

Comparación estadística de estrategias de decisión en presencia de adversarios: análisis y aplicación en búsqueda local

Pablo J. Villacorta, David A. Pelta

Dept. de Ciencias de la Computación e Inteligencia Artificial

E.T.S. de Ingenierías Informática y de Telecomunicación

Univ. de Granada

18071 Granada

olbapjo@correo.ugr.es, dpelta@decsai.ugr.es

Resumen

La decisión en presencia de adversario pretende encontrar estrategias de comportamiento para hacer frente a un adversario que observa nuestro comportamiento para intentar aprender a predecirlo. Partiendo de un modelo que captura esta situación, se ha observado la necesidad de utilizar métodos automáticos (heurísticos) para diseñar dichas estrategias. No obstante, se presenta el problema de cómo comparar entre sí dos estrategias, puesto que la evaluación de cada una implica la ejecución de una simulación no determinista. En este contexto, se ha aplicado una técnica estadística paramétrica basada en intervalos de confianza para comparar con certeza dos estrategias, y se ha estudiado su impacto al incorporarla a un algoritmo de Búsqueda Local. Los resultados obtenidos sirven de base para la aplicación de metaheurísticas más complejas orientadas al diseño automático de estrategias.

1. Introducción

La decisión en presencia de adversarios tiene como objetivo la determinación de estrategias óptimas contra un enemigo que observa nuestro comportamiento y nuestras acciones, y aprende a partir de ellas. Esta situación se da en muchas áreas de la vida real, en especial en ámbitos de lucha anti-terrorista y prevención

del crimen [3, 7].

El interés principal es el desarrollo de tecnologías para predicción de estrategias del oponente, reconocimiento de planes, descubrimiento de engaños y planificación. Los campos de aplicación no se limitan a cuestiones de seguridad sino también a la industria del videojuego, de negocios, transacciones, etc. [2]. Por ejemplo, el diseño de estrategias de patrullaje puede verse como una aplicación de la toma de decisiones en presencia de adversarios: el objetivo es diseñar rutas de patrullaje que minimicen las posibilidades de que un enemigo consiga penetrar una determinada frontera de seguridad.

Dados dos agentes S y T (el adversario), ambos pretenden maximizar sus recompensas, que son inversamente proporcionales una respecto a otra. Una forma que tiene el agente S de defenderse de los intentos de aprendizaje de T es tomar decisiones orientadas a confundir a T , pese a que esto le supondrá probablemente no elegir la mejor acción y por tanto obtener una recompensa menor que la óptima posible. El problema es averiguar cómo definir este tipo de estrategias para S . Pueden ser válidas estrategias diseñadas manualmente, pero el diseñador podría estar pasando por alto ciertas alternativas debido a las limitaciones inherentes al ser humano para la exploración de un espacio de estrategias.

En los últimos años, el diseño automático de sistemas ha ido adquiriendo cada vez más im-

portancia y focalizando mucha atención, especialmente debido al auge de las técnicas evolutivas de búsqueda y la disponibilidad de potencia computacional. Como ejemplos se puede citar el diseño de sistemas auto-ensamblados[4], cierto tipo de redes neuronales [5], controladores para robótica colectiva [1] o redes orientadas a árboles [8].

En el diseño automático de estrategias de decisión, ciertas cuestiones son especialmente relevantes, como por ejemplo la representación de una estrategia o la comparación eficiente de estrategias entre sí durante la exploración del espacio de estrategias. La evaluación de una estrategia consiste en valorar su comportamiento en un escenario particular. Esto se realiza mediante la ejecución de una simulación no determinista y por tanto, estamos frente a una función de fitness con ruido. Un esquema típico para tratar con este tipo de funciones es repetir la simulación cierto número de veces y luego, tomar la media de las evaluaciones como el fitness de la estrategia.

