

# CODEA: una herramienta para el aprendizaje de estrategias cooperativas

Juan Pedro Castro, Julio Brito, Belén Melián, José Andrés Moreno,  
José Marcos Moreno, Jonatan Ramos  
Departamento de Estadística, I.O. y Computación  
Universidad de La Laguna  
38271 La Laguna. Tenerife  
alu2952@etsii.ull.es, {jbrito,mbmelian,jamoreno,jmmoreno}@ull.es, alu3240@etsii.ull.es

## Resumen

CODEA es una herramienta diseñada para la experimentación con estrategias cooperativas descentralizadas en las que varios agentes resuelven conjuntamente un problema. CODEA es acrónimo de COoperative DEcentralized Architecture) y consta de un conjunto de clases programadas en C++ que flexibilizan al máximo la implementación de todo tipo topologías de comunicación, criterios de parada y etapas decisionales. Originalmente concebida para la implementación y el estudio de la cooperación entre metaheurísticas de búsqueda, a partir de dicho desarrollo se ha refinado con el propósito de dar soporte a la enseñanza y aprendizaje autónomo de estrategias cooperativas en varios campos de la Inteligencia Artificial como Ingeniería del Conocimiento, Agentes, Heurísticas, Sistemas de Ayuda a la Decisión y Paralelismo.

## 1. Introducción

La adaptación al Espacio Europeo de Educación Superior y a las demandas del entorno socio-económico, obligan a la Universidad a reflexionar sobre el papel que debe jugar en esta nueva etapa de cambios. Es necesario dar respuesta, desde todos los ámbitos, a los retos planteados con el fin de garantizar el cumplimiento de su misión, con el horizonte en la sociedad del conocimiento y el objetivo de una formación de calidad.

Algunas de las demandas están relacionadas con el diseño de herramientas, que sirvan de apoyo en los procesos de enseñanza-aprendizaje, y con la aplicación de las tecnologías de la información y las comunicaciones, que permitan ofertar docencia no presencial y/o ampliar la formación continua. En particular, avanzar en la implanta-

ción de herramientas que den soporte práctico al aprendizaje constructivo, autónomo, individualizado y por descubrimiento de los estudiantes.

La incorporación de tecnologías de la información y comunicaciones a la enseñanza es uno de los objetivos de las universidades y titulaciones en el nuevo contexto.

En este sentido la propuesta que se presenta pretende representar un avance en las herramientas tecnológicas flexibles que apoyan el desarrollo de competencias virtuales, la incorporación de nuevas estrategias docentes en el aula, el uso de recursos demostrativos y complementarios a la formación presencial y el aprendizaje autónomo del alumno.

El trabajo se estructura en cinco secciones. En la sección 2 se describe el entorno docente en el que se emplea la herramienta diseñada. Así, se define el concepto de estrategia cooperativa, se enumeran las asignaturas a las que va dirigida CODEA y los tópicos que pueden cubrirse con la herramienta, se listan los objetivos docentes que pretenden alcanzarse y se detalla la metodología docente. En la sección 3 se describe en detalle CODEA. La sección 4 trata sobre cómo aplicar CODEA en la práctica y la sección 5 enumera las conclusiones.

## 2. Docencia en estrategias cooperativas

La cooperación entre individuos se produce cuando interactúan para alcanzar algunos objetivos comunes. La cooperación es un fenómeno omnipresente en la *Naturaleza* [6], [7]. La cooperación y la colaboración son estrategias que mejoran la eficiencia y las competencias de los individuos y las organizaciones y forman parte de la cultura social [8]. Los estudios muestran un comportamiento colectivo inteligente, cada vez menos sorpren-

dente, en conjuntos de individuos muy simples. Las correspondientes formas de colaboración se han utilizado como inspiración para el diseño de sistemas artificiales útiles que resuelven problemas en diferentes contextos.

En la resolución cooperativa de un problema, varios agentes intercambian información relevante sobre las tareas realizadas con el propósito de mejorar la búsqueda de soluciones del problema. Una estrategia cooperativa consta de un conjunto de reglas que establecen cómo se efectúa el intercambio de información y qué acciones realiza cada agente en función de dicha información. En una estrategia cooperativa centralizada existe una entidad o agente central que dirige el intercambio de información y adopta decisiones explícitas sobre los pasos que han de realizar los diversos agentes. En una estrategia cooperativa descentralizada, cada agente tiene sus propias reglas para decidir cuándo y cómo intercambiar la información relevante con los otros agentes, y qué acciones realizar en base a la información recibida.

