Estimación de estrategias de generación de energía eléctrica mediante algoritmos genéticos coevolutivos con fitness multiobjetivo

Enrique A. de la Cal Marín
Luciano Sánchez Ramos

Resumen— **A partir de 1998, el precio de la energía eléctrica en España comenzó a determinarse a partir de las ofertas de los generadores en el mercado eléctrico, también denominado pool eléctrico, dejando de ser regulada por el gobierno.**
La evolución de este pool competitivo puede verse como un problema de teoría de juegos. En este trabajo se presenta un modelo genético para el problema anterior, y una simulación del mismo aplicada a un problema artificial.
Se han ensayado diferentes configuraciones del modelo genético, empleando tanto un esquema genético coevolutivo (múltiples poblaciones) como no-coevolutivo (una población), aplicando ambos a la simulación de mercados competitivos; además se comparan dos funciones de fitness equivalentes, una escalar y la otra multiobjetivo, aplicadas sobre los dos esquemas genéticos para determinar cual de los dos fitness es más efectivo. El método consiste en una técnica de análisis inteligente de datos (IDA), que ha sido empleado para analizar escenarios artificiales similares al mercado eléctrico español.
El objetivo final de nuestro método es aproximar las estrategias individuales de los agentes del pool. Estos resultados son un primer paso hacia la inducción de las estrategias individuales óptimas a partir de datos reales del mercado eléctrico español, suponiendo que se comportara como un mercado oligopolístico perfecto.

Palabras clave— **Mercado oligopolístico, Estrategia coevolutiva, Fitness multiobjetivo**

I. Introducción

A. Objetivo

LOS precios de la energía eléctrica en España se calculan para cada hora del día, y junto con la cantidad de energía negociada, constituyen lo que se denomina un punto de mercado. Los generadores planifican su producción y envían los precios que demandan por su energía al Operador de Mercado (OM), el cual calcula un único precio, para todos los agentes, resultado de combinar la suma de las ofertas con las demandas de los consumidores mediante el proceso de casación, [1]. El principal problema de las empresas generadoras de energía eléctrica estriba en la planificación diaria de precios y energía generada. Por ello, en este trabajo, se propone un modelo basado en algoritmos genéticos que realizará una estimación de las estrategias de oferta que cada agente ha enviado al OM, datos que no son públicos, a partir de los puntos de mercado correspondientes a un período de tiempo determinado, que sí son públicos.

Departamento de Informática, Campus de Viesques, s/n, Universidad de Oviedo, 33203 Gijón (España)
delacal@etsiig.uniovi.es, luciano@lsi.uniovi.es

B. Técnicas de análisis inteligente de datos aplicado al mercado eléctrico español

El número de firmas (cuatro firmas dominantes) y el carácter competitivo del proceso de casación español caracteriza este mercado como un modelo de mercado oligopolista. Este tipo de modelo de mercado sugiere para su simulación un modelo genético específico, un modelo coevolutivo (ver, por ejemplo, [2]), [3]), ya que cada firma es un agente competitivo diferente, con su características específicas. De esta forma, cada compañía puede representarse con una población que, al evolucionar con la estrategia genética, nos permite estimar las ofertas de generación.

Antes, se han propuesto diferentes modelos para problemas similares al nuestro: el modelo cobweb propuesto por Leontief [4] postula un modelo lineal de ofertas y demanda con un solo mercado donde n agentes, que producen el mismo producto con el mismo coste cuadrático, envían una oferta constante al mercado y la van ajustando iterativamente con el objetivo de maximizar su beneficio. Los primeros trabajos con agentes inteligentes sobre el modelo cobweb los realizan Arifovic [5], donde la cantidad ofertada por cada agente se codifica como una secuencia de bits y emplea el operador genético *elección* para reproducir mediante su algoritmo genético el precio de equilibrio del modelo cobweb. Franke [6] trabaja con un modelo muy similar donde la curva de demanda se ha perturbado mediante la función de ruido AR(1). Cabe reseñar el trabajo de Curzon [7], en el que simula, empleando un algoritmo genético coevolutivo, problemas típicos de organización industrial: Cournot, Bertrand, monopolio y un modelo simple de mercado eléctrico extraído del modelo real inglés, donde el esquema de competencia es basado en precios, al estilo Bertrand y no basado en cantidades como es el español. Y por último, comentar el reciente trabajo de Fushuan [8], donde se emplean curvas de oferta de forma lineal, $\alpha P + \beta$, para modelar el mercado eléctrico californiano.

