



Máster en Ingeniería de Sistemas y Control

Técnicas heurísticas para la búsqueda en tiempo mínimo (Comparativa)

Alumno: Francisco Javier Yáñez Zuluaga
Directores: Eva Besada Portas
Pablo Lanillos Pradas

Curso 2013-2014

Convocatoria: Septiembre

Máster en Ingeniería de Sistemas y Control

Técnicas heurísticas para la búsqueda en tiempo mínimo (Comparativa)

Proyecto específico propuesto por un profesor.

Alumno: Francisco Javier Yáñez Zuluaga

Directores: Eva Besada Portas

Pablo Lanillos Pradas

Curso 2013-2014

Convocatoria: Septiembre

Hoja de Calificaciones



Autorización

Autorizamos a la Universidad Complutense y a la UNED a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado:

A handwritten signature in blue ink, appearing to be 'Jm', written on a light-colored background.

Firma del alumno

Resumen

Este trabajo se centra en el análisis del funcionamiento de diferentes heurísticas para resolver el problema de búsqueda en tiempo mínimo en entornos inciertos. El problema requiere minimizar el tiempo en encontrar un objetivo cuya localización y dinámicas son desconocidas, empleando para ello varios agentes equipados con sensores capaces de detectar su presencia. El problema, computacionalmente complejo, se aborda por medio de diferentes técnicas heurísticas: algoritmos de estimación de distribuciones, algoritmos genéticos, y variantes mixtas de ambos. La primera técnica se incorpora de un estudio previo, la segunda se implementa para este trabajo, y la tercera se diseña especialmente para él. El análisis estadístico realizado muestra como un EDA clásico converge en menos iteraciones a las mejores soluciones mientras que los GA son más rápidos desde el punto de vista computacional.

Palabras Clave

Búsqueda en Tiempo Mínimo, Algoritmos Genéticos, Algoritmos de Estimación de la Distribución, Entornos con Incertidumbre.

Tabla de contenido

1.	Introducción	5
1.1	Objetivos	6
1.2	Estructura de la memoria.....	6
2.	Contexto.....	9
2.1	Casos históricos	9
2.2	Estado del Arte	10
3.	Técnicas	13
3.1	Algoritmos Genéticos	13
3.2	Algoritmos de Estimación de Distribuciones.....	15
3.2.1.	Redes Bayesianas (BN)	16
3.2.2.	Algoritmos de Optimización Bayesiana (BOA)	17
3.3	Algoritmos Mixtos	18
4.	Desarrollo	21
4.1	Formulación matemática.....	21
4.1.1	Variables del problema	22
4.1.2	Funciones de mérito	23
4.2	Algoritmo de optimización	25
4.2.1	Codificación	25
4.2.2	Propiedades genéricas de las heurísticas.....	26
4.2.3	Algoritmo Genético (GA)	27
4.2.4	Algoritmos de optimización bayesiana (BOA)	29
4.2.5	Algoritmos Mixtos (ROM y GOM).....	30
5.	Resultados	31
5.1	Escenarios.....	31
5.2	Heurísticas analizadas	32
5.3	Metodología del análisis estadístico.....	32
5.4	Tiempo Esperado Truncado Experimental	33
5.4	Función de mérito respecto Iteraciones.....	36
5.5	Función de mérito respecto Tiempo	39
5.6	Probabilidad Acumulada	41
5.7	Comparativa Mejores.....	44
6	Conclusiones.....	49

