

Una estrategia cooperativa adaptativa en problemas dinámicos

José M. Cadenas M. Carmen Garrido Enrique Muñoz

Dpto. de Ingeniería de la Información y las Comunicaciones

Universidad de Murcia. 30100 Espinardo. Murcia

jcadenas@um.es

carmengarrido@um.es

enriquemuba@um.es

Resumen

En la vida cotidiana aparecen muchos problemas de optimización dinámica, y para resolverlos necesitamos encontrar estrategias que puedan rastrear el óptimo según se mueve por el espacio de búsqueda. En este trabajo se propone el uso de una estrategia híbrida para tratar estos problemas. En esta estrategia, distintas metaheurísticas cooperan bajo la supervisión de un coordinador. Este coordinador es capaz de controlar la cooperación usando una colección de modelos SVM obtenidos a través de un proceso de extracción de conocimiento previo a la puesta en funcionamiento de la estrategia. Para probar su validez se ha aplicado al problema de los picos móviles. La estrategia se ha comparado con otras estrategias, tanto individuales como cooperativas, obteniendo resultados muy satisfactorios.

Palabras clave: Problemas dinámicos, Metaheurísticas cooperativas.

1. Metaheurísticas cooperativas

Las metaheurísticas [8] son estrategias inteligentes usadas para diseñar y mejorar procedimientos heurísticos muy generales con un alto rendimiento. La investigación en metaheurísticas se ha centrado tradicionalmente en problemas de optimización estáticos, sin embargo, los problemas de optimización que aparecen en la vida cotidiana son raramente estáticos, de hecho, normalmente son dinámicos, es decir, su espacio de búsqueda puede variar durante una ejecución. A este tipo de problemas se les

ha dado el nombre de Problemas de Optimización Dinámica (PODs) [2].

Se han propuesto muchos algoritmos y estrategias para afrontar los PODs, prevaleciendo aquellas estrategias que mantienen un conjunto de soluciones que evoluciona durante la ejecución, es decir, metaheurísticas basadas en poblaciones. El interés despertado por las metaheurísticas basadas en poblaciones ha dificultado el desarrollo de estrategias basadas en trayectorias. Sin embargo, trabajos recientes [7, 3] han mostrado que son aproximaciones válidas capaces de obtener resultados similares a los conseguidos por estrategias basadas en poblaciones. Estos resultados sugieren que el uso de estrategias cooperativas que combinen metaheurísticas basadas en poblaciones y trayectorias puede ayudar a resolver PODs.

El desarrollo de estas estrategias cooperativas consiste en la combinación de metaheurísticas existentes para obtener estrategias híbridas, en las que las metaheurísticas cooperan de modo paralelo. Debido a su capacidad de cooperar paralelamente para resolver una instancia de un problema, las metaheurísticas cooperativas están estrechamente relacionadas con las metaheurísticas paralelas. Algunos estudios [4, 7, 3] han demostrado que las metaheurísticas paralelas obtienen mejores resultados que sus componentes secuenciales, incluso cuando el tiempo disponible es menor. Estos estudios también han mostrado que la combinación de distintos hilos, usando cada uno una metaheurística diferente, incrementa la robustez de la búsqueda global con respecto a los cambios en la instancia del problema. Además, parece que una estrategia cooperativa basada

en una única metaheurística no cubre todas las posibilidades y se recomienda combinar distintas. Un buen ejemplo son los A-teams propuestos en [10].

Con respecto a los PODs, se han propuesto distintas estrategias que, aunque no usan una ejecución paralela, realizan cooperación entre distintas poblaciones. Los “self organizing scouts”, [1], son un claro ejemplo de estas aproximaciones donde se propone un algoritmo evolutivo multipoblación. Pero no sólo se han propuesto algoritmos evolutivos, los algoritmos de optimización por enjambre de partículas multipoblacionales se han convertido en una aproximación muy utilizada, [5]. Explorando otras direcciones se puede encontrar [7] en la que cooperan diferentes metaheurísticas basadas en trayectorias.