### 2.1. Contenidos

Los Algoritmos Genéticos (GA), la Búsqueda Dispersa (SS), la Optimización basada en Colonias de Hormigas (ACO), la Optimización basada en Enjambres (PSO) y el Re-encadenamiento de Trayectorias (PR) son algunas de las estrategias cooperativas más conocidas en Metaheurísticas [5][9].

Las estrategias cooperativas anteriores se incluyen como tópicos en diferentes asignaturas de primer, segundo y tercer ciclo impartidas por el departamento de Estadística, Investigación Operativa y Computación. Entre éstas destacamos las asignaturas *Heurísticas*, *Inteligencia Artificial* y *Agentes Inteligentes*, y el curso de doctorado *Metaheurísticas*. Además, en la asignatura *Sistemas de Ayuda a la Decisión* se emplean técnicas basadas en estrategias cooperativas para resolver diferentes problemas de decisión (problemas de decisión multicriterio, agregación de preferencias, ...).

### 2.2. Objetivos formativos

Los objetivos específicos planteados en la enseñanza de estrategias cooperativas pretenden que los alumnos adquieran las capacidades, competencias y destrezas que les permitan:

- Tener una visión amplia del campo de las estrategias cooperativas.
- Extraer y analizar la información de un problema necesaria para formalizarlo adecuadamente.
- Resolver dicho problema empleando estrategias cooperativas y tomar decisiones sobre las estrategias más adecuadas.
- Evaluar con actitud crítica las diferentes estrategias cooperativas empleadas para resolver un problema.
- Diseñar nuevos escenarios de aplicación adaptando las estrategias propuestas

### 2.3. Estrategias docentes

Las estrategias que se utiliza para alcanzar los objetivos de aprendizaje enumerados en la anterior sección combinan las exposiciones teóricas del docente con demostraciones prácticas, resolución de problemas y el aprendizaje autónomo del alumno. Este último se desarrolla durante sesiones prácticas y trabajos individuales en los que los alumnos deben diseñar, implementar y aplicar estrategias cooperativas a diferentes problemas.

Durante nuestra experiencia docente en el campo de las heurísticas, en general, y estrategias cooperativas, en particular, hemos detectado algunas dificultades de aprendizaje relacionados con la comprensión por parte de los alumnos de conceptos complejos y las habilidades necesarias para el uso de diversas estrategias. Entre las mayores dificultades específicas encontradas se hallan las siguientes:

- la complejidad inherente al diseño de estrategias cooperativas, sobre todo descentralizadas, ya que hay que especificar qué información se intercambia, a quién y cuándo se envía, qué se hace con la información recibida, ...;
- la imposibilidad de comprender correctamente el funcionamiento de algunas estrategias cooperativas, ya que éstas se basan en sucesos aleatorios o dependientes de los valores que toman ciertos parámetros;
- la tendencia de muchas implementaciones a funcionar como cajas negras, sin posibilidad de conocer, en modo paso a paso, el funcionamiento de las estrategias consideradas;
- el escaso tiempo que se puede dedicar a un alumno en las clases teóricas o prácticas. El

grado de asimilación de conceptos que varía de un alumno a otro y es difícil realizar una enseñanza personalizada con el moderado número de alumnos que cursan las asignaturas.

Por ello, desde hace tiempo, se planteó la necesidad de diseñar e implementar herramientas de apoyo a la enseñanza de estrategias cooperativas que faciliten la asimilación de conceptos difíciles en un contexto práctico. En esta línea de actuación se enmarca CODEA.

Los alumnos disponen de la herramienta, y de toda la documentación relacionada con la misma, al comienzo del curso. Durante las clases teóricas, el profesor explica las posibilidades que ofrece en relación al concepto que se esté tratando en ese momento y ofrece demostraciones prácticas.

Metodológicamente, CODEA se puede usar de dos maneras: como ayuda para impartir las clases teóricas y prácticas, y como herramienta de prácticas autoformativas por parte del alumno.

