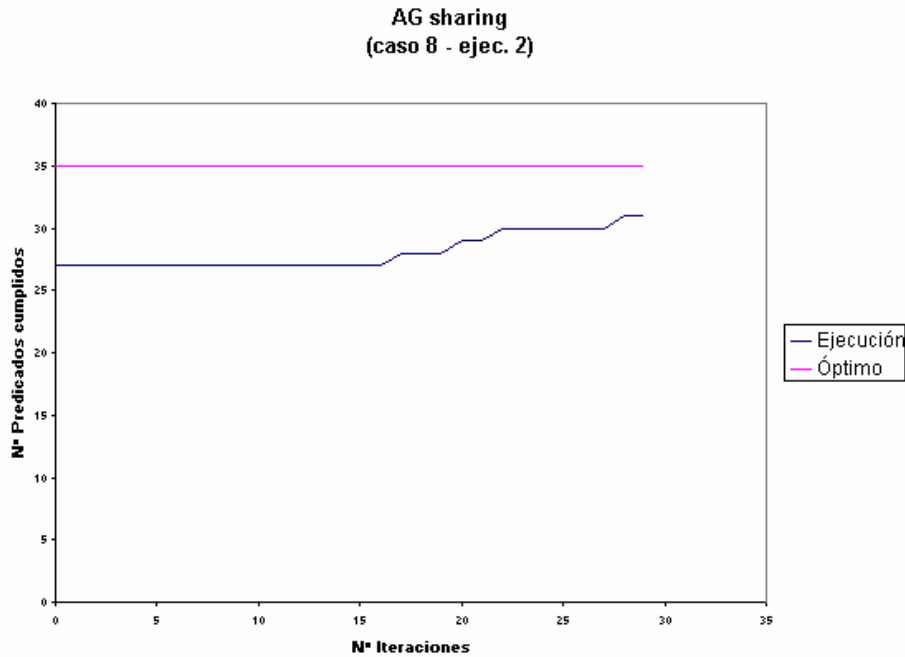


Figura 23.: Convergencia para una ejecución típica del Algoritmo Genético Multimodal de Sharing.

AG Multimodales (AG continuously updated sharing)

Resultados

Como podemos comprobar, los resultados correspondientes al *Algoritmo Genético de proporción continuamente actualizado* se mantienen dentro de una determinada homogeneidad, que rompen a veces en una corta medida ciertos resultados. Los peores resultados obtenidos para cada caso entran para muchos de ellos en un gran parecido con los mejores resultados.

Los resultados mantienen la distancia con respecto al óptimo para la totalidad de casos, siendo algunos de ellos buenos, pero en ningún momento excelentes. De esta forma, los peores resultados guardan una distancia media de 7 predicados con respecto al óptimo global, habiendo casos en los que dicha distancia es de 6 y otros en los que ésta es de 9. Tenemos un cierto retroceso con respecto a los dos algoritmos multimodales anteriores, dándose una menor garantía de la buena calidad de las soluciones obtenidas para cualquier caso y cualquier ejecución. Lo que en principio se establece como una mejora del método de proporción no acaba de cuajar en cuanto a las peores soluciones, ni tampoco en cuanto a las mejores como a continuación veremos.

Ante unas medias discretas, los mejores resultados son aceptables. Solo se consiguen buenos resultados para los casos 2 y 3 (quedándonos a 3 predicados del óptimo). Para los casos 1 y 5 son bastante aceptables, quedándonos a 4 predicados, e incluso para el caso 8 donde quedamos a 5 predicados. Para el resto de casos la distancia se amplía a los 6-7 predicados. Siendo buenos resultados hay que hablar de aceptables dado que empeoramos lo proporcionado por los dos algoritmos anteriores y hay ciertas dificultades a la hora de abordar algunos de los espacios de búsqueda. El algoritmo parece tener unas mayores facilidades para aquellos casos que presentan un menor número de predicados adicionales definidos y, aunque de forma menos clara, mayores facilidades para los casos de menor tamaño.

Parece existir una falta de despegue del algoritmo ante ciertos casos, algo que puede sorprender en principio al tratarse teóricamente de una mejora del algoritmo anterior. Una causa

muy posible puede ser la necesidad de disponer de más tiempo por parte del algoritmo a la hora de explotar sus posibilidades y capacidades plenamente, pues su elaboración es mayor que la del algoritmo anterior. Lo que sí está claro es que es una alternativa aceptable para la obtención de soluciones rápidas y no tan buena como los algoritmos anteriores.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 24).

En la convergencia del algoritmo se observa una dudosa capacidad de ascensión hasta el óptimo. Así, partiendo de una solución buena, el algoritmo es capaz de ascender tan solo dos veces hacia la conquista de mejores soluciones, siendo la segunda de ellas más tímida y alargándose cada vez más el período de estancamiento.

El algoritmo no proporciona muy buenas vibraciones en su evolución, sino con síntomas constantes de estancamiento e incapacidad de salida y ascensión a las regiones de calidad y optimales. Ante un punto de partida ideal es incapaz de evolucionar rápidamente. Esta evolución parece tener como objetivo el producirse a un más largo plazo, pues hemos de tener en cuenta que, dada la elaboración del algoritmo, el número de iteraciones durante el que discurre es bastante bajo. Es muy posible que no haya tiempo material para el correcto posicionamiento en el espacio de las soluciones que hace evolucionar el algoritmo, no se llega a la completa puesta en funcionamiento de los componentes del mismo al no llegarse a generar la información suficiente para el avance en el espacio de búsqueda.

Una mala impresión en principio que pudiera no serlo tanto en el culmen de la evolución del algoritmo, pero ya fuera del tiempo que en esta experimentación se ha establecido para su ejecución. Puede que las soluciones aceptables entren ciertamente dentro de la calidad, aunque no haya muchas esperanzas ante la gráfica.

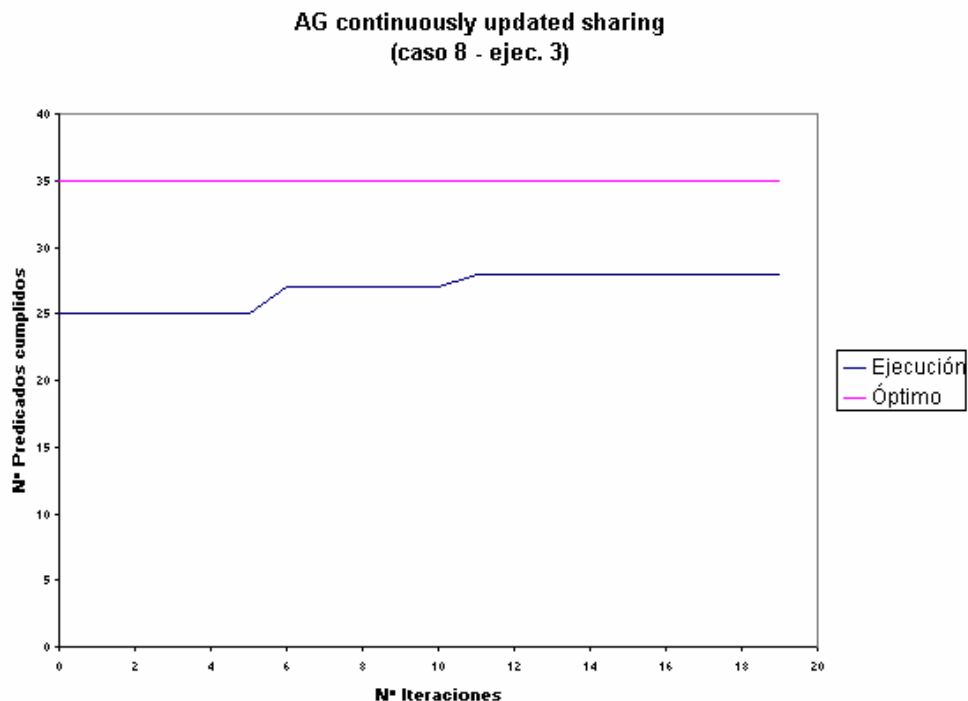


Figura 24.: Convergencia para una ejecución típica del Algoritmo Genético Multimodal de Continuously Updated Sharing.

AG Multimodales (AG nichos elitista con sharing)

Resultados

Los resultados obtenidos para el *Algoritmo Genético Multimodal de nichos elitista con sharing* presentan una gran homogeneidad. Los peores y mejores resultados obtenidos son muy parecidos y así la media. Los resultados se mantienen siempre dentro de un umbral establecido con pocas variaciones en un sentido o en el otro.

En lo que se refiere a los peores resultados obtenidos, se pueden establecer dos grupos para los distintos casos del problema. En el primero de ellos situaríamos a los casos 1, 2, 3 y 7, en el que la distancia media con respecto al óptimo global se establece en torno a los 3 predicados, algo que da idea de la gran calidad de resultados para cualquier ejecución del algoritmo sobre cada uno de estos casos. En el segundo de los grupos situaríamos a los otros cuatro casos restantes, donde la media se sitúa en torno a los 6 predicados de distancia con respecto al óptimo, siendo también un buen resultado, pero no dentro del marco de calidad cerca de la optimal como para el primero de los grupos.

Son los resultados de este algoritmo de gran calidad, excelente en unos casos y buena en otros, y solamente hemos hablado de los peores resultados obtenidos para cada caso. Sí que se puede hablar de la dificultad de ciertos espacios de búsqueda a diferencia de otros, pues tratamos con casos de todos los tamaños en cada grupo y número de predicados definidos que atiende a todos los niveles.

Analizando seguidamente los mejores resultados obtenidos podemos comprobar excelentes resultados que van desde 4 predicados de distancia para uno solo de los casos, pasando por 3 predicados para 4 casos, 1 predicado para dos casos y el óptimo para un caso. Tenemos por tanto una media de distancia para los mejores resultados y para la totalidad de casos de 2-3 predicados. No existe aquí distinción notoria entre los distintos casos del problema, ni el tamaño ni el número de predicados adicionales definidos. Los distintos espacios de búsqueda son abordados de manera satisfactoria desde este punto de vista.

Globalmente, el algoritmo se comporta de excelente forma para la práctica totalidad de los casos del problema, con gran regularidad en su evolución por los diferentes espacios de búsqueda, pero con ciertas posibilidades para algunos espacios de búsqueda de poder proporcionar soluciones que no entren dentro de la plena calidad, aunque se queden a las puertas de la misma.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 25).

La convergencia del algoritmo parte de una solución de buena calidad. Sobre la misma se mantiene durante la distribución de las soluciones en el espacio de búsqueda a la hora de abordar los picos más significativos del mismo (1/5 parte de la ejecución del algoritmo). Tras ello, el algoritmo está preparado para abordar el espacio de búsqueda con garantías y se produce una ascensión vertiginosa hacia el óptimo que conduce al corazón de la región optimal 1/5 parte de la ejecución antes de la terminación de la búsqueda del algoritmo. Ya cerca del óptimo, la última parte de la ejecución supone la preparación para la conquista del mismo, algo que no llega a producirse al finalizar el tiempo de ejecución.

Estamos ante un algoritmo capaz de proporcionar, en un corto espacio de tiempo, muy buenas soluciones. Su disposición a la hora de la búsqueda es muy buena, partiendo de un proceso de preparación que conduce al avance vertiginoso durante casi las dos terceras partes de

la ejecución del mismo. La finalización del algoritmo conduce a un nuevo momento de duda que crea esperanzas, por lo anteriormente acontecido, de llegar al óptimo. Sin embargo, la llegada a un nuevo período de estancamiento no da garantías de obtener el óptimo, pero ya dentro de una solución de excelente calidad a la que falta el paso final.

La convergencia del algoritmo da ideas de la calidad de su evolución y del enorme potencial de este algoritmo para abordar la totalidad de los espacios de búsqueda hacia la obtención de soluciones de calidad contrastada.

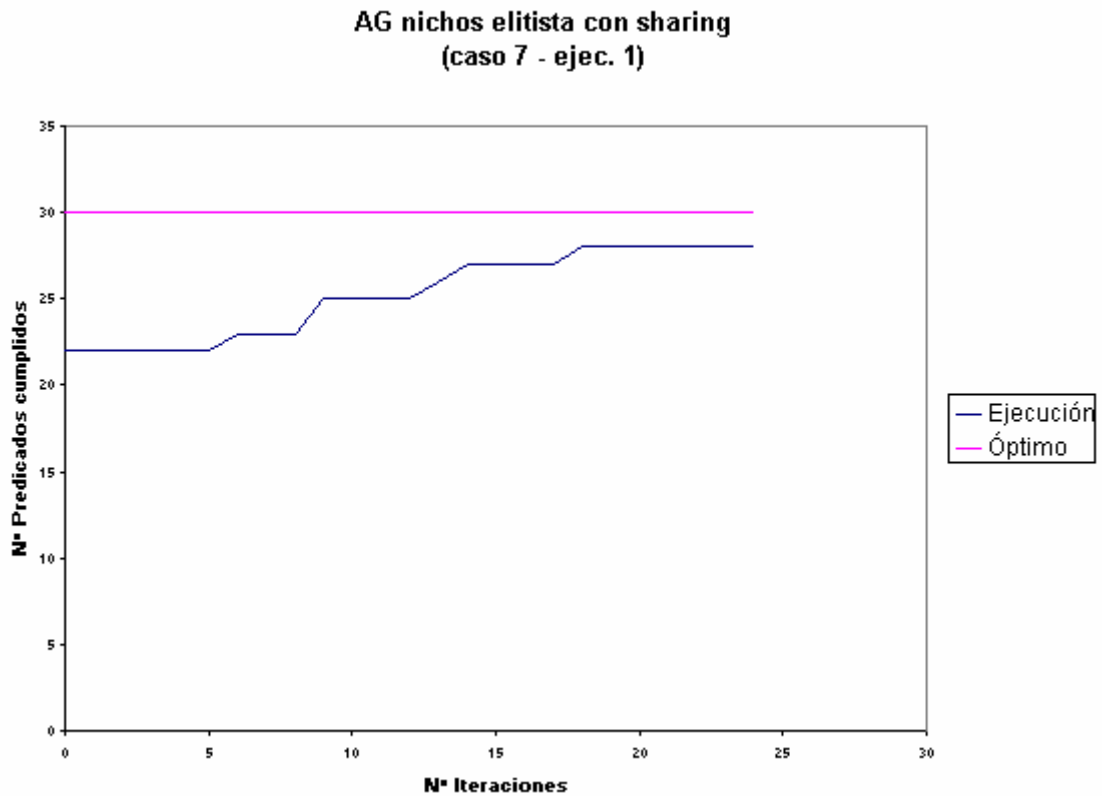


Figura 25.: Convergencia para una ejecución típica del Algoritmo Genético Multimodal de Nichos Elitista con Sharing.