En resumen, podemos decir que nuestro trabajo presenta dos diferencias importantes respecto a los trabajos anteriores: por un lado, en nuestro modelo, el equilibrio viene marcado por la bondad del ajuste del modelo a una nube de puntos de mercado y no como un equilibrio guiado por el beneficio máximo de los agentes como hacen todos los trabajos anteriores y, por otro lado, y quizás la principal aportación de

nuestro trabajo es que, la forma de nuestras curvas de oferta son muy similares a las ofertas reales del mercado eléctrico español, esto es, lineales por tramos y, en el caso de los modelos citados anteriormente las curvas de oferta son, o bien lineales, o incluso constantes.

C. Decisiones

Ya que las firmas de un mercado real oligopolista compiten de forma independiente sin interacciones directas entre ellas, de forma muy similar a los agentes de un problema de teoría de juegos; se decidió emplear un esquema genético coevolutivo, comparando este esquema con un modelo equivalente, implementado sobre un esquema genético convencional , no-coevolutivo, (con una sola población). Se demostrará empíricamente que la estrategia coevolutiva es más eficaz que la estrategia no-coevolutiva equivalente, como también corroboran Hillis [9] y Potter [10].

Respecto a la elección del fitness, podemos decir que, el problema planteado requiere para su resolución la imposición de múltiples restricciones que se detallarán en el apartado II-C, las cuales determinarán una única solución al problema de aproximación de curvas. Según esto, se plantearon dos tipos de fitness: uno en el que los objetivos para recuperar las estrategia se mapean a un único valor, obteniendo un fitness escalar, y otro, en el que los objetivos se considerarán como independientes, tratándose de un fitness multicriterio. Los resultados numéricos nos indican que este segundo fitness es más eficaz que la versión escalar en todos los casos de estudio, como también se ha demostrado en [11].

Para comprobar la calidad de nuestro método, hemos diseñado un problema artificial, inspirado en la configuración del mercado eléctrico español, en el cual las curvas de oferta son conocidas (los datos reales no se conocen directamente, ya que los agentes no publican sus estrategias de precios),y hemos aplicado diferentes métodos, tanto genéticos como clásicos, para resolver el problema de inducir las estrategias de oferta de cada agente.

D. Aplicaciones de este trabajo y sumario

Los resultados obtenidos pueden aplicarse en dos sentidos:

- Desde el punto de vista del agente generador, podremos simular el comportamiento de un nuevo pool en el que hayamos modificado ligeramente nuestra estrategia de precios. De esta forma, podremos medir la mejora de nuestro beneficio y, eventualmente, ajustar nuestra curva. Se supone que la modificación es lo suficientemente pequeña como para que el resto de agentes no reajusten sus precios, cosa que, si ocurriera, nos llevaría a otro escenario (esta última situación se discutirá en la sección sobre "Trabajo Futuros").
- Desde el punto de vista del operador de mercado, desearemos conocer las estrategias óptimas teóricas para un conjunto de puntos de mercado correspondientes a una situación oligopolística perfecta, similar al problema de Cournot, [12]. Con esta información, podríamos estimar la diferencia entre el beneficio real del pool y el beneficio máximo teórico, si la competencia fuera perfecta, y de esta forma, podríamos detectar la presencia de acuerdos entre los generadoras, práctica que no está permitida por ley.

El índice de este trabajo se organiza como sigue. En la sección II describimos la metodología propuesta. El método basado en técnicas clásicas se describe en detalle en la sección III. La sección IV incluye el diseño del problema sintético que se empleará para ilustrar la aplicación de los métodos coevolutivo, no-coevolutivo y basado en técnicas clásicas. En la sección V, se analizan los resultados correspondientes a los experimentos numéricos realizados sobre los métodos anteriores, y el trabajo finaliza con el apartado de conclusiones y trabajos futuros.