#### 2.4. CODEA como herramienta de autoaprendizaje

A continuación describiremos el uso de CODEA como herramienta de apoyo y autoaprendizaje en la asignatura *Heurísticas*. En ésta, una de las tareas que, individualmente, deben realizar los alumnos es la de analizar, resolver y evaluar el comportamiento de diferentes esquemas cooperativos en la metaheurística PSO [1]. Las variables de comportamiento empleadas son la mejor solución encontrada, el tiempo de CPU y la iteración en la que se encontró la mejor solución. Como problema base para la experimentación se emplea la minimización de la función Rastrigin. Se analizan tres topologías de comunicación: comunicación en estrella o comunicación todos a todos, comunicación en anillo (cada partícula se comunica con la siguiente y con la anterior) y comunicación k-aleatoria (cada partícula se comunica con k partículas seleccionadas aleatoriamente).

Al comenzar esta tarea, el profesor suministra a los alumnos una implementación casi completa del PSO usando CODEA. Los alumnos deben implementar las topologías de comunicación y analizarlas usando las variables de comportamiento descritas anteriormente. Para ello sólo deben describir apropiadamente la forma que tendrá la lista de agentes con los que un agente *i* puede comu-

nicarse (`coMetaheuristics[i]->setNeighborhoodList()`). En la figura 1 se muestran las líneas de código necesarias para el desarrollo de la práctica.

Descargando a los alumnos de una parte del trabajo de implementación, éstos pueden centrar su esfuerzo en la asimilación del concepto *topología de cooperación* y en las implicaciones prácticas que esta tiene en el comportamiento de un esquema cooperativo.

### 3. Estructura y funcionalidad de CODEA

Dada la complejidad del fenómeno de la cooperación el diseño de CODEA se ha diseñado siguiendo una metodología de capas virtuales. La descripción de CODEA se realiza siguiendo un esquema descendente a través de las capas virtuales. En cada capa encontramos una serie de elementos que se irán describiendo paulatinamente junto a las relaciones entre ellos. Contemplamos cuatro capas virtuales: la capa *conceptual*, la capa *estratégica*, la capa *funcional* y la capa *de implementación*.

La descripción de CODEA se hace de forma genérica. Se habla de individuos que forman parte de un grupo y que, fundamentalmente, tienen la capacidad de comunicar información. En la mente del lector debe estar presente que estos individuos pueden ser agentes, procesos o heurísticas que resuelven un problema.

#### 3.1. Capa conceptual

La cooperación se define conceptualmente como la acción de obrar conjuntamente con uno o varios miembros para lograr un fin común. De esta definición podemos subrayar tres ideas principales: idea de labor, idea de acción en grupo e idea de objetivo común. Aparece una dicotomía entre dos conceptos fundamentales: el equipo de trabajo y el trabajo en equipo. En el primero, se especifican el conjunto de elementos que trabajan conjuntamente según su papel o habilidad dentro del sistema; el segundo concepto hace referencia a qué procedimientos y metodologías se siguen para alcanzar los objetivos impuestos al sistema.

#### 3.2. Capa estratégica

En esta capa se describen las características generales de los individuos que forman parte del grupo de trabajo y cuáles son las relaciones que se pueden establecer entre ellos. Como en todo sistema

cooperativo, los individuos en CODEA tienen la capacidad de comunicar información a otros individuos. Además, nuestra arquitectura permite, en tiempo de ejecución, la modificación, eliminación y réplica de individuos.

1. Modificación: un individuo puede variar su comportamiento durante un ciclo de ejecución, si él determina que no está contribuyendo adecuadamente al buen rendimiento del grupo. Al modificar su comportamiento, conserva la información de que disponía hasta el momento.
2. Eliminación: la eliminación se produce bajo condiciones similares a las anteriores. Sin embargo, ahora el individuo se autodestruye sin dejar rastro en el sistema.

3. Réplica: si un individuo detecta que contribuye sustancialmente al rendimiento del grupo, crea una réplica de sí mismo. Con el paso del tiempo, esta réplica puede evolucionar para tener un comportamiento diferente al del individuo original.

Además de estas capacidades, un individuo puede dejar de trabajar si está entorpeciendo el rendimiento del sistema, no contribuyendo al mismo o haciéndolo de forma tímida respecto a los demás. La diferencia entre esta inactividad y la anterior capacidad de eliminación reside en el carácter definitivo, y la inactividad temporal. Si se dan las condiciones adecuadas, un individuo inactivo puede reactivarse.

