

Estudio de la robustez de las estrategias cooperativas frente a los cambios de configuración de sus componentes^{*}

Antonio D. Masegosa David Pelta

Dpto. Ciencias de la Computación e Inteligencia Artificial

Universidad de Granada

{admase,dpelta}@decsai.ugr.es

Resumen

Dentro de los distintos tipos de metaheurísticas híbridas, las estrategias cooperativas se presentan como una de las alternativas más prometedoras. Este tipo de métodos consisten en un conjunto de algoritmos que exploran el espacio de búsqueda de forma simultánea mientras intercambian información entre ellos. Con el objetivo de profundizar en el estudio de dichos métodos y utilizando el problema del p-hub mediano, en este trabajo analizamos como afecta al comportamiento de la estrategia cooperativa el hecho de cambiar de forma notoria la configuración de las metaheurísticas que la componen.

1. Introducción

La combinación de diferentes metaheurísticas es hoy en día una de las formas más exitosas de desarrollar nuevos métodos de optimización. Este tipo de algoritmos, conocidos como metaheurísticas híbridas, son cada vez más comunes en la literatura y actualmente resulta difícil encontrar problemas, tanto académicos como prácticos, en los que los mejores resultados no hayan sido obtenidos por alguna de las variantes de estos métodos.

Dentro de las diferentes formas de hibridación, las estrategias cooperativas [2, 12] se han mostrado como una de las alternativas más

prometedoras. Estas estrategias consisten en un conjunto de agentes que se ejecutan en paralelo intercambiando información entre ellos mientras llevan a cabo su propia búsqueda.

Cuando se trabaja con estrategias cooperativas, uno puede preguntarse si la cooperación seguirá funcionando aunque se utilicen diferentes configuraciones de los métodos implementados por los agentes y como afecta dicho cambio a su rendimiento. Algunos ejemplos serían la modificación del ajuste de sus parámetros con el fin de adecuarlos a un determinado tipo/tamaño de instancias o la incorporación de información específica del problema a estos métodos mediante operadores ad-hoc, generación greedy de soluciones, etc.

En esta contribución pretendemos estudiar este aspecto haciendo uso de una estrategia cooperativa, presentada en [12], en la que los agentes son controlados a través de un coordinador central dirigido por reglas. Concretamente vamos a evaluar el comportamiento de dicha estrategia con dos configuraciones distintas para sus componentes. Dichas configuraciones varían en el ajuste de los parámetros y en la incorporación o no de información específica del problema. En este sentido estamos extendiendo la investigación realizada en trabajos anteriores donde analizamos otros aspectos como la influencia de la composición de los agentes de búsqueda [10] o del esquema de cooperación [9] en el rendimiento de este tipo de estrategias.

El artículo se estructura de la siguiente manera. En primer lugar, describiremos la estrategia cooperativa utilizada. Seguidamente, en

^{*}A.D. Masegosa es beneficiario del programa de becas FPI del Ministerio de Ciencia e Innovación. Este trabajo ha sido parcialmente financiado por los proyectos TIN2008-01948 del Ministerio de Ciencia e Innovación y P07-TIC-02970 de la Junta de Andalucía.

la Sección 3, definiremos el problema empleado como banco de pruebas, el p-hub mediano. En la Sección 4 veremos los diferentes detalles de la implementación de la estrategia. La siguiente sección está dedicada a describir la experimentación realizada así como al análisis de los resultados obtenidos. Terminaremos con las principales conclusiones de este trabajo.

2. Descripción de la estrategia

Esta estrategia, presentada inicialmente en [12], consiste en un conjunto de agentes/optimizadores que pueden implementar la misma estrategia para el problema a resolver o bien una distinta. Estos optimizadores están controlados por un coordinador, que recibe información sobre su rendimiento, la procesa y haciendo uso de una base de reglas, produce sucesivamente ajustes del comportamiento de los agentes mediante el envío de ciertas órdenes. El intercambio de información se realiza a través de una arquitectura de pizarra [5].