En [9] se propuso un método de comparación de estrategias basado en técnicas estadísticas, y se comprobó que mejoraba ligeramente el rendimiento cuando se utilizaba dentro de un Algoritmo Genético (AG) para exploración del espacio de estrategias. Sin embargo, la utilización de un AG implica un gran número de decisiones de diseño, típicamente de los operadores de cruce, mutación y selección con sus respectivos parámetros, esquema de inicialización de la población y su tamaño, etc. Todos estos factores tienen gran influencia en los resultados finales. De cara a comprender mejor el impacto de la metodología de comparación de estrategias propuesta, se propone repetir el análisis de [9] utilizando una búsqueda local simple. De esta manera, se espera sentar las bases para abordar el diseño automático de estrategias utilizando algoritmos meméticos donde se combinen algoritmos genéticos con búsqueda local.

A partir de aquí, los objetivos que se plantean son: (a) aplicar un método de comparación de estrategias basado en tests estadísticos a una metaheurística de Búsqueda Local, (b) analizar la influencia de esta forma de com-

paración respecto a la de otros métodos posibles a la hora de guiar la búsqueda, y (c) comparar el comportamiento de la Búsqueda Local para diseño de estrategias con el de un Algoritmo Genético que ya se propuso en una contribución anterior, y su dependencia del método de comparación que se le incorpore.

El trabajo se organiza como sigue: en la sección 2 se explica el modelo propuesto en [6]. En la sección 3 se motiva la utilización del diseño automático de estrategias para este modelo y la utilidad de las técnicas estadísticas para comparación de estrategias. La sección 4 resume los resultados previos en este sentido obtenidos en [9]. En la sección 5 se describe la utilización de un algoritmo simple de Búsqueda Local. A continuación, en la sección 6 se explica la metodología experimental y se discuten y comparan los resultados obtenidos respecto a los ya existentes. Por último, la sección 7 contiene las conclusiones y aspectos que se pueden estudiar en el futuro.

2. Modelo de decisión en presencia de adversarios

El modelo consta de dos agentes S y T (el adversario), un conjunto de posibles entradas o eventos $I = \{i_1, i_2, \dots, i_n\}$ generados por un tercer agente R , y un conjunto de posibles respuestas o acciones $A_i = \{a_1, a_2, \dots, a_m\}$ asociadas a cada evento. Existe también una matriz de pagos P :

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

El agente S debe decidir qué acción elegir dada una entrada particular i_k 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, que en el modelo se ha representado como un agente R . Se proporciona un estímulo en cada instante de tiempo. En principio, los

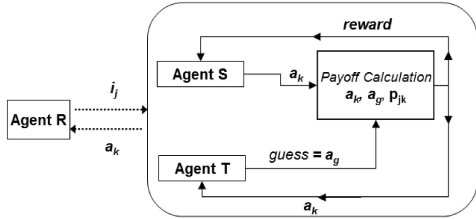


Figura 1: Representación gráfica del modelo

estímulos de una secuencia son independientes entre sí y se generan aleatoriamente.

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. E representa la longitud de la secuencia de estímulos.

```

Para  $j = 1$  hasta  $E$  hacer
  Llega un nuevo estímulo  $i_j$ .
  El agente  $T$  hace una predicción  $a_g$ 
  El agente  $S$  escoge una acción  $a_k$ 
  Se calcula el pago para  $S$ 
  El agente  $T$  registra el par  $i_j, a_k$ 
Fin

```

Algoritmo 1: Secuencia de pasos que forman el modelo

Ante un estímulo i_j , el agente S elige una acción y al mismo tiempo el agente T da una predicción sobre la acción que escogerá S . Además, T mantiene su propia matriz de observaciones, O , de dimensiones $M \times N$. 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 se define como:

$$p' = p_{jk} \times F(a_g, a_k) \quad (1)$$

siendo F :

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

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, siempre se 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 es la más difícil 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} .