II. Metodología propuesta

El equilibrio económico en mercados eléctricos competitivos no puede calcularse fácilmente por método analíticos, por lo que se han venido usando métodos de aproximación con modelos lineales, [5] [7] [6] y [8]. Estas aproximaciones introducen grandes imprecisiones en los resultados, que deben eliminarse del modelo, debido sobre todo, a que las curvas de oferta presentan formas no lineales. Se demostrará que un algoritmo genético coevolutivo (donde cada población representa una firma del pool) permite modelar un mercado eléctrico artificial y compararlo con modelos genéticos no-coevolutivos y aproximaciones lineales.

La datos que constituyen la entrada de nuestro algoritmo son un conjunto de puntos de mercado, donde cada punto de mercado incluye el precio por unidad de energía y número de unidades negociadas. Al final obtenemos un sistema de clasificación que produce una segmentación de la colección de puntos de mercado en un cierto número de clases y una oferta poligonal para cada clase. Cada una de estas ofertas trata de minimizar el error medio entre la cantidad de energía de generación estimada y las cantidades reales para los puntos de mercado estudiados. Los costes de producción se han incluido en el modelo de la siguiente forma: hemos añadido una restricción a la búsqueda, y se ha asumido que los beneficios (por unidad de energía) de todas las firmas son iguales para cada mercado. Estas dos condiciones nos llevan a un problema de optimización multiobjetivo no-lineal.

Los individuos del esquema coevolutivo se corresponden con la estrategia de un agente; por lo que tendremos tantas poblaciones como firmas compitan en el mercado y el fitness para un individuo se calculará haciéndolo competir contra una combinación de individuos del resto de poblaciones. Por lo otro lado, un individuo del esquema no-coevolutivo es un conjunto de estrategias correspondientes a cada una de las firmas del mercado, por lo que se le puede

asignar un fitness directamente. La diferencia en la representación así como el fitness y los operadores genéticos de uno y otro esquema se discuten en detalle en las siguientes secciones.

A. *Representación de los individuos de los modelos coevolutivo y no-coevolutivo*

Como se ha comentado, cada agente elige una estrategia de un conjunto de "estrategias poligonales", en función de la salida del sistema de clasificación basado en reglas, que segmenta el mercado de acuerdo a diferentes variables de entrada (día de la semana, estación, lluvia, temperatura media, etc.). Cada estrategia poligonal esta formada por una secuencia de "estrategias simples" con forma de recta. Tanto el número de estrategias poligonales de un agente, como el número de estrategias simples que las componen es fijo, de modo que un individuo está formado por una cadena de números que comprende: dos pares de números reales (los extremos de una recta) por cada estrategia simple además de la definición del clasificador. Las simulaciones llevadas a cabo han empleado una sola variable de entrada al clasificador, la hora del día, de esta forma el clasificador puede expresarse con una cadena de números, que representan el fin y comienzo de los tramos horarios que definen cada cluster de puntos de mercado, (ver figura 1).

Fig. 1. Representación genética de un individuo del esquema coevolutivo, con una oferta poligonal compuesta por 3 estrategias simples y un clasificador genérico que segmenta los mercados en 4 clusters horarios, y la representación gráfica de las estrategias poligonales

En el ejemplo de la figura 1, el clasificador (c1, c2, c3) aplicado a un punto de mercado correspondiente a la hora **h** de un día, representaría el siguiente conjunto de reglas a aplicar al competir en un pool:

```
IF (h <= c1) THEN Poligonal0
 ELSE IF (h <= c2) THEN Poligonal1
  ELSE IF (h<= c3) THEN Poligonal2
   ELSE THEN Poligonal3
```

De esta forma nuestro agente tiene una estrategia poligonal distinta para 4 franjas horarías del día, que serán las clases inducidas. Estas franjas horarias podrás ser diferentes para cada agente.

Los individuos del esquema no-coevolutivo estarán formados por la concatenación de tantos individuos del esquema anterior como agentes existan. Debe tenerse en cuenta que la segmentación de mercados puede ser distinta para cada agente, ya que no comparten los parámetros del clasificador.