AG Multimodales (AG multitud con multinichos)

Resultados

Los resultados obtenidos para el *Algoritmo Genético Multimodal de multitud con multinichos* entran dentro de la plena homogeneidad. Entre el mejor y el peor resultado no existe una diferencia superior a los 2 predicados para siete de los casos, y para el restante la diferencia es de sólo 3 predicados. La media es por tanto un fiel reflejo de las tres ejecuciones para cada caso del problema.

Junto a la homogeneidad, que resulta de especial mención, tenemos a unos peores resultados para cada caso con una distancia media al óptimo global de 4-5 predicados. Se trata de resultados excelentes que en un solo caso empeoran hasta los 7 predicados y en dos de ellos bajan a los 3 predicados. Se siguen abordando aquí unos espacios de búsqueda con más garantías que otros. Pero, para la mayoría de los casos hay facilidades para la calidad en cualquier ejecución.

Si los peores resultados eran bastante buenos, los mejores suponen la imagen viva de la calidad. Se tienen resultados excelentes para los casos 1, 2 y 7. Para el primero de ellos se alcanza el óptimo. Para los dos siguientes la distancia al óptimo global es de 1 y 2 predicados respectivamente. Tras éstos, quedan resultados bastante buenos para los casos 3, 4 y 5 (distancia al óptimo de 3 predicados). Y, por último quedan dos casos a 4 y 5 predicados, tratándose también de resultados adecuados y dentro de la calidad.

Seguimos sin encontrar una relación manifiesta con el tamaño del problema y el número de predicados definidos con respecto a los resultados. Pueden vislumbrarse mejores resultados para un tamaño más pequeño y menor número de predicados adicionales definidos, pero se establecen numerosas excepciones que señalan al espacio de búsqueda como principal determinante de la calidad de los resultados.

Tenemos ante nosotros a un algoritmo que proporciona muy buenos resultados, siempre en medio de la homogeneidad y con garantías para cualquier caso y ejecución de entrar en el marco de la calidad. Sus resultados son bastante regulares, aunque se puede comprobar la dificultad que entrañan ciertos espacios de búsqueda.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 26).

En la convergencia puede observarse el inicio del algoritmo partiendo de una buena solución. Este inicio supone el rápido ascenso en el espacio de búsqueda hasta zonas de calidad dentro de las primeras iteraciones del algoritmo. Tras introducirse en las regiones de calidad el algoritmo prosigue la subida, pero esta vez de forma más espaciada, hasta que a las puertas de la región optimal y dentro de un descanso de la ascensión espaciada acaba su ejecución. Dicha finalización puede no suponer un estancamiento, pues no discurre un tiempo suficiente como para constatarlo como tal.

Como se puede observar, el algoritmo se ejecuta durante un número poco considerable de iteraciones. Su finalización a las puertas de la región optimal y sin quedar constatado su estancamiento, puede suponer el pensar en la capacidad del algoritmo para conquistar el óptimo al tener este su pleno desarrollo a más largo plazo.

El comportamiento del algoritmo supone garantías de calidad y entra dentro de una evolución adecuada con continuas subidas y rápidas salidas de los estancamientos. No podemos juzgar la evolución del algoritmo dentro de la región optimal debido a que no discurre durante suficientes iteraciones por ésta como para poder tener capacidad de escape de la misma ante las grandes dificultades que el espacio de búsqueda comienza a ofrecer allí.

AG multitud con multinichos (caso 8 - ejec. 1)

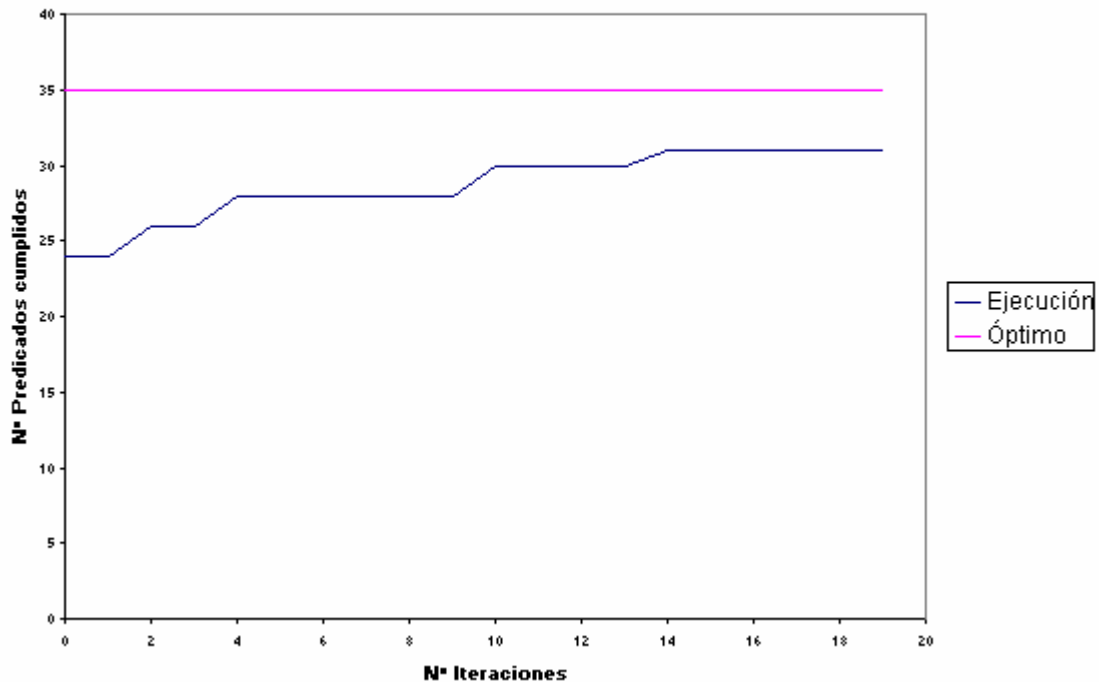


Figura 26.: Convergencia para una ejecución típica del Algoritmo Genético Multimodal de Multitud con Multinichos.

Comparación de Algoritmos Genéticos (AG)

Los *Algoritmos Genéticos* presentan en su conjunto una completa homogeneidad de resultados para todas los casos del problema. Los resultados que proporcionan son muy similares con la única excepción del *Algoritmo Genético Multimodal de proporción continuamente actualizada*. Así, de los siete *Algoritmos Genéticos*, seis de ellos constituyen un frente común con excelentes resultados y movimiento oscilante poco notorio y muy variable de un caso a otro. El único rezagado y que sigue teniendo buenos resultados, pero no dentro de la calidad continuada, es el algoritmo ya comentado.

Tenemos copadas las peores soluciones por el *Algoritmo Genético Multimodal de proporción continuamente actualizado*, con dos apariciones ocasionales e insignificantes de los otros algoritmos. Los resultados no han sido los esperados para este algoritmo, dado que no ha tenido una evolución continuada en el espacio de búsqueda, con poca ascensión y muy estancada en busca de la prosperación a largo plazo, para lo que no hay tiempo material en nuestra experimentación. Es posible que se reencontrase con el resto de algoritmos en un tiempo futuro, pero en cuanto a rapidez de soluciones de calidad queda estancado.

Las mejores soluciones se las reparten los otros seis algoritmos: la práctica totalidad de los *Algoritmos Genéticos* y además dando una idea importante de homogeneidad siendo capaces todos ellos de coincidir con la mejor solución encontrada para dos casos del problema (caso 2 y caso 4). El resto de casos se los reparten sin predominio absoluto de ningún algoritmo en particular. Así, dentro de estos 6 casos, el *AG nichos elitista con sharing* y *AG clearing* se encuentran presentes como mejores en tres de ellos, *CHC* y *AG multi-crowding* en dos de ellos, y *AG básico* y *AG sharing* en uno de ellos. Así mismo, para 5 de los 8 casos el mejor resultado obtenido lo comparten varios algoritmos. Aparece por tanto una gran competitividad entre los distintos *Algoritmos Genéticos*.

Con respecto a la bondad de las soluciones de gran calidad para todos los casos del problema. Se alcanza el óptimo para 3 de los casos (1, 3 y 7). Nos quedamos a un solo predicado para el caso 2. Y, para el resto de casos la distancia es de 3 predicados. Hablamos de soluciones siempre presentes en el corazón de la región optimal para cada caso del problema, sin distinción del número de predicados adicionales definidos ni del tamaño del caso concreto del problema.

Tenemos un conjunto de algoritmos, salvo la excepción comentada, que, con una gran regularidad abordan con éxito los diferentes espacios de búsqueda, compensando unos con otros las dificultades para cada caso. Nadie es el perfecto dominador con escasas diferencias entre unos y otros y con multitud de coincidencias en los resultados. Las medias y el resalto de cada algoritmo para un caso concreto establecen el equilibrio prácticamente entre estos seis algoritmos. Los menos elaborados, menos recargados y más clásicos parecen guardar un poco más la distancia con respecto al resto, pero nunca sin dejar de dar la talla. La conquista del óptimo o sus inmediatos alrededores para cualquier caso del problema está asegurada, a pesar de desentonar uno de los algoritmos. Cualquier ejecución de uno de los algoritmos del grupo masivo comentados llega arropada por la calidad.

4.2.2. Evolución de Poblaciones mediante Aprendizaje Incremental. PBIL.

PBIL Básico (Actualización a partir del mejor)

Resultados

Los resultados proporcionados por parte del algoritmo PBIL Básico son muy homogéneos: es escasa la diferencia entre el mejor y el peor resultado obtenido para cada caso del problema. Todos los resultados son de calidad, y se mantienen a las puertas de la región del óptimo con una media de diferencia de 4 predicados. Se llega en un caso a alcanzar el óptimo (caso 1), pero para el resto de casos mantenemos la distancia ya comentada y que apenas varía de un caso a otro y de una ejecución a otra. Independientemente del tamaño del problema y del número de predicados adicionales el algoritmo discurre bien y sus resultados se mantienen en medio de la homogeneidad al borde de la región del óptimo.

Las iteraciones del algoritmo son rápidas, el algoritmo es capaz de generar una gran cantidad de soluciones y de actualizar los componentes adicionales de la evolución en poco tiempo. Se desarrolla un número de iteraciones adecuado e incluso se echa mano de la reinicialización. Se van poniendo todos los componentes en el asador a la hora de evolucionar en la búsqueda y he ahí los resultados que invitan al optimismo. Se necesita un pequeño empuje adicional a las puertas de la región optimal, pero vamos en el buen camino.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 27).

El algoritmo se inicia desde soluciones de buena y más que buena calidad, y realiza un pequeño ascenso hacia las regiones de calidad. Una vez en las mismas permanece en el estancamiento hasta que sale del mismo una vez que éste se ha prolongado entrando de lleno en las regiones optimales del espacio de búsqueda. Al final de la ejecución del algoritmo nos encontramos ante soluciones de buena calidad y con la posible esperanza de penetrar en el interior de las regiones optimales hacia la conquista del óptimo global.

A partir de la generación de soluciones y actualización de los mecanismos de evolución de forma vertiginosa llegamos incluso a poder usar la reinicialización y escapar de estancamientos confirmados y asociados a regiones bastante amplias.

Tenemos aquí una buena línea de evolución que directamente entra dentro del núcleo del espacio de búsqueda y allí desarrolla su trabajo con una alta capacidad de escape y navegación dentro de las dificultades que entrañan las regiones de calidad.

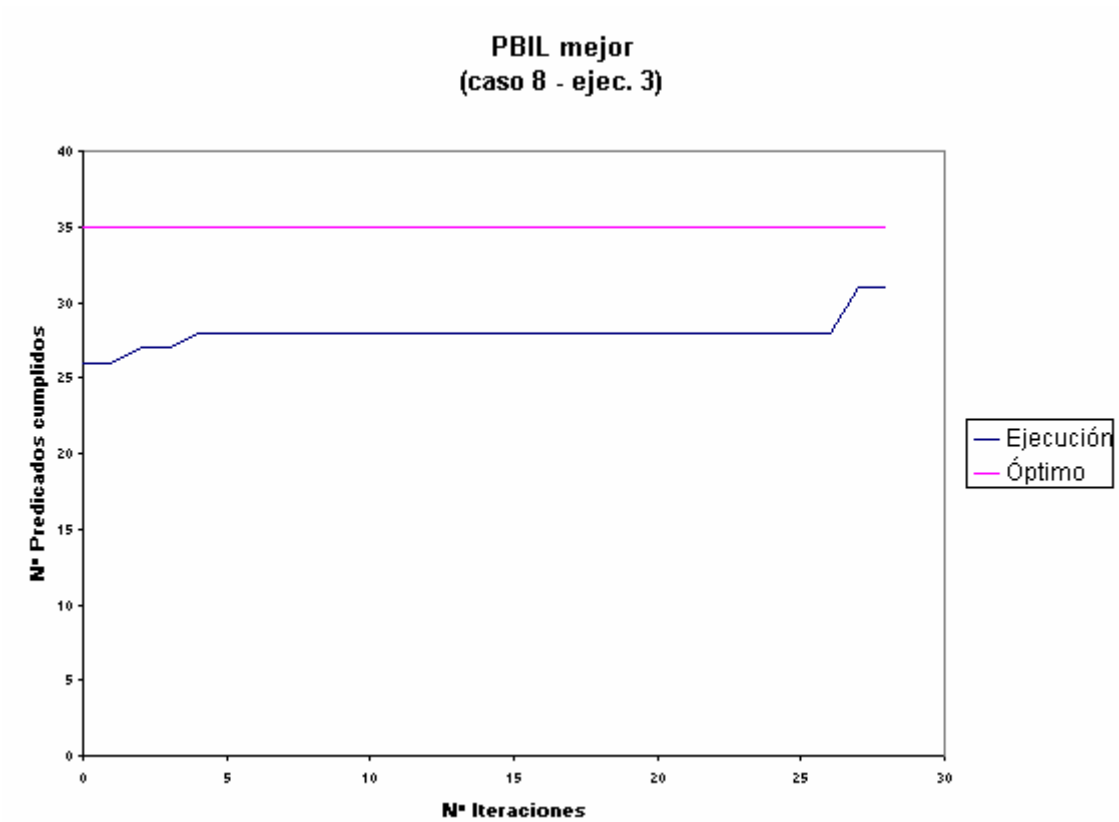


Figura 27.: Convergencia para una ejecución típica del algoritmo de PBIL con actualización a partir del mejor.