Una parte importante de esta estrategia es el flujo de información, que se divide en tres etapas: 1) envío de reportes por parte de los agentes al coordinador, 2) los datos son almacenados y procesados por el coordinador y 3) el coordinador envía órdenes a los optimizadores. Cada uno de los reportes enviados en la primera etapa contiene:

- identificación del agente
- un marcador de tiempo t
- la solución actual del agente s^t
- la mejor solución obtenida hasta ahora por éste s_{mejor}

El coordinador almacena los dos últimos reportes de cada agente. A partir de estos dos reportes, el coordinador calcula su ratio de mejora como $\Delta_f = \frac{f(s^t) - f(s^{t'})}{t - t'}$, donde $t - t'$ representa el tiempo transcurrido entre dos reportes consecutivos, $s^{t'}$ es la solución actual enviada por el solver en el reporte anterior y f es la función objetivo. Los valores Δ_f y $f(s^t)$ se almacenan en dos memorias ordenadas de longitud fija, una para los ratios de mejora y

otra para los costes, cuyo tamaño se estableció a $3 \times \text{número de optimizadores}$.

El comportamiento de los agentes se controla mediante un conjunto de reglas que permiten al coordinador detectar patrones de mal comportamiento en los optimizadores, además de determinar la acción a realizar para corregir dicho comportamiento. Estas reglas son del tipo:

si condición **entonces** acción.

Las acciones consisten en el envío de una solución al agente para que este la use como punto de reinicio de su búsqueda (esta puede ser la mejor solución vista hasta ahora por el coordinador C_{mejor} , una aleatoria, etc.). De esta manera, el coordinador controla la localización de las hebras dentro del espacio de búsqueda.

2.1. Reglas de control

Como se comentó anteriormente, la base de reglas se utiliza para determinar cuando debe alterarse el comportamiento/posición de un determinado agente. Las reglas de control usadas en este trabajo se diseñaron de forma “manual” siguiendo el principio: *Si el funcionamiento del agente es malo, entonces modifica su comportamiento*. Dicho principio se modeló de dos formas diferentes que se describen a continuación.

2.1.1. Regla de control básica (RB)

Se trata de una regla de control extremadamente simple e intuitiva cuya definición es la siguiente:

si la solución actual reportada por $agente_i$ es peor que C_{mejor}
entonces enviar C_{mejor} a $agente_i$

C_{mejor} denota la mejor solución vista por el coordinador hasta el momento.

2.1.2. Regla de control difusa (RD)

Esta regla se propuso inicialmente en [12] y, tanto en dicho trabajo como en otros posteriores, ha llevado a buenos resultados sobre diferentes problemas. Su diseño está basado en conjuntos difusos, siendo esta su definición:

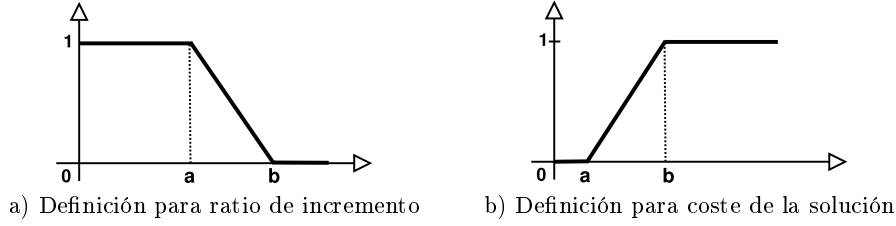


Figura 1: Funciones de pertenencia para las dos etiquetas usadas en la regla de control difusa.

si la calidad de la solución reportada por $agente_i$ es *baja* y el ratio de incremento de $agente_i$ es *bajo* entonces enviar C_{mejor} a $agente_i$

Las etiquetas *baja* y *bajo* son conjuntos difusos cuyas funciones de pertenencia $\mu(x)$ pueden verse en la Figura 1. La variable x corresponde con la posición relativa (semejante a la noción de percentil) de un valor (un ratio de mejora o un coste) en las muestras almacenadas en las respectivas memorias. Los parámetros (a,b) se fijaron a $(80,100)$ y $(0,20)$ para los ratios de mejora y los costes, respectivamente.