B. *Operadores genéticos en los esquemas coevolutivo y no-coevolutivo*

En el esquema coevolutivo el cruce de dos individuos debe afectar a las dos partes (el conjunto de estrategias simples y el clasificador):

- Las estrategias simples: Para prevenir una excesiva variación en los individuos y la consecuente dispersión de los datos hemos determinado empíricamente que es mejor seleccionar aleatoriamente una de las estrategias simples de los individuos y cruzarla, que cruzarlas todas sistemáticamente. El cruce de una estrategia simple está basada en la media de los coeficientes de los padres, con una ligera variación aleatoria. Para que la estrategia resultante del cruce sea válida deberá ser creciente en precio y cantidad.

  ```
  i=random(No. de estrategias simples)
  hijo.a = 0.5*(madre[i].a + padre[i].a)
           *random[0.8..1.20]
  hijo.b = 0.5*(madre[i].b + padre[i].b)
           *random[0.8..1.20]
  ```

  Donde a será precio y b cantidad.
- Parámetros del clasificador: Es necesario producir un clasificador válido. Como hemos restringido los parámetros a una sola variable de entrada, la hora del día, esto significará que la lista de parámetros deberá ser creciente. El algoritmo que se ha empleado es el siguiente:

  ```
  for i:= 1 to c do
   hijo.clas[i] =
      1/2*(madre.clas[i] + padre.clas[i])
         *random[0.8, 1.2]
  ```

No ha sido necesario incluir el operador mutación dado que el propio operador de cruze incluye un factor aleatorio.

La extensión de estas operaciones al esquema no-coevolutivo consiste en seleccionar aleatoriamente un agente de un individuo no-coevolutivo y cruzar su clasificador y colección de estrategias simples.

C. *Cálculo del Fitness multiobjetivo para un agente*

Se han considerado dos criterios para ordenar dos individuos. Un individuo k_1 es mejor que otro k_2, cuando la energía total generada por el pool de k_1 para un precio de mercado, está más cerca de la energía real del punto de mercado correspondiente que la energía total generada por el pool en el que haya competido el individuo k_2 y además el beneficio unitario de las firmas seleccionadas para competir contra k_1 es más parecido entre ellos que los de k_2. Consecuentemente, el fitness multiobjetivo propuesto está basado en dos objetivos.

Sea n_f el número de firmas, n_m el número de puntos de mercado, q(p) la estrategia agregada de generación real, q'(p') la estrategia agregada de generación estimada, p_m el precio marginal real para el punto de mercado m y p'_m el precio marginal estimado para el punto de mercado m. Según esto el primer objetivo será Fitness_a:

$$Fitness_a = \frac{\sum_{m=1}^{n_m}(\sum_{f=1}^{n_f} q_f(p_m) - q'_m(p'_m))^2}{n_m} \quad (1)$$

En otras palabras, el Fitness_a, es la suma media de los cuadrados de las diferencias entre la energía predicha y la energía realmente vendida.

El segundo objetivo, Fitness_b, minimiza las diferencias entre los beneficios unitarios (Bu_f^m) de cada firma f respecto a cada punto de mercado m. Siendo q'_f la estrategia estimada de generación para la firma f y C_f la función de coste para la firma f:

$$Bu_f^m = (q'_f(p'_m) * p'_m - C_i(q'_f(p'_m))/q'_f(p'_m) \quad (2)$$

Hemos decidido medir las diferentes entre beneficios unitarios con su varianza

$$\sigma^2(\text{Bu}^\text{m}) = \frac{\Sigma_{f=1}^{n_f}(Bu_f^m - media(Bu^m)^2)}{n_f} \quad (3)$$

$$Fitness_b = \sum_{m=1}^{n_m} \sigma^2(\text{Bu}^\text{m}) \quad (4)$$

D. *Fitness multiobjetivo para el esquema coevolutivo*

El fitness para los individuos del esquema coevolutivo (tantas poblaciones como firmas) es diferente de los individuos no-coevolutivos. En el esquema coevolutivo, el fitness se calcula haciendo competir a nuestro individuo contra individuos seleccionados aleatoriamente del resto de poblaciones (parásitos), de modo que el fitness de los individuos seleccionados será diferente.