---

```

/* Topología de comunicación estrella */

for (size_t i = 0; i < numberOfCoMetaheuristics; i++) {
    coMetaheuristics[i]->setNeighborhoodList(new
        all2allNeighborhood(coMetaheuristics));
}

/* Topología de comunicación anillo */

for (size_t i = 0; i < numberOfCoMetaheuristics; i++) {
    vector<coMetaheuristic*> ringgedNeighborhood;
    ringgedNeighborhood.push_back(coMetaheuristics[(i + 1)
        % numberOfCoMetaheuristics]);
    ringgedNeighborhood.push_back(coMetaheuristics[(i - 1)
        % numberOfCoMetaheuristics]);
    coMetaheuristics[i]->setNeighborhoodList(new
        all2allNeighborhood(ringgedNeighborhood));
}

/* Topología de comunicación: k-aleatorios */

for (size_t i = 0; i < numberOfCoMetaheuristics; i++) {
    vector<coMetaheuristic*> randNeighborhood;
    vector<unsigned> agentsId;
    unsigned k = 1 + mrandl.randInt(((int)sqrt(numberOfCoMetaheuristics)));
    while (randNeighborhood.size() < k) {
        unsigned newIdAgent = mrandl.randInt(numberOfCoMetaheuristics);
        if ((find(agentsId.begin(), agentsId.end(), newIdAgent)
            != agentsId.end()) && (newIdAgent != i)) {
            agentsId.push_back(newIdAgent);
            randNeighborhood(coMetaheuristics[newIdAgent]);
        }
    }
    coMetaheuristics[i]->setNeighborhoodList(new
        all2allNeighborhood(randNeighborhood));
}

```

---

Figura 1. Código para la práctica

El esquema de comunicación más sencillo contempla la existencia de una serie de elementos que reciben y generan un flujo de información. Las salidas de unos elementos son las entradas de otros y viceversa. Sin embargo, existen otros muchos modelos en los que no se mantiene una comunicación continua y en los que, además, se establecen jerarquías para comunicar la información.

Aunque los sistemas cooperativos implementados en CODEA pueden no estar sujetos a un patrón fijo en el flujo de operaciones, sí deben producir algún tipo de información que se pueda comunicar. La comunicación tampoco está sometida a ninguna restricción lo que permite crear varios modelos de interrelación entre individuos y, además, ésta no ha de ser necesariamente bidireccional. Es decir, si un individuo envía cierta información a otro, el segundo no tiene porqué realizar un procedimiento reflejo.

### 3.3. Capa funcional

En esta capa describiremos qué roles pueden desempeñarse en el sistema, cómo se desempeñan estos roles y cómo se produce la comunicación.

Cada individuo desempeña un rol dentro del grupo de trabajo. Las habilidades de cada individuo vienen caracterizadas por una serie de fases o estadios finitos en los que puede encontrarse cada individuo. Algunas de estas fases son comunicación, resolución, decisión, etcétera. Esto permite tener elementos en el sistema con roles bien definidos o, por el contrario, roles híbridos en los que cada elemento posee una combinación de fases. Además, no existe una restricción en cuanto a la sincronía de las fases en los elementos del sistema. Se podrían tener, por ejemplo, elementos resolutores, cuya única función es la de aportar nuevas soluciones al sistema; elementos comunicadores, encargados de enviar y recibir flujos de información entre una lista variable y restringida de candidatos, etc.

Todos estos elementos del grupo de trabajo desarrollan sus actividades en un bucle que se ejecuta iterativamente hasta alcanzar una condición general de parada, siendo ésta no preestablecida ni impuesta por el sistema. En dicho bucle, cada componente del grupo posee un turno para ejecutar su fase actual. Durante cada turno, no todos los individuos están obligados a desarrollar la misma actividad (por ejemplo, resolutora). Debido a que

cada elemento ejecuta la fase que desea en su turno, el sistema flexibiliza tanto una comunicación síncrona como asíncrona. Esto es posible porque cada miembro del grupo de trabajo posee un buzón de mensajes a modo de buffer en el que se van almacenando los mensajes que llegan de otros miembros del grupo. Cuando el individuo lo considere oportuno, atiende estos mensajes y realiza las operaciones pertinentes sobre ellos.

