Diseño de estrategias mediante Evolución Diferencial en un juego de imitación iterado con dependencia entre los turnos

Pablo J. Villacorta y David A. Pelta

Resumen— **La decision en presencia de adversarios trata de obtener estrategias que tengan en cuenta por adelantado comportamiento de un oponente que intenta aprender a predecir nuestras acciones. Una posible manera de defenderse es tomar decisiones orientadas a confundirlo, a pesar de que nuestra propia recompensa inmediata pueda disminuir. Esta idea ha sido capturada en un modelo con adversarios introducido en un trabajo previo, en el que dos agentes dan una respuesta por separado a una secuencia desconocida a priori de estímulos externos. En este trabajo extendemos el modelo introduciendo dependencia estadística entre la respuesta de uno de los agentes y el siguiente estímulo de la secuencia. Se han aplicado varias variantes del algoritmo de Evolución Diferencial para diseñar estrategias para este modelo extendido, utilizando simulaciones durante el proceso de optimización para evaluar la bondad de cada estrategia. Los resultados indican que el método de Evolución Diferencial es una buena técnica y las estrategias obtenidas mejoran a otras más simples.**

Palabras clave— **Toma de decisiones, adversarios, estrategias, manipulación.**

I. Introducción

La decisión en presencia de adversarios tiene que ver con entender la mente y las acciones de un oponente. Es relevante en una amplia gama de dominios donde los actores están activamente y de forma consciente compitiendo por algunos objetivos [1].

En su forma más básica, involucra a dos participantes, cada uno de los cuales elige una acción como respuesta a un estímulo externo, sin saber cómo actuará el otro. Como resultado de sus elecciones, se le asigna un pago a cada uno. Cuando esta situación se repite muchas veces, resulta más difícil actuar ya que los participantes tienen la posibilidad de aprender la estrategia del otro en base a las acciones pasadas. Ejemplos donde esto sucede pueden encontrarse en el ámbito militar, pero también en juegos de estrategia en tiempo real, conflictos entre gobiernos, en el ámbito económico [2] y en deportes de equipo en general.

Es un área multidisciplinar por naturaleza en la que intervienen técnicas provenientes de diferentes campos como planificación, redes bayesianas y redes de creencia, conceptos de teoría de la decisión y teoría de juegos, entre otros. La teoría de juegos en particular ha sido aplicada para modelar interacciones estratégicas entre gobiernos, empresas y también entre terroristas y fuerzas de seguridad. Como disciplina matemática, proporciona herramientas para determinar la manera más racional de decidir ante ciertas situaciones. Se pueden encontrar aplicaciones en campos relacionados con la seguridad, como por ejemplo Sistemas de ayuda a la decisión para la seguridad en aeropuertos [3] y modelos de patrullaje de perímetros mediante robots autónomos [4], [5].

Dpto. de Ciencias de la Computación e IA, CITIC-UGR, Universidad de Granada. C/Periodista Daniel Saucedo Aranda s/n, 18071 Granada. E-mail: {pjvi,dpelta}@decsai.ugr.es

Como se ha dicho, se trata de encontrar estrategias para hacer frente a un adversario observador y adaptativo (agente T). Una posible manera de defenderse es tomar decisiones orientadas a confundirle, aunque a corto plazo puedan resultar perjudiciales. A pesar de ello, esta manera de decidir resulta muy interesante para el razonamiento frente a adversarios puesto que lo que el agente S trata de hacer es comportarse de una manera lo más impredecible posible, pero a la vez, racional. En otras palabras, S desea forzar la presencia de incertidumbre para confundir al adversario, pero a la vez, tratando de que su propio beneficio se vea lo menos afectado posible por la aleatoriedad.

En un trabajo previo [6] se propuso un modelo para estudiar el balance entre el pago obtenido y el nivel de confusión inducido en el adversario. La conclusión era que una manera de introducir incertidumbre es comportarse siguiendo reglas parcialmente aleatorizadas. En [7] se dieron expresiones teóricas que representaban algunas de estas reglas o estrategias, aunque el modelo tratado en ambos casos no contenía dependencia estadística entre una acción y el siguiente evento.

El objetivo de esta contribución es diseñar y analizar estrategias de decisión para el agente S que no son constantes en el tiempo sino que cambian en ciertos momentos del proceso iterado. Más específicamente, abordamos el problema de diseño de estrategias como un problema de optimización no lineal con restricciones cuya solución nos da tanto el momento exacto del cambio como el nuevo comportamiento que se debe seguir. Se toman como punto de partida los resultados de [8] pero se introduce una dificultad adicional al considerar dependencia estadística entre la acción elegida por S y el próximo evento al que habrá que responder.

Esta extensión al modelo provoca que la optimización sea imposible de llevar a cabo con técnicas exactas, y por ello se recurre a métodos heurísticos para encontrar la mejor estrategia, basados en Evo-

lución Diferencial. La popularidad de los algoritmos de este tipo motivó numerosos trabajos en los que se propusieron mejoras y variantes al algoritmo básico. Aquí se aplicarán varias de estas propuestas y se compararán los resultados en el problema de diseño de estrategias.

El resto del trabajo se estructura como sigue. La Sección II describe en detalle el modelo de interacción entre dos agentes que se ha anticipado ya en la introducción, incluyendo la extensión correspondiente a la relación de dependencia entre respuestas y eventos. La Sección III describe qué se entiende por estrategia aleatorizada y explica los conceptos de estrategia mixta estática y dinámica. En la Sección IV se presenta brevemente el algoritmo de Evolución Diferencial y se mencionan las variantes que se han aplicado y algunas de sus características más relevantes. Los experimentos realizados y los resultados obtenidos se describen en la Sección V, y finalmente las conclusiones junto con trabajos futuros se discuten en la Sección VI.