3. Diseño automático

Una posible mejora es la definición de estrategias variables en el tiempo. Se define una *táctica* (o plan) como una sucesión de estrategias más simples. La longitud de una táctica es el número de estrategias simples de que consta. Además, las tácticas pueden utilizarse de una manera cíclica, de forma que para responder al estímulo n -ésimo con una táctica de longitud L , se utilizará la estrategia $(n \bmod L)$ de dicha táctica. La utilización de una estrategia distinta (aunque se agrupen en períodos) en cada instante de tiempo dificulta el aprendi-

zaje por parte de un observador.

Cada estrategia consta de dos fases. Primero se elige un conjunto de k acciones candidatas (siendo k un parámetro de la estrategia) según un determinado criterio (las mejores, o aleatoriamente, o con probabilidad proporcional al pago), y después, del conjunto de acciones candidatas, se selecciona una (utilizando otro criterio diferente) que será la que finalmente usaremos para responder al estímulo. En un modelo con K_{max} posibles acciones, el número de estrategias posibles es (ver [9] para una explicación detallada):

$$ESTRATEGIAS = 8 \times (K_{max} - 2) + 3 \quad (3)$$

Si utilizamos tácticas de longitud L , el número de tácticas posibles será

$$TACTICAS = (8 \times (K_{max} - 2) + 3)^L \quad (4)$$

Suponiendo una instancia simple de nuestro modelo, donde $K_{max} = 5$, el número de tácticas posibles es de 531441. Lógicamente, es imposible evaluarlas todas exhaustivamente ya que la evaluación consiste en ejecutar el modelo con una secuencia de estímulos suficientemente larga haciendo que el agente S utilice la táctica que queremos evaluar. El tiempo empleado para evaluar una cantidad tan grande de tácticas distintas hace que sea necesario emplear técnicas de búsqueda heurística en el espacio de tácticas.

3.1. Esquema de Representación

Como una estrategia puede describirse con 3 elementos, es directo codificarla computacionalmente como un número entero de 3 cifras donde cada cifra representa uno de estos 3 elementos. A su vez, una táctica puede representarse como un vector de estrategias (enteros). Esta codificación discreta facilita la tarea de búsqueda en el espacio de tácticas posibles.

3.2. Comparación de tácticas

La búsqueda en el espacio de tácticas presenta una peculiaridad en relación a otros problemas de optimización combinatoria. El *gap* explicado anteriormente es distinto en cada evalua-

ción de una misma táctica, ya que las estrategias que forman la táctica contienen en sí mismas criterios probabilísticos. Esto implica que la bondad o *fitness* de una solución es no determinista, lo cual dificulta mucho la búsqueda.

En [9] se propuso un método estadístico para la comparación de tácticas. El método más simple consiste en repetir muchas veces la simulación con el modelo, y tomar el *gap* medio resultante de todas las simulaciones realizadas. No obstante, esto puede llevar demasiado tiempo para obtener resultados suficientemente precisos, y además, en el caso de tácticas similares, no da resultados claros.

El método propuesto consistía en enfocar la comparación de tácticas como un problema estadístico de comparación de dos muestras tomadas de dos poblaciones distintas. Cada muestra sería el resultado de una simulación. El objetivo es realizar la comparación construyendo un intervalo de confianza para la diferencia entre las medias muestrales de ambas poblaciones (tácticas). Si el intervalo de confianza construido no contiene al 0, entonces es posible determinar con certeza cuál de las dos poblaciones (tácticas) que estamos comparando es mejor. La fórmula utilizada es la siguiente:

$$\bar{Z}(n) \pm t_{n-1, 1-\frac{\alpha}{2}} \cdot \sqrt{\frac{s^2(n)}{n}} \quad (5)$$

donde

$$\bar{Z}(n) = \frac{1}{n} \sum_{j=1}^n Z_j, \quad Z_j = X_j - Y_j \quad (6)$$

es decir, la diferencia entre cada par de muestras de las tácticas X e Y a comparar, y

$$s^2(n) = \frac{1}{n-1} \sum_{j=1}^n (Z_j - \bar{Z}(n))^2 \quad (7)$$

El nivel de confianza viene dado por $100(1-\alpha)\%$ y se ha fijado en el 95%. t es una variable aleatoria que sigue una distribución t-Student con $n-1$ grados de libertad, siendo n el número de muestras (en nuestro caso, 30, 60 o 90 según sean necesarias).