El trabajo en equipo exige la comunicación entre los individuos que lo conforman. Ya hemos visto de qué forma se comunican los mensajes, pero CODEA soporta que cada individuo pueda enviar en su mensaje la información que desea. Si el receptor recibe una información que no puede interpretar o no le es útil, envía unos códigos de retorno al emisor que le indican a éste que su mensaje no tiene ningún valor o simplemente no tiene sentido.

En una organización o sistema de trabajo en el que la cooperación es fundamental que los miembros diseñen estrategias para comunicarse.

En muchos casos, la comunicación “todos a todos” supone una carga demasiado pesada para el grupo. Por ello, en CODEA se ha diseñado a modo de módulo la libre implementación del criterio para enviar la información que desea comunicarse. Una posibilidad (contemplada actualmente en CODEA) es considerar un sistema de puntuaciones basado en el provecho que se ha obtenido de la información recibida. El emisor de un mensaje puntúa satisfactoriamente al receptor si éste, al operar con la información recibida, mejora su comportamiento. En posteriores fases de comunicación, los individuos envían la información de que disponen a sus compañeros mejor puntuados, pues son éstos los que más provecho han obtenido de la misma. También envían la información a un subconjunto del resto de individuos, porque, al tratarse de un sistema que cambia muy rápidamente, no existe la certeza de que la información que no fue útil en un momento tampoco lo sea posteriormente.

### 3.4. Capa de implementación

A continuación, se describe cómo se han implementado los anteriores principios básicos de comunicación. Tal y como se describió en las capas anteriores, los miembros o componentes del grupo de trabajo o sistema están inmersos en un bucle cuya condición de parada no está delimitada ex-

plicítamente. En su lugar, existe una clase virtual que implementa un método abstracto de parada. A partir de dicha clase, se extienden los criterios de parada elegidos por el desarrollador. Entre los criterios de parada que podrían utilizarse destacamos el tiempo máximo de vida del grupo, el número de soluciones evaluadas o el nivel de estancamiento del grupo.

Los mensajes se definen formalmente como cajas negras de información. Esto es, si un elemento desea enviar una estructura de datos determinada, primero crea un paquete en el que inserta la información, añade un encabezado muy simple y finalmente lo envía al buzón del destinatario. Las estructuras de datos se empaquetan utilizando un objeto instancia de una clase de propósito general que permite alojar dentro cualquier tipo de información. Una vez que el destinatario revisa su buzón, debe desempaquetar la información y devolver un código de error en caso de que la información no le sea útil.

Cuando un individuo ha recibido y usado la información, ha de ponderar si la misma le ha sido fructífera o no. En el primer caso, le enviará un mensaje al emisor de la información para que éste aumente su puntuación. Este sistema de recompensas crea enlaces fuertes dentro de los elementos del grupo. De esta forma, la próxima vez que el proceso entre en una fase de comunicación, revisará las puntuaciones y establecerá la lista de candidatos a los que enviar la nueva información. Además, como ya se dijo en la capa funcional, a la lista de candidatos se añaden algunos destinatarios adicionales elegidos al azar.

La comunicación, la decisión de destinatarios, la resolución,... forman parte del conjunto de fases que puede desarrollar cada individuo. Éste posee un vector de fases por las que transita cada vez que se le concede un turno. A su vez, cada fase es una instancia de una clase que extiende una clase virtual que define qué estructuras han de tener en común todas las clases. El polimorfismo inherente de esta práctica, permite a cada elemento crear y destruir fases en tiempo de ejecución, instanciando y borrando objetos de su vector de fases.

#### 4. Aplicación

La importancia de CODEA estriba en la originalidad de mostrar en un primer plano de cómo se trabaja con modelos cooperativos en la resolución

de muy diversos problemas. Así mismo, el estudio de estrategias cooperativas, en este caso descentralizadas, no se reduce a la obtención de soluciones con mayor calidad, sino que conlleva una reflexión de las topologías de comunicación emergentes que reflejan comportamientos que pueden ser observados en la naturaleza.