E. *Equivalencia entre el fitness multiobjetivo y el monoobjetivo*

En este trabajo demostramos que la propuesta multiobjetivo es mejor que la equivalente monoobjetivo para los problemas estudiados. Aún cuando existen herramientas suficientes [13], que permitan mapear fitness multiobjetivo a fitness monoobjetivo, hemos elegido la suma como función de mapeo por su simplicidad de aplicación. En particular, una media ponderada de Fitness_a y Fitness_b será siempre menos eficaz que la alternativa multiobjetivo. Sean,

$$Fitness_{multi} = < Fitness_a, Fitness_b > \quad (5)$$

$$Fitness_{mono} = Fitness_a + Fitness_b \quad (6)$$

donde < > representan "concatenación".

Fitness_{multi} explorará ambos objetivos en paralelo, alcanzando más rápido un espacio de búsqueda mayor que Fitness_{mono}. Esta característica hace de Fitness_{multi} una función más eficiente y eficaz que la equivalente Fitness_{mono}. En la sección V-B se demostrará que Fitness_{mono} no alcanza la solución global óptima sobre nuestro problema sintético en la mayoría de los casos, cosa que si hace Fitness_{multi}

III. Método clásico

El método alternativo propuesto está basado en la clasificación de una serie de puntos de mercado en base a sus características, y un ajuste de un modelo simple para cada cluster. Una solución simple a este problema consiste en realizar una clasificación previa de los puntos de mercado, por ejemplo con el algoritmo K-means, y a continuación un ajuste genético a cada cluster con individuos sin clasificador, ya que la clasificación se ha realizado previamente mediante el algoritmo K-means (ver Figura 2). Este método tiene el gran inconveniente de que obliga a todas las firmas a emplear la misma estrategia de clasificación, que vendrá impuesta por el clustering de los puntos de mercado. Cosa que no ocurre en nuestro esquema genético donde el clasificador evoluciona con cada firma.

Fig. 2. Aproximación de curvas de generación a partir de a) nube de puntos de mercado mediante la técnica KGM: b) Clustering + optimación genética

En general esta solución no es evidente porque los criterios que debe tener en cuenta el método de clasificación no son evidentes. Se planteará un nuevo problema derivado del anterior donde los puntos se clasificarán *a priori* para justificar lo dicho. Este algoritmo se denominará KGM (K-means Genetic Model). Los modelos no-coevolutivo y coevolutivo con clasificador integrado se denominarán GM y CGM respectivamente.

IV. Problema artificial de estudio

Se ha generado un problema artificial muy sencillo que nos permite aplicar los tres métodos de estudio, GM, CGM y KGM, para poder recuperar una aproximación de las curvas de generación individuales. Los puntos de mercado sintéticos se han generado agrupados en dos clusters (ver figura 3): un cluster corresponde a las horas de la 1 a las 10 (MP1-MP10) y otro de las 11 a las 20 horas (MP11-MP20), ambos grupos de puntos se encuentran alineados sobre dos rectas. Se ha seleccionado los puntos agrupados en dos clusters, con el fin de permitir la aplicación del modelo KGM, pero se han distribuido de manera

que haga patente la dificultad de la segmentación, mediante el algoritmo K-means, de dicha nube de puntos empleando las dimensiones precio y energía

Nuestro objetivo es recuperar las 4 curvas de oferta individuales que originaron los puntos de mercado MP1-MP20. Para ello nuestro método parte de los siguientes datos:

1. Se han definido 20 puntos de mercado, $MP_{i=1}^{20}$=(precio, energía). Y se sabe que están ordenados de izquierda a derecha con respecto a su número, que representa la hora del día, (ver figura 3).

Fig. 3. Nube de puntos de mercado para problema artificial formada por dos clusters que se cortan en forma de cruz

2. Las funciones de demanda que han generado la nube de puntos de mercado tienen la misma elasticidad, -2. Por ejemplo, la función de demanda para el punto de mercado (precio=16.67, energía=66.67) es -2*precio+100.
3. Las 4 compañias presentan las siguientes funciones de costo lineal respectivamente,

$$\begin{aligned} C_0(q) &= 8q + 10 \\ C_1(q) &= 10q + 20 \\ C_2(q) &= 12q + 30 \\ C_3(q) &= 11q + 25 \end{aligned}$$

4. También se sabe que el beneficio unitario de todas las firmas es el mismo para cada punto de mercado (aunque puede ser diferentes entre mercados).
5. Se sabe que la estrategia de cada firma clasifica los puntos de mercado en dos clusters y que a cada cluster le asocia una poligonal de 3 estrategias simples.