3. El problema del p-hub mediano

Los problemas de localización de hubs pueden considerarse como la formalización de aquellos problemas en los que se desea guiar un determinado flujo desde un conjunto de orígenes a un conjunto de destinos, resultando muy costoso tener conexiones entre cada par origen-destino. El objetivo de este tipo de problemas se compone de dos pasos: (1) **Localización de los hubs**: determina que nodos deben ser los hubs y su número total, para así distribuir el flujo a través de ellos. (2) **Asignación de nodos no-hub a hubs**: asigna el resto de nodos a los hubs. Generalmente, estas tareas se realizan mediante la minimización de una función objetivo que depende del flujo intercambiado y de su coste. Una revisión general de estos problemas puede verse en [3].

Este trabajo se centra en aquellos problemas donde los hubs no tienen restricciones de capacidad en cuanto a cantidad de tráfico, los

nodos pueden ser asignados a un único hub y hay un número fijo p de hubs.

Como modelo del problema hemos utilizado la formulación cuadrática entera dada por O'Kelly en [11]. Sea N un conjunto de n nodos. Definimos W_{ij} como la cantidad de flujo del nodo i al j , y C_{ij} como el costo de transportar una unidad de flujo entre los nodos i y j . Cada flujo W_{ij} , tiene tres componentes diferentes: recogida, transferencia y distribución. La recogida se corresponde con el transporte desde el nodo i hasta su hub. La transferencia representa el movimiento entre los hubs i y j . Y la distribución ocurre desde este segundo hub hasta j . Cada una de estas actividades puede modificar el costo por unidad de flujo en función de una determinada constante. Tales constantes se denotan como χ (generalmente $\chi = 1$), α (generalmente $\alpha < 1$) y δ (generalmente $\delta = 1$).

Sea X_{ij} una variable de decisión cuyo valor es 1 si el nodo i se asigna al hub j y 0, en caso contrario. Este problema se formula como:

$$\min \sum_{i,j,k,l \in N} W_{ij} (\chi C_{ik} X_{ik} + \alpha C_{kl} X_{ik} X_{jl} + \delta C_{jl} X_{jl}) \quad (1)$$

sujeito a

$$\sum_{j=1}^n X_{jj} = p \quad (2)$$

$$\sum_{j=1}^n X_{ij} = 1, \forall i = 1, \dots, n \quad (3)$$

$$X_{ij} \leq X_{jj}, \forall i, j = 1, \dots, n \quad (4)$$

$$X_{ij} \in \{0, 1\}, \forall i, j = 1, \dots, n \quad (5)$$

La función objetivo (1) minimiza la suma de los costes del flujo origen-hub, hub-hub y hub-destino. La restricción (2) asegura que únicamente existen p hubs. La condición (3) indica que un nodo puede únicamente ser asignado a un hub. El hecho de que un punto no-hub tiene que estar asignado a un hub viene determinado por (4) y finalmente, (5) es la clásica restricción binaria.

Las instancias usadas en este trabajo pertenecen al benchmark AP y se obtuvieron del recurso ORLIB [1]. Estas pueden dividirse en dos grupos: aquellas con 20, 25, 40 y 50 nodos donde $p \in \{2, 3, 4, 5\}$ y aquellas con 100 y 200 nodos, en la que $p \in \{2, 3, 4, 5, 10, 15, 20\}$. El valor de las constantes χ , α y δ se fijó a 3, 0.75 y 2, respectivamente, siguiendo las indicaciones dadas en este recurso. El óptimo de las instancias con 50 o menos nodos fue obtenido de ORLIB, mientras que para las demás instancias se consideró la mejor solución reportada por uno de los algoritmos del estado del arte para este problema [8]. La calidad de las soluciones se midió como: $error(\%) = 100 \times \frac{\text{costo obtenido} - \text{optimo}}{\text{optimo}}$.