II. Descripción del modelo

El modelo consta de dos agentes S y T (el adversario), un conjunto de posibles entradas o eventos $E = \{e_1, e_2, \ldots, e_n\}$ generados por un tercer agente R, que representa el entorno externo al modelo, y un conjunto de posibles respuestas o acciones $A = \{a_1, a_2, \ldots, a_m\}$ con las que los agentes pueden responder a cada evento. Existe también una matriz de pagos P:

$$P(n \times m) = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ p_{31} & p_{32} & \cdots & p_{3m} \\ \\ p_{n1} & p_{n2} & \cdots & p_{nm} \end{pmatrix}$$

tal que p_{ij} es el pago que obtendrá S si elige la acción a_j como respuesta al evento e_i y T no es capaz de adivinar esta respuesta.

El agente S debe decidir qué acción elegir dada una entrada particular e_i y con un perfecto conocimiento de la matriz de pagos P. Su objetivo es maximizar la suma de los pagos obtenidos tras una secuencia de entradas o estímulos. Los estímulos proceden del entorno externo. Se proporciona un estímulo en cada instante de tiempo. Los estímulos se generan de forma estocástica como se describirá más adelante.

El agente T no conoce la matriz de pagos P, pero está observando el comportamiento de S con el fin de aprender de sus acciones. Su objetivo es reducir el beneficio del agente S adivinando qué acción elegirá como respuesta al estímulo de la secuencia recibido en cada instante. El siguiente algoritmo describe estos pasos. L representa la longitud de la secuencia de estímulos.

Secuencia de pasos que forman el modelo

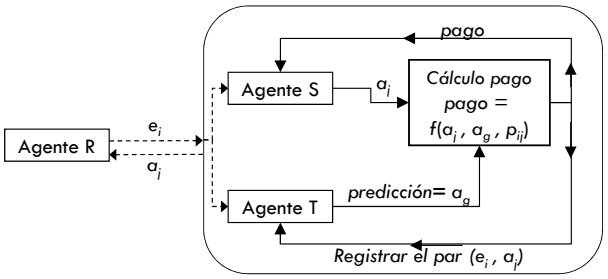


Fig. 1. Representación gráfica del modelo

Para l = 1 hasta L hacer
    Llega un nuevo estímulo e_i.
    El agente T hace una predicción a_g
    El agente S escoge una acción a_j
    Se calcula el pago para S
    El agente T registra el par (e_i, a_j)
    en su memoria
Fin

Ante un estímulo e_i, el agente S elige una acción en base a sus propias reglas de elección (denominadas genéricamente *estrategia*). Al mismo tiempo el agente T, utilizando también su propia estrategia, da una predicción sobre la acción que escogerá S. En el momento de la elección, ninguno de los dos agentes sabe lo que va a elegir el otro. Además, T mantiene su propia matriz de observaciones, O, de dimensiones $n \times m$. O_{ij} representa el número de veces que, hasta el momento actual, el agente S decidió tomar la acción i cuando el estímulo era j. El cálculo de la recompensa para S ante el evento e_i se define como:

$$pago = p_{ij} \cdot F(a_g, a_j) \tag{1}$$

siendo F:

$$F(a,b) = \begin{cases} 0 & \text{si a = b} \\ 1 & \text{en otro caso} \end{cases} \tag{2}$$

Esto significa que el agente S no obtiene ninguna recompensa cada vez que el agente T consiga adivinar correctamente su respuesta.

El patrón de comportamiento de ambos agentes puede ser muy diverso. El agente S puede comportarse de manera totalmente determinista, eligiendo siempre la acción que, según la matriz de pagos que él conoce, le dará un mayor beneficio. Este comportamiento, si se repite a lo largo del tiempo, es muy fácil de aprender ya que para cada estímulo posible, S siempre responde de la misma manera y por tanto es fácil predecir correctamente la acción para cada estímulo. El otro extremo es comportarse de manera totalmente aleatoria, aunque esto puede suponer una pérdida de beneficio debido a que elijamos continuamente acciones cuyo pago es mucho menor que el óptimo. Mayor aleatoriedad implica mayor dificultad de aprendizaje para un observador, pero también

mayor riesgo de pérdida de beneficio por elegir acciones no óptimas.

Análogamente, el agente T puede hacer su predicción eligiendo de manera determinista la acción que más veces ha observado como respuesta a un estímulo dado, o puede hacer una predicción probabilística en función de la frecuencia observada para cada acción. En lo que sigue, T usará precisamente esta última estrategia, puesto que se ha demostrado en [6] que muy dificil de contrarrestar para S. Más precisamente, la probabilidad de dar como predicción una acción a_i para el estímulo e_j es proporcional a O_{ij}.

A. Dependencia estadística en los eventos

En el modelo original [6] los eventos que provenían del exterior eran independientes y se generaban aleatoriamente siguiendo una distribución de probabilidad uniforme. La independencia significa que no hay relación entre el evento actual, la respuesta actual y el evento siguiente. Eventos uniformemente generados significa que cada vez que se va a generar un evento, todos ellos son igualmente probables. Nótese que el considerar otras distribuciones de probabilidad distintas a la uniforme no cambiaría la metodología de los trabajos existentes [8], [7] sino que sólo requeriría repetir los experimentos y ajustar los resultados. Las conclusiones de dichos estudios no dependen de la distribución de probabilidad empleada.