Si las poblaciones siguen una distribución normal, entonces el intervalo es exacto, y el procedimiento es equivalente a realizar un test estadístico t-student para intentar rechazar la hipótesis de igualdad de las medias poblacionales. Las condiciones previas requeridas para poder realizar un test t-student son la normalidad muestral y la homoscedasticidad (varianzas similares en las poblaciones que se están comparando).

Este método permite que el número de repeticiones de una simulación para evaluar una táctica sea adaptativo, es decir, distinto en función de la comparación que estemos haciendo. La idea es evaluar cada táctica un número pequeño de veces y construir el intervalo de confianza para la diferencia de las medias. Si no contiene al 0, la comparación es concluyente y no necesitamos tomar más muestras. En cambio, si contiene al 0, aún no tenemos certeza de cuál de las dos tácticas es mejor, por lo que añadimos más muestras (es decir, repetimos la simulación más veces) a cada táctica y después volvemos a construir el intervalo de confianza, hasta un determinado máximo número de muestras por táctica. Así, cuando las tácticas que comparamos son muy diferentes, bastarán unas pocas muestras (al menos 30) para que el intervalo de confianza construido no contenga al 0 y que la comparación sea concluyente. Reservaremos un mayor número de repeticiones sólo para comparaciones más dudosas. Si tras llegar al máximo permitido de muestras por táctica la comparación sigue siendo dudosa, simplemente nos guiamos por el gap medio de cada táctica para determinar cuál es mejor.

4. Resultados previos

En primer lugar, en [9] se comprobó que la utilización de tácticas periódicas mostró mejores resultados en media respecto a tácticas más primitivas que no variaban en el tiempo: el conjunto de todas las tácticas de longitudes 2 y 3 (que se evaluaron exhaustivamente) tenía un gap medio menor que las tácticas de longitud 1 en experimentos con una instancia simple del modelo, con 5 estados (5 estímulos

posibles y 5 acciones). Esto confirmaba que el enfoque de tácticas variables en el tiempo es beneficioso, y por tanto queda justificada una experimentación más detallada.

A continuación se comprobó experimentalmente que la población formada por las muestras (gap) obtenidas al repetir varias veces la evaluación de una misma táctica sigue una distribución normal en un porcentaje muy alto de los casos (95 % de las tácticas evaluadas). En el caso de la homoscedasticidad, un 85 % de las comparaciones 2 a 2 mostraban que las muestras de las 2 tácticas comparadas en cada caso presentan varianzas similares. Por tanto, se cumplen la normalidad y la homoscedasticidad, por lo que es factible utilizar tests paramétricos para la comparación, y el intervalo de confianza para la diferencia de las medias será exacto.

Se evaluaron 3 métodos distintos de comparación: uno basado en intervalos de confianza al 95 %, ya descrito; otro basado simplemente en la comparación de las medias muestrales, y otro basado en la comparación de las medianas muestrales.

El método fue incorporado a una heurística de búsqueda, en concreto un Algoritmo Genético Generacional (AG) ejecutado 100 veces con cada uno de los 3 métodos de comparación. Los resultados mostraban claramente que el método de las medianas era peor. En cambio, aunque parecía que el método de intervalos de confianza daba soluciones ligeramente mejores tras las 100 ejecuciones del AGG respecto al método que sólo comparaba las medias muestrales, esta ligera mejora aparente no era significativa al aplicar un test estadístico no paramétrico para muestras independientes (Mann-Whitney).