6.1	Consecución de objetivos.....	49
6.2	Trabajo futuro	49
7	Siglas y abreviaturas.....	51
8	Bibliografía	53

Índice de Figuras

Figura 2. 1	Cuadrícula empleada en la búsqueda del USS Scorpion	10
Figura 3. 1	Operador de cruce de 1 punto	14
Figura 3. 2	Operador de mutación	14
Figura 3. 3	Pseudocódigo de GA.....	15
Figura 3. 4	Pseudocódigo EDA.....	16
Figura 3. 5	Ejemplo de gráfico de dependencia	17
Figura 3. 6	Representación BOA.....	17
Figura 3. 7	Representación Mixto	19
Figura 3. 8	Variación Mixta.....	19
Figura 3. 9	Pseudocódigo ROM	20
Figura 3. 10	Pseudocódigo GOM.....	20
Figura 4. 1	Esquema del problema de búsqueda en tiempo mínima	21
Figura 4. 2	Relación entre las posiciones y acciones de los agentes.....	22
Figura 4. 3	Probabilidad de que el objetivo se desplace.....	22
Figura 4. 4	Esquema del funcionamiento del modelo del sensor ideal	23
Figura 4. 5	Mapa de creencia	25
Figura 4. 6	Codificación direcciones de los agentes	26
Figura 4. 7	Visualización codificación	26
Figura 4. 8	Cálculo del DD para un escenario estático sobre un recorrido de un agente	27
Figura 4. 9	Cálculo del TET para un escenario estático sobre un recorrido de un agente	27
Figura 4. 10	Selección por Torneo	28
Figura 4. 11	Cruce de 1 punto	28
Figura 4. 12	Cruce de 2 puntos.....	28
Figura 4. 13	Mutación	29
Figura 4. 14	Selección padre ROM y GOM	30
Figura 4. 15	Variación Mixta.....	30
Figura 5. 1	Escenarios.....	31
Figura 5. 2	Heurísticas empleadas	32
Figura 5. 3	Tiempo Experimental para DD	34
Figura 5. 4	Tiempo Experimental para TET	35
Figura 5. 5	Función de mérito respecto Iteraciones para DD.....	37
Figura 5. 6	Función de mérito respecto Iteraciones para TET	38
Figura 5. 7	Función de mérito respecto Tiempo para DD.....	39
Figura 5. 8	Función de mérito respecto Tiempo para TET.....	40
Figura 5. 9	Probabilidad Acumulada para DD	42
Figura 5. 10	Probabilidad Acumulada para TET	43

Figura 5. 11 Escenario 1 DD.....	44
Figura 5. 12 Escenario 2 DD.....	45
Figura 5. 13 Escenario 3 DD.....	45
Figura 5. 14 Escenario 4 DD.....	45
Figura 5. 15 Escenario 5 DD.....	46
Figura 5. 16 Escenario 1 TET.....	46
Figura 5. 17 Escenario 2 TET.....	46
Figura 5. 18 Escenario 3 TET.....	47
Figura 5. 19 Escenario 4 TET.....	47
Figura 5. 20 Escenario 5 TET.....	47

1. Introducción

Uno de los objetivos de la tecnología ha sido realizar todo tipo de tareas de una forma más rápida y eficiente. Una de esas tareas y que se abordará en este Trabajo Fin de Máster (TFM) es el problema de búsqueda.

Un problema de búsqueda es una situación en la que necesitamos encontrar un objetivo, desconociendo su posición exacta. Esto ocurre, por ejemplo, en el caso de accidentes marítimos. Una accidente clásico fuente de estudio es el del USS Scorpion [ISBA, 2009], un submarino que se hundió cubriendo una ruta entre el Mediterráneo y Virginia (EE.UU.). Fue encontrado 6 días después de su hundimiento. Más adelante se hablará de cómo se realizó la búsqueda, pero se puede adelantar que en ella se utilizaron mapas de probabilidad.

En nuestro caso, además de querer conocer la posición del objetivo, queremos hacerlo en el menor tiempo posible. Esto se debe a que el tiempo es un factor importante. En el ejemplo anterior puede significar la diferencia entre la vida o la muerte debido a las bajas temperaturas. Un ejemplo real de la relevancia del tiempo se puede apreciar en el caso de la plataforma Hibernia (Canadá) [Ljunggren, 2009]. El 12 de Marzo de 2009, un helicóptero que transportaba trabajadores se precipitó sobre las aguas del Atlántico; sólo sobrevivió uno de los dieciocho ocupantes ya que el resto pasó demasiado tiempo en el agua.

