© 2007 Thomson. Personal use of this material is permitted. Permission from Thomson must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Minería de datos aplicada a la modelización de un sistema híbrido

J.M. Cadenas, M.C. Garrido, E. Muñoz, E. Serrano

Dpto. de Ingeniería de la Información y las Comunicaciones

Universidad de Murcia, 30007 Murcia.

jcadenas@um.es, mgarrido@dif.um.es, enriquemuba@dif.um.es, emilioserra@gmail.com

Resumen

Cuando se propone diseñar un sistema híbrido centralizado de metaheurísticas para resolver problemas de optimización combinatoria, uno de los problemas que surge es la modelización del agente coordinador del sistema de cara a que haga lo más efectiva posible la coordinación entre las distintas metaheurísticas. Una de las formas de poder atacar este problema es mediante la extracción del conocimiento que se pueda obtener de distintas ejecuciones previas de las metaheurísticas a la hora de resolver instancias del problema o problemas planteados.

Para comprobar la validez de esta aproximación se han construido dos sistemas híbridos basados en un modelo fuzzy para resolver el problema de la mochila. El primer sistema coordina dos metaheurísticas, un algoritmo genético y una búsqueda tabú. El segundo añade a las dos anteriores el temple simulado, con objeto de comprobar la robustez del sistema y la capacidad de obtener soluciones de más alta calidad. Se muestra la aplicación del proceso así como los resultados obtenidos por los prototipos comparándolos con los obtenidos por las metaheurísticas individuales.

Palabras clave: Minería de Datos, Sistema Cooperativo, Metaheurísticas.

1. Introducción

La minería de datos se define en [4] como el proceso de extraer conocimiento nuevo, útil y comprensible de datos almacenados en diferentes formatos. En otras palabras, la tarea fundamental de la minería de datos es la de encontrar modelos inteligibles, empezando desde los datos. Esto hace de la extracción inteligente de conocimiento una herramienta muy potente a la hora de obtener el modelo de un sistema híbrido cooperativo centralizado de metaheurísticas para resolver problemas de optimización.

En los problemas de optimización, generalmente, buscamos la mejor solución (o una suficientemente buena) a un problema entre muchas alternativas. Esto puede parecer trivial, sin embargo estos problemas pueden crecer a grandes escalas, lo que hace que se requiera de algoritmos computacionales para abordarlos. Por ello para resolver estos problemas surgen diversas estrategias, de las cuales unas de las más destacadas son las metaheurísticas, que se pueden definir como [9]:

> *Estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento*

Las metaheurísticas nos proporcionan estrategias efectivas para encontrar soluciones aproximadas a problemas de optimización. Al trabajar con ellas, sin embargo, no es extraño encontrarnos con el problema de la selección del algoritmo, [11]. Este problema consiste en determinar qué algoritmo se debe seleccionar para resolver una instancia maximizando una medida de rendimiento. El enfoque más común consiste en medir el rendimiento de un conjunto de algoritmos sobre un conjunto de instancias, y utilizar aquel que presente el mejor. Sin embargo, existe la posibilidad de inducir un sistema que sugiera un algoritmo, o conjunto de algoritmos, a partir de la evidencia teórico-experimental conseguida de la ejecu-

ción de diferentes algoritmos sobre conjuntos de instancias con diferentes características.

Por tanto, habría que esperar que una combinación inteligente de diferentes metaheurísticas proporcionara enfoques más eficientes y mayor flexibilidad cuando abordáramos estos problemas, obteniendose robustez por el uso de diferentes metaheurísticas que cooperan entre sí, alcanzando soluciones de muy alta calidad para diferentes clases de instancias, [6].

En este tipo de sistema cooperativo, surge el problema de la descripción del coordinador, de manera que controle y modifique el comportamiento de las metaheurísticos de forma eficiente y que suavice los problemas planteados anteriormente relacionados con el comportamiento de los algoritmos metaheurísticos de forma individual. Como se indicó, la minería de datos se presenta como una opción interesante para dar solución a este problema. Por ello en [1, 2] se presentó un proceso de extracción de conocimiento, con el que obtener un modelo del coordinador basado en reglas fuzzy. Y en [3] se mostraron los resultados obtenidos en cada fase y el coordinador que finalmente se obtuvo para el problema de la mochila.

En este trabajo se presenta una visión metodológica de la aplicación de este proceso de extracción de conocimiento a la construcción de dos prototipos del sistema anteriormente obtenido y los resultados que se obtuvieron con su aplicación. En la sección 2 se explica como es el sistema que se pretendía construir, en la sección 3 se explica la aplicación del proceso de extracción del conocimiento desglosado en las distintas fases que lo componen, preparación de los datos, minería de datos y evaluación del modelo, en la que se mostrarán y compararán los resultados obtenidos por los dos prototipos. Para terminar en la sección 4 se mostrarán las conclusiones y los posibles trabajos futuros.