Otra aproximación interesante dentro del campo de las metaheurísticas híbridas es la utilización de técnicas de minería de datos para mejorar metaheurísticas. Ese campo ha logrado una gran popularidad en los últimos años, mostrando resultados prometedores. En [6] se puede encontrar una interesante revisión de este campo. En este trabajo se presenta una estrategia cooperativa adaptativa para resolver PODs. En esta estrategia se ejecutan distintas metaheurísticas mientras cooperan para resolver una instancia de un POD, estando controlada la cooperación por un coordinador. Para dirigir la cooperación paralela el coordinador es capaz de adaptar su comportamiento durante la resolución del POD. La adaptatividad se consigue mediante una base de reglas fuzzy y unos modelos SVM obtenidos mediante un proceso de extracción de conocimiento preliminar. Los modelos SVM contienen conocimiento extraído acerca del POD, y ayudan al coordinador a elegir la combinación de valores de parámetros para cada metaheurística y, junto con la base de reglas fuzzy, decidir cuando una metaheurística está obteniendo un rendimiento pobre y necesita información del resto de metaheurísticas.

En este trabajo presentamos (Sección 2) una estrategia cooperativa, su proceso de construcción y la evaluamos (Sección 3) para probar su validez al resolver PODs.

2. Estrategia cooperativa KEPS+D

Para describir la estrategia cooperativa, que denominamos KEPS+D, vamos a describir tres aspectos principales: la arquitectura del sistema, el esquema de cooperación que permite crear una sinergia entre las diferentes estrategias, y el proceso de extracción del conocimiento que otorga adaptabilidad al sistema.

Previamente introduciremos el contexto de la técnica, tratando de clasificarla. La estrategia propuesta puede ser incluida en diversas taxonomías. Si consideramos la taxonomía de metaheurísticas híbridas propuesta en [11], la estrategia se encuadra dentro de las estrategias HTH. Sin embargo, si la clasificamos como una metaheurística paralela se debe incluir en las de tipo 3, metaheurísticas paralelas o paralelismo multi-búsqueda, de la taxonomía propuesta en [4]. Una característica importante del sistema cooperativo de metaheurísticas es el uso de un proceso de extracción del conocimiento para configurar el control de la cooperación. Teniendo en cuenta la taxonomía propuesta en [6] se puede definir la estrategia como *a priori* atendiendo al tipo de conocimiento usado, debido a que su conocimiento se adquiere antes de la ejecución de la estrategia, el objetivo de la cooperación es mejorar la calidad de las soluciones encontradas y la parte de la metaheurística a la que se aplica el conocimiento son los parámetros.

2.1. Arquitectura

La ejecución de la estrategia propuesta se realiza ejecutando diferentes metaheurísticas en paralelo para explorar el espacio de búsqueda. Modelamos la estrategia usando un sistema multiagente que contiene dos tipos de agentes, *agentes de optimización* y *agente coordinador*. Cada agente de optimización ejecuta una metaheurística y el agente coordinador tiene dos tareas principales: recolectar la información acerca del rendimiento de cada uno de los agentes de optimización y enviar órdenes que afectarán su comportamiento (Fig. 1).

El agente coordinador es el responsable de iniciar el proceso de optimización activando paralelamente los agentes y además coordi-