Un factor especialmente importante a la hora de abordar el problema de búsqueda de este TFM es que trabajamos con múltiples fuentes de incertidumbre. Además de no conocer la posición del objetivo, tenemos incertidumbre tanto en la dinámica del objetivo (en el caso anterior los naufragos estarían a merced de las mareas y no fijos en un punto) como en los sensores (en el caso ejemplo podemos pasar por el lugar del accidente y no detectar a los naufragos).

La mayoría de estas tareas de búsqueda, a día de hoy, las realiza el ser humano de forma “manual”, pero gracias a distintos avances tecnológicos se intenta conseguir formas de realizar dichas búsqueda de forma autónoma.

Si queremos automatizar el proceso de búsqueda, debemos tener un buen sistema de recogida de información y un sistema de toma de decisiones a la hora de elegir el camino a seguir. En este TFM nos centraremos en el análisis del funcionamiento de dos familias de técnicas diferentes dentro del proceso de toma de decisiones: la optimización del tiempo de búsqueda mediante Algoritmos Genéticos y mediante Algoritmos de Estimación de Distribuciones. Adicionalmente, también se analizará el funcionamiento de una técnica mixta que auna las propiedades de ambas familias.

Por tanto, y a modo de resumen, este TFM abordará el estudio de las prestaciones de dos familias de algoritmos para realizar la búsqueda de objetivos en los que el tiempo es un factor esencial en base a un mapa de probabilidades. Además, se propondrá y analizará el funcionamiento de una técnica nueva adicional, inspirada en las características de ambas familias.

1.1 Objetivos

El objetivo principal de este TFM es el análisis estadístico del funcionamiento de diferentes técnicas de optimización para resolver el problema de búsqueda en tiempo mínimo (Minimum Time Search, MTS). Las técnicas a analizar serán, al menos, dos heurísticas capaces de resolver problemas de búsqueda y optimización complejos: los Algoritmos Genéticos (Genetic Algorithm, GA) y los Algoritmos de Distribución de Probabilidades (Estimation Distribution Algorithms, EDAs).

Para llevar a cabo este objetivo será necesario:

- Estudiar el problema de búsqueda en tiempo mínimo, prestando especial atención a la comprensión de los elementos que lo constituyen y su formulación matemática.
- Familiarizarse con las características y en el manejo de un EDA (en concreto un Algoritmo de Optimización Bayesiana) ya existente para resolver el problema de MTS.
- Implementar un GA que resuelva el problema de MTS para poder comprobar si esta técnica, computacionalmente más sencilla que el EDA disponible, permite resolver el problema.
- Opcionalmente, implementar alguna técnica adicional (Mixta), que combine características de los EDAs y los GA.
- Familiarizarse con el uso de las herramientas desarrolladas para realizar un análisis estadístico del funcionamiento de los distintos algoritmos. Adaptar estas herramientas para realizar el análisis de diferentes versiones de los GA y EDA.
- Realizar el análisis estadístico, necesario para determinar si alguna de estas heurísticas con soporte estocástico, que evita que siempre se obtenga la misma solución ante el mismo escenario de búsqueda, tiene habitualmente un comportamiento mejor que el resto.

1.2 Estructura de la memoria

El resto de esta memoria se divide en 5 bloques, cada uno recogido en uno de los capítulos que la conforman.

En el capítulo 2, Contexto, se presenta brevemente el contexto histórico de los problemas de búsqueda. En una primera aproximación, se repasan los casos reales y las aplicaciones más conocidas para los problemas de búsqueda para después hacer un recorrido por las diferentes técnicas empleadas hasta nuestros días.

El capítulo 3, Técnicas, sirve de introducción teórica a las técnicas de optimización que se van a emplear para resolver el problema. Se explican los conceptos más importantes que ayuden a comprender el funcionamiento de los GAs y de los EDAs. Además, presenta las Técnicas Mixtas que hemos desarrollado, con el objeto de combinar las características de las otras técnicas.

En el capítulo 4, Desarrollo, se detalla la formulación matemática del problema de búsqueda en tiempo mínimo, las funciones objetivo que hay que optimizar para resolverlo, la

codificación del problema de búsqueda en tiempo mínimo y las características de los algoritmos de optimización.