Dado que la cooperación es un fenómeno complejo en el que intervienen muchas variables, el diseño de una herramienta que permita una ejecución eficiente y la configuración flexible de todas las variables puede no ser fácil de entender. El desarrollo se acompaña de tutoriales en los que se explica desde la naturaleza de CODEA hasta cómo operar con ella. Este es el punto clave sobre el que se sostiene nuestra propuesta: ofrecer CODEA como recurso didáctico en el que los alumnos aprenden cómo operar con un sistema cooperativo descentralizado.

Según el diseño actual de CODEA, los usuarios podrán operar sobre:

- *Tipo de problema:* no existirán restricciones sobre el tipo de problemas con el que ejecutar el sistema. Tan sólo se exige que la implementación de estos problemas sigan una serie de normas generales de la plataforma inherentes a la programación orientada objetos (POO).
- *Topología de comunicación:* se pueden crear tantas topologías de comunicación como se deseen. Se han probado satisfactoriamente topologías estándar como la topología en estrella, topología en anillo, topologías k-aleatorias, así como topologías más complejas basadas en puntuaciones. Además, dichas topologías pueden ser modificadas en tiempo de ejecución o bien ser nulas. En este último caso, los agentes no se comunicarían, lo que sirve como banco de pruebas para comparar las soluciones obtenidas por sistemas individualistas y cooperativos.
- *Información objeto de la comunicación:* CODEA fue diseñado siguiendo la filosofía de que cada agente puede decidir qué enviar. De esta forma, se define un mensaje como un contenedor de información arbitraria. Se puede enviar un mensaje que contenga cualquier estructura de datos, incluyendo objetos u objetos que contenga otros objetos.
- *Criterios de parada:* Tanto el sistema como los agentes poseen objetos que les confieren la

capacidad de parar usando criterios predeterminados por el usuario. Dichos criterios pueden ser programados ajustándose a las necesidades de cada problema y también pueden variar el tiempo de ejecución.

- *Fases*: cada agente transitará por un conjunto de estados como lo haría un autómata finito; dichos estados son considerados fases. Tanto la cantidad de fases como el tipo son de libre implementación y modificables también en el curso de ejecución del programa. Se puede considerar que el conjunto de estados o fases de un agente es lo que determina su comportamiento o rol en el sistema. Es posible crear fases nulas en la que los agentes no ejecutan acción alguna y, además, el curso de los agentes por las fases puede ser tanto síncrono, en el que todos los agentes cursan una misma fase, como asíncrono, en el que, por ejemplo, un agente puede recibir información mientras ejecuta una fase resolutoria.
- *Agentes*: Tampoco existen restricciones sobre qué tipo de agentes crear. Es posible crear un sistema homogéneo con agentes idénticos que realizan operaciones colaborativamente o, por el contrario, hacer agentes especializados que sólo ejecuten algunas actividades concretas (por ejemplo, data mining). Así mismo, lo usuarios pueden crear reglas para que los agentes desactiven o supriman su actividad, creen replicas de sí mismos o nuevos agentes o muten a otros agentes, todo esto en tiempo de ejecución.

Aunque pueda parecer complejo implementar todas estas opciones y estrategias, una vez aprendidas las premisas que rigen su funcionamiento, es bastante rápida la tarea de implementar módulos como agentes, topologías de comunicación, etcétera. Por otro lado, la herramienta se ofrece sobre una plantilla y, por lo tanto, todas estas variables no se han de empezar a programar desde cero. Es decir, se ofrece un marco de trabajo en el que se pueden ir implementando estas operaciones paulatinamente, según las necesidades y el enfoque que se le pretenda dar al uso de la herramienta para la docencia.

Hasta la fecha, CODEA ha sido ejecutada con búsquedas locales resolviendo el problema del viajante de comercio con topologías de todo tipo, entre ellas una basada en puntuaciones. En ésta,

los agentes valoraban a sus emisores según la utilidad de la información recibida; de esta forma, en la etapa o fase en la que se eligen los destinatarios, cada emisor enviaba información a aquellos destinatarios que obtenían más productividad de la información. Esto generaba que la topología de comunicación emergiese de las necesidades de los agentes como en una sociedad real. Los resultados de estos experimentos fueron presentados en el Congreso Internacional de Metaheurísticas (MIC07) [3].