2. El sistema metaheurístico cooperativo

Existen diversas maneras de realizar la ejecución cooperativa de las distintas metaheurísticas. La que utilizaremos consiste en emplear un sistema multiagente, en el que cada metaheurística es un agente que tiene por objetivo resolver el problema mientras se coordina con el resto (Fig. 1). Una aproximación para realizar la coordinación es el uso de un agente coordinador que controle y modifique el comportamiento de los agentes. El coordinador por lo tanto tendrá dos tareas fundamentales: recopilar la información del rendimiento de cada uno de los algoritmos metaheurísticos y enviarles órdenes que afecten a sus comportamientos de búsqueda, [6]. Para la comunicación de las metaheurísticas con el coordinador se usará un modelo de pizarra adaptado, en el que cada metaheurística va dejando en su espacio reservado las soluciones del problema que va encontrando y el agente coordinador va consultado la pizarra para monitorizar el comportamiento de las metaheurísticas y tomar decisiones acerca de como modificar su comportamiento.

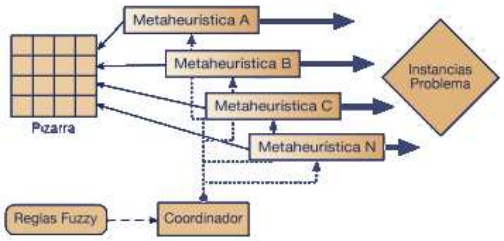

Figura 1: Sistema metaheurístico multiagente

El problema que surge inmediatamente es como modelar este agente coordinador. Proponemos que la inteligencia del coordinador esté condensada en un conjunto de reglas fuzzy (reglas de tipo Mamdani, [8]). Las reglas fuzzy dotarán al coordinador de inteligencia y esta a su vez surgirá como resultado de un proceso de extracción inteligente del conocimiento [1, 2], el cual se muestra en la Fig. 2.

Este proceso comienza en la fase de preparación de los datos donde queremos obtener una base de datos que contenga la información útil para aplicarle técnicas de minería de datos. A continuación viene la fase de minería de datos en la que se espera obtener el modelo del coordinador del sistema metaheurístico utilizando las técnicas apropiadas. Y se finaliza con la fa-