En el capítulo 5, Resultados, se recogen y comparan estadísticamente los resultados obtenidos por medios de las diferentes técnicas propuestas sobre diferentes escenarios de búsqueda.

Finalmente en el último capítulo se extraen las conclusiones y se proponen algunas líneas de trabajo futuro.

2. Contexto

A continuación analizamos la importancia que han tenido los problemas de búsqueda en base a distintos casos históricos o desde la exposición de los diferentes trabajos desarrollados para resolverlos.

2.1 Casos históricos

El primer caso, muy conocido en España, es el de Palomares, en el año 1966 [Melson, 1967]. Un bombardero procedente de Carolina del Norte cargado con cuatro bombas de hidrógeno colisionó con un avión cisterna durante la maniobra de repostaje, arrojando dichas bombas al mar. Tres de las cuatro bombas fueron encontradas pocas horas después pero una no pudo ser encontrada a pesar de los esfuerzos. Durante la búsqueda, se realizaron fotografías aéreas del accidente, recopilando así los datos necesarios para que un comité de expertos de las fuerzas Navales de EE.UU. buscara solución al problema.

Esta búsqueda se realizó recurriendo a la teoría de búsqueda de Bayes, que consiste en dividir el espacio de búsqueda en una serie de casillas de forma discretizada, en forma de malla. A cada casilla se le asigna una probabilidad de encontrar el objetivo deseado y se van actualizando los valores según vamos buscando. Las probabilidades requieren unos valores iniciales. En este caso, esos valores se asignaron en base a la declaración de un pescador local que presenció el accidente. Se tardaron 80 días en encontrar la última bomba.

Otro accidente en el que se utilizaron mapas de probabilidad es el caso de USS Scorpion [ISBA, 2009], un submarino nuclear que se hundió el 22 de Mayo de 1968 y fue encontrado el 28 del mismo mes. En este caso el mapa de probabilidades se creó utilizando datos acústicos y teniendo en cuenta la probabilidad de cada uno de los fallos que se pudieran haber dado en el submarino. La importancia de este caso radica en que el USS Scorpion se encontró a unos 200 metros de la casilla en que la probabilidad era mayor.

El método seguido en la búsqueda del USS Scorpion se describe a continuación:

1. Se adquirieron con sónar los datos que formarán el mapa de probabilidad inicial. En la Figura 2.1 [Richardson et al. 1971] se observan las señales del sónar, así como la localización real del USS Scorpion y la casilla cuya probabilidad era más elevada.
2. Utilizando métodos de Montecarlo, se construyó el mapa de probabilidad, teniendo en cuenta la incertidumbre de la localización del accidente y el tipo de movimiento seguido por el submarino.
3. Se definió una probabilidad de detección, ya que los detectores usados no eran ideales y porque debido a las dimensiones de las casillas empleadas, 1×0.84 millas, pasar por una casilla no tiene porqué significar que se encuentre el submarino.
4. Se realizaron diferentes planes de búsqueda en base a los mapas de probabilidad y se eligió aquel en que la probabilidad total era mayor. Se calculó la probabilidad de éxito de un plan de búsqueda propuesto.
5. A medida que se ejecutaba el plan elegido, se actualizaba continuamente el mapa de probabilidad con las celdas ya exploradas.