PBIL Básico (Actualización a partir del mejor y del peor)

Resultados

Con un nuevo algoritmo que presenta una sola variación con respecto al anterior, variación consistente en la actualización del vector de probabilidades también alejándonos del peor individuo generado en las posiciones distintas con el mejor, mantenemos en primer lugar la homogeneidad con soluciones mejores y peores de calidad, y realizamos, en segundo lugar y más importante, el impulso definitivo hacia el óptimo con una mayor profundización e intensificación en las regiones optimales donde antes quedábamos en la periferia.

Se obtienen resultados de excelente calidad para la gran mayoría de los casos: seis de los casos, excepto el caso 4 y el caso 6, para los que nos mantenemos en la misma línea que el algoritmo anterior. Se superan significativamente los registros del algoritmo anterior, dado que exige mucho esfuerzo el avance dentro de una región optimal.

Se realiza el avance definitivo, el asalto al óptimo para la mayor parte de los casos del problema. Se resisten un par de espacios de búsqueda y en las puertas de la región del óptimo. A partir de unas características anteriores esperanzadoras hemos constatado esta esperanza en la aproximación al óptimo con el único añadido del alejamiento del peor en la actualización. Seguimos desarrollando todo el potencial del algoritmo con iteraciones rápidas, sirviéndonos de reinicialización y con un mecanismo de evolución sencillo y rápido de actualizar.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 28). Aparece en la convergencia de este algoritmo una línea escalonada que parte de regiones de muy buena calidad y que penetra en el interior de las regiones optimales en busca del óptimo global. El algoritmo va eludiendo las distintas zonas de estancamiento más o menos prolongadas consiguiendo culminar su ejecución en una actitud ascendente hacia el óptimo global.

El comportamiento es excelente y supera al del algoritmo anterior con mayor capacidad de penetración hacia el óptimo y mayor capacidad de escape de óptimos locales. Es posible que en una posible prolongación de la ejecución del algoritmo se alcanzase el óptimo. Refinamos aún más las posibilidades y el comportamiento del algoritmo anterior.

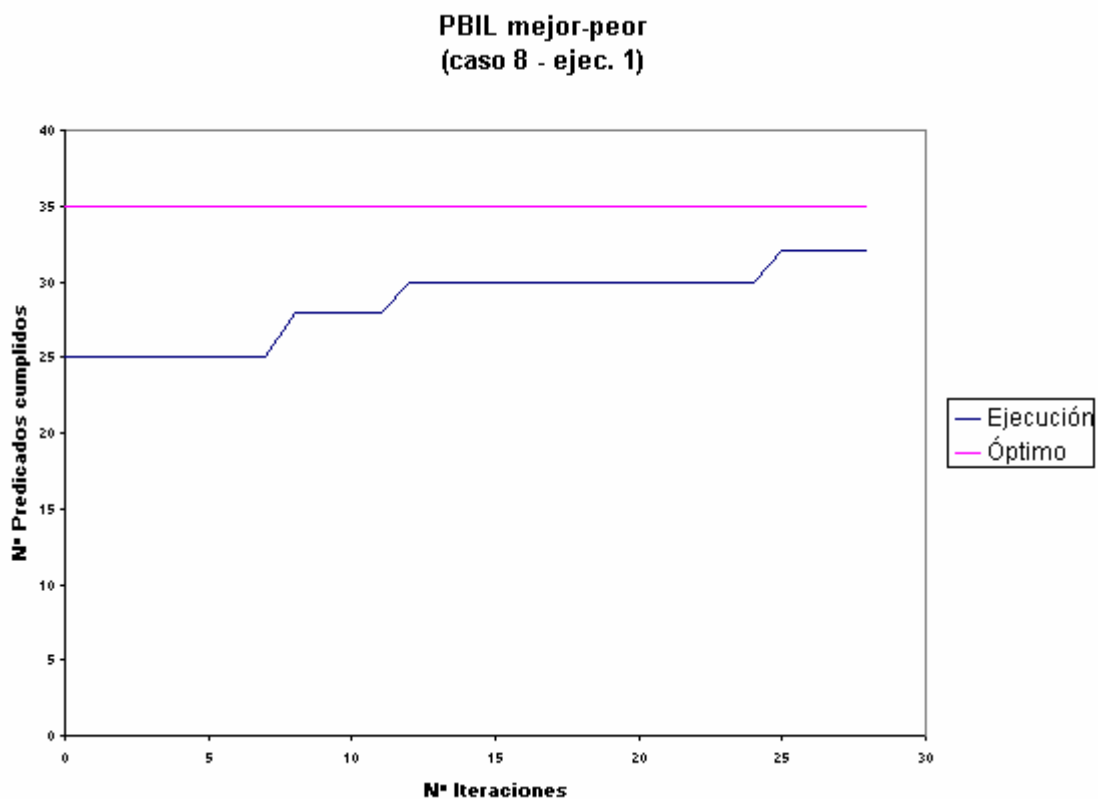


Figura 28.: Convergencia para una ejecución típica del algoritmo de PBIL con actualización a partir del mejor y del peor.

PBIL Básico (Actualización a partir de las M mejores soluciones: con todas equitativamente)

Resultados

Nuevamente se realiza un cambio en el algoritmo *PBIL* correspondiente a la regla de actualización del vector de probabilidades. El resto se mantiene. Si para el primer algoritmo la actualización era con el mejor y para el segundo de ellos era a partir del mejor y el peor, en este caso será a partir de los M mejores de manera equitativa (actualizando con todos).

Los resultados vuelven a moverse dentro de la homogeneidad, aunque surgen ciertas variaciones no muy significativas para los casos de mayor tamaño del problema. Se prosigue

nuevamente en la misma línea que anteriormente, con prácticamente todos los resultados correspondientes a la profundización dentro de las regiones optimales del espacio de búsqueda. La mayor parte de los resultados son excelentes, alcanzándose en primer lugar el óptimo para el caso 3, y quedándonos a solo dos predicados para los casos 1,2,7 y 8. Para los casos 4 y 5 las soluciones son también buenas, quedándose a las puertas de penetrar en el núcleo de la región optimal del espacio de búsqueda.

Las peores soluciones se encuentran en un nivel de calidad y siempre de manera holgada por encima de los 20 predicados cumplidos para la figura solución. Se dan incluso tres casos en los que la mejor y la peor solución son idénticas: casos 1, 2 y 6, donde la homogeneidad llega por supuesto a su culmen.

Independientemente del tamaño o del número de predicados asociados al caso del problema, el algoritmo se mantiene en sus resultados sin prácticamente variarlos con una media de distancia al óptimo de dos predicados. Nos encontramos nuevamente ante un algoritmo que supone una aportación a tener en cuenta a la hora de obtener garantías de calidad en todas las ejecuciones sobre los casos del problema.

Hemos de tener en cuenta que nos enfrentamos al mismo algoritmo que en los dos casos anteriores y seguimos con un mismo comportamiento, comportamiento que recibe uno u otro tipo de direccionamiento final dentro de las regiones optimales, uno u otro tipo de profundización, que lo hará reaccionar de una u otra forma para los diferentes casos del problema pero siempre dentro de la optimalidad y la garantía de calidad asegurada. Recordemos que el algoritmo se mantiene en sus características, las ya indicadas en los dos anteriores apartados, y que el cambio viene dado solo y exclusivamente en la regla de actualización de la distribución de probabilidades que permite la generación de la población en cada iteración.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 29).

Nuevamente el algoritmo sigue una línea escalonada en la convergencia, e incluso esta vez más prolongada sin la existencia de estancamientos significativos y partiendo en su fase inicial de una zona de soluciones buenas. El algoritmo se dispone en una continua subida hacia el óptimo, introduciéndose rápidamente al alcance del mismo sobre las regiones de alta calidad del espacio de búsqueda. La ascendencia no se detiene en la totalidad de la ejecución, y finalizada la misma se establece la esperanza de su posible alcance con si esta siguiera.

Seguimos otra vez en la disposición de las ejecuciones anteriores, refinando un poco más y perfilando la profundización y el escape de óptimos locales.

**PBIL M-mejores(todas a la vez)
(caso 8 - ejec. 2)**

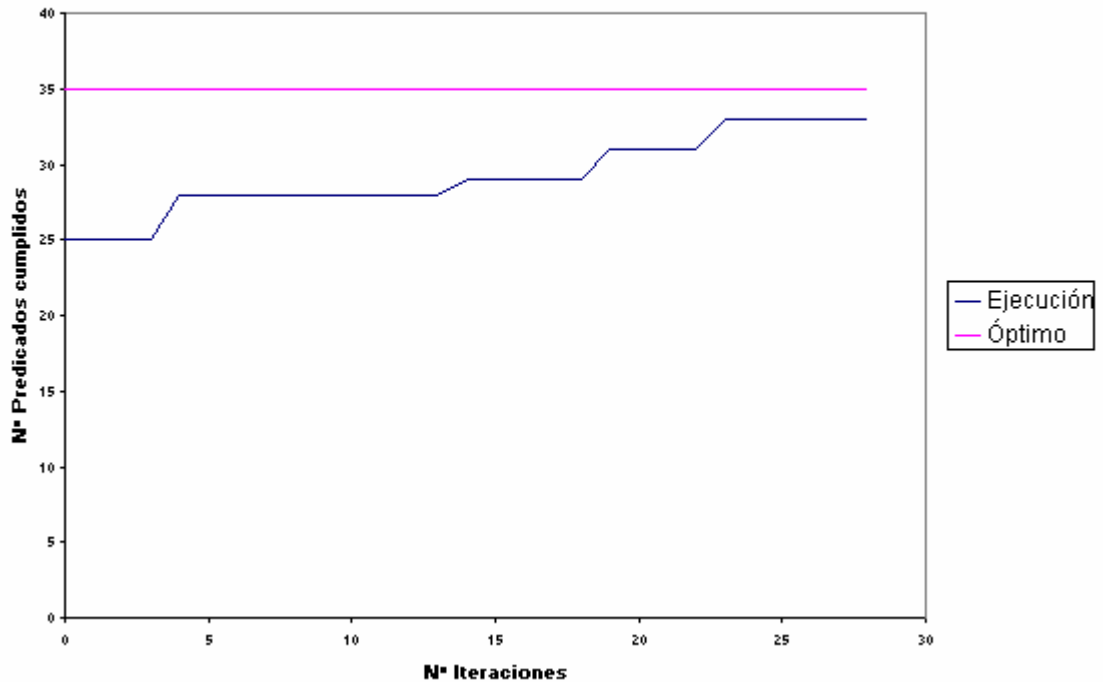


Figura 29.: Convergencia para una ejecución típica del algoritmo de PBIL con actualización a partir de las M mejores soluciones (con todas equitativamente).

PBIL Básico (Actualización a partir de las M mejores soluciones: solución de consenso)

Resultados

Los resultados de este último algoritmo básico de Evolución de Poblaciones a partir de Aprendizaje Incremental presentan una menor homogeneidad que los anteriores, aunque siguen manteniéndose en ella firmemente en la mitad de los casos del problema. Seguimos en la misma filosofía donde esta vez la regla de actualización del vector de probabilidades tiene como objetivo el actualizar a partir de una solución de consenso construida a partir de las M mejores soluciones.

Se siguen observando peores resultados que entran dentro de la calidad y mejores resultados que suponen la entrada en las regiones óptimas. Existen pequeñas diferencias en los resultados arriba y abajo para los distintos casos del problema, pero no cambia la forma de actuar ni el resultado global de la aplicación del algoritmo.

Se obtienen excelentes soluciones para los casos del problema 1, 2 (en el que incluso se alcanza el óptimo global) y 3. Para el resto de casos se mantiene en las puertas de la región óptima consiguiendo penetrar en algunos y quedando algo más rezagado en otros; se tiene así una media de 4 predicados de distancia al óptimo global. Los peores resultados por otra parte tienen una media de distancia al óptimo de 5 predicados, salvo la excepción notoria para el caso 8. Comprobamos así como existen muchas garantías de obtener buenas soluciones para las distintas ejecuciones sobre los diferentes casos.

Los resultados se mantienen en la línea de los otros tres algoritmos básicos. No existe dependencia manifiesta con respecto al tamaño del caso del problema ni al número de

predicados adicionales definidos. Es el espacio de búsqueda el que dispone las posibilidades de evolución que luego serán aprovechadas o no por el algoritmo.

Las conclusiones a extraer de este algoritmo nos llevan al terreno ya allanado por los otros tres.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 30).

En la gráfica de convergencia se puede observar el comportamiento que venimos comprobando para los otros tres algoritmos de *PBIL* básicos. Una subida escalonada que queda finalmente en las regiones óptimas. No existe variación en la convergencia con respecto al resto de algoritmos. Ya se ha comprobado que las variaciones son puntuales en los resultados, siendo las ascensiones algo más pequeñas y existiendo un mayor estancamiento en el tramo final, pero la esencia es la misma que en los casos anteriores.

Tratamos ante una repetición de lo ya comprobado y constatado que solo supone confirmarlo nuevamente a pesar de las pequeñas diferencias puntuales ya aclaradas.