4. Implementación de la estrategia

Cuando se implementan estrategias cooperativas multi-hebra, se puede recurrir a implementaciones paralelas, o bien, simular el paralelismo en una máquina monoprocesador. En este trabajo hemos optado por esta última estrategia y para ello construimos un array de optimizadores que son ejecutados siguiendo un esquema round-robin, es decir, a cada uno de ellos se le asignan sucesivamente slots de tiempo de procesador que vienen dados por un cierto número de evaluaciones de la función objetivo. Este número se genera aleatoriamente en el intervalo [1000,1500] y [14000,15000] para instancia con 50 o menos nodos y más de 50 nodos, respectivamente. Una vez que un optimizador ha consumido su slot, tiene lugar la comunicación con el coordinador. Estos pasos se repiten hasta que se cumple una determinada condición de parada.

Otro aspecto importante de esta estrategia es el número de agentes que la componen y

las metaheurísticas que estos implementan. En este caso hemos trabajado con una estrategia formada por un Enfriamiento Simulado (ES), una búsqueda Tabú (Tabu) y una búsqueda de Descenso por Entornos Variables (DEV). Para la implementación de estos métodos se siguieron las directrices básicas descritas en [6]. El operador de vecindario utilizado por las heurísticas hace uso de dos mecanismos distintos: cambio de asignación de nodos no-hub y cambio de localización de hubs. De una forma más detallada, los pasos que lleva a cabo el primero de los mecanismos son los siguientes:

1. Elegir aleatoriamente un grupo G_j , donde $G_j = \{i | X_{ij} = 1, i \neq j\}$ es el grupo de aquellos nodos asignados al hub j
2. Escoger aleatoriamente un nodo $i \in G_j$
3. Elegir aleatoriamente un grupo $G_k, k \neq j$
4. Asignar el nodo seleccionado al nuevo grupo: $X_{ij} \leftarrow 0, X_{ik} \leftarrow 1$

El segundo mecanismo cambia la localización de un hub j a otro nodo que está asignado actualmente al hub i . Si no hay nodos asignados a j , se escoge un nodo diferente como hub y j se asigna a otro grupo. Estos son los pasos que realiza:

1. Elegir aleatoriamente un grupo G_j
2. Si tiene al menos un nodo ($|G_j| > 0$):
 - a) Seleccionar un nodo aleatorio $i \in G_j$
 - b) Asignar todos los nodos en G_j y su nodo hub j al nuevo hub i . $\forall k \in G_j$: $X_{kj} \leftarrow 0, X_{ji} \leftarrow 1, X_{ki} \leftarrow 1, X_{ji} \leftarrow 1$ y $X_{ii} \leftarrow 1$
3. Si el grupo no tiene nodos ($|G_j| = 0$):
 - a) Elegir aleatoriamente otro grupo $G_k, k \neq j$ con al menos un nodo
 - b) Elegir un nodo aleatorio $i \in G_k$
 - c) Hacer un nuevo grupo con el nodo seleccionado i . $X_{ii} \leftarrow 1$
 - d) Asignar el hub j como un nodo normal a otro hub seleccionado aleatoriamente. $X_{jr} \leftarrow 1$ donde r es un hub aleatorio

4.1. Configuraciones de los optimizadores

Se definieron dos configuraciones distintas para las metaheurísticas implementadas por los agentes que variaban tanto en términos del ajuste de los parámetros como en la incorporación o no de información específica del problema.