Sin embargo, introducir dependencia estadística entre eventos de enfrentamientos consecutivos es diferente. Podemos pensar en varios tipos de dependencia. La primera es la dependencia entre el estímulo de una etapa y el estímulo de la siguiente. La pregunta es, ¿sería útil esta información para alguno de los agentes de algún modo? En otras palabras: cuando un agente va a elegir una acción, ¿es relevante para su decisión saber que el evento siguiente será probablemente de cierto tipo y no de cualquier otro? La respuesta es negativa: cada vez que se produce un evento, el agente debe intentar hacerlo lo mejor posible para obtener el máximo beneficio, tanto si ese evento es muy frecuente como si es poco frecuente, sin preocuparse de cuál va a ser el siguiente evento ya que ni su decisión en la etapa actual ni en ninguna etapa pasada va a influir en ello.

El segundo tipo de dependencia es la relación entre la acción elegida en la etapa actual y el próximo evento que aparecerá. Esta dependencia será estudiada en el resto de la presente contribución dado que tiene implicaciones interesantes. La más importante es que, antes de tomar una decisión frente al evento actual, un agente debería considerar que su elección no sólo va a afectar al pago que obtendrá en esta etapa sino también al máximo pago que podría obtener en la etapa siguiente, ya que la mejor acción para un cierto evento puede llevar asociado un pago más alto que la mejor acción para otros eventos. Esta consideración refleja el hecho de que algunos eventos pueden ser críticos así que llevan asociado un pago muy alto, mientras que otros llevan asociados pagos más bajos en general para todas las acciones. Con esto en mente, se puede definir una buena acción como aquella que no solamente proporciona un pago alto para la etapa actual (pago inmediato) sino que además provoca que en la siguiente etapa surja un evento cuyos pagos son mayoritariamente altos.

Asumimos que la información acerca de la dependencia estadística entre la acción actual y el próximo evento sólo está disponible para el agente S, en forma de matriz de probabilidades condicionadas C de dimensiones $m \times n$, como se muestra a continuación.

$$C(m \times n) = \begin{pmatrix} P[X = e_1|Y = a_1] & \dots & P[X = e_n|Y = a_1] \\ P[X = e_1|Y = a_2] & \dots & P[X = e_n|Y = a_2] \\ \vdots & \ddots & \vdots \\ P[X = e_1|Y = a_m] & \dots & P[X = e_n|Y = a_m] \end{pmatrix}$$

El valor C_{ij} es la probabilidad condicionada $P[X = e_j|Y = a_i]$, y por lo tanto $\sum_{j=1}^{n} P[X = e_j|Y = a_i] = 1$ para cada fila $i = 1, ..., m$. En la expresión anterior, X es la variable aleatoria discreta que representa al siguiente evento, e Y es la variable aleatoria discreta que representa la acción actual tomada por el agente S. Finalmente, sea $(\pi_1, ..., \pi_n)$ el vector de probabilidades de que se produzca cada evento en la primera etapa de la simulación. No se pueden dar probabilidades condicionadas en esta etapa puesto que no existe una acción previa en ese punto. En los experimentos se ha tomado el vector $(1/n, ..., 1/n)$ como probabilidades iniciales.

III. Comportamiento de los agentes

A. Estrategias para T

Se considerarán dos posibles estrategias a partir de la matriz de observaciones del agente T:

- Proporcional a la Frecuencia (PF): T elige una acción de manera estocástica, donde la probabilidad de elegir la acción a_j como predicción ante el evento e_i es proporcional a O_{ij}.
- Más frecuente (MF): T elige de forma determinista la acción que ha observado más veces en el pasado, es decir, la acción a_j con el mayor valor de O_{ij}.

B. Estrategias para S

B.1 Estrategias mixtas estáticas

El agente S debe utilizar una estrategia aleatorizada para evitar en la medida de lo posible que su comportamiento sea aprendido fácilmente. No debe utilizar una estrategia uniformemente aleatoria sobre sus acciones porque elegiría demasiadas veces acciones con poco pago. Por tanto, es necesario encontrar la mejor estrategia aleatorizada. En términos de teoría de juegos, una aleatorización sobre las acciones posibles se conoce como una *estrategia mixta*, que puede ser representada como una distribución de probabilidad (vector de pesos) sobre las acciones

y que es utilizada por el jugador cada vez que debe elegir una acción. Para mayor flexibilidad, consideraremos que el agente S mantiene una estrategia mixta diferente $\alpha_i = (\alpha_{i1}, ..., \alpha_{im})$ específica para cada evento e_i, donde $\sum_{j=1}^{m} \alpha_{ij} = 1 \quad \forall i = 1, ..., n$.

B.2 Estrategias mixtas dinámicas

Extendemos las estrategias mixtas descritas en la sección anterior, permitiendo ahora que se utilice para cada evento, varias estrategias mixtas, según el momento de la simulación en el que nos encontremos. Las estrategias mixtas estáticas implican utilizar siempre el mismo vector de pesos. Ahora se propone cambiar el vector de pesos utilizado en ciertos momentos. Definimos un *período* como una serie de eventos consecutivos ante los cuales utilizamos el mismo vector de pesos para responder. La idea ahora es definir varios períodos y calcular la estrategia mixta dinámica que maximiza el pago del agente S. La longitud de cada período, es decir, el número de eventos consecutivos de un mismo tipo e_i durante los cuales se utilizará el mismo vector de pesos, también se debe ajustar para conseguir dicha maximización. Sea N_i^h la longitud del h-ésimo período de la estrategia mixta dinámica para los eventos de tipo e_i. Este valor indica que se utilizará una misma estrategia mixta para N_i^h apariciones del evento e_i. De nuevo, el agente mantendrá una estrategia mixta dinámica diferente para cada tipo de evento, para adaptarse mejor a la situación permitiendo mayor flexibilidad. Por simplicidad, y aunque se podría definir un número de períodos diferente para cada una de estas n estrategias, supondremos que el número de períodos es igual para todas ellas. Dicho número se ha fijado en $H = 4$, pero deben hacerse estudios más detallados sobre su valor. La Figura 2 muestra un ejemplo de 4 estrategias dinámicas ante 4 posibles eventos. Cada estrategia puede tener definidos un número diferente de períodos; en el caso del ejemplo, hay estrategias con 3 períodos y también con sólo 2.