Las curvas de oferta originales empleadas para generar los puntos de mercado artificiales (MP) son las de la tabla I.

TABLA I

Curvas de oferta individuales originales compuestas de una poligonal con tres estrategias simples para cada cluster, se representan cuatro puntos de la poligonal. Estas curvas se corresponden con el problema en forma de cruz expuesto en esta sección. En = EMPRESA n

	MP_{1-10}
E0	(16.67,2.13)(26.67,2.17)(36.67,2.21)(46.67,2.25)
E1	(16.67,7.39)(26.67,7.69)(36.67,7.99)(46.67,8.29)
E2	(16.67,42.52)(26.67,51.02)(36.67,59.52)(46.67,68.02)
E3	(16.67,14.55)(26.67,15.65)(36.67,16.75)(46.67,17.85)
	MP_{11-20}
E0	(16.67,1.93)(26.67, 2.08)(36.67,2.23)(46.67,2.38)
E1	(16.67,6.40)(26.67,7.33)(36.67,8.26)(46.67,9.19)
E2	(16.67,29.35)(26.67,46.45)(36.67,63.55)(46.67,80.65)
E3	(16.67,10.63)(26.67,14.33)(36.67,18.03)(46.67,21.73)

V. Resultados numéricos

A. Framework y medida de errores

Todos los ensayos se ha realizado empleando los parámetros genéticos de la tabla II (se ha seguido la nomenclatura del paquete "Genetic Programming Kernel" de GPs coevolutivos desarrollado por Thomas Weinbrenner, [14]).

TABLA II

Configuración de los parámetros genético empleados en los experimentos desarrollados en esta sección.

Parámetro	Valor
Tipos de selección:	Torneo
Tamaño del torneo:	4
Numero de poblaciones:	4 poblaciones en CGM
Individuos por población:	500-1000
Número de iteraciones:	200-500
Probabilida de cruce:	Desc.lineal desde 100.0%

Con el fin de obtener una información más completa de los resultados obtenidos, se han empleado dos funciones de error para medir la distancia entre las curvas estimadas y las curvas de oferta originales:

- Error_1 mide el error porcentual medio de la energía estimada para todas las firmas para cada mercado.
- Error_2 mide el error porcentual medio de la cuota de energia estimada para cada mercado para todas las firmas.

B. Fitness monoobjetivo vs. Fitness multiobjetivo para los modelos GM y CGM

Los ensayos realizados con el modelo CGM empleando Fitness_{multi} y Fitness_{mono} los encontramos en la tabla III, donde los valores numéricos son la media de 10 ejecuciones con cada conjunto de parámetros: tipo de fitness (Fit), número de individuos por población (Pop) y número de iteraciones (Iter).

Se observa que empleando el Fitness_{multi}, tanto el error absoluto (Error_1) como el de distribución de cuotas de mercado (Error_2), son mucho mejores (más bajos) que empleando el Fitness_{mono}, en todos los tests. Esto debido principalmente a que, el fitnes multiobjetivo converge paralelamente a un

TABLA III

Errores de estimación resultantes de aplicar el fitness multiobjetivo ($\text{Fitness}_{\text{multi}}$) frente al fitness monoobjetivo ($\text{Fitness}_{\text{mono}}$), en el modelo CGM para estimar las curvas de oferta individuales del problema planteado. F = Fitness

Fit	Pop	Iter	$Error_1, \sigma_{error_1}$	$Error_2, \sigma_{\%error_2}$
F_{multi}	500	200	1.631,0.748	0.332,0.176
F_{mono}	500	200	246.641,96.271	20.926,5.624
F_{multi}	1000	200	1.118,0.523	0.197,0.088
F_{mono}	1000	200	255.751,62.520	21.408,4.483
F_{multi}	500	500	1.327,0.782	0.270,0.181
F_{mono}	500	500	215.066,112.140	17.916,6.920
F_{multi}	1000	500	1.168,0.640	0.253,0.139
F_{mono}	1000	500	223.648,126.607	19.170,8.001