En la primera configuración, los optimizadores cuentan con un ajuste más adecuado para instancias de 50 o menos nodos y estos parten en cada reinicio de una solución generada de forma totalmente aleatoria. Por el contrario, en la segunda configuración, los parámetros de los métodos tienen un ajuste más apropiado para las instancias de mayor tamaño, además de que la inicialización de las soluciones se realiza mediante un proceso greedy aleatorizado que hace uso de información específica del problema. La descripción de este proceso sería la siguiente: en primer lugar se determinan qué hubs y qué asignaciones de nodos no-hub pueden ser potencialmente buenos. Con este fin, se calcula para cada nodo i su flujo potencial como $\hat{\omega}_i = \sum_{j \in N} (W_{ij} + W_{ji})$. Tras esto, se llevan a cabo los dos siguientes pasos:

1. **Localización de hubs:** se construye una Lista Restringida de Candidatos (LRC) con los l nodos ($l \geq p$) que tienen un mayor flujo potencial. De esta lista se escogen aleatoriamente los p hubs.
2. **Asignación de nodos no-hub:** se asocia una LRC con tamaño r ($r \leq p$) a cada nodo no-hub. Esta LRC contiene los r hubs con menor costo de transporte al punto considerado. Cada uno de los nodos no-hub es asignado de forma aleatoria a uno de los hubs de su lista.

Los parámetros l y r se fijaron a *número de hubs + 1* y *2*, respectivamente.

5. Experimentación y resultados

La experimentación realizada en este trabajo esta dirigida a evaluar el comportamiento de la estrategia descrita anteriormente frente a la variación de la configuración de los optimizadores que la componen. Se consideraron

dos modos distintos de cooperación que se corresponden con las reglas de control vistas en la Sección 2.1. Con este fin, comparamos para ambas configuraciones los tres métodos individuales correspondientes a cada uno de los optimizadores utilizados (ES, Tabu y DEV) frente a dos versiones de la estrategia cooperativa que utilizan una de las dos reglas de control presentadas (RB y RD).

Antes de comenzar con el análisis de los resultados, vamos a describir algunos aspectos de la experimentación. Las condiciones de parada se fijaron a 8000 evaluaciones para instancias con 20 y 25 nodos, a 20000 para las de 40 y 50 nodos, y a 250000 para las de 100 o más nodos. Es importante remarcar que tanto los métodos individuales como las estrategias cooperativas realizan el mismo número de evaluaciones de la función objetivo. Además de esto, hay que señalar que cada método se ejecutó 30 veces sobre cada una de las instancias y que para la experimentación se utilizaron las herramientas SIGMA [7] y DACOS [4].

En el Cuadro 1 mostramos el porcentaje promedio de error obtenido sobre las 30 ejecuciones por cada uno de los 5 métodos comparados con las dos configuraciones utilizadas, C1 y C2. Si nos centramos en los métodos individuales, en esta tabla podemos comprobar, como se apuntó anteriormente, que C1 es más apropiada para instancias con 50 o menos nodos, mientras que la segunda muestra un mejor comportamiento para instancias de mayor tamaño. Si bien este hecho no está demasiado claro para Tabu, debemos notar que dicha mejora tiene lugar en las instancias de mayor dificultad ($\{100,200\}$ nodos - $\{15,20\}$ hubs).

Entrando más de lleno en el objetivo de esta experimentación, vamos a comparar las dos estrategias cooperativas frente a los métodos individuales de una forma global, es decir, teniendo en cuenta ambas configuraciones. Con el fin de no llegar a conclusiones erróneas, hicimos uso de test pareados no-paramétricos que utilizan como muestras las medias expuestas en el Cuadro 1. El test pareado no-paramétrico de Friedman establece que la hipótesis de igualdad puede rechazarse para estos métodos a un nivel de significación de 0.05.