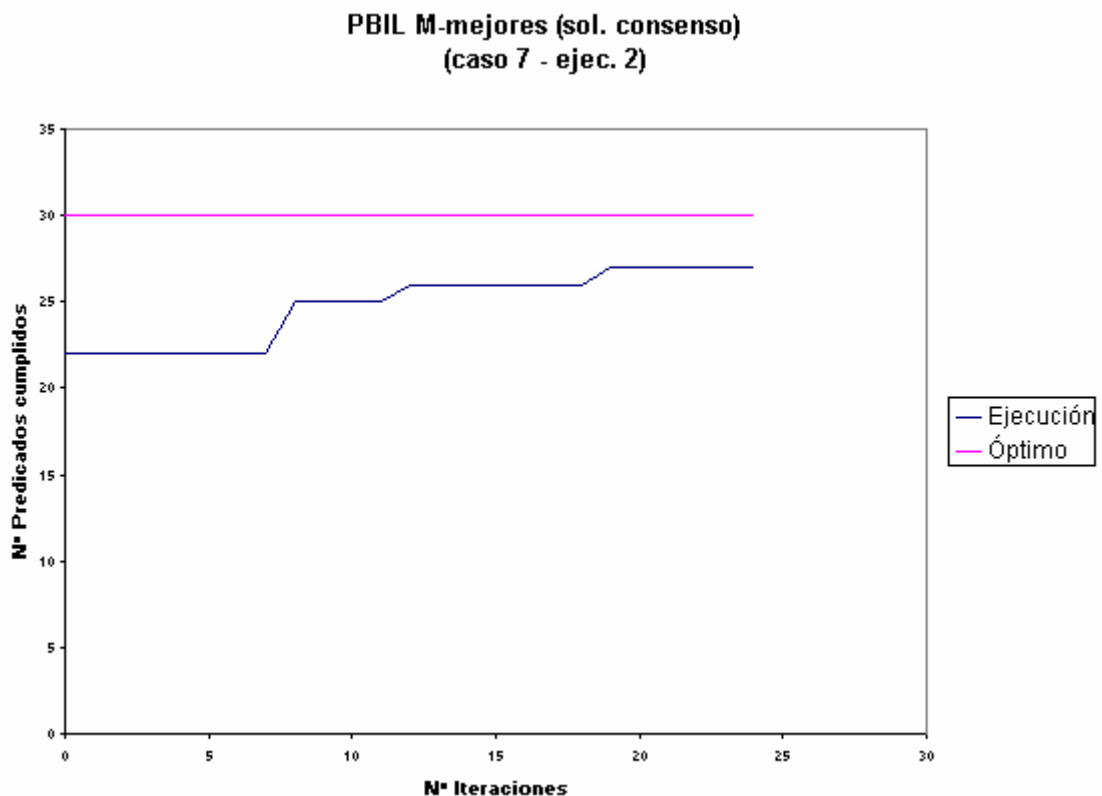


Figura 30.: Convergencia para una ejecución típica del algoritmo de PBIL con actualización a partir de las M mejores soluciones (solución de consenso).

Compact-GA Básico

Resultados

Son muy homogéneos los resultados obtenidos para el algoritmo *Compact-GA*, algoritmo que discurre sobre la plena simplicidad de la filosofía *PBIL*. En este caso se generan dos soluciones en cada iteración y se actualiza a partir de la mejor de ellas. No hay más. Es

esperable por tanto que este simple mecanismo conduzca a regiones de buenos resultados al igual que los algoritmos anteriores, pero que necesite de un pequeño empuje debido a las decisiones tomadas tan vertiginosas y con tan poca información.

En efecto, la homogeneidad supone el parecido completo entre mejores y peores soluciones, soluciones que mantienen la distancia con respecto al óptimo global para cada caso, quedando en regiones de cierta calidad a una media de 4 predicados para los casos 1, 2 y 3 y a una media de 6-7 predicados para el resto de los casos. En unos casos a las puertas de las regiones optimales y en otros en regiones de buenas soluciones pero a la espera de dar el salto definitivo a las regiones importantes de verdad.

El acercamiento para los diferentes casos es muy parecido al de los algoritmos anteriores, pero en este caso aumenta el radio de alcance al óptimo. El algoritmo puede desarrollarse plenamente en este caso al ser muy simple y poder trabajar durante un gran número de iteraciones. La reinicialización es utilizada en más de una ocasión y se explotan todas las capacidades de este simple algoritmo.

El tamaño del problema sigue teniendo poco que decir, pero parece ser que sí el número de predicados adicionales definidos, de tal forma que el algoritmo proporciona mejores soluciones cuanto menor es dicho número, aunque no queda esto manifiesto de forma muy clara.

Tenemos un algoritmo simple pero efectivo que puede expresar todas sus capacidades y que se puede ver como la raíz de los cuatro anteriores analizados.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 31).

Este algoritmo es, como sabemos, un algoritmo muy simple, que juega con la filosofía general de *Evolución de Poblaciones mediante Aprendizaje Incremental*. Si esto se manifestaba en los resultados, así mismo lo va a hacer en la convergencia.

Dado que cada iteración es muy simple, el algoritmo discurrirá a lo largo de un número considerable de iteraciones. Pero, a lo largo de dicho número de iteraciones tiene muy poca capacidad de movimiento o ascensión en la convergencia. De esta forma, comenzando por soluciones de cierta calidad rápidamente asciende hasta el estancamiento (5-8 iteraciones de movimiento); permanece en dicho estancamiento sin posibilidad de movimiento hasta la reinicialización, momento a partir del cual, tras seguir en el estancamiento mientras se alcanzan nuevas regiones a inspeccionar, se vuelve a ascender durante un cierto periodo algo mayor, para quedar finalmente estancado sin posibilidades de salida (ahora más difícil) la mitad de iteraciones que el algoritmo finalmente se ejecuta.

Al final quedamos con soluciones de calidad, pero debido a la simplicidad del algoritmo, la evolución se da de manera más retardada y bajo el continuo estancamiento. La escalera es ahora más espaciada, con menos escalones identificativos y grandes subidas zigzagueantes puntuales. Tenemos la esencia pero sin estar previamente cargada de ornamento para la evolución, lo que pasa factura en la convergencia.

Compact-GA básico (caso 8 - ejec. 1)

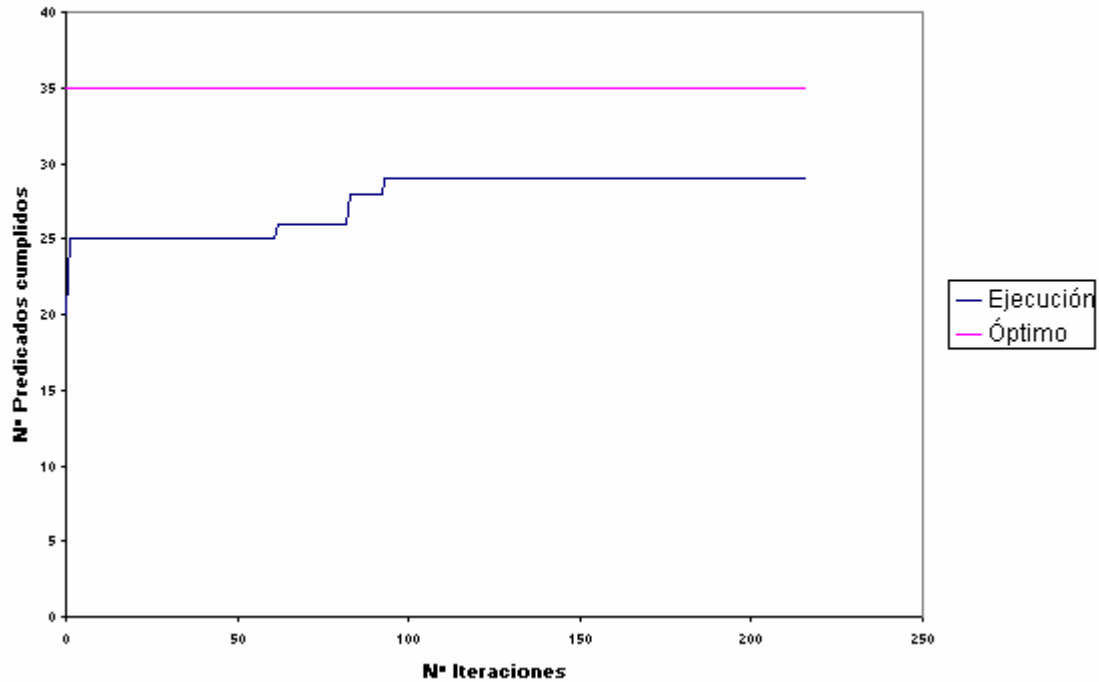


Figura 31.: Convergencia para una ejecución típica del algoritmo Compact-GA básico.

Compact-GA Extendido

Resultados

Proseguimos con otro algoritmo simple, pero esta vez con un pequeño cambio: generación en cada iteración de un determinado número de soluciones y actualización con la mejor ponderando su importancia con la comparación con el resto de soluciones. Esta pequeña elaboración con respecto al anterior algoritmo sigue manteniendo las soluciones dentro de la homogeneidad y de manera bastante palpable, como lo demuestra la igualdad en los resultados de las tres ejecuciones para dos de los casos del problema (casos 1, 6 y 8).

Los efectos que tiene el comentado cambio con respecto a las soluciones suponen un pequeño avance. Seguimos manteniendo la distancia con respecto al óptimo con buenas soluciones para algunos casos que no acaban de cuajar dentro de la calidad esperada de una solución lo suficientemente satisfactoria. Pero en lo que se refiere a otros, se obtienen soluciones propias de las regiones optimales.

Tenemos soluciones excelentes para los casos 2 y 8 (quedándonos a tan solo dos predicados del óptimo) y resultados de calidad para los casos 1 y 3. Para el resto se dista de 6 a 8 predicados del óptimo. Aparece un mayor acercamiento hacia soluciones de calidad para ciertos casos del problema, manteniendo la línea establecida por el algoritmo anterior en otros e incluso retrasándola un poco.

Más que establecer una dependencia manifiesta con respecto a número de predicados adicionales definidos o tamaño del problema hemos de indicar una cierta habilidad para abordar ciertos espacios de búsqueda, que no es tan efectiva para otros.

Se ha dispuesto un cambio que dirigido a ciertos casos supone mejora y que visto desde el punto de vista de otros ningún efecto o incluso retraso.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 32).

El algoritmo *Compact-GA extendido* se acerca al escalonado perfecto de los *PBIL básicos*, aunque la ascensión que lleva a cabo por cada escalón le lleva cada vez a un estancamiento mayor, que queda con la culminación de su ejecución en un callejón sin salida. Se alcanzan resultados de calidad, pero sin la actitud de penetrar en las regiones óptimas quedándose el algoritmo estancado a las puertas de las mismas.

No existe una finalización del algoritmo con expectativas a la subida, como ocurría en otras ocasiones. En vistas de la gráfica de convergencia, en la que el estancamiento crece, prolongándose el último hasta el final habiendo permanecido vigente más de la mitad de la ejecución, quedan pocas esperanzas de una posterior ascensión si el tiempo de ejecución se prolongase.

Seguimos partiendo de una buena solución como salida, pero con pocas ascensiones y sólo una de ellas muy significativa es muy difícil plantarse en el objetivo final de la ejecución de nuestro algoritmo. Tenemos calidad pero falta un pequeño retoque para el culmen.

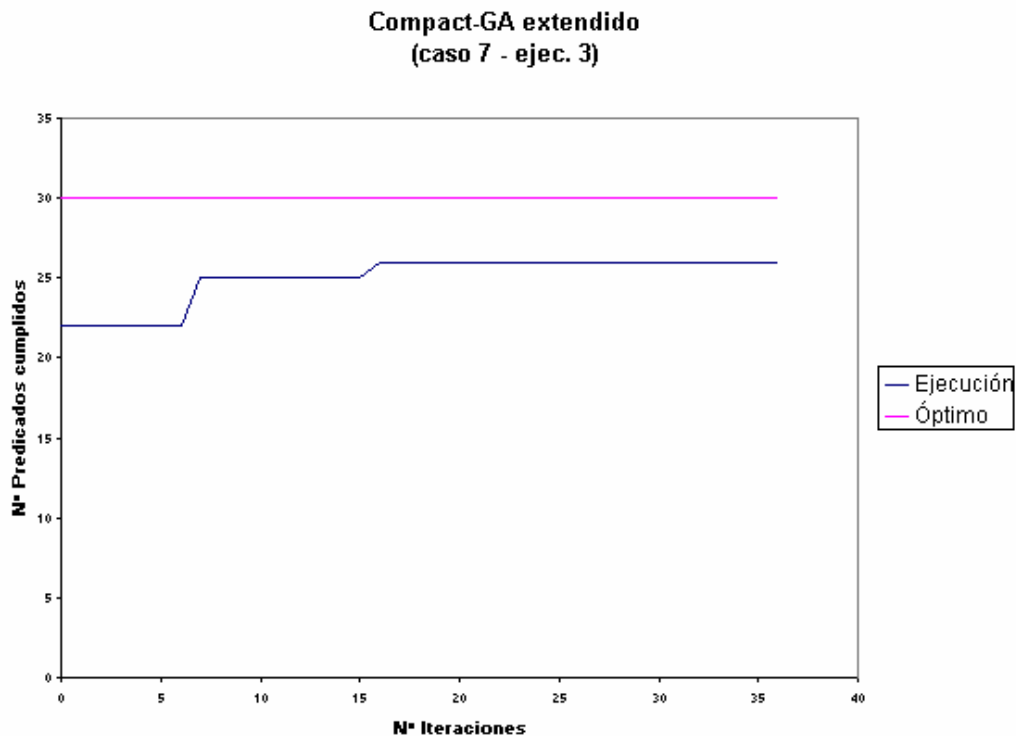


Figura 32.: Convergencia para una ejecución típica del algoritmo Compact-GA extendido.

Comparación de algoritmos de Evolución de Poblaciones mediante Aprendizaje Incremental

Los algoritmos de *Evolución de Poblaciones mediante Aprendizaje Incremental* constituyen dos grandes grupos tanto en concepto como en evolución y resultados: el primero

está constituido por los cuatro algoritmos *PBIL* básicos, mientras que el segundo de ellos lo conforman los algoritmos de *Compact-GA*. Todos ellos han proporcionado unos resultados de homogeneidad manifiesta y dentro de la calidad, pero existen grandes diferencias entre estos dos grupos.

El grupo de algoritmos englobados bajo el nombre de *Compact-GA* se caracteriza por proporcionar unos resultados de calidad pero en general asociados a regiones limítrofes a la optimal para cada caso. Profundizan dentro de la región optimal para ciertos casos del problema, pero mantienen las distancias para el resto. Al ser muy homogéneos incluso más que los del otro grupo, siempre vamos a tener garantía con su ejecución de obtener resultados de cierta bondad más o menos adecuada. De este grupo destaca el algoritmo *Compact-GA extendido* que se acerca más que su homólogo más simple a las zonas más prometedoras del espacio de búsqueda para un mayor número de casos del problema. El algoritmo *Compact-GA básico* mantiene más las distancias y profundiza para un número muy reducido de casos.