B.3 Optimización de las estrategias

El planteamiento de este tipo de estrategias sofisticadas persigue aumentar el pago que obtiene el agente S. La hipótesis principal es que el uso de estrategias mixtas dinámicas puede ayudar a incrementar el pago, basándose en que el comportamiento de S en un cierto período, a pesar de dar pagos inmediatos más bajos, puede dar lugar a mejores pagos en períodos siguientes ya que el adversario se encuentra parcialmente *confundido* por lo que observó en el pasado y además, es poco probable que pueda detectar este tipo de cambios en el comportamiento de S, al menos de manera simple.

Para comprobar la potencia de este enfoque, se debe primero encontrar la mejor de las estrategias estáticas y la mejor de las dinámicas, y comparar el pago que son capaces de dar para S, tanto cuando T utiliza la estrategia PF como cuando utiliza MF. La mejor estrategia es aquella que maximiza el pago obtenido por S tras una secuencia de eventos, y debe buscarse dentro del espacio de estrategias existentes. Una estrategia mixta estática está formada por m variables reales (pesos) entre 0 y 1 pero, dado que el comportamiento ante un evento influye en el el siguiente, no es posible calcular por separado la estrategia mixta óptima para cada evento sino que deben buscarse todos los pesos de las n estrategias mixtas al mismo tiempo, en un solo problema de optimización que tendrá $m \cdot n$ variables reales $\alpha_{ij} \quad i = 1, ..., n, \quad j = 1, ..., m \in [0, 1]$, sometidas a las restricciones de que $\sum_{j=1}^{m} \alpha_{ij} = 1 \quad i = 1, ..., n$. Para un modelo con $n = 5$ eventos y $m = 5$ acciones posibles, se obtiene un problema con 25 variables reales. En el caso de estrategias mixtas dinámicas ocurre lo mismo: todos los pesos de todas las estrategias mixtas de todos los períodos deben buscarse en un solo problema de optimización puesto que están relacionados entre sí. El número de variables que se buscan son $m \cdot n \cdot H + n \cdot H$ siendo H el número de períodos de las estrategias. El primer sumando indica el número de pesos entre 0 y 1 que se buscan, y el segundo, el número de variables enteras positivas correspondientes a las longitudes óptimas que debe durar cada uno de los períodos. Dado que hay n estrategias dinámicas distintas con H períodos cada una, es necesario buscar $n \cdot H$ longitudes de período. Tomando $n = m = 5$ y $H = 4$ se obtiene un total de 120 variables.

El pago obtenido que se pretende maximizar se mide mediante una simulación como la indicada en el Algoritmo de la Sección II, que sirve para evaluar cómo de buena es una estrategia. No pueden emplearse técnicas exactas porque dicha función no es una expresión matemática, y además una simulación es un proceso no determinista debido a la naturaleza estocástica de las estrategias que buscamos y a la secuencia de estímulos que se van generando, que depende también de modo estocástico de las acciones que S va eligiendo. Por ambos motivos, se propone el empleo de metaheurísticas para optimización de variables reales, y en concreto el algoritmo conocido como Evolución Diferencial, como se explicará en detalle en la siguiente sección.

IV. Optimización mediante Evolución Diferencial

La optimización mediante técnicas heurísticas ha sido aplicada a muchos campos de las ciencias y las ingenierías donde, por algún motivo, las técnicas analíticas de optimización no pueden emplearse. Las causas para ello pueden ser muy diversas: que la función que se va a optimizar no sea diferenciable, que existan restricciones arbitrarias en el espacio de búsqueda, que el número de variables sea muy grande o incluso que aquello que se pretende optimizar no sea fácilmente expresable como función matemática. Muchas situaciones reales presentan este tipo de in-

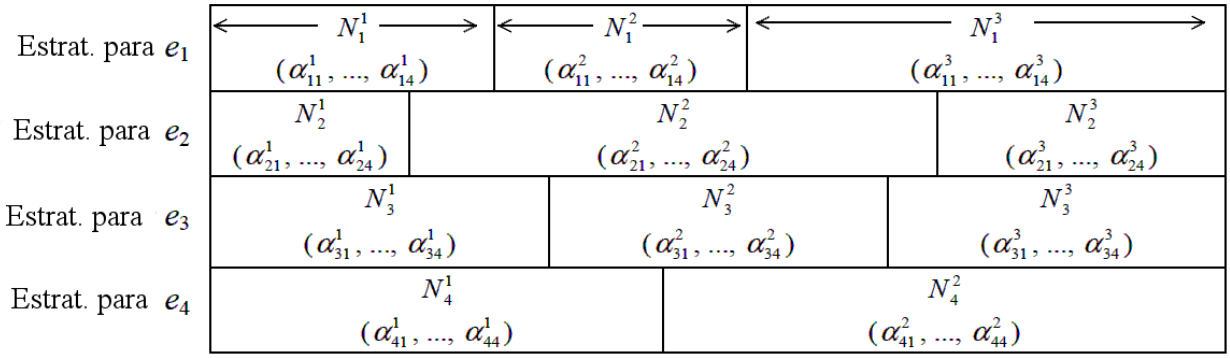


Fig. 2. Ejemplo de 4 estrategias mixtas dinámicas independientes para 4 eventos distintos