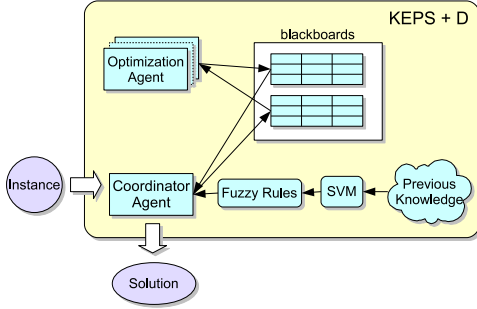


Figura 1: Un sistema multiagente cooperativo

na la cooperación entre los agentes de optimización eligiendo los valores de sus parámetros y controlando su intercambio de soluciones de un modo adaptativo. La adaptabilidad del coordinador se consigue considerando el conocimiento obtenido por medio de aprendizaje computacional y codificado como una colección de modelos SVM. En particular, los modelos SVM (a partir de la instancia a resolver) ayudan al coordinador a: 1) elegir los mejores valores de parámetros para cada metaheurística y 2) ordenar la idoneidad de cada metaheurística para resolver la instancia que se esté considerando, utilizando un conjunto de pesos en el rango $[0, 1]$. A través del análisis de pesos, el coordinador comprobará si alguna metaheurística está comportándose pobremente y, en consecuencia, puede actualizar sus soluciones añadiendo un conjunto de soluciones mejores obtenido por otra metaheurística con un mejor comportamiento. Este proceso permite a las metaheurísticas pobres reiniciarse en un punto más interesante del espacio de búsqueda.

Después del proceso de inicialización realizado por el agente coordinador, los agentes de optimización ejecutan asincrónicamente sus estrategias de optimización mientras envían y reciben información. El intercambio de información se realiza a través de un esquema de pizarra. Concretamente se utilizan dos pizarras, la primera la utilizan los agentes de optimización para publicar información sobre su comportamiento, la segunda la utiliza el coordinador para indicar las instrucciones para cada agente.

Específicamente, después de un periodo especificado, cada metaheurística para y publica su solución/población actual (dependiendo de si es basada en poblaciones o trayectorias). Entonces el agente coordinador analiza el comportamiento de cada metaheurística, teniendo en cuenta el conocimiento previo, y decide si alguna estrategia necesita mejorar su rendimiento. En este caso, el coordinador escribe en la segunda pizarra la nueva solución/población que las metaheurísticas tendrán que usar. Los algoritmos 2.1 y 2.1 ilustran el comportamiento de los agentes de optimización y coordinador, respectivamente.

```

Input( $q$ :instancia,  $m_i$ :Metaheurística,
       $n$ :periodo ejecución)
Begin
  While (No condición finalización)
    Read instrucciones de la pizarra;
    If se reciben instrucciones
      reemplazar actual solución/población con
      la contenida en las instrucciones;
    end
  end
  solucion=obtenida por metaheurística  $m_i$  durante  $n$ ;
  Write solucion en la pizarra
end

```

Algoritmo 1: Agentes de optimización

```

Input( $SVM$ :modelos SVM,  $n$ :periodo ejecución,
       $q$ :instancia,  $M$ :metaheurísticas)
Output(solucion:mejor solución encontrada)
Begin
  Analyze  $SVM$  y seleccionar los mejores valores
  para los parámetros de cada metaheurística en  $M$ ;
  While (No condición finalización)
    Compute cada agente;
    Exploit reglas fuzzy,  $SVM$  y pizarra para
    seleccionar POOR (agentes con mal rendimiento);
    ForEach (agente  $a$  en POOR)
      Create instruccionesa conteniendo una solución
      /población más adecuada;
      Write instruccionesa en la pizarra
    end
  end
  Return mejor solución obtenida por los agentes
end

```

Algoritmo 2: Agente coordinador

Con la intención de adaptar el sistema a los cambios que se pueden producir en un POD, el agente coordinador monitoriza en todo momento la instancia que se está resolviendo. En cuanto se da cuenta de que la instancia ha

cambiado, revisa su configuración. Para hacer esto, utiliza los modelos SVM mencionados anteriormente, que son independientes de la instancia que se esté resolviendo, y obtiene nuevos pesos y valores de parámetros para cada metaheurística. De esta manera se puede adaptar mejor al entorno cambiante.