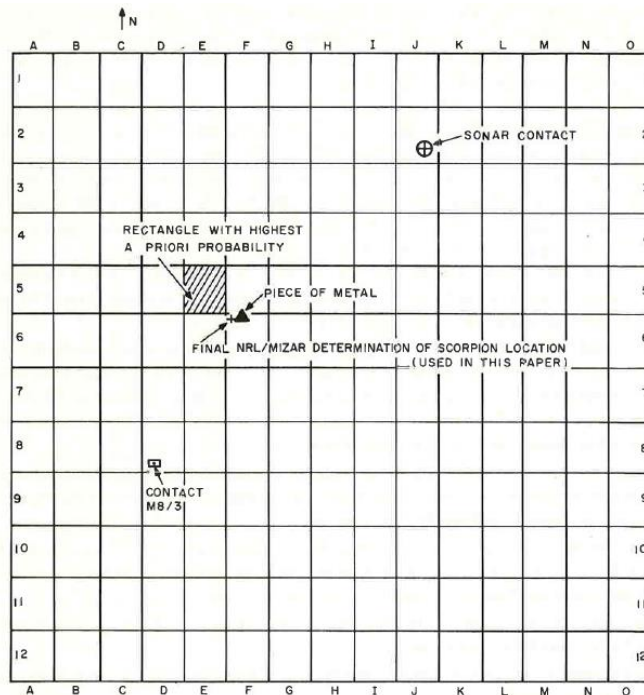


Figura 2. 1 Cuadrícula empleada en la búsqueda del USS Scorpion

Este método es muy parecido al que se seguirá en las técnicas de búsqueda de tiempo mínimo en este TFM que se describen más adelante.

2.2 Estado del Arte

El primer estudio sobre métodos de búsqueda que cabe destacar es el de Stone [Stone, 1975], que recoge los métodos arriba mencionados y otros empleados por la marina estadounidense en otros problemas de búsqueda. Estos estudios, limitados en parte por los ordenadores de la época, tratan sobre todo problemas relacionados con el tiempo que se debe dedicar a cada región de búsqueda.

Más adelante, Eagle [Eagle, 1984] observó que si se introducían restricciones a las trayectorias a los métodos propuestos por Stone, éstos no funcionaban y decidió emplear algoritmos de resolución de Procesos de Decisión Parcialmente Observables de Markov (Partially Observable Markov Decision Process, POMDP) para resolver aquellos problemas de búsqueda en que había que restringir el camino por alguna razón. Eligió este tipo de técnicas debido a que son aplicables para resolver procesos en los que se desconoce un estado (locación del objetivo) que es observado parcialmente por los sensores del agente. Sin embargo, y debido a la complejidad de dichos problemas, relacionada con el número de casillas principalmente, y a la, todavía, baja capacidad de los computadores, sólo se resolvieron problemas pequeños: en escenarios con pocas casillas y con pocas restricciones.

Simultáneamente, algunos autores analizaron con éxito la complejidad del problema de búsqueda de un objetivo mediante un agente con dinámica restringida [Trummel et al., 1986] y demostraron que tenía la complejidad de un problema NP-duro o NP-completo.

Más tarde, Eagle, Yee y Washburn [Eagle et al, 1990; Washburn, 1998] limitaron el número de estados posibles, redujeron el tiempo de computación y convirtieron el problema POMDP en una optimización determinista donde sólo hay un tipo de observación: la no detección. Para ello, establecen que si el problema es encontrar un objetivo perdido, entonces dicha tarea termina una vez que el objetivo es encontrado y se puede asumir que el objetivo no es detectado durante todo el plan de decisión.

El desarrollo de los vehículos no tripulados, la reducción de costes y la miniaturización de sensores [Ross, 2011] ha dado un nuevo impulso al desarrollo de múltiples algoritmos para diferentes problemas de búsqueda [Yang et al., 2002; Blum et al., 2003; Bourgault et al, 2004; Hsu et al, 2008; Mathews et al., 2008; Sarmiento et al., 2009; Lanillos et al., 2012; Lanillos et. al, 2013]. Sin embargo, ninguno de estos trabajos resuelve el problema de forma óptima, y en la mayoría de estos casos [Yang et al., 2002; Blum et al., 2003; Bourgault et al, 2004; Hsu et al, 2008; Mathews et al., 2008] únicamente se maximiza la probabilidad de detección, y por lo tanto, no se minimiza necesariamente el tiempo de búsqueda.

Entre los trabajos en los que se optimiza el tiempo de búsqueda, es importante resaltar la Tesis de Lanillos [Lanillos, 2013] y el artículo [Lanillos et al., 2013]. En la primera se proponen dos funciones objetivo para formular el problema de optimización del tiempo de búsqueda de un objetivo por varios agentes y se utilizan varios métodos de optimización (entropía cruzada, programación con restricciones y gradiente descendiente) para resolverlo. En el segundo se realiza una comparativa estadística del funcionamiento de los Algoritmos de Optimización Bayesiana (Bayesian Optimization Algorithm, BOA) sobre diferentes escenarios. Ambos se encuentran directamente relacionados con este TFM, ya que en él se utilizan las funciones objetivo propuestas en la Tesis para formular el problema de la búsqueda y la metodología de la comparativa estadística del artículo para realizar una comparativa de un conjunto de algoritmos más amplio (Algoritmos Genéticos, Algoritmos de Estimación de la Distribución y Algoritmos Mixtos).