CODEA también se ha utilizado para el desarrollo de un sistema de Optimización por Enjambre de Partículas (PSO) en el que cada partícula era simulada por un agente. Dicho esquema fue diseñado para probar su eficacia y eficiencia sobre varias funciones de optimización continua: la función de Rastrigin, función de Rosenbrock, función de Schwefel, la función de Ackley y la función de Griewangk. Así como tres topologías para comparar sus resultados: en estrella, anillo y topología de  $k$ -aleatorios. Se obtienen muy buenos resultados, alcanzando la solución óptima en casi todos los casos y los mismos se presentaron en el segundo Workshop de Estrategias Cooperativas para la Optimización Inspiradas en la Naturaleza (NICSO07) [2].

## 5. Conclusiones y futuros trabajos

CODEA ofrece suficiente información y soporte como para diseñar prácticas o proyectos sobre ella. Dado que la cooperación y distribución de procesos es uno de los temas importantes y recientes del momento, el uso de CODEA como soporte para la docencia produce una significativa diferencia en los resultados de aprendizaje autónomo por parte de los alumnos de aquellos elementos indispensables con los que se tiene que enfrentar cualquier esquema cooperativo. CODEA ofrece un sistema flexible y simple de implementación de estrategias y resultó útil para su estudio por partes y como un conjunto. A su vez, ofrece una 'arena' en la que se puede experimentar con infinitas combinaciones de parámetros.

Actualmente, se está estudiando la aplicación de CODEA a problemas reales de localización, especialmente del diseño de rutas de distribución para problemas VRPTW (Problemas de Ruta de Vehículos con Ventanas de Tiempo) con incertidumbre en los tiempos de recorrido [1].



Todo esto demuestra la importancia de operar con una herramienta que ofrece un marco de trabajo real, cuyo uso no puramente académico le confiere un aliciente extra con el que los usuarios sentirán que su tiempo utilizando la misma es aprovechado.

Además, es nuestra intención que los usuarios puedan colaborar en el futuro desarrollo de la herramienta, bien con sugerencias de cómo mejorarla o el informe errores en la misma. Esto, creemos que invitará a los alumnos a sentirse partícipes y al desarrollo y utilización de estrategias docentes basadas en proyectos en las distintas materias.

### Agradecimientos

Este trabajo está parcialmente financiado a través de los proyectos TIN2005-08404-C04-03 (70% de fondos FEDER) y PI042005/044.

### Referencias

- [1] Brito, J., Campos, C., J.P. Castro, F.J. Martínez, B. Melián, J.A. Moreno, J.M. Moreno. Fuzzy Vehicle Routing Problem with Time Windows. *12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2008)*, Málaga, 22 al 27 de Junio de 2008.
- [2] Castro Rodríguez, J.P., Martínez, F.J., Melián Batista, B., Moreno Pérez, J.A., Moreno Vega, J.M. CODEA: An architecture for Nature-inspired designing Cooperative Decentralized Heuristics. *II Internacional Workshop on Nature Inspired Cooperative Strategies for Optimization, NICSO 2007*. Acireale, Sicilia, Italia, 11 al 14 de No-viembre de 2007
- [3] Castro Rodríguez, J.P., Ramos, J., Melián Batista, B., Moreno Pérez, J.A., Moreno Vega, J.M. CODEA based on Metaheuristics. *V Metaheuristics International Conference, MIC 2007*. Montreal, Canadá, 25 al 29 de Junio de 2007.
- [4] Clerc, M. *Particle Swarm Optimization*. ISTE, 2006.
- [5] Duarte Muñoz, A., Pantrigo Fernández, J.J. Gallego Carrillo, M. *Metaheurísticas*. Dykinson, 2007.
- [6] Eberhart, R.C., Shi, Y., Kennedy, J. *Swarm Intelligence*. Kaufman, 2001
- [7] Engelbrecht, A.P. *Fundamentals of Computational Swarm Intelligence*. Wiley, 2005
- [8] Grundel, D., Murphey, R., Pardalos, P.M. (eds.) *Theory and Algorithms for Cooperative Systems*. Series on Computers and Operations Research - Vol. 4., World Scientific 2004
- [9] Melián Batista, B., Moreno Vega, J.M., Moreno Pérez, J.A. Metaheurísticas: una visión global. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial* 2003. Número 19, Volumen 2, páginas 7-28.