2.2. Esquema de cooperación

El esquema de cooperación de esta estrategia está definido principalmente por las reglas de control usadas por el coordinador y los modelos SVM que guían su mecanismo de adaptación. Estos componentes tienen que ayudar al agente coordinador a:

- Elegir los mejores valores de parámetros para cada metaheurística.
- Descubrir cuando una metaheurística con un comportamiento pobre tiene que recibir nuevas soluciones de otras metaheurísticas con mejor comportamiento.

La primera tarea se realiza utilizando un conjunto de modelos SVM obtenidos de la aplicación de un proceso de aprendizaje supervisado. Cada uno de estos modelos está relacionado con un parámetro de una metaheurística. El agente coordinador realiza la segunda tarea considerando la información contenida en los modelos SVM restantes. Estos modelos analizan la instancia que se va a resolver y ordenan las diferentes metaheurísticas cooperantes, devolviendo una colección de pesos $\Omega = \{\omega_i, i = 1, \dots, nm\}$ donde nm es el número de metaheurísticas, $\omega_i \in [0, 1]$ and $\sum_{i=1}^{nm} \omega_i = 1$. Cada peso ω_i está asociado con una metaheurística m_i y representa sus idoneidad para resolver la instancia actual.

Para controlar el intercambio de soluciones el coordinador utiliza el siguiente conjunto de reglas fuzzy que explota Ω , esta colección se replica una vez por cada metaheurística:

si $(\omega_1 \cdot \delta_1)$ es *suficiente* entonces
 enviar soluciones de m_1 a m_h
 ...
 si $(\omega_{nm} \cdot \delta_{nm})$ es *suficiente* entonces
 enviar soluciones de m_{nm} a m_h

donde:

- nm es el número de metaheurísticas.
- m_h es la metaheurística evaluada.
- $\delta_i = (\xi_{m_i} - \xi_{m_h}) / \text{maximum}(\xi_{m_i}, \xi_{m_h})$, donde ξ es una medida de rendimiento.
- $\omega_{m_i} \in [0, 1]$ donde $\sum_{i=1}^{nm} \omega_{m_i} = 1$ y representa el peso de la metaheurística m_i .
- *suficiente* es un conjunto fuzzy con función de pertenencia trapezoidal (con soporte $\in [0, 1]$) definido por:

$$\mu(x, a, b, c, d) = \begin{cases} 0 & x \leq a \text{ or } x \geq d \\ \frac{x-a}{b-a} & x \in (a, b] \\ \frac{d-x}{d-c} & x \in [c, d) \\ 1 & x \in [b, c] \end{cases}$$

con $(a, b, c, d) = (0, 0.1, 1, 1)$.

Este conjunto de reglas indica al coordinador cuando tiene que enviar nuevas soluciones a una metaheurística dada. En concreto, cambia la posición en el espacio de búsqueda de una metaheurística que está mostrando un comportamiento pobre por una posición cercana a la posición de otra metaheurística con un comportamiento mejor. El disparo de la regla está controlado por un α -corte definido por el usuario. En el caso de que se dispare más de una regla el coordinador aplica todas ellas.

Para cambiar la solución de una metaheurística se tiene en cuenta lo siguiente:

- $m_{receptor}$ es basada en trayectorias. Se envía una solución “cercana” a la mejor solución obtenida entre las metaheurísticas que han disparado la regla. “cercana” se obtiene aplicando $\frac{1}{2 * \omega_{m_{emisor}}}$ veces un operador de mutación.
- $m_{receptor}$ es basado en poblaciones.
 - ◊ m_{emisor} es basada en trayectorias. Una proporción de los peores individuos de $m_{receptor}$ igual a $\omega_{m_{emisor}}$ se sustituye por un conjunto de soluciones cercanas a la mejor solución de m_{emisor} .
 - ◊ m_{emisor} es basada en poblaciones. Una proporción de los peores individuos de $m_{receptor}$ igual a $\omega_{m_{emisor}}$ se sustituye por un conjunto de los mejores miembros (soluciones) de la población de m_{emisor} .