El otro grupo, el constituido por los algoritmos *PBIL* básicos, se caracteriza por su asalto a las regiones optimales para la mayoría de los casos del problema y por quedarse a las puertas para un número insignificante de los mismos (dos, en concreto). Aparecen un conjunto de algoritmos que proporcionan soluciones excelentes y que son capaces de desarrollar todo su potencial al igual que los del grupo comentado anteriormente, pero más elaborados en cuanto a la actualización del vector de probabilidad y la generación de soluciones en base a éste. Así, ante una homogeneidad un poco menor que los algoritmos del grupo anterior, son capaces de profundizar por completo en la región optimal de los casos del problema al menos para una de sus ejecuciones. Logran alcanzar tres óptimos (casos 1, 2 y 3) y dejan al alcance otros tres: uno de ellos a un solo predicado (caso 5) y otros dos a dos predicados (casos 7 y 8). Los casos restantes quedan a tan solo 4 predicados.

Dentro del grupo de algoritmos *PBIL* se pueden incluso establecer dos subgrupos en cuanto a resultados proporcionados. El grupo que proporciona los mejores resultados está constituido por dos algoritmos: el que actualiza a partir de la mejor y la peor solución y el que actualiza a partir de las *M* mejores soluciones por igual. Ambos algoritmos se disponen en cinco ocasiones dentro de las mejores soluciones para los distintos casos del problema que tratamos. En lo que se refiere al otro grupo hemos de decir que consta de los otros dos algoritmos. De esos dos algoritmos el que proporciona mejores soluciones es el que se actualiza a partir del mejor, sin menospreciar las soluciones que permite obtener el otro.

Tenemos por tanto que las peores soluciones, globalmente hablando, las proporcionan los algoritmos *Compact-GA*, mientras que las mejores soluciones, también globalmente hablando, las proporcionan los algoritmos *PBIL* básicos. Esto no es infalible, pues existe trasiego para ciertos casos del problema, como lo demuestra que para el caso 7 el algoritmo *Compact-GA extendido* proporcione una de las mejores soluciones o que para el caso 8 sea el algoritmo *PBIL* con actualización a partir de solución de consenso, el que proporcione la peor solución.

Los algoritmos se complementan unos con otros, sobre todo los *PBIL básicos*, para proporcionar una solución excelente a prácticamente todos los casos del problema, quedando para un número insignificante de ellos a las puertas de la optimalidad. Sus homólogos sencillos, los *Compact-GA* hacen todo lo posible dentro de las capacidades tan simples que presentan. Todo ello independientemente de número de predicados adicionales definidos e independientemente del tamaño del caso del problema; el espacio de búsqueda definirá las posibilidades, siendo aquellos que más se resisten los correspondientes a los casos 4 y 6.

4.2.3. Optimización a partir de Colonias de Hormigas

Sistema de Hormigas (SH)

Resultados

Los resultados obtenidos para el *Sistema de Hormigas* son muy homogéneos, a excepción de una de las ejecuciones para cada uno de los casos de mayor tamaño del problema. Dentro de dicha homogeneidad, los resultados se mantienen guardando las distancias con respecto al óptimo. El algoritmo presenta poca capacidad de profundización, lo que supone la no penetración significativa en zonas atractivas del espacio de búsqueda. Todas las ejecuciones se mantienen en una misma línea, sin el empuje definitivo que proporcione el acercamiento al interior de cada una de las regiones descubiertas. Existe muy poca diferencia entre el mejor y el peor de los resultados para todos y cada uno de los casos, de tal forma que, los peores resultados obtenidos para cada caso entran dentro de un radio de cierta calidad y garantías.

Tanto el tamaño como el número de predicados adicionales definidos para cada caso del problema no influyen en los resultados obtenidos, pues éstos se mantienen en una misma distancia prácticamente al óptimo, con ligeras variaciones tanto para casos grandes como pequeños, dándose bastante el caso de que se obtienen mejores resultados para casos de más tamaño y mayor número de predicados definidos y peores resultados para casos de menor tamaño y menor número de predicados adicionales definidos. Todo ello tiene lugar en la línea de homogeneidad ya comentada.

Se dispone de diversificación que permite abordar la búsqueda con garantías, pero se echa de menos el componente de intensificación que permita profundizar en las diferentes zonas localizadas. También, dado que el tiempo disponible para la ejecución es muy corto, el número de iteraciones alcanzado marca así mismo esta pauta y no existe por tanto tiempo suficiente para explotar la comunicación existente entre las hormigas proporcionada por la matriz, en este caso vector, de feromona. Tampoco el número de iteraciones permite la posibilidad de un número considerable de reinicializaciones que permitan abordar de otra forma el espacio de búsqueda. Es esperable la disposición de mejores resultados a un mayor tiempo por parte del *Sistema de Hormigas*.

Las limitaciones de tiempo junto a la necesidad de un componente de intensificación hacen al *Sistema de Hormigas* ser una aportación apropiada pero necesitada de complementos que la empujen hacia la auténtica calidad y la alejen de permanecer en la orilla de las regiones atractivas. La oportunidad de un mayor tiempo de ejecución no la tratamos aquí y hemos de pasar sin ella.

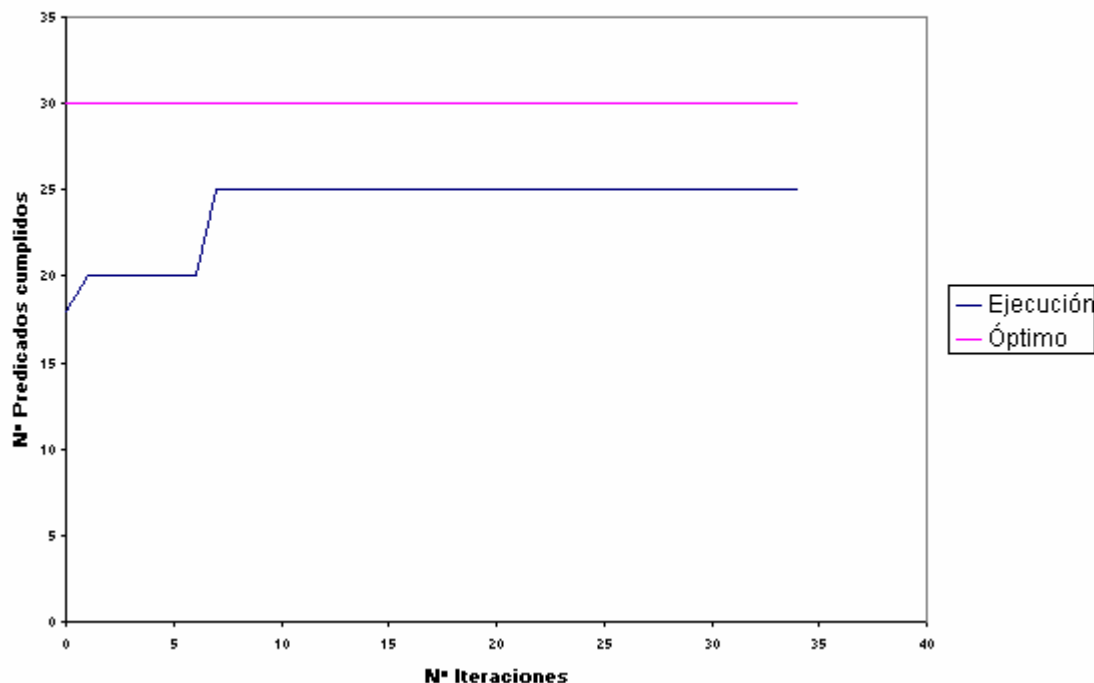
Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico de una ejecución para un caso concreto (figura 33).

El *Sistema de Hormigas* discurre durante un número de iteraciones que se establece como insignificante para el desarrollo de su pleno potencial: reinicialización y utilización de vector de feromona como patrón de comunicación. Así, tras un inicio que pronto se dispone en la zona de buenas soluciones, se estanca hasta el final sin posibilidad de salida.

El algoritmo queda en una solución a las puertas de la región optimal pero sin capacidad de reacción a corto plazo ante una región donde se extienden las dificultades de escapar. La

Sistema de Hormigas (caso 7 - ejec. 1)



reacción rápida ni la profundización entran dentro de la filosofía de este algoritmo que se detiene en la periferia.

Figura 33.: Convergencia para una ejecución típica del algoritmo del Sistema de Hormigas.

Sistema de Hormigas con Búsqueda Local (SHBL)

Resultados

Estamos ante un algoritmo réplica del anterior, pero que intenta incluir un componente de intensificación adicional proporcionado por una *Búsqueda Local* para cada hormiga antes de la actualización de feromona. Dicha *Búsqueda Local* no es en ningún caso exhaustiva, sino que tiene las mismas características que la utilizada en los algoritmos anteriores: generación del 5% de vecinos en cada iteración.

Los resultados esperables en principio supondrían la mejora del algoritmo anterior. Pero, sin embargo, nos mantenemos en una misma línea. La homogeneidad es la misma y las mejoras en ciertos resultados suponen el empeoramiento en otros, siendo las diferencias con los anteriores de apenas un solo predicado y en ocasiones de dos. No tenemos el aporte que necesitábamos aquí, pues el tiempo que se pierde en intensificar por cada iteración, supone un menor aprovechamiento de la comunicación, al discurrir el algoritmo un menor número de iteraciones.

Las características de los resultados obtenidos obedecen a lo ya comentado para el anterior algoritmo, con unas insignificantes variaciones, obteniéndose incluso peores resultados. Seguimos pensando en la necesidad de disponer de un mayor tiempo para que pueda fraguar el

algoritmo, en este caso tiempo necesario tanto para comunicación como para intensificación a partir de búsqueda local.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico para una ejecución sobre un caso concreto (figura 34).

Cuando intentamos refinar el algoritmo del *Sistema de Hormigas* introduciendo una mayor intensificación se produce la reducción del número de iteraciones que discurre el algoritmo y perdemos parte de la eficacia ganada por el cambio por la disminución del tiempo en el que las hormigas se desplazan por el espacio de búsqueda.

No se dispone de tiempo suficiente para que la acción de las hormigas y su comunicación pueda llegar a resultados concluyentes en un posible estancamiento. Menos todavía existe posibilidad de explotar la posibilidad de las reinicializaciones que permita profundizar en las regiones prometedoras y explorar otras que lo puedan ser.

Hemos de contentarnos con progresar con la generación de soluciones por parte de las hormigas y la pequeña intensificación posterior sobre las mismas. Esto supone partir de buenas soluciones en un inicio e inmediatamente subir a internarse dentro de las regiones de calidad. Queda por supuesto la convergencia del algoritmo en un sentido de posible subida posterior a la profundización sobre las regiones óptimas. Se necesita un poco de más tiempo para ver en acción toda la potencialidad del algoritmo.

Con lo obtenido en las 7 iteraciones iniciales se pueden prever resultados dentro de las regiones óptimas, pero sólo eso, sin información suficiente del discurrir del algoritmo en su ejecución. Aunque hemos de tener en cuenta que los resultados corresponden a buenas soluciones.

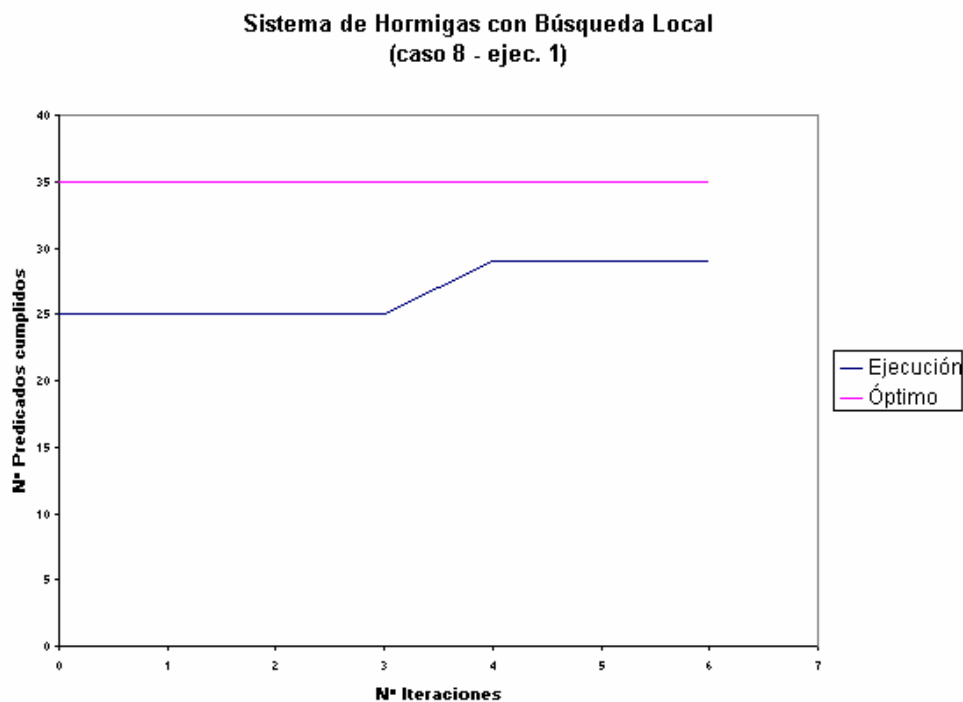


Figura 34.: Convergencia para una ejecución típica del algoritmo del Sistema de Hormigas con Búsqueda Local para un caso del problema.

Sistema de Hormigas de la Mejor y la Peor Hormiga (SHBW)

Resultados

Los resultados para el *Sistema de Hormigas de la Mejor y la Peor Hormiga* siguen entrando dentro de una gran homogeneidad salvo insignificantes excepciones, como ocurre para la tercera ejecución del caso 8. Dentro de esa gran homogeneidad suben el listón de buenos resultados y proporcionan un ápice de calidad en sus peores soluciones, que ciertamente mejoran para las mejores soluciones acercándose considerablemente al óptimo en muchos casos.

Los nuevos componentes introducidos por el *Sistema de Hormigas de la Mejor y la Peor Hormiga* proporcionan su resultado, pero todavía manteniendo las distancias con respecto al óptimo en ciertos casos del problema. Seguimos necesitando de tiempo para que el algoritmo desarrolle todo su potencial. No se llegan a cuajar resultados espléndidos, nos quedamos a las puertas.

Los resultados obtenidos no tienen ningún tipo de dependencia con respecto al número de predicados adicionales definidos ni con respecto al tamaño del caso del problema. La forma del espacio de búsqueda tiene aquí la última palabra.