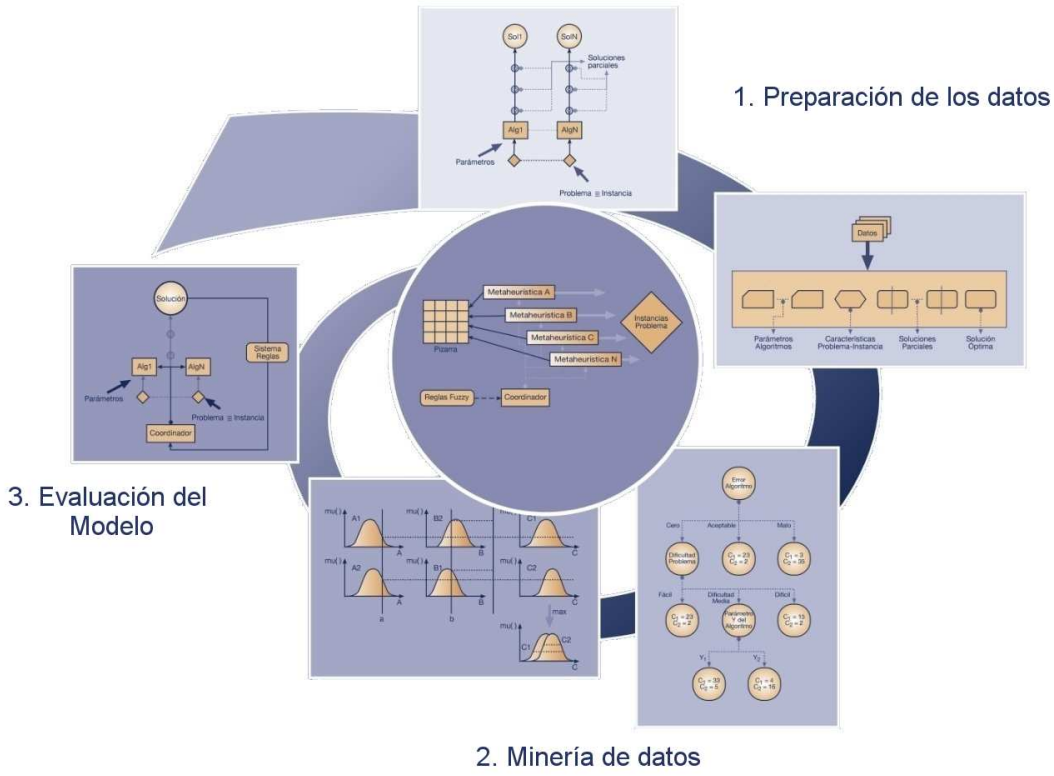

Figura 2: Proceso de extracción del conocimiento

se de evaluación del modelo donde se quiere comprobar la eficacia del conjunto de reglas fuzzy que modelan el coordinador.

3. Aplicación del proceso de extracción del conocimiento

A continuación se mostrará cómo se ha aplicado cada una de las fases del proceso de extracción del conocimiento y los resultados que se han obtenido durante la construcción de dos prototipos del sistema híbrido de metaheurísticas para resolver el problema de la mochila, utilizando para el primero (a partir de ahora Coor2Met) dos metaheurísticas, un algoritmo genético y una búsqueda tabú y para el segundo (a partir de ahora Coor3Met) tres metaheurísticas, las dos anteriores y un temple simulado.

3.1. Preparación de los datos

En la fase de preparación de los datos queremos obtener una base de datos que contenga la información útil a la cual podamos aplicar las técnicas de minería de datos apropiadas para obtener el modelo. Para ello, el primer paso es la recopilación de información. Nuestras fuentes de información provienen de la aplicación de distintos algoritmos metaheurísticos a conjuntos de instancias de un problema.

El primer paso dado en esta fase fue la selección del problema a resolver, escogiendose el problema de la mochila, ya que por su sencillez nos permite experimentar con el proceso antes de aplicarlo a problemas más complejos. De este problema se usó una base de instancias

compuesta por 4000 instancias del problema las cuales varían tanto en tamaño (número de objetos) como en la forma en la que han sido generadas.

Como siguiente decisión se escogieron las metaheurísticas que se utilizarían, pretendiéndose tanto que hubieran metaheurísticas basadas en trayectorias, como basadas en poblaciones, puesto que esta clasificación [5] es muy usada y queríamos ver como el coordinador podría manipular cada tipo. Por ello se seleccionaron una búsqueda tabú y un temple simulado como metaheurísticas basadas en trayectorias y un algoritmo genético para las basadas en poblaciones.

Una vez elegido el problema y las metaheurísticas que lo resolverían se pasó a recopilar la información que se utilizaría para obtener la definición del coordinador. De tal forma que se aplicaron las distintas metaheurísticas elegidas a la base de instancias, extrayéndose de estas ejecuciones cierta información interesante con la que construir la base de datos. Para ello se seleccionaron de cada instancia las ejecuciones de cada metaheurística que mejor resultado obtuvieron y se fusionaron en un vector, de tal manera que un ejemplo está compuesto por los siguientes atributos:

- Una descripción de la instancia del problema que incluye: nombre de la instancia, peso máximo de la mochila, beneficio óptimo, número de objetos disponibles, tipo de la instacia y una etiqueta lingüística que indique la dificultad de la instancia.
- La configuración de parámetros de cada metaheurística que mejor resultado dio.
- Información obtenida durante las ejecuciones de cada metaheurística que incluye: beneficio de la solución inicial, dos soluciones intermedias y la final, así como la proporción de error con respecto al óptimo de la solución final.

Esta base de datos contenía toda la información de las ejecuciones, pero fue necesario realizar un preproceso de tal forma que se seleccionaran aquellos atributos e instancias más relevantes teniendo como objetivos:

1. Realizar un estudio comparativo del comportamiento de las distintas metaheurísticas implementadas. De tal forma que se pudiera determinar:
   - Cómo se comportaba cada metaheurística a lo largo del tiempo con respecto al resto de metaheurísticas.
   - Cuál era el comportamiento de cada metaheurística con cada tipo de instancia.
   - Una mezcla de las dos anteriores.
2. Realizar un estudio de los parámetros de cada metaheurística de tal forma que pudieramos observar:
   - Qué combinaciones de parámetros funcionaban mejor.
   - Qué conjunto de parámetros era más adecuado para cada tipo de instancia.
   - Qué conjuntos de parámetros podían sustituir a los que se consideran los mejores.

Para realizar este preproceso principalmente se hizo uso del conocimiento que se posee del problema apoyado por técnicas automáticas como las que ofrece la herramienta Weka [12]. De esta manera se prepararon subconjuntos de la base de datos con los objetivos antes mencionados.