3. Técnicas

A continuación se describen brevemente las técnicas que se emplean en este trabajo para encontrar una solución a nuestro problema de búsqueda: los métodos de optimización estándar denominados Algoritmos Genéticos y Algoritmos de Estimación de Distribuciones, y dos nuevos métodos que combinan las técnicas anteriores, denominados Algoritmos Mixtos.

3.1 Algoritmos Genéticos

Los Algoritmos Genéticos (Genetic Algorithms, GAs) [Mitchell, 1996; Goldberg, 1989; Vose, 1999] son una de las técnicas más conocidas para resolver problemas de búsqueda y optimización. Se basan en el proceso genético de los organismos vivos. A lo largo de distintas generaciones, una población de soluciones potenciales evoluciona según la selección natural, es decir, favoreciendo la supervivencia de los más aptos. En base a esto, los GAs crean soluciones cada vez mejores, aunque su capacidad para obtener buenas soluciones depende en gran medida de la codificación empleada.

Los GAs emplean una población de individuos, cada uno de los cuales representa una solución de un problema. A cada individuo se le asigna un valor relacionado con su bondad. En la naturaleza este valor reflejaría cuán efectivo es un organismo para sobrevivir. Cuanto mejor sea este valor, mayor será la probabilidad de que el individuo sea seleccionado para reproducirse, cruzando su información con otro individuo seleccionado de la misma forma. Este cruce de información dará lugar a nuevos individuos, descendientes de otros y que por tanto compartirán características con los primeros. Los individuos peor preparados, tienen menos probabilidades de reproducirse y por tanto su información se propagará menos que la de los mejor preparados.

De esta manera, se van obteniendo nuevas poblaciones de soluciones, que reemplazan a las anteriores y que contienen una mayor proporción de características buenas para nuestro problema. Estas buenas características se propagan a lo largo de las generaciones, y al cruzar los individuos bien preparados, exploramos las áreas más prometedoras del espacio de búsqueda. Si nuestro GA está bien diseñado, la población convergerá hacia una solución óptima.

La importancia de los GAs se debe al hecho de ser una técnica robusta y flexible, que puede tratar una gran variedad de problemas, incluso algunos en que otros métodos no tienen éxito. El GA habitualmente no encuentra la solución óptima del problema, pero empíricamente se puede comprobar que suelen aportar soluciones de un nivel aceptable en tiempos relativamente bajos comparados con otras técnicas, si bien es verdad que técnicas especializadas para ciertos problemas pueden dar soluciones mejores en tiempos menores. Por tanto, los GAs parecen una buena solución para problemas para los que no existan técnicas especializadas o como una posible mejora de dichas técnicas.

Las soluciones del problema (individuos) se representan como un conjunto de parámetros (genes), los cuales se agrupan como una serie de valores (cromosomas). Al conjunto de parámetros representando en un cromosoma se llama fenotipo, que contiene la información necesaria para construir un organismo, denominado genotipo. La adaptación de un individuo al problema depende de la evaluación del genotipo.

Esta adaptación se evalúa por medio de la función de utilidad, también denominada función de mérito. Ésta asigna a cada cromosoma particular un número, que refleja el nivel de adaptación al problema de dicho cromosoma.

Durante la fase de reproducción, se seleccionan los individuos de la población que se usarán para generar la siguiente generación. Esta selección de padres se efectúa al azar de manera que se favorezca a los individuos mejor preparados, asignando una probabilidad de ser seleccionado en función de la función de utilidad. De esta forma, individuos mejor preparados se elegirán más a menudo que aquellos peor preparados.