No podemos utilizar la búsqueda local exhaustiva ni tampoco aprovechar en demasía la reinicialización ni la capacidad de comunicación a partir del vector de feromona. El discurrir del algoritmo permite mayor profundización, pero se le resisten una serie de espacios de búsqueda, tales como los proporcionados por los casos 4, 5 o 6. Son muy pocas iteraciones las que discurre el algoritmo: el beneficio esperable debería venir a más largo plazo. Se ha progresado con respecto a los dos algoritmos anteriores, y prueba de ello lo tenemos en buenos resultados para los casos 7 y 8 de gran tamaño y 1 y 2 de tamaño más pequeño. Sin embargo, todavía queda frente de avance.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico para una ejecución sobre un caso concreto (figura 35).

Aparece aquí lo que ocurría en el algoritmo anterior, pero sin embargo nos encontramos ante un algoritmo diferente, un algoritmo que dispone un mayor número de componentes de evolución para las hormigas y que refina por tanto los básicos. La convergencia supone una ascensión vertiginosa hacia las regiones óptimas sin prácticamente detención.

Se acrecientan aquí las esperanzas aún más de alcanzar el óptimo, pues con 6 iteraciones quedamos a las puertas de las regiones óptimas y con la consiguiente falta de explotación de la potencialidad de los distintos mecanismos tanto externos como internos del algoritmo.

Partiendo de buenas soluciones convergemos a las regiones óptimas sin estancamiento. Los resultados son buenos, pero la limitación de tiempo evita el desarrollo completo del algoritmo.

Sistema de Hormigas de la Mejor y la Peor Hormiga (caso 8 - ejec. 1)

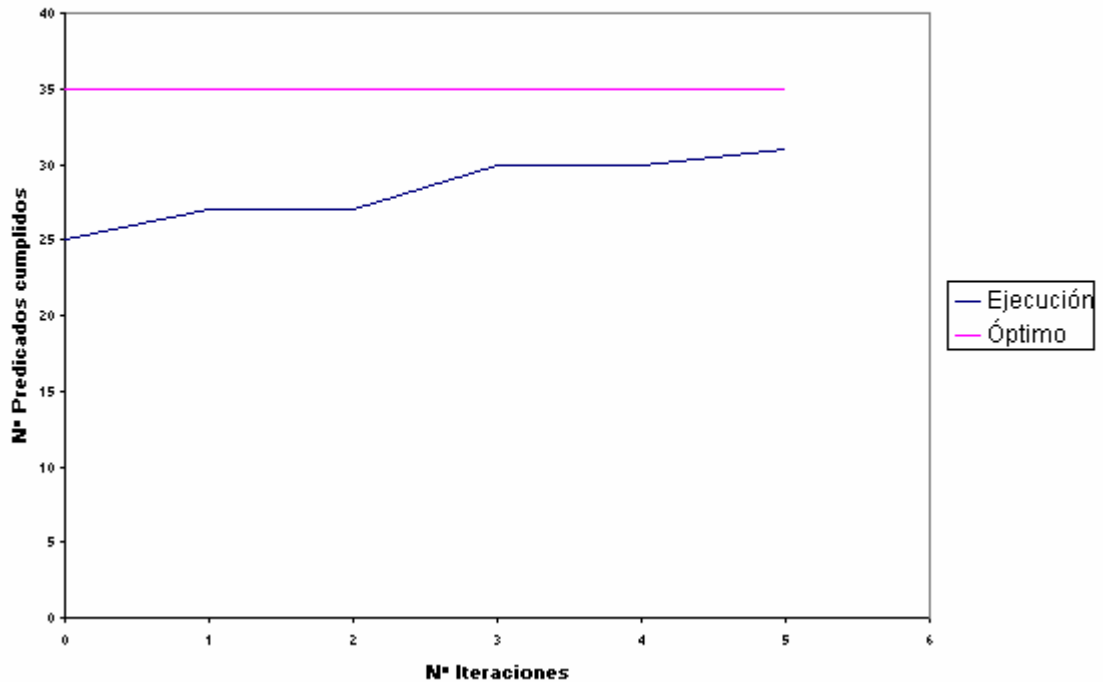


Figura 35.: Convergencia para una ejecución típica del algoritmo del Sistema de Hormigas de la Mejor y la Peor Hormiga para un caso del problema.

Sistema de Hormigas Basado en Rangos (SHRB)

Resultados

El *Sistema de Hormigas Basado en Rangos* proporciona así mismo resultados muy homogéneos, con sus correspondientes pequeñas excepciones, como ha ocurrido para el resto de algoritmos. Entramos nuevamente en la dimensión de la calidad pero guardando la distancia para ciertos casos del problema. Somos capaces de profundizar lo suficiente en algunos de los espacios de búsqueda y necesitamos un empuje para otros, pues los resultados del algoritmo quedan a la orilla de las regiones atractivas.

Se obtienen excelentes resultados para los casos del problema de menor tamaño: el 1 (en el que se obtiene el óptimo) y el 2 y el 5 en el que nos quedamos a dos predicados de obtenerlo. A estos siguió un buen resultado para el caso 7 y unos aceptables en la periferia de las regiones atractivas para el resto de casos. Ha ocurrido con respecto a los mejores resultados obtenidos lo mismo que para el algoritmo anterior, pero siendo los buenos resultados mucho mejores para el algoritmo que ahora tratamos.

Con respecto a los peores resultados que se obtienen hay que resaltar su presencia dentro de un radio de calidad, de manera que todos se establecen a partir de 20-21 predicados cumplidos para los casos de menor tamaño y de 24-25 para los de mayor tamaño. Son unos peores resultados que suponen garantías a la hora de la ejecución del algoritmo sobre los distintos casos.

El tamaño del problema para este algoritmo determina la bondad de los resultados obtenidos, de tal forma que aparecen mejores resultados para casos del problema de menor tamaño y va descendiendo la diferencia con respecto al óptimo a medida que asciende el tamaño de los casos del problema. En lo que se refiere al número de predicados adicionales definidos

ocurre lo mismo que para el tamaño, pero de forma menos acentuada, existiendo un pequeño mayor número de excepciones (caso 2 y 7).

Existen espacios de búsqueda que requieren de una mayor profundización, pero donde se ha llegado un poco más lejos que en el algoritmo anterior. La principal causa puede ser el tiempo utilizado para la ejecución del algoritmo que ni siquiera permite la convergencia total del mismo, y por tanto ni una posible reinicialización ni la total explotación de sus posibilidades. Todas las características del algoritmo que suponen dejar discurrir el algoritmo hacia una estabilización no son prácticamente aplicables y el resto conducen en la medida de lo posible hacia buenas regiones, consiguiendo penetrar en algunas áreas atractivas y quedándose a la orilla de otras. El espacio de búsqueda ante esta situación tiene la última palabra.

Tenemos buenos resultados teniendo en cuenta las características dirigidas a medio-largo plazo de la ejecución de nuestro algoritmo y la entidad de los casos a resolver.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico para una ejecución sobre un caso concreto (figura 36).

Para el *Sistema de Hormigas basado en Rangos* se recorta aún más el número de iteraciones durante el que se desarrolla la ejecución del algoritmo. Pero, sin embargo la ascensión es mucho más vertiginosa que para el caso anterior, introduciéndose el algoritmo en el núcleo de las regiones óptimas del espacio de búsqueda con solo 5 iteraciones.

La convergencia del algoritmo acaba en ascenso, al igual también que el algoritmo anterior y con las esperanzas intactas de alcanzar el óptimo, pero en una posterioridad donde se puedan utilizar en todo su auge las componentes del algoritmo con un mayor tiempo disponible.

Cada vez el despegue hacia el óptimo se hace más manifiesto, pero parece que tenemos poco tiempo para mucho algoritmo.

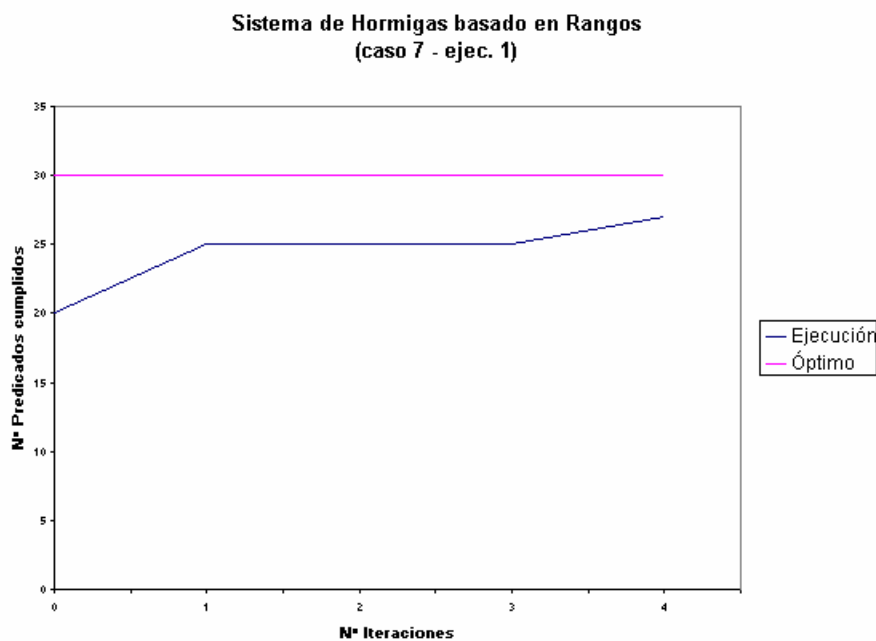


Figura 36.: Convergencia para una ejecución típica del algoritmo del Sistema de Hormigas basado en Rangos para un caso del problema.

Sistema de Hormigas MAX-MIN (SHMM)

Resultados

Para el *Sistema de Hormigas MAX-MIN* la homogeneidad la podemos establecer como completa y sin excepciones, puesto que la máxima diferencia en predicados cumplidos por parte de la solución final que existe entre el peor resultado y el mejor resultado para cada caso del problema es de 4. En medio de esa homogeneidad, los resultados guardan la distancia con respecto al óptimo sin llegar a profundización notoria. Los mejores resultados se establecen a partir de 4 predicados de diferencia con respecto al óptimo (con la excepción del caso 7 en el que el mejor resultado tiene una diferencia de 3 predicados con respecto al óptimo). Los peores resultados se mantienen en esa misma línea con un pequeño desplazamiento y pueden considerarse por tanto de cierta calidad, pudiéndose hablar de garantías de un peor resultado posible bueno en la ejecución del algoritmo.

Si se tiene en cuenta el tamaño del problema, parece que para los cuatro primeros casos a medida que aumenta el tamaño del problema, mayor distancia existe con respecto al óptimo en el resultado; sin embargo, para el resto de casos no está tan clara dicha relación. Así mismo, en lo que se refiere al número de predicados adicionales definidos, hemos de hablar de la no existencia de relación manifiesta alguna con respecto a los resultados.

Hemos podido comprobar que los resultados suponen una intensificación no lo suficientemente cuajada por parte del algoritmo. Hay que tener en cuenta el número de iteraciones que el algoritmo es capaz de desarrollar en el poco tiempo dispuesto para cada experimento. El algoritmo discurre durante un insignificante número de iteraciones, de tal forma que, prácticamente sólo se limita a generar soluciones y utilizar en una primera instancia la memoria de comunicación (vector de feromona). El método de reinicialización basado en la lista élite, método muy elaborado, no tiene oportunidad en algunos casos y prácticamente alguna en otros. No hay suficiente tiempo para llegar al estancamiento y cambiar la dirección del algoritmo. Se necesita tiempo para que el algoritmo entre dentro del caso del problema a tratar. Junto a esto, el único componente interior de intensificación, constituido por la búsqueda local, tiene las mismas características de pequeña intensificación que en el resto de algoritmos ya analizados.

Tenemos ante nosotros un algoritmo de un enorme potencial, pero que la práctica no deja desarrollarse por completo. Al no explotar lo suficiente sus características, nos vemos obligados a permanecer en la periferia de las regiones de calidad sin ser capaces de introducirnos en las mismas para su exploración.

Convergencia

Para poder comprobar la convergencia de este algoritmo, veremos un ejemplo típico para una ejecución sobre un caso concreto (figura 37).

Nuevamente nos encontramos con el mismo comportamiento del algoritmo aunque en este caso el ascenso es un poco menos pronunciado. Seguimos con un número de ejecuciones bastante pequeño: 6 ejecuciones. Partimos de buenas soluciones y llegamos a la puerta de las regiones optimales finalizando en ascenso y haciendo entrever posibilidades futuras que pudieran conducirnos al óptimo.

Hemos tratado con cuatro algoritmos con una convergencia prácticamente idéntica, más pronunciada en unos casos que en otros, pero muy similar. Dicha convergencia supone el acercamiento a buenas soluciones, pero la no finalización de la entrada en las regiones optimales del espacio de búsqueda. En ningún caso explotamos completamente todos los mecanismos asociados al algoritmo, pues materialmente no existe tiempo para que todas las

estructuras cuajen con información de la que extraer conclusiones importantes de cara al avance sobre el espacio de búsqueda.

En un comportamiento similar desde el punto de vista de la convergencia, los algoritmos se distinguen entre ellos por las soluciones proporcionadas en poco tiempo, pero siempre dentro de la calidad.