TABLA IV

Errores de estimación resultantes de aplicar el fitness multiobjetivo ($\text{Fitness}_{\text{multi}}$) frente al fitness monoobjetivo ($\text{Fitness}_{\text{mono}}$), en el modelo GM para estimar las curvas de oferta individuales del problema planteado. F = Fitness

Fit	Pop	Iter	$Error_1, \sigma_{error_1}$	$Error_2, \sigma_{\%error_2}$
F_{multi}	500	200	12.742,4.032	2.245,0.885
F_{mono}	500	200	34.516,18.940	6.841,2.328
F_{multi}	1000	200	10.551,2.566	1.704,0.708
F_{mono}	1000	200	28.389,4.608	6.540,0.985
F_{multi}	500	500	5.708,1.880	1.252,0.443
F_{mono}	500	500	14.173,10.343	3.101,1.954
F_{multi}	1000	500	4.822,1.163	1.003,0.237
F_{mono}	1000	500	11.437,8.126	2.483,1.592

mínimo global en ambas componentes del fitness Fitness_{multi}=<Fitness_a,Fitness_b >, de modo que no se establece una prioridad en ninguna de las componentes. En cambio al existir una dependencia en las componentes Fitness_a y Fitness_b de Fitness_{mono}: aquella que primero converge en la evolución del algoritmo sesga los individuos, ocasionando un estancamiento de la otra componente en un mínimo local, cosa que se puede ver en la figura 4.

Fig. 4. Comparación de la progresión de los fitness $\text{Fitness}_{\text{mono}}$**a)** vs. $\text{Fitness}_{\text{multi}}$**b)** en dos simulaciones con la misma semilla aleatoria

Los mismos ensayos con ambos fitness se realizaron con el esquema no-coevolutivo concluyendo nuevamente la superioridad del Fitness_{multi} frente a Fitness_{mono}, para todos los tests realizados, (ver tabla IV).

C. Comparativa final

Las pruebas realizadas con el modelo KGM se efectuaron tomando 1 cluster (1-KGM), es decir, sin aplicar segmentación de la nube de puntos, y también segmentando la nube de puntos en dos clusters (2-KGM) y para ambas variantes se aplicó a posteriori el modelo CGM sin clasificador, con 1000 individuos y 500 iteraciones. Dado que el resultado con 1000 individuos y 500 iteraciones ajustaba perfectamente la regresión de la oferta agregada de cada cluster, se optó por no realizar ensayos con más configuraciones.

A la vista de los resultados obtenidos con cada uno de los métodos, CGM, GM y n-KGM (ver tabla V), puede concluirse que el modelo coevolutivo, CGM, con fitness multiobjetivo es más eficaz que el modelo no-coevolutivo equivalente GM, en todos los casos y, también que la estrategia subóptima KGM, a la hora de aproximar las ofertas de generación individuales de un pool, a partir de una nube de puntos de mercado.

TABLA V

Comparación de los mejores resultados obtenidos con los modelos CGM, GM y KGM. F = Fitness

Model	Fit	Pop	Iter	$Error_1, \sigma_{error_1}$	$Error_2, \sigma_{error_2}$
1-KGM	-	1000	500	3.716, 0.0	0.423, 0.0
2-KGM	-	1000	500	3.580, 0.0	0.515, 0.0
GM	F_{multi}	500	200	7.844,1.744	0.793,0.333
CGM	F_{multi}	500	200	1.631,0.748	0.332,0.176
GM	F_{multi}	500	500	4.268,1.746	0.899,0.410
CGM	F_{multi}	500	500	1.327,0.782	0.270,0.181
GM	F_{multi}	1000	200	7.230,1.276	0.720,0.346
CGM	F_{multi}	1000	200	1.118,0.523	0.197,0.088
GM	F_{multi}	1000	500	3.875,1.871	0.750,0.340
CGM	F_{multi}	1000	500	1.168,0.640	0.253,0.139

VI. Conclusiones y trabajos futuros

Los modelos genéticos coevolutivos se emplean a menudo para simular sistemas naturales con múltiples agentes de comportamiento independiente, donde no es posible aplicar modelos matemáticos.