De entre las técnicas que ofrece Weka para seleccionar atributos se utilizo principalmente la técnica supervisada AtributeSelection que es un filtro que puede ser usado para seleccionar atributos y subconjuntos de estos. Es bastante flexible y permite varios métodos de búsqueda de subconjuntos y de evaluación de atributos y subconjuntos de estos que pueden combinarse. En la aplicación de esta técnica se hizo uso de diversas combinaciones tanto de métodos de búsqueda como de evaluación. Construyéndose distintas bases de datos según cada una de ellas.

En cuanto a técnicas de selección de instancias se hizo uso de herramientas de submuestreo aleatorio.

Tras todo este proceso ya se disponía de diversas bases de datos sobre las que aplicar la minería de datos.

3.2. Fase de minería de datos

Selección de la técnica de minería de datos

Para poder aplicar esta fase el primer paso necesario era escoger una técnica de minería de datos. Existe una amplia variedad de técnicas de minería de datos, basadas en diferentes propuestas teóricas. Dado que las observaciones imperfectas aparecen de forma inevitable en dominios y situaciones realistas y que a priori no sabíamos los tipos de datos de partida con los que trabajaríamos, nos centramos en aquellas técnicas que permiten en la actualidad un mayor y más completo tratamiento de la información imperfecta.

Se han realizado algunos esfuerzos para incorporar incertidumbre e imprecisión en las fases de aprendizaje e inferencia de técnicas de minería de datos muy conocidas. Algunas técnicas que permiten el tratamiento de ejemplos con atributos desconocidos son las técnicas de construcción de árboles de regresión/clasificación. Además, en las técnicas de construcción de árboles también se permiten observaciones de atributos nominales que presentan valores inciertos desde el punto de vista probabilista, así como atributos continuos expresados mediante valores crisp. También se incorpora el tratamiento de atributos expresados mediante valores difusos. En general, podemos decir que una de las técnicas que en la actualidad realiza un tratamiento más completo de información imperfecta es la de árboles de decisión.

Los árboles además destacan por su sencillez, legibilidad e interpretabilidad. Y esto fue precisamente lo que finalmente nos llevó a utilizar esta técnica de minería (nos interesaban modelos lo más claro posibles para poder comprenderlos).

Una vez se decidió que se usaría una técnica de árbol de decisión, el siguiente paso fue decidir que técnica de árboles sería más adecuada. Existen diferentes alternativas disponibles, de las cuales se consideraron CART, ID3, C4.5 y FID3.4. Las dos primeras (CART e ID3) fueron descartadas puesto que no incorporan tratamiento de datos imperfectos. C4.5 aparece como respuesta a esto, ya que generaliza el algoritmo básico de construcción de árboles ID3 para tratar datos tanto nominales como continuos y que además permite el tratamiento de datos con un cierto tipo de imperfección. Entre estos tipos de imperfección sin embargo no se encuentran los conjuntos fuzzy (etiquetas lingüísticas). En este sentido, FID3.4, [7], mezcla la representación fuzzy y sus capacidades de razonamiento aproximado con la generación de árboles de decisión simbólicos uniendo las ventajas de ambos: el manejo de datos imperfectos junto con el alto grado de popularidad, comprensibilidad y fácil aplicación de esta técnica.

FID 3.4

FID3.4 permite trabajar con atributos nominales y continuos, siendo los dominios de los atributos continuos particionados en un conjunto de etiquetas lingüísticas. Esta técnica permite el tratamiento de los siguientes tipos de valores:

- Valores crisp (nominales y continuos).
- Valores missing (nominales y continuos).
- Valores expresados mediante distribuciones de probabilidad (sólo nominales).
- Valores expresados mediante un intervalo crisp (sólo para atributos continuos).
- Valores expresados mediante un conjunto fuzzy.

FID 3.4 modifica el algoritmo básico ID3 tanto en la construcción del árbol como en la inferencia, para incorporar la representación fuzzy y el razonamiento aproximado. En ID3, al estar basado en conjuntos clásicos, un ejemplo satisface solamente una de las condiciones de salida de un nodo y por lo tanto cae exactamente en un subnodo y consecuentemente en una sola hoja. En cambio en FID3.4 un ejemplo puede cumplir más de una condición,

donde ahora las condiciones son restricciones fuzzy basadas en conjuntos fuzzy. Por esto, un ejemplo puede pertenecer a más de un hijo de un nodo y por lo tanto puede pertenecer o alcanzar más de un nodo hoja con algún grado [0,1].