5. Utilización de la Búsqueda Local

Para profundizar en el estudio de esta técnica de comparación de soluciones en el caso de fitness obtenidos mediante simulaciones no deterministas, se ha utilizado ahora como algoritmo de búsqueda una Búsqueda Local del Primer Mejor en el espacio de tácticas. Se pretende estudiar la influencia de las compara-

ciones a la hora de guiar la búsqueda en un contexto diferente: una metaheurística basada en entornos y no en poblaciones. Como se trata del mismo problema, se puede asumir que el método de intervalos de confianza es aplicable porque se siguen cumpliendo las mismas condiciones.

La hipótesis de partida es que, por la propia naturaleza de la técnica de búsqueda, puede ser más importante acertar en las comparaciones cuando nos basamos en entornos para irnos moviendo por el espacio, ya que cada decisión errónea nos irá entorpeciendo y llevando por un camino cada vez más equivocado. En una heurística basada en poblaciones, puede no ser tan crucial la selección de los padres adecuados para el cruce. Es posible que la selección de padres erróneos se compense en cierta medida a lo largo del tiempo gracias a que (a) ciertos cruces de individuos no tan buenos pueden sin embargo generar un individuo mejor, y (b) la mutación puede influir también favorablemente en este proceso.

```

Generar una táctica inicial  $S_m$  aleatoria
Mientras evaluaciones < TOPE hacer
     $S_{vec} \leftarrow$  Seleccionar aleatoriamente
        una estrategia de  $S_m$  y mutarla
    Si  $MEJ(S_{vec}, S_m)$  hacer
         $S_m \leftarrow S_{vec}$ 
    Fin-si
    Incrementar evaluaciones según el
    método de comparación utilizado
Fin

```

Algoritmo 2: Búsqueda Local del Primer Mejor

El operador $MEJ(\cdot, \cdot)$ implementa el método de comparación. En el caso de los intervalos de confianza, se encarga además de ejecutar simulaciones adicionales y repetir la construcción del intervalo de confianza para la diferencia de las medias en caso de que la comparación sea dudosa.

6. Experimentos y resultados

La metodología seguida ha sido la misma empleada en [9]. De nuevo se han utilizado las 3 técnicas ya descritas, en las mismas condiciones (tramos de 30 simulaciones que se añaden

Cuadro 1: Valores de gap de las tácticas obtenidas mediante BL con cada método de comparación (datos medios sobre 100 ejecuciones)

	BL_{media}	BL_{ic}	$BL_{mediana}$
Media	39,684	39,647	40,020
Desv. típ.	0,0987	0,0679	0,3727
Mejor	39,537	39,521	39,578
Peor	40,110	39,911	41,565

si es necesario, hasta un máximo de 90, en el caso de intervalos de confianza; sólo 30 simulaciones en el caso de los métodos basados en la comparación de la media muestral y de la mediana muestral). El límite de evaluaciones se fijó en 150.000 para que los resultados fuesen comparables con los obtenidos por el AGG de [9], y se utilizó la misma matriz de pagos. Las secuencias de estímulos utilizadas fueron las mismas que en [9], y cada nueva simulación fue ejecutada con una secuencia de estímulos diferente, siendo todas ellas generadas previamente a la ejecución del algoritmo de búsqueda para que todas las tácticas se evaluaran en las mismas condiciones.

Se realizaron 100 ejecuciones de la Búsqueda Local con cada método de comparación y se almacenó la solución obtenida en cada una. Las 300 soluciones obtenidas se evaluaron posteriormente todas en las mismas condiciones, mediante 8000 nuevas simulaciones para cada solución y utilizando las mismas 8000 secuencias de estímulos de [9], y se anotó el gap medio de estas 8000 simulaciones. Finalmente se obtuvo la media de las 100 soluciones dadas por la BL con cada método de comparación (media, intervalos de confianza (*ic*) y mediana), como muestra el 1.