- ◊ Distintas metaheurísticas tienen que enviar soluciones. Una proporción de los peores individuos de $m_{receptor}$ igual a la suma de los ω_i de los emisores se sustituye por un conjunto de soluciones donde cada m_{emisor} (situaciones antes descritas).

2.3. Obteniendo los modelos SVM

Para obtener los modelos de pesos y parámetros se utiliza un proceso de extracción como el que se muestra en la Fig. 2, compuesto de dos fases diferentes, preparación de datos y minería de datos. Se debe notar que este proceso es supervisado y previo a la ejecución de la estrategia.

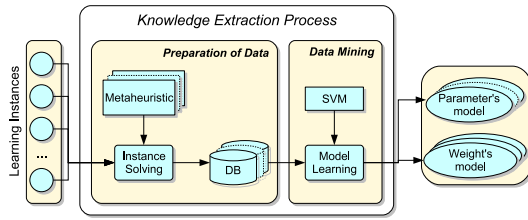


Figura 2: Proceso de extracción del conocimiento

Fase de preparación de datos

En esta fase las distintas metaheurísticas se ejecutan varias veces usando distintos valores de parámetros para obtener una colección de conjuntos de datos que describan el comportamiento de cada metaheurística. Estos conjuntos de datos son los utilizados para obtener los modelos que controlan los parámetros y los pesos. La obtención de estos conjuntos de datos se realiza en dos pasos, en el primero se obtienen conjuntos de datos que contiene información cruda. En el segundo paso esta información es procesada para obtener los conjuntos de datos finales.

Para obtener la primera colección de conjuntos de datos, llamados *datasets-crudos*, las diferentes metaheurísticas se ejecutan para resolver distintas instancias del problema que se esté considerando usando distintos valores de parámetros, repitiendo cada ejecución varias veces. La información obtenida de estas eje-

cuciones se incluye en los *datasets-crudos*. Se almacena un *dataset-crudo* distinto por cada metaheurística, y por cada ejecución se almacena la siguiente información: una descripción de la instancia que se está resolviendo, los valores para cada parámetro y el valor de la función objetivo obtenido.

Sin embargo, antes de aplicar minería de datos es necesario refinar los *datasets-crudos* para obtener los *datasets-refinados*. Estos conjuntos de datos contienen información refinada útil para obtener los modelos SVM para parámetros y pesos. Existen dos tipos diferentes de *datasets-refinados*, *datasets-refinados* de parámetros y *datasets-refinados* de pesos. El primer tipo está relacionado con la elección de los valores de parámetros de las metaheurísticas y existe un conjunto de datos para cada parámetro de cada metaheurística, el segundo tipo está relacionado con la importancia de cada metaheurística para una instancia dada, y existe un conjunto de datos para cada metaheurística.

El proceso para obtener todos los *datasets-refinados* de parámetros asociados a una metaheurística m_i es el siguiente:

1. Para cada instancia del *datasets-crudos* y para cada combinación de parámetros se calcula el valor de función objetivo obtenido medio.
2. Para cada instancia del *datasets-crudos* se obtiene la combinación de parámetros que obtiene el mejor valor medio de función objetivo.
3. Para cada instancia en el *datasets-crudos* y para cada parámetro asociado a m_i sus *datasets-refinados* de parámetros son actualizados con una nueva entrada que contiene la descripción de la instancia y el valor del parámetro incluido en la mejor combinación de valores de parámetros.

El proceso para obtener todos los *datasets-refinados* de pesos es el siguiente:

1. Para cada metaheurística m_i , para cada instancia y para cada combinación de valores de parámetros se calcula el valor medio de función objetivo obtenido.

2. Para cada m_i y para cada instancia, se calcula el *peso de la metaheurística* como:

$$\omega_{m_i} = \frac{fit_{best}^{q,i}}{\sum_{l=1}^{nm} fit_{best}^{q,l}}$$

donde: $fit_{best}^{q,i}$ es el mejor valor de la función objetivo asociado a la metaheurística i y la instancia q .