El procedimiento de construcción del árbol, [7], es el de ID3, salvo que ahora la información de los atributos se evalúa usando conjuntos fuzzy, funciones de pertenencia y razonamiento aproximado. Esto significa, desde el punto de vista práctico, que el número de ejemplos que caen en un nodo (que satisfacen las condiciones que conducen a ese nodo) está basado en los grados de satisfacción de los conjuntos fuzzy apropiados por parte de las características individuales de esos ejemplos. Para alcanzar una hoja se deben de satisfacer un número determinado de restricciones fuzzy. Además, como el procedimiento de decisión de una simple hoja está descrito como la implicación "Si las restricciones fuzzy que conducen a la hoja se satisfacen entonces inferir la clasificación correspondiente de los datos de aprendizaje en dicha hoja", entonces el nivel de satisfacción anterior debe de ser pasado hasta esta implicación.

En el procedimiento de inferencia, [7], el primer paso es calcular el grado con el que un nuevo ejemplo satisface cada hoja del árbol. Como en los árboles fuzzy el número de hojas alcanzadas por un ejemplo se incrementa es necesario usar operadores para resolver posibles conflictos. Sin embargo, los conflictos no sólo aparecen entre distintas hojas sino que es frecuente que aparezcan en una hoja individual, ya que es frecuente que los ejemplos pertenecientes a una hoja dada pertenezcan a clases distintas. El otro nivel de conflicto aparece cuando se alcanzan distintas hojas del árbol. Esto surge debido a que los conjuntos fuzzy se solapan y a la aparición de valores desconocidos e incompletos. Por todo ello no podemos hablar de clasificar al nuevo ejemplo asignándole un valor del dominio de la clase sino más bien de asignar grados de pertenencias del ejemplo a las etiquetas lingüísticas correspondientes a la clase.

Aplicación de la técnica

Una vez elegida la técnica de minería de datos se aplicó a los distintos conjuntos de la base de datos previamente seleccionados en la fase de preparación de los datos. Se debe notar que FID3.4 sólo es capaz de capturar dependencias parciales, puesto que genera modelos para un atributo determinado, en lugar de capturar dependencias globales. Por ello para cada base de datos se infirieron todos los atributos que la componían de tal manera que se capturaran todas las dependencias.

Como resultado de la aplicación de FID3.4 sobre los distintos subconjuntos de la base de datos se obtuvieron varios conjuntos de árboles, de cuya interpretación se obtuvo un extenso conjunto de reglas compuesto por 37 reglas fuzzy, que hemos denominado de bajo nivel o directas. Para crear estas reglas previamente tuvieron que definirse distintos conjuntos fuzzy para los distintos valores que pueden tomar: las diferencias entre los beneficios obtenidos por cada metaheurística, el tiempo y los parámetros que puede tomar cada metaheurística.

Usando estos conjuntos fuzzy se crearon las reglas, ejemplos de las cuales pueden verse en la tabla 1, que modelaron los siguientes comportamientos:

- Cuándo y de qué manera se puede cambiar la solución de una metaheurística por la solución de otra metaheurística que se está comportando mejor.
- Con qué parámetros se debe iniciar cada metaheurística dependiendo del tipo de instancia que se desee resolver.
- Cuándo y cómo se pueden cambiar los parámetros de una metaheurística que no está obteniendo el comportamiento esperado.
- Qué regla debe seleccionarse en caso de que más de una se active.

Tabla 1: Ejemplos de reglas obtenidas

SI [Tiempo ES $Reinicio1$ Y Dificultad ES $Dificil$ Y p_gen_tem ES (gp O mgp)] ENTONCES Reiniciar temple con solución de genético.
SI [Tiempo ES $Inicio$] ENTONCES pcruce ES b
SI [p_gen_tab ES (pn O gn O mgn) Y Tiempo ES $cp6$ Y Dificultad ES $muyfacil$] ENTONCES pcruce ES a
.....

Modelo final obtenido

Las reglas obtenidas en el punto anterior son perfectamente válidas al venir directamente del conocimiento conseguido en la fase de minería de datos. No obstante, estas reglas tienen algunas carencias y pueden mermar la potencia del sistema, puesto que son demasiado numerosas y poco abstractas como para servir de modelo del coordinador. Por lo que se decidió crear un conjunto de reglas más general obtenido a partir de las reglas anteriores y del comportamiento que se espera del coordinador. De esta manera se obtuvieron dos reglas:

- SI [($peso_1 * d_1$ O ... O $peso_n * d_n$) ES $suficiente$] ENTONCES cambiar la solución actual de la peor metaheurística.
- SI la Metaheurística *esMuchoPeorQueTodas* Y *esMomentoDeCambiarParametros* ENTONCES *cambiarParametros* de Metaheurística.