Nodos	Hubs	DEV		ES		Tabu		RB		RD	
		C1	C2	C1	C2	C1	C2	C1	C2	C1	C2
20	2	0.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	3	0.74	1.35	0.6	2.45	1.06	2.30	0.38	0.57	0.45	0.87
	4	3.09	4.14	1.56	4.01	2.19	2.73	2.39	1.64	1.99	2.66
	5	2.02	3.52	1.24	3.48	1.44	1.93	1.48	1.31	0.77	2.00
25	2	0.01	0.09	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
	3	2.12	4.57	1.03	3.18	2.12	3.43	2.03	1.36	0.98	2.47
	4	3.01	6.46	1.55	4.85	2.17	2.46	1.70	2.06	1.87	2.28
	5	5.95	7.78	1.88	7.27	4.43	4.25	3.31	2.41	4.34	3.75
40	2	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00	0.00
	3	0.47	0.34	0.17	0.63	0.05	1.19	0.20	0.39	0.22	0.28
	4	1.60	1.46	0.95	2.58	0.37	2.19	1.12	0.21	0.69	0.21
	5	3.45	2.46	1.19	2.54	2.22	2.14	2.21	1.50	2.34	1.69
50	2	0.01	0.02	0.03	0.00	0.00	0.02	0.00	0.00	0.01	0.00
	3	1.25	3.03	1.14	2.19	1.38	3.00	1.98	1.03	1.34	1.65
	4	3.88	5.37	3.28	3.76	2.97	2.82	3.76	0.84	2.78	1.22
	5	5.80	6.07	3.51	5.53	4.33	3.12	5.69	3.80	5.13	3.44
100	2	0.00	0.00	0.21	0.00	0.00	0.01	0.00	0.00	0.00	0.00
	3	0.19	0.09	0.9	0.00	0.07	0.17	0.02	0.00	0.06	0.09
	4	0.51	0.64	1.51	0.00	0.28	0.48	0.38	0.42	0.32	0.18
	5	2.72	2.23	3.65	1.49	2.12	1.05	2.25	1.30	2.41	1.30
	10	9.62	6.69	6.54	2.90	5.48	6.26	4.86	2.32	5.67	3.62
	15	14.35	6.79	10.91	5.52	9.15	7.20	9.40	4.37	9.77	6.32
	20	14.87	4.62	10.00	3.98	7.68	5.98	8.42	3.46	8.97	4.84
200	2	0.05	0.05	1.37	0.00	0.06	0.08	0.05	0.01	0.06	0.01
	3	0.32	0.00	3.61	0.00	0.16	0.29	0.09	0.00	0.19	0.00
	4	0.75	0.08	4.51	0.01	0.39	0.53	0.35	0.04	0.35	0.11
	5	2.10	0.62	6.22	0.65	1.26	1.31	1.24	0.85	1.32	0.73
	10	11.28	8.88	14.64	5.59	6.78	10.70	6.94	3.80	8.24	6.97
	15	19.07	8.58	23.36	7.99	12.13	11.41	14.31	7.17	14.86	8.25
	20	23.33	7.97	27.36	5.19	15.06	9.92	16.17	6.45	17.45	6.03

Cuadro 1: Porcentaje promedio de error para DEV, ES, Tabu, RB y RD con cada una de las dos configuraciones consideradas.

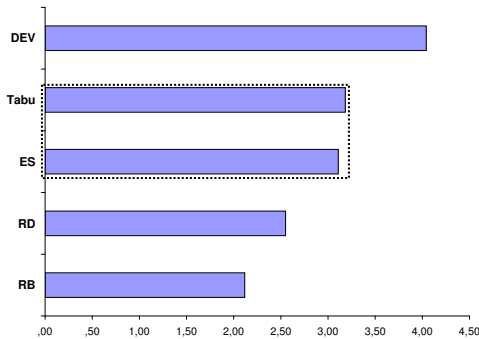


Figura 2: Cada barra representa el rango promedio devuelto por el test pareado no-paramétrico de Friedman para DEV, ES, Tabu, RB y RD sobre ambas configuraciones. Los métodos mostrados en el eje Y están ordenados por este valor (a menor rango promedio, mejor rendimiento). El rectángulo punteado agrupa los pares de métodos cuyo rendimiento no es significativamente diferente (test pareado no-paramétrico de Wilcoxon con $\alpha < 0.05$)