La tabla muestra que la tendencia con la BL es la misma que la del AG: los resultados con el método de intervalos de confianza son ligeramente mejores, lo que indica que esta manera de "distribuir" el número total de simulaciones disponibles es más adecuada. A su vez, el método de la mediana es mucho mejor que los otros dos. La gran diferencia (que no se aprecia en la tabla) respecto a los experimentos previos con un AGG es que, en esta ocasión,

Cuadro 2: Valores de gap de las tácticas obtenidas mediante AGG con cada método de comparación (datos medios sobre 100 ejecuciones). Tabla reproducida de [9]

	AG_{media}	AG_{ic}	$AG_{mediana}$
Media	39,617	39,604	39,761
Desv. típ.	0,049	0,046	0,207
Mejor	39,539	39,515	39,540
Peor	39,768	39,771	40,569

las diferencias entre BL_{ic} y BL_{media} sí son significativas desde un punto de vista estadístico. El test de Mann-Whitney para muestras independientes realizado a las dos poblaciones de los gap correspondientes a las 100 soluciones de cada uno de estos dos métodos rechaza la hipótesis de igualdad de medias con un p-valor de 0,000. Esta novedad confirma nuestra hipótesis de partida: en un algoritmo basado en entornos como la BL, es crucial determinar correctamente la dirección en la que nos movemos al explorar el espacio, y comparaciones falseadas llevan a malos resultados. El reparto adaptativo de las simulaciones permite aprovecharlas mejor, lo que a la larga conduce a mejores soluciones.

En la Tabla 2, se ha reproducido la tabla de resultados de [9], para poder comparar los resultados. Nótese que la BL obtiene en todos los casos resultados peores en media que los obtenidos por el AGG. Esto da una interesante pista sobre el empleo de metaheurísticas basadas en poblaciones en este problema y su capacidad de explorar el espacio completo, siendo especialmente más eficaces para escapar de óptimos locales.

La figura 2 muestra la distribución por cuartiles de los gap de las 300 soluciones obtenidas por la BL evaluadas a posteriori. Se indica el porcentaje de soluciones de cada método que entran en cada cuartil. Se confirman los resultados medios de la tabla, ya que tras ordenar de menor (mejor) a mayor (peor) gap las 300 soluciones, la mayoría de las estrategias en el primer y segundo cuartil, provienen del método de intervalos de confianza.

La figura 3 muestra el impacto del uso de la

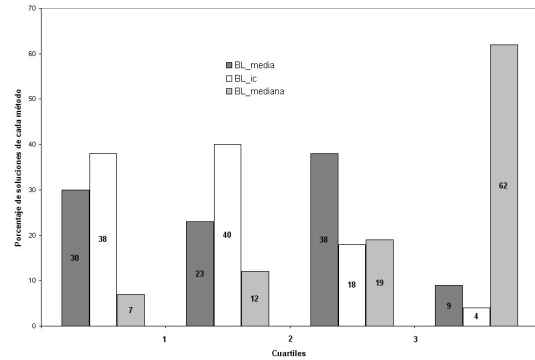


Figura 2: Distribución de las soluciones de la BL por cuartiles

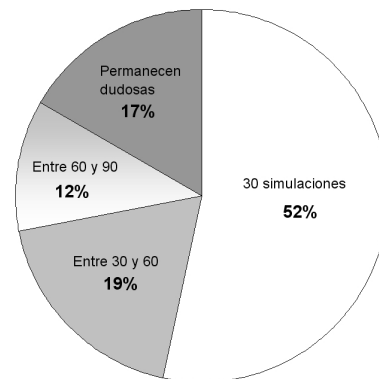


Figura 3: Porcentaje de comparaciones que requirieron 30, 60 o 90 simulaciones. Resultados medios sobre 100 ejecuciones independientes de BL_{ic}

técnica de comparación por intervalos de confianza durante la búsqueda, en media sobre las 100 ejecuciones del método de IC. Sólo la mitad (52%) de las comparaciones son claras con sólo 30 simulaciones. Esto explica que el método de las medias con sólo 30 simulaciones funcione significativamente peor. Además, añadir simulaciones clarifica los resultados en un 31% adicional de comparaciones (19% + 12%), y sólo un 17% permanecen dudosas. Esto confirma que los intervalos de confianza mejoran considerablemente la exploración del espacio en la Búsqueda Local.