convenientes. A pesar de dichos problemas, las metaheurísticas han sido capaces de obtener soluciones de gran calidad y por ello siguen siendo usadas con mucha frecuencia. A diferencia de la optimización matemática, no garantizan la obtención del óptimo pero, una vez más, en la mayoría de problemas reales importa más obtener una buena solución en un tiempo razonable que obtener la óptima empleando un tiempo demasiado alto o simplemente no poder obtener niguna solución por no disponer de herramientas matemáticas (analíticas) apropiadas.

En el problema descrito en la sección anterior, no es posible expresar el pago para S como una función matemática de las variables de la estrategia empleada, ya que dicho pago se obtiene a través de una simulación no determinista que evalúa la estrategia. Las metaheurísticas también son capaces de resolver este tipo de problemas con funciones que no están claramente definidas. Además, las variables que buscamos son en realidad distribuciones de probabilidad, por lo que son continuas y su rango está comprendido entre 0 y 1 ambos inclusive, y además deben sumar 1. En el caso de las estrategias dinámicas, se tiene también un conjunto de variables enteras positivas cuya suma debe ser igual al número de estímulos que ha durado la simulación y que se supone conocido por los agentes.

Se propone emplear el algoritmo heurístico conocido como *Evolución Diferencial* (DE), que fue introducido por primera vez en [9] y se desarrolla en [10]. Ha sido utilizado especialmente en problemas de optimización de variables reales con muy buenos resultados [11], y desde su aparición se han propuesto numerosas variantes y mejoras. Aquí se utilizarán algunas de ellas, descritas en detalle a continuación, para comparar su rendimiento en este problema.

A. Algoritmo básico de Evolución Diferencial

Se comentará a grandes rasgos el proceso general de Evolución Diferencial. Pueden encontrarse más detalles en [12], [9]. Sigue el esquema general de un Algoritmo Evolutivo. Empieza con una población de NP soluciones candidatas o individuos, comúnmente representados como vectores D-dimensionales. Cada individuo de la generación G se representa como $X_{i,G} = \{x_{i,G}^1, ..., x_{i,G}^D\}$, y son llamados también vectores objetivo. La población inicial se genera aleatoriamente. A continuación, se aplica sobre cada individuo $X_{i,G}$ un operador de mutación para obtener un individuo mutado $V_{i,G}$ asociado, combinando otros individuos de la población. Algunas de las numerosas variantes para conseguir esto son:

- DE/Rand/1:
$V_{i,G} = X_{r_1^i,G} + F \cdot (X_{r_2^i,G} - X_{r_3^i,G})$
- DE/Best/1:
$V_{i,G} = X_{best,G} + F \cdot (X_{r_1^i,G} - X_{r_2^i,G})$
- DE/RandToBest/1:
$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r_1^i,G} - X_{r_2^i,G})$
- DE/Best/2:
$V_{i,G} = X_{best,G} + F \cdot (X_{r_1^i,G} - X_{r_2^i,G}) + F \cdot (X_{r_3^i,G} - X_{r_4^i,G})$
- DE/Rand/2:
$V_{i,G} = X_{r_1^i,G} + F \cdot (X_{r_2^i,G} - X_{r_3^i,G}) + F \cdot (X_{r_4^i,G} - X_{r_5^i,G})$
- DE/RandToBest2: $V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r_1^i,G} - X_{r_2^i,G}) + F \cdot (X_{r_3^i,G} - X_{r_4^i,G})$

Los índices r_j^i se refieren a otros individuos de la población distintos entre sí y distintos al propio individuo $X_{i,G}$, y son generados aleatoriamente para cada mutación. F es un factor de escala de la diferencia entre individuos.

A continuación, se lleva a cabo la operación de cruce del individuo original con su individuo mutado asociado, según algún operador de cruce tal como *cruce exponencial, cruce binomial o cruce aritmético*, para obtener un individuo cruzado $U_{i,G}$ asociado al individuo original $X_{i,G}$. Finalmente, para obtener los individuos de la siguiente generación, se selecciona entre cada individuo $X_{i,G}$ y su cruzado asociado $U_{i,G}$ aquel que tenga mejor valor de la función objetivo.

B. Variantes del algoritmo básico

- Evolución diferencial auto-adaptativa (SADE [13], *Self-adaptive Differential Evolution*): implementa varias de las estrategias de mutación anteriores, y selecciona una de ellas diferente para cada individuo, de manera estocástica con una probabilidad que va

variando en función de cómo de exitosa resultó en el pasado esa estrategia de mutación. También adapta los factores de escala para cada individuo.

- Evolución diferencial adaptativa con archivo externo (JADE [14]): utiliza una nueva estrategia de mutación llamada DE/RandTo*p*Best/1 donde uno de los individuos que intervienen no pertenece necesariamente a la generación actual sino a la unión de ésta con un conjunto de soluciones inferiores obtenidas mediante exploraciones recientes.
- Búsqueda local del factor de escala en Evolución Diferencial (SFLSDE [15], *Scale factor local search in differential evolution*): inspirado en los algoritmos meméticos, utiliza dos algoritmos distintos de búsqueda local para encontrar el factor de escala más apropiado.