La primera regla indica cómo se puede cambiar la posición en el espacio de búsqueda de la metaheurística que está obteniendo un "peor resultado" por una posición "más cercana" a otra metaheurística con "mejor comportamiento". De esta manera a cada metaheurística se le asigna un peso utilizando el conocimiento obtenido en la fase de minería de datos y este peso se multiplica por la diferencia entre la peor solución y la solución de la metaheurística. Si alguno de estos productos es "suficientemente grande" entonces se debe cambiar la solución de la metaheurística con peor resultado, por una cercana a la de la metaheurística o metaheurísticas que hicieron saltar la regla.

Por otro lado la segunda regla muestra como modificar los parámetros de las distintas metaheurísticas. De esta forma si una metaheurística obtiene unas soluciones con un "beneficio mucho menor" que las del resto y hace "bastante tiempo" desde que se le cambiaron los parámetros, entonces se le puede dar un nuevo conjunto de parámetros utilizando las reglas que se obtuvieron en la fase de minería de datos.

3.3. Evaluación

La última fase del proceso es la fase de evaluación del sistema. Para llevarla a cabo se implementaron los dos prototipos del sistema (Coor2Met y Coor3Met) de tal manera que pudiéramos comprobar la eficacia del modelo del coordinador y si éste realiza su trabajo de forma eficiente con respecto al coste computacional y a la bondad de las soluciones obtenidas. Además de esto también se propuso comprobar la robustez del sistema ante la posibilidad de añadirle una nueva metaheurística. Para ello se compararon los prototipos implementados con las metaheurísticas que los componían y entre ellos.

Prototipos implementados

Los dos prototipos del sistema fueron implementados como sistemas síncronos, es decir, todas las comunicaciones se realizaban en intervalos previamente especificados, escribiendo todos sus soluciones parciales en el mismo instante, y entrando en juego el coordinador en ese instante para comprobar que acciones debía realizar.

Para la implementación del coordinador se utilizó la herramienta de modelado de sistemas fuzzy XFuzzy 3.0 [10]. Con esta herramienta se modelaron las diferentes reglas y se obtuvieron los motores que finalmente fueron modificados ligeramente para adaptarlos a nuestros propósitos.

Experimentación

Para realizar las comparativas previstas en esta fase se utilizó una base de instancias que contenía 80 instancias del problema las cuales variaban tanto en tamaño (número de objetos) como en la forma en la que habían sido generadas. Además se decidió que cada uno de los sistemas coordinados empleara en la ejecución de cada instancia 100 segundos. De la misma manera cada metaheurística individual se ejecutó durante 100 segundos.

Para empezar mostraremos los resultados que se obtuvieron con Coor2Met, que coordinaba un algoritmo genético y una búsqueda tabú. La tabla 2 muestra la proporción de error obtenida en la resolución de los distintos tipos de instancia tanto por el sistema coordinado como por las metaheurísticas individuales. En la primera parte de la tabla se muestra la proporción de error media obtenida con cada tipo de instancia, y en la segunda con cada tamaño de instancia. En la Fig. 3 se muestran los gráficos correspondientes a estas tablas.

Tabla 2: Coo2Met vs. metaheuristicas indiv.

Tipo	Sist. c.	Alg. Gen.	B. Tabu
unc span	0,00506	0,08245	0,00930
wea span	0,00449	0,07497	0,00830
str span	0,00637	0,06492	0,01485
pceil	0,00070	0,05911	0,00095
circle	0,04068	0,09790	0,11748
Items			
500	0,00340	0,00406	0,02674
1000	0,01021	0,05689	0,03130
1500	0,01453	0,10405	0,03137
2000	0,01771	0,13848	0,03130

Como se puede ver el sistema coordinado siempre obtiene como mínimo soluciones iguales que las del mejor algoritmo independiente, y esto es importante puesto que era lo mínimo que se exigía al sistema. Pero también se puede observar como en la mayoría de casos la mejoría es sensible reduciendo en la mayoría el error a la mitad. Esta mejora sobre todo es importante en las instancias de tipo circle, que son las más difíciles de resolver y en las que se reduce el error de un 10 % a un 4 %.

Una vez vistos estos resultados pasamos a mostrar los resultados que se obtuvieron con Coor3Met, que coordina un algoritmo genético, una búsqueda tabú y un temple simulado. En la tabla 3 se muestra la proporción de error obtenida tanto por el sistema coordinado como por las metaheurísticas individuales. En la primera parte de la tabla se muestra la proporción de error media obtenida con cada tipo de instancia, y en la segunda con cada tamaño de instancia. En la Fig. 4 se muestran los gráficos correspondientes a estas tablas.