7. Conclusiones y trabajos futuros

Se ha aplicado un método de comparación de individuos basado en estadística paramétrica a un problema de optimización cuya función de costo es “ruidosa” y se ha mostrado que este aspecto influye de manera decisiva en el caso de metaheurísticas basadas en entornos, ya que los resultados mejoran considerablemente respecto a la utilización de técnicas tradicionales de comparación de medias muestrales.

En un trabajo previo, se comprobó que los Algoritmos Genéticos son una buena herramienta para diseño automático de este tipo de estrategias, y además, las soluciones obtenidas son mejores que los obtenidos mediante Búsqueda Local del Primer Mejor en una instancia simple del modelo.

Ambos resultados abren la puerta a la incorporación de la Búsqueda Local a un Algoritmo evolutivo con el fin de obtener un Algoritmo Memético que explore mejor el espacio de posibles tácticas. Considerando que la comparación de estrategias mediante intervalos de confianza afecta de forma crucial a la Búsqueda Local y no tanto al Algoritmo Genético, creemos que es un punto interesante incorporar dicha técnica estadística sólo a la fase de Búsqueda Local dentro de un Algoritmo Memético.

Otra posibilidad es aplicar la técnica propuesta a otro tipo de problemas de optimización con funciones ruidosas y extender la metodología a aquellos casos donde los datos no sigan una distribución normal.

Agradecimientos

Este trabajo se ha realizado con el apoyo de los proyectos TIN2008-01948 del Ministerio de Ciencia e Innovación y P07-TIC-02970, de la Consejería de Innovación, Ciencia e Industria, Junta de Andalucía.

Referencias

[1] G. Baldassarre and S. Nolfi. Strengths and synergies of evolved and designed controllers: A study within collective robotics. *Artificial Intelligence*, 173(7-8):857–875, 2009.

- [2] A. Kott and W. M. McEneaney. *Adversarial Reasoning: Computational Approaches to Reading the Opponents Mind*. Chapman and Hall/ CRC Boca Raton, 2007.
- [3] A. Kott and M. Ownby. Tools for real-time anticipation of enemy actions in tactical ground operations. In *Proceedings of the 10th International Command and Control Research and Technology Symposium*, 2005.
- [4] N. Krasnogor, G. Terrazas, D. Pelta, and G. Ochoa. A critical view of evolutionary design of self-assembly system. In *Proceedings of the Conference on Artificial Evolution (EA'05)*, volume 3871 of *Lecture Notes in Computer Science*, pages 179–188. Springer, 2005.
- [5] H.-S. Park, W. Pedrycz, and S.-K. Oh. Evolutionary design of hybrid self-organizing fuzzy polynomial neural networks with the aid of information granulation. *Expert Systems with Applications*, 33(4):830–846, 2007.
- [6] D. Pelta and R. Yager. On the conflict between inducing confusion and attaining payoff in adversarial decision making. *Information Science*, 179:33–40, 2009.
- [7] R. Popp and J. Yen. *Emergent Information Technologies and Enabling Policies for Counter-Terrorism*. John Wiley and Sons Hoboken, NJ, 2006.
- [8] S. Salcedo-Sanz, M. Naldi, A. M. Perez-Bellido, A. Portilla-Figueras, and E. G. Ortiz-Garcia. Evolutionary design of oriented-tree networks using cayley-type encodings. *Information Sciences*, 179(20):3461–3472, 2009.
- [9] P. J. Villacorta and D. A. Pelta. Evolutionary design and statistical assessment of strategies in an adversarial domain. In *Proceedings of the IEEE Conference on Evolutionary Computation (CEC'10)*, 2010. In press.