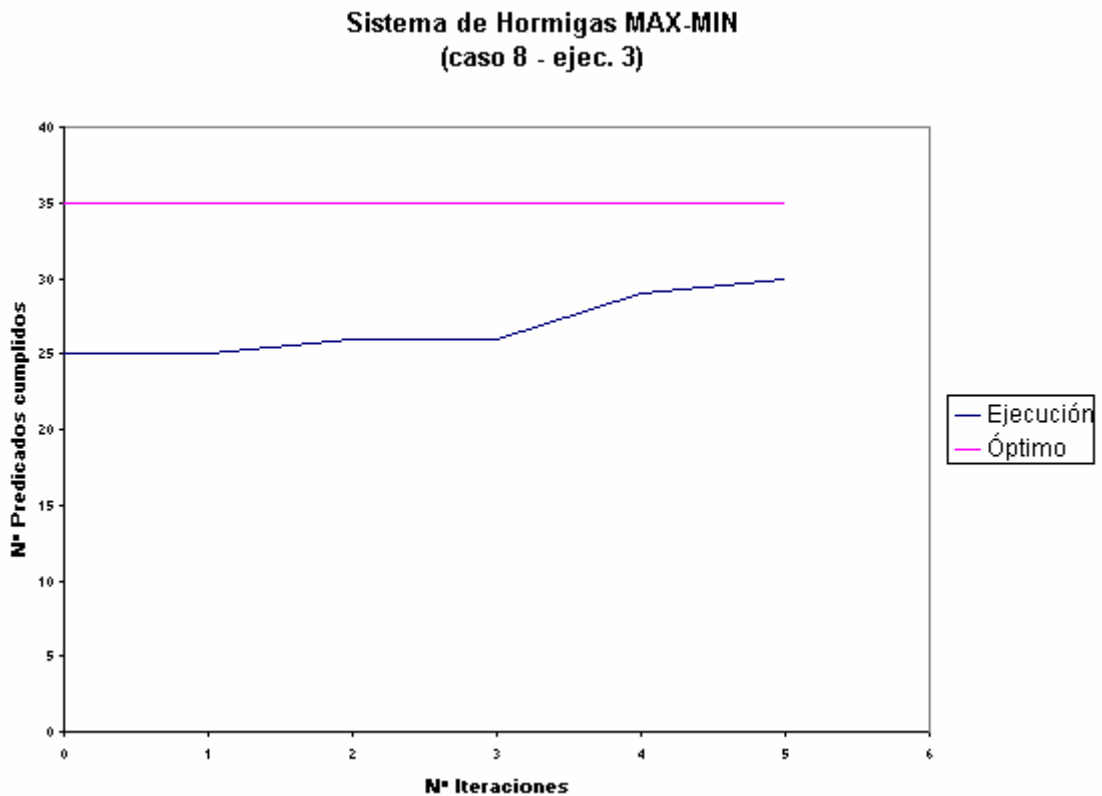


Figura 37.: Convergencia para una ejecución típica del algoritmo del Sistema de Hormigas MAX-MIN para un caso del problema.

Comparación de algoritmos de Optimización a partir de Colonias de Hormigas

Los resultados obtenidos por parte de los diferentes algoritmos de *Optimización a partir de Colonias de Hormigas* para todos los casos del problema suelen ser bastante homogéneos para cada caso como ya hemos ido indicando para cada uno de ellos. Dentro de dicha homogeneidad tenemos los peores resultados por parte del *Sistema de Hormigas* y el *Sistema de Hormigas con Búsqueda Local*, a los que sigue en el camino hacia la obtención de buenos resultados el escalón constituido por los otros tres algoritmos. La homogeneidad tiene su excepción en el prácticamente menos homogéneo de todos: el *Sistema de Hormigas de la Mejor y la Peor Hormiga*, que proporciona para uno de los casos el mejor y el peor resultado en solitario.

No tenemos ciertamente un algoritmo dominador entre todos los analizados. Cada algoritmo tiene más facultades para abordar un espacio de búsqueda que otros e incluso coinciden en los mejores resultados dos o más para la mitad de los casos del problema. Ni un algoritmo produce todos los mejores resultados ni un algoritmo produce todos los peores resultados: todos los algoritmos se encuentran presentes en los peores resultados globales alguna vez y casi todos excepto el *Sistema de Hormigas con Búsqueda Local* se encuentran presentes alguna vez en los mejores resultados globales. Depende del caso del problema que el

algoritmo sea capaz de rendir con nivel en unas ocasiones o se quede bastante rezagado en otras ocasiones.

Lo único que queda claro es la existencia de dos grandes escalones en los resultados de los algoritmos, pero no se puede establecer una distinción clara entre los algoritmos que componen éstos. Parece ser que para el primero de ellos: el constituido por el *Sistema de Hormigas* y el *Sistema de Hormigas con Búsqueda Local* (es el algoritmo que está más presente en los peores resultados), el primero de los mismos se adelanta mínimamente con respecto al otro, pero manteniéndose la mejor disponibilidad de uno de los algoritmos para ciertos casos y del otro para otros.

Con respecto al segundo escalón, constituido por el resto de algoritmos, hay que decir que se hace así mismo difícil observar un orden, pues también se establecen las mayores capacidades por parte de unos algoritmos para abordar unos casos que quedan rezagadas para otros casos en los que otros algoritmos son los que triunfan. Podemos hablar en media y, como algoritmo más presente en los mejores resultados y menos en los peores, del *Sistema de Hormigas basado en Rangos*. Tras este se establece el *Sistema de Hormigas de la Mejor y la Peor Hormiga*. Finalmente queda un poco más cerca del primer escalón, el *Sistema de Hormigas MAX-MIN*. Sin embargo, no hay que hablar de una diferencia muy manifiesta, sino ligeramente perceptible.

Lo que queda claro es que, los resultados globales, aún siendo muy buenos para ciertos casos del problema: casos 1, 2 y 5, y buenos para otros de gran tamaño (casos 7 y 8), necesitan un ápice de intensificación que supone el quedar al borde de las áreas de calidad para los otros cuatro casos del problema. Para los resultados globales también se manifiestan las posibilidades de abordar de manera satisfactoria algunos casos del problema y no tanto otros, faltando un pequeño empuje. Se hace necesario para ciertos espacios de búsqueda un mayor tiempo para el desarrollo de todas las capacidades de cada algoritmo. Los algoritmos de *Optimización a partir de Colonias de Hormigas* son algoritmos que proporcionan resultados a medio-largo plazo, por lo que en una fase prematura puede que no hayan alcanzado lo deseado como aquí ocurre. Todos los algoritmos para todos los casos invierten un número muy pequeño de iteraciones, en las que apenas se pueden servir de la comunicación (vector de feromona) y donde la reinicialización se puede establecer como un factor olvidado, dado que apenas se da oportunidad al estancamiento. Así mismo, la búsqueda local en los algoritmos que se aplica, no tiene posibilidad de exhaustividad, generando como sabemos una insignificante parte del vecindario. Se echa por tanto de menos un mayor tiempo para el desarrollo completo de las facultades de los algoritmos, pero en conjunto todos ellos, los unos por los otros, obtienen buenos resultados para todos los casos y de mucha calidad para buena parte de ellos.

Los resultados no parecen tener dependencia manifiesta del tamaño del caso del problema, pues se obtienen mejores y peores resultados para problemas de tamaño grande y lo mismo para problemas de tamaño más pequeño. Hay que tener en cuenta que la diferencia de tamaño no es muy grande entre los distintos casos. En lo que se refiere al número de predicados adicionales definidos, hay que decir que tenemos buenos resultados tanto para un número grande como pequeño de predicados, pero manteniendo las distancias para los casos en los que hay un número intermedio de predicados adicionales definidos. Todo esto hace pensar que es el espacio de búsqueda el que, ante el tiempo disponible, define las posibilidades de los algoritmos de abordarlo y proporcionar mejores resultados.

Los algoritmos de *Optimización a partir de Colonias de Hormigas* resultan una buena alternativa para abordar los casos del problema, pudiendo seleccionar en cada caso el más adecuado, sin ser ninguno de ellos infalible y a falta de mayor espacio en el que desarrollar todo su potencial.

4.2.4. Comparación de Métodos basados en poblaciones

Dentro de los *Métodos basados en Población* se establecen desde el punto de vista teórico y conceptual tres grandes grupos: *Algoritmos Genéticos*, *Algoritmos de Evolución de Poblaciones Mediante Aprendizaje Incremental* y *Algoritmos de Optimización a partir de Colonias de Hormigas*. Sin embargo, desde el punto de vista práctico, teniendo en cuenta los resultados de los experimentos realizados, los grupos que quedan constatados son diferentes. Así, en cuanto a los resultados de los experimentos, se establecen dos grandes grupos: el conformado por todos los *Algoritmos Genéticos* excepto el de *Shared Continuously Updated* y todos los algoritmos *PBIL básicos*, y el conformado por los algoritmos de *Optimización a partir de Colonias de Hormigas*, los *Compact-GA* y el *Shared Continuously Updated*.

El primero de los grupos se caracteriza por una mayor homogeneidad general de resultados, algo más favorable para los *Algoritmos Genéticos*. Así mismo, todos los resultados de los algoritmos que lo componen siguen una misma línea de calidad muy poco descompensada para cada caso concreto, pero muy variable a lo largo de los distintos casos del problema, no habiendo un único dominador nato, en cuanto a mejores resultados se refiere. Los resultados son excelentes para los distintos algoritmos del grupo y entre unos y otros se ocupan de copar las mejores soluciones obtenidas, salvo una pequeña excepción protagonizada por el algoritmo del *Sistema de Hormigas basado en Rangos* y para un caso del problema pequeño (el caso 1).

Las mejores soluciones obtenidas para cuatro de los casos del problema no tienen un único algoritmo protagonista, sino varios, manifestándose de forma más significativa para los casos 1 y 4 en los que tenemos 5 y 6 algoritmos respectivamente copando la mejor solución. Tenemos aquí una prueba fundamental para afirmar la gran competitividad existente entre los algoritmos que se incluyen dentro del primer grupo que resulta del estudio experimental. Además, el número de veces de todos los algoritmos dentro de la mejor solución es prácticamente coincidente.

Las soluciones obtenidas por este grupo son extraordinarias, alcanzándose el óptimo para cuatro de los casos del problema, los correspondientes a los espacios de búsqueda que, según los experimentos, han resultado más abordables: 1, 2, 3 y 7. Seguimos con resultados excelentes para el caso 5, en el que quedamos a un solo predicado del óptimo. El caso 8 queda a dos predicados y el 6 a tres cerrando la gran y exitosa penetración dentro de la región optimal para cada uno de los casos del problema. Prácticamente no existe influencia manifiesta ni dependencia directa por parte del número de predicados definidos y el tamaño del problema. Como vemos, cabe muy poca mejoría dentro de estos algoritmos para superar lo casi insuperable.

En cuanto a la media, dentro de este grupo podemos destacar a los *Algoritmos Genéticos no clásicos*: *AG clearing*, *AG multi-crowding* y *AG nichos elitista con sharing* y a los *PBIL con actualización a partir del mejor y el peor* y *PBIL con actualización a partir de todas las M soluciones equitativamente*.

Tenemos en el grupo de algoritmos comentado a una alternativa definitiva a la hora de abordar los diferentes casos de nuestro problema de forma rápida y con soluciones de calidad de la región optimal. Aunque, eso sí, hemos de tener en cuenta la no infalibilidad de ningún algoritmo, pero sí la regularidad de cada uno de ellos y la garantía de soluciones de calidad contrastada.

El segundo de los grupos está compuesto por algoritmos de menor homogeneidad, aunque sin mucha diferencia con respecto al otro grupo. Los algoritmos que lo componen presentan una menor regularidad en lo que se refiere a la disposición de soluciones de calidad. Este grupo lo constituyen por una parte aquellos algoritmos elaborados con una actitud

vinculada a una evolución a más largo plazo, como son los algoritmos de *Optimización a partir de Colonias de Hormigas* y el *Algoritmo Genético Multimodal de Proporción Continuamente Actualizada*; y por la otra algoritmos simples establecidos bajo el marco de la *Evolución de Poblaciones Mediante Aprendizaje Incremental*: los *Compact-GA*. Los primeros necesitan de mayor tiempo de experimentación para desarrollar todo su potencial, caracterizándose por una ejecución con un número de iteraciones muy bajo. Los segundos, dado el poco entramado de manejo de información que presentan, no tienen más posibilidad que la proporcionada por un mecanismo muy simple de evolución.

De todos los algoritmos, quedan más rezagados en cuanto a peores resultados obtenidos y media el *Sistema de Hormigas* y el *Sistema de Hormigas con Búsqueda Local*. El resto de algoritmos se reparten una bondad mucho menos significativa que la del primer grupo de algoritmos ya analizado, pero que se acerca a ellos para los espacios de búsqueda más abordables del problema. En la comparación de los resultados de ese conjunto de algoritmos restantes para una misma medida y un mismo caso del problema hay que resaltar la existencia de una mínima diferencia que en su máxima expresión no supera en media los 4 predicados.

Tratamos con un conjunto de algoritmos, los de este segundo grupo, en los que hay severos problemas de estancamiento, para unos casos más que para otros (los ya conocidos de espacios de búsqueda menos abordables). A pesar de partir desde muy buenas soluciones en las iteraciones iniciales, no existe una convergencia continuada, fraguándose ésta en una continua lentitud ascendente sin estancamiento pleno.

El proporcionar soluciones rápidas de gran calidad les es muy difícil y ponen su esperanza en la posteridad que no llega dado que el tiempo de experimentación es bastante corto.

Los *Métodos basados en Población* presentan un buen conjunto de excelentes alternativas para abordar el problema que tratamos. No tenemos en ningún caso la panacea, pero sí que nos movemos en una gran homogeneidad y regularidad para la mayor parte de los algoritmos. Las soluciones rápidas invitan a ciertos algoritmos a proporcionar mejores resultados, pero las soluciones de las iteraciones iniciales son coincidentes para la práctica totalidad de los algoritmos, estableciéndose la diferencia fundamental en la convergencia escalonada rápida para el primer grupo de algoritmos comentados y escalonada suave para el segundo. Los algoritmos más clásicos y con menor infraestructura proporcionan los peores resultados dentro de cada grupo, y así mismo los de evolución a mayor largo plazo teniendo en cuenta el tiempo para la experimentación en cada ejecución sobre cada caso. Tenemos para todos los algoritmos espacios más abordables que otros, reduciéndose la diferencia para el primer grupo y contrastándose para el segundo, independientemente del tamaño y del número de predicados adicionales definidos para cada caso.

Se alcanzan soluciones magníficas y las peores soluciones entran dentro de un radio de cierta calidad sin llegar en ningún momento a grandes carencias. Hay de todo y para todos, sin descartar ninguna alternativa dadas las características de los propios experimentos y de los resultados obtenidos. Resalta un gran grupo de algoritmos a tener en cuenta y otros que quedan en un término medio que puede suponer interesante y que no se debe en ningún momento descartar.

4.3. Comparación entre todos los algoritmos (resumen y conclusiones de todos los resultados obtenidos)

Los resultados para los algoritmos asociados a las distintas *Metaheurísticas* sobre los que se han realizado experimentos han dejado entrever la multitud de posibilidades y soluciones para todos los gustos que se encierran dentro del gran abanico de conceptos y filosofías diferentes que abarca el marco de las *Metaheurísticas*. Se ha hecho un recorrido por todos los

posibles resultados, las distintas dificultades que se presentan gracias a la multiplicidad de espacios de búsqueda y la potencialidad de los diferentes algoritmos demostrada en su regularidad o media y en su capacidad de llegar al óptimo.