3. Para cada metaheurística m_i y para cada instancia el dataset refinado de pesos de m_i se actualiza con una nueva entrada que contiene, la descripción de la instancia y el peso de la metaheurística.

Fase de minería de datos

Una vez se ha reunido la información y se han obtenido los *datasets-refinados*, la fase de minería de datos extrae la colección de modelos de conocimiento a través de minería de datos. Nuestra aproximación de minería de datos utiliza la técnica SVM [9] para extraer conocimiento de los *datasets-refinados* y construir las colecciones de modelos antes mencionadas que guiarán al coordinador. Se pueden distinguir dos tipos de modelos, *modelos de parámetros* y *modelos de pesos*. El primer tipo está relacionado con la elección de valores de parámetros realizada por el coordinador y existe un modelo por cada parámetro de cada metaheurística. Por otro lado el segundo tipo de modelos está relacionado con la obtención del conjunto de pesos Ω , y existe un modelo por cada metaheurística.

Una vez finalizada esta fase, obtenemos un conjunto de modelos SVM que guían al coordinador, y está disponible para controlar la ejecución de los agentes de optimización.

3. Análisis experimental

En esta sección se realizarán tests para comprobar la validez de la estrategia cooperativa propuesta. Para probar la efectividad de la propuesta se utilizará el problema de los picos móviles (PPM).

3.1. El problema de los picos móviles

El PPM [2] es un POD ampliamente utilizado que define un horizonte de búsqueda n -

dimensional en el que se definen un número prefijado de picos con localizaciones (X), alturas (H), y anchuras (W) específicas. Los picos están distribuidos aleatoriamente en un área restringida. Cada pico puede variar su altura, anchura y posición a lo largo del tiempo.

Se han generado distintos entornos para PPM de acuerdo a estándares¹:

- Escenario 1: el horizonte de fitness está definido por un espacio de búsqueda 5-dimensional estando acotada cada dimensión entre $[0, 100]$. En este dominio existen 5 picos móviles que varían su altura aleatoriamente en el intervalo $[30, 70]$, anchura entre $[0,0001, 0,2]$ y posición por una distancia máxima de 1.
- Escenario 2: el horizonte de fitness está definido por un espacio de búsqueda 5-dimensional estando acotada cada dimensión entre $[0, 100]$. Sin embargo, en el escenario 2 se tienen 5 configuraciones, que tienen 10, 20, 30, 40 y 50 picos móviles que varían su altura aleatoriamente en el intervalo $[30, 70]$, anchura entre $[1, 12]$ y posición por una distancia máxima de 1.

3.2. Configuración del sistema

Usaremos un sistema compuesto por tres metaheurísticas diferentes: un algoritmo genético (AG), una búsqueda tabú (BT) y un temple simulado (TS).

El siguiente aspecto que debe ser tratado es el proceso de extracción del conocimiento. Se debe notar que, aunque el problema de prueba es dinámico, el sistema tiene que aprender de instancias estáticas que son aquellas que se pueden caracterizar, sin embargo, esto no es problemático puesto que un problema dinámico puede ser considerado como una sucesión de problemas estáticos. Para realizar el aprendizaje se generaron 150 instancias (25 por tipo). Cada una de estas instancias se resolvió utilizando 4000 evaluaciones de la función objetivo usando diferentes combinaciones de valores de parámetros con cada una de las metaheurísticas. Con la información generada se rellenaron

¹<http://www.aifb.uni-karlsruhe.de/~jbr/MovPeaks/>

los *datasets-crudos* y por medio de su procesamiento se obtuvieron los *datasets-refinados*. Estos últimos conjuntos de datos fueron la entrada de la fase de minería de datos, cuya salida son los modelos SVM tanto de parámetros como de pesos.