El rango promedio devuelto por este test para cada uno de los algoritmos estudiados puede verse en la Figura 2. En este gráfico los métodos se encuentran ordenados por dicho valor (puesto que estamos minimizando, a menor rango promedio mejor rendimiento) y la función del rectángulo punteado es agrupar a aquellos pares de algoritmos cuyo rendimiento no es significativamente diferente (test pareado no-paramétrico de Wilcoxon con $\alpha < 0.05$). Centrándonos en los métodos individuales vemos que DEV es significativamente el peor de todos ellos seguido por Tabu y ES, si bien la hipótesis nula no puede rechazarse para estas dos últimas metaheurísticas. Fijándonos en las estrategias cooperativas, podemos comprobar que ambas superan de forma significativa a los tres métodos individuales, o lo que es lo mismo, muestran un comportamiento más robusto frente a los cambios de configuración. Cuando se comparan los modos de cooperación entre

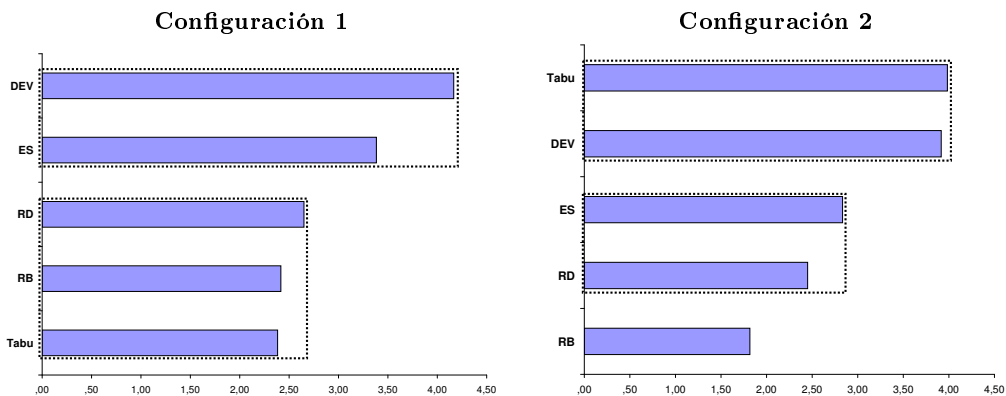


Figura 3: Cada barra representa el rango promedio devuelto por el test pareado no-paramétrico de Friedman para DEV, ES, Tabu, RB y RD con cada una de las correspondientes configuraciones. Las gráficas son análogas a las mostradas en la Figura 2.

ellos, en contra de lo que se puede pensar, la regla más básica (RB) muestra un comportamiento significativamente mejor que RD.

Una vez vistos los resultados de forma global, vamos a estudiar que ocurre en cada una de las configuraciones. Esta información puede verse en la Figura 3 donde mostramos, para ambas configuraciones, gráficas análogas a la anterior. El test pareado no-paramétrico de Friedman para múltiples comparaciones establece que se puede rechazar la hipótesis nula en ambos casos ($\alpha < 0.05$). Comenzando por los métodos individuales, vemos que para la primera configuración la búsqueda Tabu es significativamente el mejor algoritmo, seguido por ES y DEV en ese orden. Si observamos las dos gráficas, podemos comprobar que el cambio de la configuración de estos métodos altera dicho ranking. En el segundo caso ES es de forma significativa el método que muestra un mejor rendimiento mientras que Tabu pasa a ser la peor metaheurística individual, aunque la hipótesis de igualdad no puede rechazarse cuando se compara con la segunda, DEV.