Los resultados son de bastante calidad para la mitad de los algoritmos y van disminuyendo en su calidad para el resto hasta llegar a la mera aceptabilidad. Los *Métodos basados en Poblaciones* establecen su dominio sobre los *Métodos de Trayectoria*, copando con sus algoritmos las diez primeras posiciones en la regularidad de resultados para la media y las doce primeras en la regularidad del mejor. Así mismo, las últimas cuatro y cinco posiciones respectivamente, son ocupadas por otra parte por *Métodos de Trayectoria*. Los mejores resultados globales obtenidos para cada caso inscriben a gran cantidad de *Métodos basados en Población*, mientras que sólo aparece en ellos un solo *Método de Trayectoria* (la *Búsqueda por Vecindarios Cambiantes*).

La gran decepción la encontramos en el *Enfriamiento Simulado* cuyos algoritmos proporcionan los peores resultados con diferencia, tanto en media como en mejor obtenido. Algo mejor lo hace la *Búsqueda Tabú* y la *Búsqueda Local Iterativa*, pero quedando en una manifiesta carencia de calidad dentro de los más rezagados tanto en media como en mejor obtenido. Quedan mejor algoritmos más simples y clásicos como la *Ascensión de Colinas* o la *Búsqueda Local Múltiple*; en la media de resultados quedan por encima de la secuencia de algoritmos más rezagados, pero en el mejor obtenido escalan puestos y se sitúan cerca de la mitad de la tabla, algo que realiza de forma más significativa la *Ascensión de Colinas*. Nos podemos dar cuenta como un primer vistazo a la tabla deja entrever el hundimiento de los *Métodos de Trayectoria* y la superación de algoritmos complejos y elaborados por parte de los algoritmos simples y clásicos.

Los algoritmos que abandonan la tónica seguida por los distintos *Métodos de Trayectoria* son la *Búsqueda Local Iterativa basada en Poblaciones* y la *Búsqueda basada en Vecindarios Cambiantes*. El primero de los algoritmos tiene un cierto carácter poblacional, y se podría haber incluido dentro de los *Métodos basados en Poblaciones*, y el segundo de ellos es un método natural de trayectoria. Estos dos algoritmos se sitúan prácticamente en la mitad de la tabla. Aparecen de la mitad de la tabla hacia abajo con la *Búsqueda basada en Vecindarios Cambiantes* proporcionando los mejores resultados. Sus resultados han de ser tenidos en cuenta dentro del marco de calidad necesario para nuestro problema. Hay que añadir y notar también la presencia del algoritmo de *Búsqueda basado en Vecindarios Cambiantes* dentro de dos de las mejores soluciones globales obtenidas: casos 5 y 6, obteniendo para el primero de ellos el óptimo.

Los *Métodos de Población* comienzan desde abajo con una representación discreta por parte del *Sistema de Hormigas*, el *Sistema de Hormigas con Búsqueda Local* y el *Algoritmo Genético Multimodal de Continuously Updated Sharing*. Estos tres algoritmos se sitúan en la parte baja de la tabla tanto para la media como para el mejor. Tienen por debajo a los *Métodos de Trayectoria* del *Enfriamiento Simulado*, *Búsqueda Local Iterativa* y *Búsqueda Tabú*. Se intercalan con algoritmos muy simples como son los de la *Búsqueda Local* y *Búsqueda Local Múltiple*. Se trata de algoritmos de trabajo a largo plazo que han sido incapaces de explotar todas sus posibilidades, capacidades, mecanismos y potencialidad en el tiempo establecido para la experimentación. El más prometedor de todos ellos tanto en media como en mejor resultado es el *Algoritmo Genético*.

El siguiente peldaño de la escalera lo constituyen para las dos clasificaciones los algoritmos *Compact-GA*. Se trata de dos algoritmos con una filosofía muy simple pero potente a la hora de proporcionar buenos resultados. El más regular de los dos es el *Compact-GA extendido*, pero viendo los mejores resultados que permiten extraer, los dos son muy similares.

Un poco más arriba, ya en la mitad de la tabla tanto de resultado medio como de mejores resultados, encontramos a los algoritmos más elaborados del *Sistema de Hormigas* que conviven junto a los mejores algoritmos de los *Métodos de Trayectoria*. De ellos los resultados más discretos los proporciona el *Sistema de Hormigas MAX-MIN*. Resultados un poco mejores son los asociados al *Sistema de Hormigas de la Mejor y la Peor Hormiga*, situados más entorno a los dos algoritmos mencionados que se encuentran dentro de los *Métodos de Trayectoria*. El algoritmo que entra dentro de la zona de verdadera calidad, a las puertas para la media de resultados y dentro para los mejores resultados es el *Sistema de Hormigas basado en Rangos*.

Llegamos tras abandonar los últimos algoritmos de *Optimización a partir de Colonias de Hormigas* a las primeras posiciones, copadas todas ellas por los *PBIL básicos* y los *Algoritmos Genéticos* a excepción del *Continuously Updated Sharing*. Dentro de los algoritmos dominadores de los *Métodos de Población* resaltan los más elaborados y menos clásicos: así, para los *PBIL* tenemos al *PBIL de actualización a partir del mejor y el peor* y el *PBIL de actualización a partir de las M mejores soluciones equitativamente*, y para los *Algoritmos Genéticos* tenemos al de *Nichos elitista con sharing*, *Clearing* y *Multitud con Multinichos*. El algoritmo que queda más descolgado de todos los que copan los primeros puestos es el *PBIL* más básico: el que permite la actualización a partir del mejor, sobre todo para el mejor obtenido.

Los mejores resultados obtenidos para todos los casos del problema corresponden a este conjunto de algoritmos que copan las primeras posiciones. Todos ellos sin excepción se incluyen dentro de estos mejores resultados, interviniendo en todos los casos del problema excepto en uno de ellos para el que el *VNS* proporciona la mejor solución posible. Así mismo, para varios de los casos del problema no es un único algoritmo el que alcanza la mejor solución, lo que deja entrever la gran competitividad existente entre este conjunto de algoritmos que proporcionan las soluciones de mayor calidad. Esos mejores resultados no quedan en mejores globales para todos los algoritmos con los que se experimenta sino que prácticamente son resultados óptimos: así, se consiguen 4 óptimos globales (casos 1, 2, 3 y 7) y quedamos en los otros tres a escasa distancia del óptimo en el corazón de la región óptima: tres predicados para los casos 4 y 6, y dos predicados de distancia para el caso 8. Son sólo estos tres espacios de búsqueda, los correspondientes a los casos 4, 6 y 8, los más difíciles de abordar por parte de la totalidad de los algoritmos, algo que impide obtener el óptimo para los mismos en el poco tiempo destinado para cada ejecución.

Dentro de la media, los mejores resultados globales son para el *PBIL mejor peor* seguido de cerca por su compañero de filosofía *PBIL M mejores equitativamente*. A continuación se sitúan los *Algoritmos Genéticos* más elaborados ya comentados quedando *CHC* y *Sharing* cerrando el grupo al que siguen los otros dos *PBIL básicos* y el *Algoritmo Genético básico*.

Dentro del mejor, los mejores resultados globales están más compartidos y cambia el orden anterior. Así, *PBIL mejor peor* continúa a la cabeza, pero esta vez compartiendo puesto con *AG nichos elitista con sharing*. El *CHC* se acerca aquí más a los primeros puestos, siguiendo a los dos primeros junto al *AG clearing*. Quedando un poco más rezagado aquí continuamos con el *PBIL M mejores equitativamente* y prácticamente en la línea anterior con *AG multi-crowding*. Con muy poca distancia entre ellos continúan *AG básico*, *PBIL M mejores consenso* y *AG sharing*, quedando ahora *PBIL mejor* un poco más rezagado, dejando por delante a *Sistema de Hormigas basado en Rangos* entre otros.

Hemos podido comprobar cómo se alcanzan excelentes resultados y cómo no se resiste ningún espacio de búsqueda, donde el número de predicados definidos y el tamaño del problema no determina de forma manifiesta el comportamiento de cada algoritmo. Sólo ciertos espacios de búsqueda quedan sin haber alcanzado el óptimo, pero con perspectivas de hacerlo teniendo en cuenta la convergencia de los mejores algoritmos. Tenemos un gran grupo de algoritmos de calidad que proporcionan una nueva perspectiva para abordar el problema que tratamos y que

ha sido representado mediante ocho casos distintos. Se ha podido comprobar cómo en poco tiempo se pueden alcanzar magníficas soluciones y cómo también algoritmos que en principio parecen tener más potencialidad teórica que otros, sucumben ante los distintos espacios de búsqueda. Está claro que no debemos descartar ningún algoritmo, pero los resultados aportan una valiosa información a la hora de la consideración de las *Metaheurísticas* en la resolución del problema de la selección de la solución deseada. Así mismo, ninguna *Metaheurística* es infalible y es la conjugación de las mismas la que permite abordar con garantías los diferentes espacios de búsqueda, sobre todo la fusión de los algoritmos del grupo comentado en un frente común.

5. Conclusiones Generales

Nuestro objetivo fundamental fue el de verificar los resultados de una aplicación práctica de las *Metaheurísticas* para resolver el problema de la selección de la solución deseada para una posible aplicación práctica real. De esta forma, partiendo del solver anteriormente comentado, hemos integrado los diferentes algoritmos en el mismo para abordar la solución a un conjunto significativo de casos planteados: un total de 8 casos de gran tamaño y con un número variable de restricciones adicionales definido. Para cada algoritmo, se han realizado un total de 3 ejecuciones sobre cada caso y cada ejecución ha tenido como condición de finalización el alcance de los 5 minutos (debemos verificar las capacidades de cada algoritmo para proporcionar buenas soluciones en un corto espacio de tiempo). En total tenemos 648 ejecuciones repartidas en 54 horas.

Los resultados de la experimentación, que pueden comprobarse en las secciones anteriores, junto a los análisis correspondientes para cada algoritmo, técnica, filosofía, métodos y global, dejan entrever las grandes posibilidades que presentan las *Metaheurísticas* a la hora de abordar el problema de la selección de la solución deseada. Pasamos a resumir las conclusiones principales que se extraen de tales resultados.

En general, los resultados obtenidos son excelentes. Se alcanza el óptimo para cinco de los ocho casos planteados y se queda en el corazón de la región optimal para los tres restantes. Existen resultados de toda índole, pero predomina la calidad. No existe ningún algoritmo infalible, pero sí ciertos algoritmos que en media se comportan mejor y que proporcionan las mejores soluciones para la mayoría de los casos del problema. El tamaño del problema y el número de predicados adicionales definidos para cada caso no tienen una actuación determinante en los resultados y pasan a un segundo plano, siendo el espacio de búsqueda el que tiene la última palabra en la evolución de los distintos algoritmos, como así lo demuestran aquellos casos para los que no se ha alcanzado el óptimo en su comparación con los que sí han permitido su alcance. Hay que tener en cuenta la no existencia de una diferencia muy amplia en el tamaño de los casos y en el número de predicados definidos.

Las *Metaheurísticas* que dan mejor resultado son las asociadas a los *Métodos basados en Población*, quedando los *Métodos de Trayectoria* con diferencia tras éstos tanto en lo que se refiere a mejores resultados obtenidos como en lo tocante a la regularidad o las garantías de soluciones de calidad.

Hay que resaltar dentro de los *Métodos de Trayectoria* la decepción que supone el *Enfriamiento Simulado* en todas sus variantes con los peores resultados globales, y las pocas aportaciones de *Búsqueda Local Iterativa Simple* y *Búsqueda Tabú* que apenas suponen poco más que el *Enfriamiento Simulado*. Se han de tener en cuenta así mismo los resultados considerables de la *Ascensión de Colinas* y la *Búsqueda Local Múltiple* que son capaces de superar a algoritmos mucho más elaborados. Aún así, los únicos que se salvan dentro de los *Métodos de Trayectoria* entrando ya dentro del radio de calidad tanto desde el punto de vista de mejores soluciones como de regularidad son la *Búsqueda Local Iterativa basada en Poblaciones* y la *Búsqueda basada en Vecindarios Cambiantes*.

Los *Métodos basados en Población* proporcionan muy buenos resultados que se convierten en excelentes para muchos de los algoritmos que se establecen dentro de las filosofías que los componen. Así, tenemos unos discretos resultados de los *Compact-GA*, los *Sistemas de Hormigas básico y con búsqueda local* y el *AG continuously updated sharing* que quedan a continuación de los *Métodos de Trayectoria* comentados, excepto los dos algoritmos de calidad: *VNS* e *ILS basada en Poblaciones*. Las causas de esos discretos resultados las encontramos en la simplicidad y poca capacidad de evolución para los *Compact-GA*, y en la capacidad de evolución a más largo plazo para los *Sistemas de Hormigas* y el *AG* indicado (se trata de muy poco tiempo de ejecución para cada caso). Pero, tras comenzar a abrir camino con los *Sistemas de Hormigas MAX-MIN, Mejor y Peor Hormiga y basado en Rangos*, algoritmos que todavía acusan la necesidad de una evolución a más largo plazo, se sitúa un grupo amplio de algoritmos en la cumbre de la calidad y capacidad para abordar nuestro problema: los *PBIL básicos* y todos los *Algoritmos Genéticos* a excepción del ya comentado. Llegamos aquí al óptimo para la práctica totalidad de los casos del problema, alcanzamos las mejores medias sea cual sea el espacio de búsqueda a abordar y existe una competitividad plena sin un claro dominador.

El *problema de la selección de la solución deseada* adopta una nueva dimensión de tal forma que, para un conjunto de casos significativos del problema, existe un conjunto amplio de algoritmos que proporciona una excelente solución en un tiempo que puede resultar interesante para la aplicación práctica: tan solo 5 minutos. Se abre por tanto la puerta a una futura profundización en una solución al problema que empieza a ser palpable desde el punto de vista de la realidad.

El punto de partida ha resultado satisfactorio. Se abren nuevos caminos hacia la resolución de casos más complejos y con nuevas dimensiones y hacia la utilización de nuevos algoritmos que integren las capacidades de aquellos que mejores aportaciones han supuesto en la resolución de nuestro problema, sin descartar por supuesto ninguno de ellos.

Referencias

- [YLB06] E.Yeguas, M.V. Luzón, E. Barreiro y R. Joan-Arinyo. Estudio e implementación de metaheurísticas para solucionar el problema de la selección de la solución deseada. Technical Report LSI-08-25-R. Departamento de Lenguajes y Sistemas Informáticos. Universitat Politècnica de Catalunya. Septiembre 2008.