Es en este trabajo se ha demostrado que un modelo genético coevolutivo (CGM) es más eficaz que un modelo no-coevolutivo equivalente (GM) y un algo-

ritmo híbrido basado en el algoritmo de clasificación K- means más un algoritmo genético (KGM), para estimar las curvas de oferta de un pool de empresas a partir de una serie de puntos del mercado eléctrico (precio, demanda). Por otro lado también se ha constatado que un fitness multiobjetivo (Fitness_{multi}) aplicado al problema propuesto es más efectivo que el fitness monoobjetivo equivalente (Fitness_{mono}), ya que el Fitness_{multi} explora los objetivos en paralelo evitando su obstrucción mutua, y por ello la convergencia prematura a un óptimo local.

Desde el punto de vista de una firma que obtiene una estimación de las curvas de oferta del pool, este método nos permite conocer la influencia de una ligera variación en su estrategia individual en una situación dada del mercado. De esta forma, una firma puede ajustar su estrategia con el fin de mejorar su beneficio, y simular el ajuste empleando nuestra herramienta. Además, un pool estimado con el método CGM, puede emplearse para detectar la presencia de acuerdos entre generadores, comparándolo con el pool óptimo (en términos del problema de Cournot). Estamos desarrollando un nuevo algoritmo para estimar el pool óptimo, donde puede fijarse la estrategia de una firma para simular la reacción del resto de firmas.

Referencias

[1] Ministerio de Industria y Energía, "Protocolo para el establecimiento de una nueva regulación del sistema electico nacional," Tech. Rep., Ministerio de Industria y Energía, 1996.

[2] J. Paredis, "Coevolutionary computation," *Artificial Life*, vol. 2, no. 4, pp. 355–375, 1995.

[3] C.D. Rosin and R.K. Belew, "New methods for competitive coevolution," *Evolutionary Computation*, vol. 5, no. 1, pp. 1–29, 1997.

[4] W.W. Leontief, "Verzögerte angebotsanpassung und partielles gleichgewicht," *Zeitschrift für Nationalökonomie (Alemán)*, vol. 5, pp. 670–676, 1934.

[5] J. Arifovic, "Genetic algorithm learning and the cobweb model," *Journal of Economic Dynamics and Control*, vol. 18, pp. 3–28, 1994.

[6] R. Franke, "Coevolution and stable adjustments int the cobweb model," *Journal of Evolutionary Economics*, vol. 8, pp. 383–406, 1998.

[7] T. Curzon Price, "Using co-evolutionary programming to simulate strategic behaivour in markets," *Journal of Evolutionary Economics*, vol. 7, pp. 219–234, 1997.

[8] W. Fushuan and D. A. Kumar, "A genetic algorithm based method for bidding strategy coodination in energy and spinning reserve markets," *Artificial Intelligence in Engineering, 15(2001)*, pp. 71–79, 2001.

[9] W. D. Hillis, "Co-evolving parasites improves simulated evolution as an optimization technique," *Artificial Life II*, pp. 41–47, 1991.

[10] K.A. De Jong M.A. Potterm and J.J. Grefenstette, "A coevolutionary approach to learning sequential decision rules," *ICGA95*, 1995.

[11] B.J. Reynolds S. Azar and S. Narayanan, "Comparison of two multiobjective optimization techniques with and within genetic algorithms," *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, 1999.

[12] A. Cournot, *Recherches sur les Principies Mathématiques de la Théorie des Richesses*, Hachette, París, 1838.

[13] Y.-J. Lai and C.-L. Hwang, *Fuzzy Multiple Objetive Decision Making: Methods and Applications*, Springer-Verlag, New York, 1996.

[14] T. Weinbrenner, "Genetic programming kernel version 0.5.2," Tech. Rep., http://www.emk.e-technik.tu-darmstadt.de/ thomasw/gpkernel_toc.html, 1997.