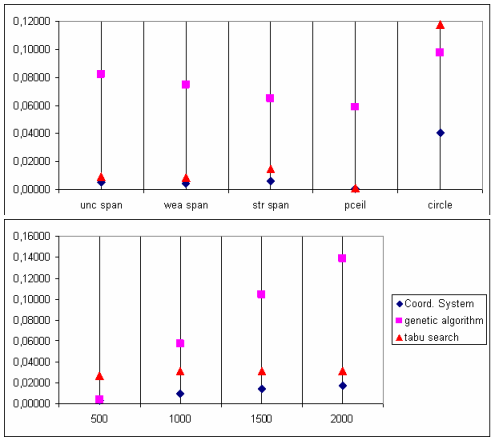

Figura 3: Coor2Met vs. metaheuristicas indiv.

Tabla 3: Coor3Met vs. metaheuristicas indiv.

Tipo	Sist. c.	A. Gen.	B. Tabu	Tem. S.
unc span	0,0005	0,0824	0,0092	0,0092
wea span	0,0043	0,0749	0,0082	0,0082
str span	0,0062	0,0648	0,0148	0,0148
pceil	0,0006	0,0590	0,0009	0,0009
circle	0,0330	0,0978	0,1174	0,0978
Items				
500	0,0031	0,0039	0,0266	0,0039
1000	0,0078	0,0568	0,0312	0,0312
1500	0,0116	0,1040	0,0313	0,0313
2000	0,0132	0,1384	0,0312	0,0312

Como puede observarse al igual que con el coordinador más sencillo este coordinador cumple el requisito de obtener por lo menos los mismos resultados que la mejor de las metaheurísticas individuales. Además sigue obteniendo una mejora sustancial en los resultados por lo que podemos ver que el sistema es robusto ante la adición de nuevas metaheurísticas ya que no empeora sus resultados, de hecho los mejora, como puede verse en la Fig. 5, que

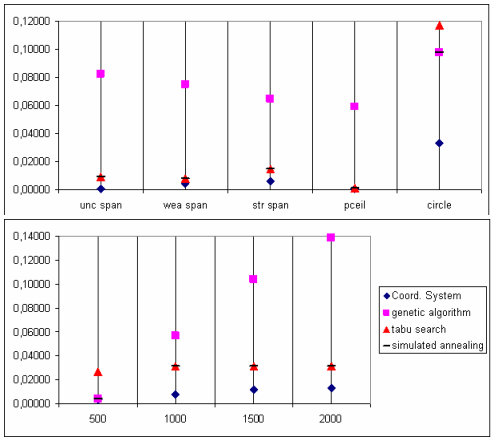

Figura 4: Coor3Met vs. metaheuristicas indiv.

muestra una comparativa entre los dos sistemas coordinados. En esta figura podemos ver como al aumentar el tamaño de las instancias, y por lo tanto su dificultad, Coor3Met va superando a Coor2Met. En cambio si atendemos al tipo de la instancia, las diferencias no son tan pronunciadas en todas, sin embargo, en las instancias de tipo circle, que son las más difíciles, y en las de tipo spanner no correlacionadas las diferencias siguen siendo apreciables.

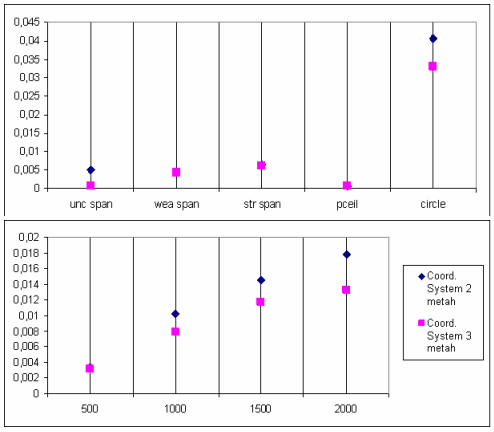

Figura 5: Coor2Met vs. Coor3Met

4. Conclusiones y trabajos futuros

Durante el desarrollo de este artículo se ha mostrado desde un punto de vista metodológico la aplicación de un proceso de extracción del conocimiento para realizar la construcción de Coor2Met y Coor3Met, dos prototipos de un sistema híbrido cooperativo centralizado de metaheurísticas para resolver el problema de la mochila. El proceso de extracción del conocimiento ha comprendido las siguientes fases:

- Preparación de los datos: Se ha realizado un análisis de la estructura de la base de datos requerida, estudiado las metaheurísticas que más interesaban, aplicado estas metaheurísticas a instancias y extraído información interesante de estas ejecuciones, realizando finalmente un preproceso para prepararla para la siguiente fase.
- Minería de datos: En la fase de minería de datos se ha realizado un estudio de las técnicas de minería de datos que pueden ser más adecuadas, explicado aquella escogida, FID 3.4, aplicado la técnica para comparar el comportamiento de las metaheurísticas y realizar un estudio de los parámetros de estas metaheurísticas, y producido un conjunto de reglas fuzzy, que finalmente se han adaptado con el fin de disponerse para la integración inmediata en el coordinador.
- Evaluación del modelo: Se han construido dos prototipos síncronos del sistema, ejecutado sobre una base de pruebas con distintas instancias y comprobado como el modelo obtenido era válido en cuanto a las soluciones obtenidas, ya que siempre mejoraba o igualaba las soluciones obtenidas por las metaheurísticas individuales, siendo la mejoría más palpable conforme se aumentaba la dificultad de la instancia. Además el modelo también se mostró valido desde el punto de vista de la robustez, puesto que la adición de una metaheurísticas incurrió en la mejora de los resultados y no al contrario.

Respecto a los trabajos futuros, sería interesante revisar el proceso de tal manera que fuera más automático, puesto que tanto la obtención de las reglas de bajo nivel como del modelo final requieren demasiada atención del experto. De igual manera se plantea la aplicación de otras técnicas de minería de datos, distintas de FID3.4, de tal manera que se pueda comprobar las distintas reglas que se pueden obtener y las mejoras posibles.

En otra línea de avance se espera poder realizar el mismo proceso que se ha seguido pero para un problema más complejo como es el de la p-mediana. Con estas instancias se seguirá el mismo proceso aquí expuesto para obtener la descripción de un agente coordinador de un sistema híbrido cooperativo centralizado de metaheurísticas para resolver el problema de la p-mediana.

Agradecimientos

Los autores agradecen al "Ministerio de Educación y Ciencia" y al "Fondo Europeo de Desarrollo Regional" (FEDER) por el soporte, bajo el proyecto TIN2005-08404-C04-02, para el desarrollo de este trabajo. Así como a la Fundación Séneca, Agencia de Ciencia y Tecnología de la Región de Murcia, que financia a través de una ayuda del Programa Séneca.

Referencias

[1] Cadenas, J.M., Garrido, M.C., Hernández, L.D., Muñoz, E., *Towards a definition of a Data Mining process based on Fuzzy Sets for Cooperative Metaheuristic systems*, IPMU2006, 2828-2835, 2006.

[2] Cadenas, J.M., Díaz-Valladares, R.A., Garrido, M.C., Hernández, L.D., Serrano, E., *Modelado del Coordinador de un sistema MetaHeurístico Cooperativo mediante SoftComputing*,CMPI2006,679-689, 2006.

[3] Cadenas, J.M., Garrido, M.C., Liern, V., Muñoz, E., Serrano, E., *Un prototipo del coordinador de un Sistema Metaheurístico Cooperativo para el Problema de la Mochila*, MAEB07, 811-818, 2007.

[4] Clark, P., Boswell, R., *Data mining. Practical machine learning tools and techniques with java implementations*, Morgan Kaufmann Publ., 2000.

[5] Corne, D., Dorigo, M., Glover, F. (editores), *New Ideas in Optimization*. McGraw-Hill, 1999.

[6] Cruz, C.A., *Estratégias coordinadas paralelas basadas en soft-computing para la solución de problemas de optimización*, Tesis doctoral, Dpto. Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, 2005.

[7] Janikow, C.Z., *Fuzzy Decision Tree/Forest. FID3.4*, URL: http://www.cs.umsl.edu/∼janikow/fid.

[8] Mamdani, E., *Applications of fuzzy logic to approximate reasoning using linguistic synthesis*, IEEE trans. on computers 26(2):1182-1191, 1997.

[9] Melián, B., Moreno, J.A., Moreno, J.M., *Metaheurísticas: una visión global*, Revista Iberoamericana de Inteligencia Artificial 19:7–28, 2003.

[10] Moreno-Velo, F.J., Baturone, I., Sánchez -Solano, S., Barriga, A., *XFUZZY 3.0: A Development Environment for Fuzzy Systems*, Proc. Int. Conference in Fuzzy Logic and Technology, 93-96, 2001.

[11] Rice, J.R., *The algorithm selection problem*, Adv. in Computers 15:65-118, 1976.

[12] University of Waikato *Weka, Data Mining with Open Source Machine Learning Software in Java*, URL: http://www.cs.waikato.ac.nz/ml/weka/.