V. Experimentos y resultados

Los experimentos pretenden responder a las siguientes cuestiones:

1. ¿Tiene sentido la utilización de estrategias mixtas dinámicas?
2. ¿Mejora el pago de S con las estrategias mixtas dinámicas frente a las estáticas cuando ambas han sido obtenidas mediante un proceso de optimización?
3. ¿En qué medida afecta el algoritmo de optimización utilizado a la calidad de las estrategias y, por consiguiente, al pago obtenido por S?

Para ello se han llevado a cabo experimentos con los siguientes ajustes.

A. Parámetros del modelo

- Número de eventos y acciones: $n = m = 5$
- Longitud de las secuencias de eventos en una simulación: $L = 500$.
- Estrategia para el agente T: se experimentó por separado usando las estrategias PF y MF.
- Matriz de probabilidades condicionadas para la generación del siguiente evento en función de la acción actual elegida por S:

$$C = \begin{pmatrix} 0.2 & 0.5 & 0.15 & 0.1 & 0.05 \\ 0.4 & 0.1 & 0.25 & 0.05 & 0.2 \\ 0.15 & 0.2 & 0.4 & 0.1 & 0.15 \\ 0.1 & 0.1 & 0.2 & 0.5 & 0.1 \\ 0.3 & 0.4 & 0.3 & 0 & 0 \end{pmatrix}$$

- Matrices de pago empleadas: se utilizaron 5 matrices de pagos distintas. Una matriz posee, en cada fila, una permutación diferente de un mismo conjunto de m pagos, donde dichos conjuntos son los que se muestran en la Tabla I. Nótese que la diferencia es que hay un pago cada vez más destacado que el resto.

TABLA I
Conjunto de pagos asociado a cada una de las 5 matrices de pago empleadas

Matriz de pagos	Primera fila				
M_1	0,8	0,6	0,4	1,5	1
M_2	0,8	0,6	0,4	1,75	1
M_3	0,8	0,6	0,4	2	1
M_4	0,8	0,6	0,4	2,25	1
M_5	0,8	0,6	0,4	2,5	1

B. Parámetros de los algoritmos de DE

Se utilizó una implementación en Java de todos los algoritmos que está disponible públicamente[1] y que ha sido utilizada previamente en otros estudios de minería de datos que en último término requerían optimización de variables reales [12]. Se han fijado los siguientes valores para los parámetros:

- Función fitness: media aritmética del porcentaje de pago obtenido en 100 simulaciones independientes de 500 eventos cada una utilizando la estrategia que se esté evaluando. El porcentaje se mide como el pago real acumulado entre el pago máximo que se podría haber acumulado si ante cada evento se elige la acción que da el mayor pago y se asume que T no adivinó dicha accion.
- Tamaño de la población: 50 individuos.
- Criterio de parada: 50000 generaciones o estar 1000 generaciones sin mejorar la solución. En todos los casos el criterio efectivo de parada fue este último. Se habían agotado en torno a 3000 generaciones.
- Factor de escala: F = 0.5.
- Probabilidad de cruce: CR = 0.9.
- Estrategia de mutación en caso no adaptativo: DE/RandToBest/1.
- Memoria de aprendizaje (SADE): 30 generaciones.
- Porcentaje de individuos considerados para mutación (JADE): p = 0.05.
- Representación de soluciones: el cromosoma utilizado contiene valores reales acotados en el intervalo [0, 1]. Los algoritmos se aseguran de mantener dicho rango tras las mutaciones. Antes de evaluar, se normalizan los valores adecuadamente para asegurarse de que sean una distribución de probabilidad.

C. Resultados obtenidos

Para responder a las preguntas anteriores se ha hecho un conjunto de experimentos común. Después se han presentado los resultados desagregados de manera distinta, por un lado para responder a las dos primeras preguntas, y por otro para responder a la tercera. Los experimentos fueron los siguientes. Para cada matriz de pagos, se realizaron 30 ejecuciones independientes de cada una de las 4 variantes de Evolución Diferencial presentadas, utilizando para T la estrategia Más Frecuente, y se tomó la media de ellas. Después se hizo lo mismo cuando T aplica la estregia Proporcional a la Frecuencia, obteniendo así otros 4 resultados. Por tanto, para cada matriz de pagos, se ejecutaron 8 experimentos, 4 con cada estrategia para T. A su vez, el proceso se ha realizado por separado para las estrategias estáticas y las estrategias dinámicas.