3.3. Configuración del experimento

Los diferentes escenarios disponibles se configuraron para cambiar después de 5000 evaluaciones de la función objetivo, y por cada ejecución ocurrieron 100 cambios. Los resultados que se muestran son la media de 100 ejecuciones. Para medir el rendimiento de cada estrategia se utiliza el error offline [2]:

$$error_offline = \frac{1}{T} \sum_{t=1}^T (optimo - mejorSolucion)$$

Una vez mostrados los resultados experimentales², se hace un análisis usando técnicas estadísticas. Utilizamos el test de *Wilcoxon signed-rank* para comparar dos métodos, y el test de *Friedman*, y el test post-hoc *Benjamin-Hochberger*, para comparaciones múltiples.

3.4. Utilidad de la cooperación

Los primeros tests realizados fueron ejecutados para conocer como afecta la cooperación entre las diferentes metaheurísticas desde el punto de vista del rendimiento. Los parámetros del sistema que controlan el número de comunicaciones entre metaheurísticas, y por tanto la cooperación, son el número de evaluaciones de la función objetivo antes de realizar la cooperación y el α -corte que controla el disparo de las reglas. Se debe notar que estos tests son preliminares y que por tanto las medias calculadas son realizadas sobre 10 ejecuciones.

En la figura 3 se puede ver una clara tendencia, el ratio de cooperación no debe ser ni muy alto ni muy bajo. Si nos movemos de izquierda a derecha de la figura, incrementando el ratio de comunicación, se puede ver como el porcentaje de error inicialmente decrece ligeramente hasta que llegado un punto comienza a crecer.

²Tests realizados usando AMD Opteron Dual Core (8 GB) - cluster UGRGrid - <http://ugrgrid.ugr.es>

De estos resultados se puede extraer que demasiado cooperación hace el sistema inestable, sin embargo, poca cooperación también empeora el rendimiento, puesto que no hay suficiente información para las metaheurísticas. El mejor resultado corresponde al intervalo [800, 900] de evaluaciones y con un $\alpha - cut = 0,5$, obteniendo un error offline de 3.77, desviación típica 0.69 y 262 disparos medios de las reglas.

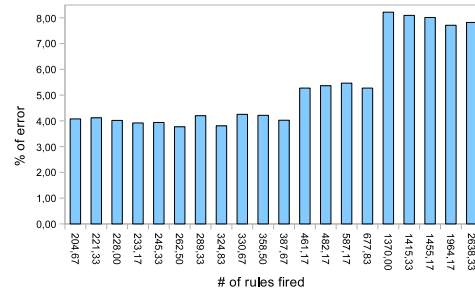


Figura 3: Tests de cooperación

3.4.1. Comparando KEPS+D

Comparamos los resultados obtenidos por la estrategia con aquellos que obtienen las metaheurísticas individuales que la componen y con la estrategia cooperativa CS-WB. Esta última estrategia se obtiene usando el mismo sistema de metaheurísticas pero sustituyendo el coordinador por uno más simple que solamente lee la mejor solución obtenida por cada metaheurística y envía la mejor solución obtenida hasta el momento a aquella metaheurística que obtenga el peor valor de la función objetivo. La Tabla 1 muestra estos resultados.

Tabla 1: Resultados para PPM

# picos	GA	SA	TS	CS-WB	KEPS+D
5	12.45 _{1,74}	14.85 _{1,28}	7.03 _{0,63}	6.00 _{0,77}	4.01 _{1,11}
10	10.89 _{1,98}	13.07 _{1,11}	7.66 _{0,32}	5.17 _{0,30}	4.56 _{0,31}
20	10.08 _{1,86}	8.04 _{0,73}	5.42 _{0,29}	3.57 _{0,26}	2.96 _{0,23}
30	13.51 _{2,02}	9.85 _{0,84}	7.80 _{0,33}	5.37 _{0,32}	4.53 _{0,36}
40	12.61 _{2,15}	7.05 _{0,63}	5.71 _{0,38}	3.86 _{0,21}	3.40 _{0,26}
50	11.31 _{1,54}	7.29 _{0,68}	6.12 _{0,24}	4.24 _{0,20}	3.92 _{0,26}