Una vez tenemos seleccionados los padres, se combinan sus cromosomas por medio de operadores de mutación y cruce.

Con el operador de cruce, se cortan los cromosomas de los padres en bloques y se intercambian, por ejemplo, como se muestra en la figura 3.1, produciéndose cromosomas completos que heredan genes de los padres. Hay diferentes tipos de cruce, distinguibles por el número de puntos de corte empleados.

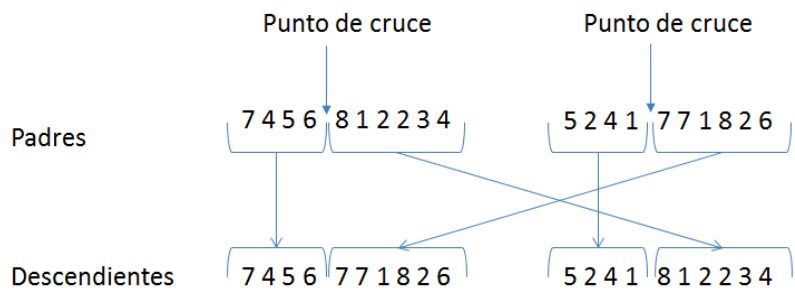


Figura 3. 1 Operador de cruce de 1 punto

El operado de mutación se aplica a cada hijo de forma individual y simplemente consiste en la alteración aleatoria, con probabilidad pequeña, de cada gen. En la figura 3.2 se observa la mutación del quinto gen. Este operador favorece que regiones del espacio de búsqueda que podrían quedar inexploradas tengan alguna probabilidad de ser alcanzadas.

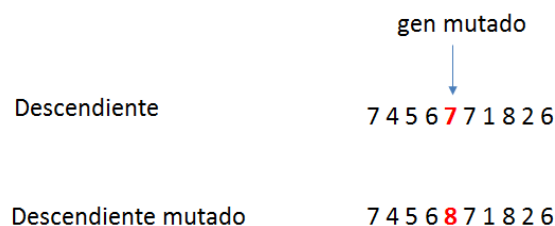


Figura 3. 2 Operador de mutación

Finalmente, es habitual elegir, entre los individuos de la generación anterior y los recién creados, un número de individuos fijo para que las poblaciones sucesivas tengan un tamaño constante.

Las etapas habituales de un GA y el orden en el que éstas se realizan se presenta en la Figura 3.3.

- Inicializar la población
- For i=0 hasta G generaciones
 - Evaluación individuos
 - Selección de Padres
 - Cruce
 - Mutación
 - Crear nueva población tamaño deseado
- Mejor individuo

Figura 3. 3 Pseudocódigo de GA

Si el GA ha sido correctamente implementado, la población evoluciona a lo largo de las generaciones de forma que la media de la población y el valor del mejor individuo mejoran en base a la función de utilidad. Al final, cada gen converge hacia un valor óptimo, haciendo que finalmente el cromosoma converja hacia una solución óptima del problema.

En el trabajo, se analizarán dos versiones para el GA, una empleando un solo punto de cruce y otra empleando dos puntos de corte. La selección de los padres en ambos casos se realizará del mismo modo, por medio de método del torneo.

3.2 Algoritmos de Estimación de Distribuciones

Al igual que los GAs, los Algoritmos de Estimación de la Distribución (Estimation Distribution Algorithms, EDAs) [Larrañaga et al., 2002; Pelikan, 2005] están basados en poblaciones que evolucionan. Sin embargo, se diferencian de los GAs, en que en los EDAs la evolución de las poblaciones no se lleva a cabo por medio de los operadores de cruce y mutación. En lugar de ello, la nueva población de individuos se muestrea a partir de un modelo gráfico probabilístico. Este modelo representa una factorización de la distribución de probabilidad estimada a partir del conjunto de individuos seleccionados de entre los que constituyen la generación anterior. A través de esta distribución de probabilidad, los EDAs tienen en cuenta las interrelaciones que existen entre las variables representadas por los individuos seleccionados en cada generación.