[1] http://sci2s.ugr.es/EAMHCO/srcnew/advancesDEs.zip

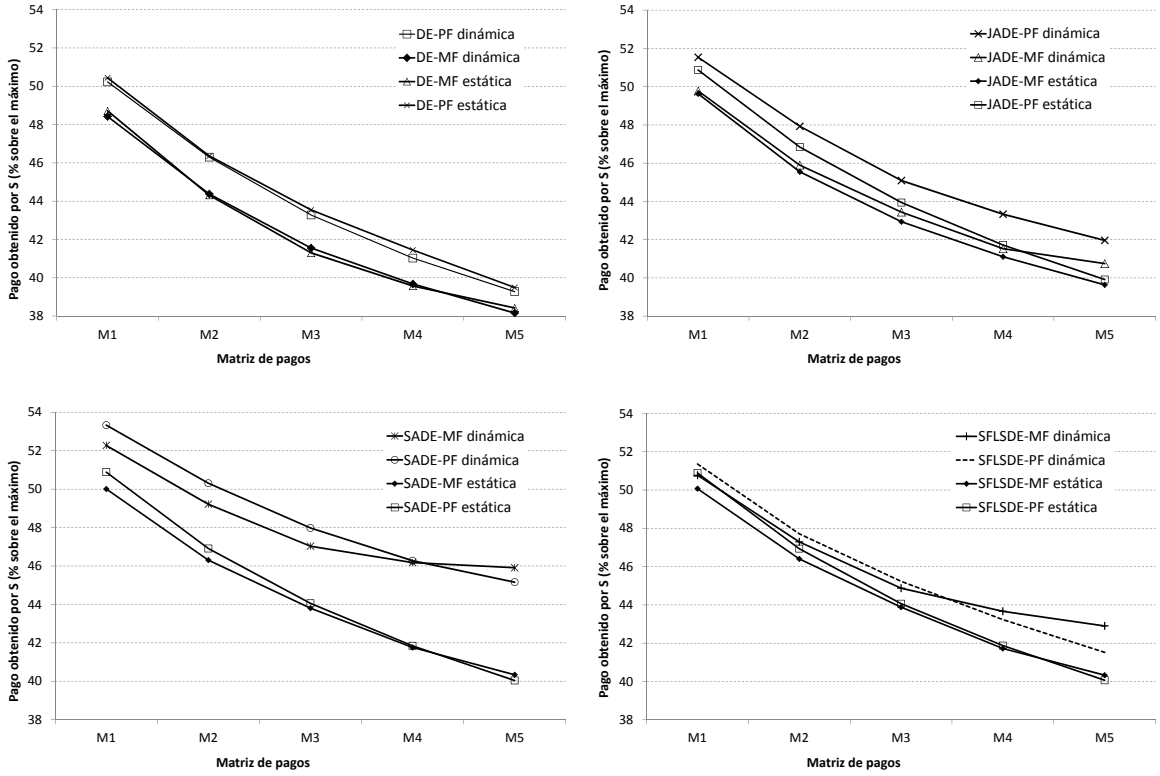


Fig. 3.Comparación del pago obtenido con estrategias estáticas y dinámicas, por separado para cada algoritmo. Todos los resultados son medias de 30 ejecuciones independientes.

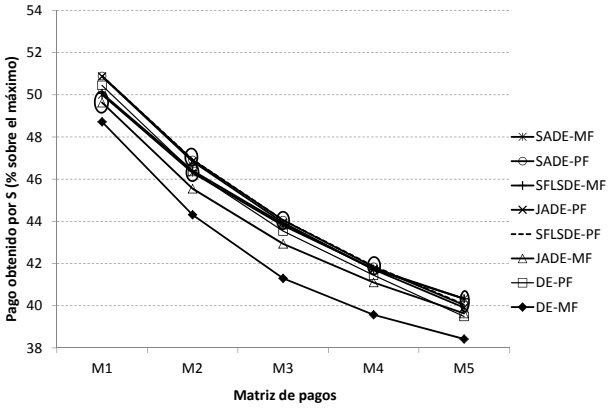
Con objeto de responder a las dos primeras preguntas, la Figura 3 muestra una comparación entre cada variante de algoritmo en el caso estático y dinámico frente a las dos posibles estrategias del agente T. En todos los casos se aprecia que las estrategias dinámicas dan un mayor pago a S, lo cual confirma que el enfoque de las estrategias mixtas dinámicas es válido a pesar de suponer un problema de optimización de mayor dificultad con más variables. La mejora es especialmente importante cuando T juega con PF dado que dicha estrategia es más fácilmente manipulable debido a que la acumulación de un tipo de acciones observadas durante un período hace que después sea más difícil cambiar el total de la proporción de respuestas observadas para predecir con otra acción distinta. En otras palabras, S es capaz de asegurarse que estará "a salvo"durante un período posterior si antes ha actuado de manera tal que el contenido de la memoria de T continuará inclinándose por cierta acción debido al alto número de veces que observó esa acción en un período inicial, a pesar de que actualmente S ya no se comporte de esa manera.

La respuesta a la tercera pregunta se deduce de la Figura 4. En el caso de las estrategias dinámicas, donde el problema de optimización es más complejo, se puede comprobar que, a pesar de tener sólo 5 ejecuciones independientes de cada variante, los resultados son consistentes para las 5 matrices de pago y apenas hay cruces entre las líneas. Como en cierto modo cabía esperar, tanto en el caso estático como en el dinámico, la versión básica de Evolución Diferencial es la que peor resuelve el problema, tanto cuando T juega PF como cuando juega MF. El algoritmo adaptativo SADE fue el que consiguió mejores resultados. Por otro lado, las diferencias entre los algoritmos son más notorias en el caso de las estrategias dinámicas debido a que el problema de maximización es más difícil y es donde se le saca mayor partido a las variantes más sofisticadas, como confirma tambien el test estadístico de Mann-Whitney realizado a las muestras de ejecuciones independientes obtenidas [2]. El problema de maximización de estrategias estáticas es relativamente simple y por ello todas las variantes dan resultados similares excepto el DE básico que funciona ligeramente peor que el resto. Tambien se pudo comprobar que las estrategias dinámicas funcionan mejor que las estáticas mediante dicho test, lo cual confirma la validez del enfoque de estrategias dinámicas propuesto.