Analizemos los resultados de las metaheurísticas individuales. En este caso aquella que

muestra el mejor rendimiento es la búsqueda tabú (validado por los tests estadísticos con un 98 % de confianza), las otras dos metaheurísticas obtuvieron resultados estadísticamente equivalentes, aunque a simple vista parece que los del temple simulado son algo mejores. Si se comparan los resultados obtenidos por las metaheurísticas individuales con los de las cooperativas, son estas últimas las que obtienen mejores resultados con un grado de confianza del 98 %. Por lo que se puede afirmar que la colaboración introducida ayuda eficientemente en la resolución de problemas dinámicos. Finalmente la estrategia que mostró el mejor rendimiento fue KEPS+D (al 98 % de confianza). Por lo que también podemos afirmar que el conocimiento introducido en el sistema también presenta una gran utilidad.

4. Conclusiones

En este trabajo se ha presentado una estrategia metaheurística cooperativa llamada KEPS+D para resolver problemas de optimización dinámica. En esta estrategia, diferentes metaheurísticas cooperan para resolver una instancia de un problema bajo la supervisión de un coordinador. Este coordinador está modelado usando el conocimiento extraído de un proceso de aprendizaje supervisado aplicado a datos históricos de la ejecución de las metaheurísticas individuales.

Para probar la propuesta se ha elegido el problema de los picos móviles. Los resultados obtenidos se han comparado con los obtenidos por las metaheurísticas individuales que la componen y con otra estrategia cooperativa obteniendo resultados muy favorables.

Agradecimientos

Los autores agradecen al MICINN y al FEDER por el financiación dada a este trabajo bajo el proyecto TIN2008-06872-C04-03. También agradecen el programa para grupos de investigación de excelencia 04552/GERM/06 de la "Fundación Séneca". También se agradece a la "Fundación Séneca" por la financiación ofrecida a E. Muñoz a través de un programa FPI.

Referencias

- [1] Branke, J., Kauler, T., Schmidt, C., Schmeck, H., *A multi-population approach to dynamic optimization problems*, Proc. Adaptive Computing in Design and Manufacturing, 299–308, 2000.
- [2] Branke, J., *Evolutionary Optimization in Dynamic Environments*, Kluwer Academic Publishers, 2002.
- [3] Cadenas, J.M., Garrido, M.C., Muñoz, E., *Using machine learning in a cooperative hybrid parallel strategy of metaheuristics*, Inf. Sci. 179(19), 3255–3267, 2009.
- [4] Crainic, T.G., Toulouse, M., *Parallel Strategies for Metaheuristics*, Handbook of Metaheuristics, Kluwer Academic Publisher, 475–513, 2003.
- [5] Du, W., Li, B., *Multi-strategy ensemble particle swarm optimization for dynamic optimization*, Inf. Sci. 178, 3096–3109, 2008.
- [6] Jourdan, L., Dhaenens, C., Talbi, E.G., *Using datamining techniques to help metaheuristics: a short survey*, Hybrid Metaheuristics (LNCS), 57–69, 2006.
- [7] González, J.R., Masegosa, A.D., García, I.J., *A cooperative strategy for solving dynamic optimization problems*, Memetic Computing (10.1007/s12293-010-0031-x).
- [8] Melián, B., Moreno, J.A., Moreno, J.M., *Metaheurísticas: una visión global*, Intelligenci Artificial 19, 7–28, 2003.
- [9] Shevade, S.K., Keerthi, S.S., Bhattacharyya, C., Murthy, K.R.K., *Improvements to the SMO Algorithm for SVM Regression*, IEEE Trans. Neural Networks 11(5), 1188–1193, 2000.
- [10] Souza, P., Talukdar, S., *Asynchronous organizations for multi algorithm problems*, ACM Symp. Applied Computing, 1993.
- [11] Talbi, E.G., *A Taxonomy of Hybrid Metaheuristics*, Heuristics 8, 541–564, 2002.