A diferencia de los métodos individuales, ambos modos de cooperación muestran un comportamiento más estable frente al cambio de configuración. En el primer caso, tanto RB como RD son significativamente mejores que los dos peores algoritmos individuales, y aun-

que el test no-paramétrico devuelve un menor rango para Tabu, ninguna de las dos estrategias cooperativas funciona peor que dicho método de una forma significativa. En cuanto a la segunda configuración, los resultados realzan aún más los beneficios de la cooperación, ya que ambas estrategias cooperativas mejoran de forma significativa a las tres metaheurísticas individuales, a excepción de RD cuando se compara con SA. Esto indica que la cooperación lleva resultados similares o mejores para ambas configuraciones de los optimizadores.

6. Conclusiones

En este artículo hemos analizado la influencia que tiene en una estrategia cooperativa la variación de la configuración de las metaheurísticas implementadas por los agentes. Para ello se consideraron dos configuraciones que diferían tanto en términos de ajuste de los parámetros como en el uso o no de información específica del problema. Con el fin de tener una referencia del funcionamiento de la estrategia, esta se comparó frente a la versión individual de las heurísticas usadas como optimizadores.

La experimentación realizada con dos modos de cooperación distintos mostró que la estrategia cooperativa presentaba un comportamiento bastante robusto frente a los cambios

de configuración. Ambos modos de cooperación mejoraban de forma significativa a las metaheurísticas individuales cuando se consideraban de forma global los resultados obtenidos. Analizando cada configuración por separado, comprobamos que a pesar de las variaciones en el rendimiento de los algoritmos individuales, tanto RB como RD llevaban a resultados similares o mejores que los obtenidos por tales métodos.

Otro aspecto interesante de los resultados estriba en el hecho de que la regla más básica mejorara a la difusa, sobre todo teniendo en cuenta que en trabajos como [12] se muestra justamente lo contrario. Esto requiere una mayor investigación para evaluar si este hecho se debe únicamente al problema p-hub o bien a otros factores.

Referencias

- [1] J.E. Beasley. Obtaining test problems via internet. *Journal of Global Optimization*, 8(4):429–433, 1996.
- [2] A. Le Bouthillier and T. G. Crainic. A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers and Operations Research*, 32(7):1685–1708, 2005.
- [3] J.F. Campbell, A. Ernst, and M. Krishnamoorthy. *Facility Location: Applications and Theory*, chapter Hub location problems, pages 373–406. Springer-Verlag, Berlin, 2002.
- [4] I. G. del Amo, D. A. Pelta, A. D. Masegosa, and J. L. Verdegay. A software modeling approach for the design and analysis of cooperative optimization systems. *Software: Practice and Experience*, 2009. En prensa.
- [5] J. Ferber. *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. Addison-Wesley Longman Publishing Co., Inc., Boston, USA, 1999.
- [6] F. W. Glover and G. A. Kochenberger. *Handbook of Metaheuristics*. International Series in Operations Research & Management Science. Springer, 2003.
- [7] J. R. González, D. A. Pelta, and A. D. Masegosa. A framework for developing optimization-based decision support systems. *Expert Systems With Applications*, 36(3P1):4581–4588, 2009.
- [8] J. Kratica, Z. Stanimirović, and V. Filipović D. Tovšić. Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. *European Journal of Operational Research*, 182(1):15–28, 2007.
- [9] A. D. Masegosa, F. Mascia, D. Pelta, and M. Brunato. Cooperative strategies and reactive search: A hybrid model proposal. In *Learning and Intelligent Optimization (LION 3)*, volume 5851 of *Lecture Notes in Computer Science*, pages 206–220. Springer Berlin, 2009.
- [10] A. D. Masegosa, D. Pelta, I. G. del Amo, and J. L. Verdegay. On the Performance of Homogeneous and Heterogeneous Cooperative Search Strategies. In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2008)*, volume 236 of *Studies in Computational Intelligence*, pages 287–300. Springer Berlin, 2009.
- [11] M. O’Kelly and E. Morton. A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404, 1987.
- [12] D. Pelta, A. Sancho-Royo, C. Cruz, and J.L. Verdegay. Using memory and fuzzy rules in a co-operative multi-thread strategy for optimization. *Information Sciences*, 176(13):1849–1868, 2006.