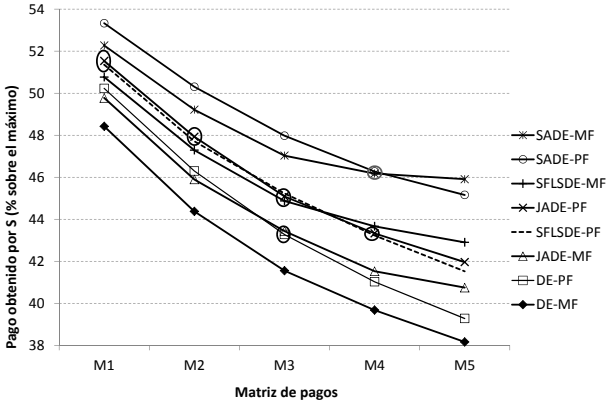
VI. Conclusiones y Trabajos futuros

Se han aplicado 4 algoritmos basados en Evolución Diferencial para el problema de diseño de estrategias aleatorizadas en un modelo con adversarios en el que una secuencia de estímulos externos depende de las

[2]Un estudio previo mediante un test de Kolmogorov-Smirnov al 95 % descartó la normalidad para aproximadamente la cuarta parte de los conjuntos de muestras obtenidos con las posibles combinaciones de algoritmos y estrategias para T, por lo que se prefirió no utilizar un test ANOVA



(a) Estrategias estáticas



(b) Estrategias dinámicas

Fig. 4.Comparación del pago obtenido con las estrategias obtenidas por cada algoritmo. Todos los resultados son medias de 30 ejecuciones independientes. Las pequeñas elipses indican diferencias no significativas según un test de Mann-Whitney; el resto de diferencias sí son significativas

respuestas dadas por los agentes. Se ha motivado la utilización en este modelo de estrategias aleatorizadas que varían en el tiempo, como una manera de evitar el aprendizaje del comportamiento de uno de los agentes por parte del otro, y se ha comprobado que tiene sentido la utilización de este tipo de estrategias dinámicas ya que dan mejores resultados que las estáticas, y estos resultados son significativos desde un punto de vista estadístico (test de Mann-Whitney). Se planteó el diseño de ambos tipos de estrategias como un problema de optimización en el que la función objetivo es una simulación no determinista. Se ha mostrado que los algoritmos funcionaron bien a pesar de esto, en un contexto en el que no habría sido posible una optimización mediante técnicas analíticas. Asimismo, se ha comprobado que hay diferencias notables en la calidad de las soluciones, especialmente en el caso de estrategias dinámicas que plantean un problema de optimización más complejo y donde las variantes más sofisticadas son capaces de mostrar todo su potencial.

Como trabajos futuros se considerará analizar el modelo desde una perspectiva más formal que permita obtener expresiones exactas o aproximadas para predecir el pago, y utilizar dichas expresiones como función objetivo en lugar de simulaciones.

Agradecimientos

Este trabajo ha sido financiado parcialmente por los proyectos TIN2008 - 01948 y TIN2008 - 06872 - C04 - 04 del Ministerio de Ciencia e Innovación y P07 - TIC - 02970 de la Junta de Andalucía.

Referencias

[1] A. Kott and W. M. McEneany, *Adversarial Reasoning: Computational Approaches to Reading the Opponents Mind*, Chapman and Hall/ CRC Boca Raton, 2007.

[2] H. Ishibuchi, C.H. Oh, and T. Nakashima, "Designing a decision making system for a market-selection game," Computing in Economics and Finance 1999 1131, Society for Computational Economics, 1999.

[3] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed ARMOR Protection: The Application of a Game Theoretic Model for Security at the Los Angeles International Airport," in *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS'08) - Industry and Applications Track*, 2008, pp. 125–132.

[4] F. Amigoni, N. Gatti, and A. Ippedico, "A game-theoretic approach to determining efficient patrolling strategies for mobile robots," in *Proceedings of the International Conference on Web Intelligence and Intelligent Agent Technology (IAT'08)*, 2008, pp. 500–503.

[5] F. Amigoni, N. Basilico, and N. Gatti, "Finding the optimal strategies for robotic patrolling with adversaries in topologically-represented environments," in *Proceedings of the 26th International Conference on Robotics and Automation (ICRA'09)*, 2009, pp. 819–824.

[6] D. Pelta and R. Yager, "On the conflict between inducing confusion and attaining payoff in adversarial decision making," *Information Sciences*, vol. 179, pp. 33–40, 2009.

[7] P. J. Villacorta and D. A. Pelta, "Theoretical analysis of expected payoff in an adversarial domain," *Information Sciences*, vol. 186, no. 1, pp. 93–104, 2012.

[8] P. Villacorta and D. Pelta, "Expected payoff analysis of dynamic mixed strategies in an adversarial domain," in *Proceedings of the 2011 IEEE Symposium on Intelligent Agents (IA 2011). IEEE Symposium Series on Computational Intelligence*, 2011, pp. 116 – 122.

[9] R. Storn and K.V. Price, "Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 10, pp. 341–359, 1997.

[10] K.V. Price, R.M. Storn, and J.A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer - Natural Computing Series, 2005.

[11] U. K. Chakraborty (Ed.), *Advances in Differential Evolution*, vol. 143 of *Studies in Computational Intelligence*, Springer, 2008.

[12] I. Triguero, S. Garcia, and F. Herrera, "Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification," *Pattern Recognition*, vol. 44, no. 4, pp. 901–916, 2011.

[13] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Trans. on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[14] J. Zhang and A. C. Sanderson, "Jade: adaptive differential evolution with optional external archive," *IEEE Trans. on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958, 2009.

[15] F. Neri and V. Tirronen, "Scale factor local search in differential evolution," *Memetic Computing*, vol. 1, no. 2, pp. 153–171, 2009.