De nuevo, debemos generar una población inicial, por ejemplo aleatoriamente, que evaluaremos con una función de utilidad. En base a esta función de utilidad, seleccionamos un subconjunto de individuos formado habitualmente por los individuos mejor preparados.

A continuación, se debe aprender el modelo probabilístico asociado al subconjunto seleccionado. Para ello, se utiliza un modelo probabilístico que captura la distribución de los valores y las interdependencias existentes entre las variables, es decir, entre los genes. Más adelante se profundizará en cómo se aprenden estas dependencias.

Posteriormente se muestrean los nuevos individuos en base a la distribución de probabilidad aprendida anteriormente. De esta forma las nuevas soluciones siguen una distribución probabilística similar a las mejores anteriormente encontradas. Este proceso se repite hasta que al algoritmo finaliza, bien por converger a una población fija o por realizar un número de iteraciones predeterminado.

Las etapas habituales de un EDA y el orden en el que éstas se realizan se presenta en la Figura 3.4.

- Inicializar la población
- For $i=0$ hasta G generaciones
 - Evaluación individuos
 - Selección mejores individuos
 - Aprender distribución de probabilidad $p(x)$
 - Crear individuos nuevos en base a $p(x)$
 - Crear nueva población tamaño deseado
- Mejor individuo

Figura 3. 4 Pseudocódigo EDA

Uno de los modelos probabilísticos para variables discretas usados en EDAs es la Red Bayesiana (Bayesian Network, BN), ya que es un modelo gráfico capaz de codificar dependencias de una serie de variables que toman valores finitos. Entre los EDAs que utilizan BN, nos encontramos con los Algoritmos de Optimización Bayesiana (Bayesian Optimization Algorithms, BOAs) [Pelikan et al., 1999]. A continuación explicamos brevemente que es una BN y cómo funcionan los BOAs que se analizarán en este TFM.

3.2.1. Redes Bayesianas (BN)

La red bayesiana representa la probabilidad conjunta de las variables (genes) que constituyen el cromosoma. La BN tiene dos partes, un grupo de afirmaciones de independencia condicional y un grupo de distribuciones condicionales locales, que nos permiten construir la probabilidad conjunta global de la distribución de cromosomas a partir de las distribuciones locales de genes.

El conjunto de afirmaciones se expresa por medio de un gráfico de dependencia, donde cada gen corresponde con un nodo del gráfico. Si la probabilidad del valor de un cierto gen X_1 se ve afectado por el valor de otro gen X_2 , decimos que “ X_1 depende de X_2 ” o que “ X_2 es una variable padre de X_1 ”. Esta afirmación se expresa por la existencia de un arco en el gráfico de dependencia. Al conjunto de todas las variables padre de un nodo corresponde al conjunto de todos los nodos cuyos arcos van a parar a dicho nodo.

Las distribuciones condicionales locales representan la probabilidad de cada una de las variables condicionadas a todas las combinaciones de valores de las variables padre. Estas probabilidades se almacenan en una tabla multidimensional y, junto con el modelo gráfico, permite obtener la probabilidad conjunta de todas las combinaciones de los valores de las variables como el producto de las probabilidades condicionales locales. Además, las distribuciones locales y la estructura de la red también permiten generar fácilmente muestras de la probabilidad conjunta.

La Figura 3.5 muestra un ejemplo de BN con tres variables, en la que los genes X_0 y X_2 son independientes y el valor de X_1 depende de X_0 y X_2 . Bajo esta afirmación podemos escribir la probabilidad del cromosoma $X = (X_0, X_1, X_2)$ como el producto de las distribuciones locales: $p(X_0, X_1, X_2) = p(X_0) \cdot p(X_2) \cdot p(X_1|X_0, X_2)$.

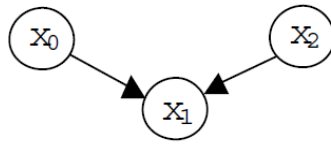


Figura 3. 5 Ejemplo de gráfico de dependencia

Suponiendo además que los genes del ejemplo toman valores binarios, para los genes X_0 y X_2 la distribución local es no condicionada y se representa con los valores $p(X_0 = 0)$, $p(X_0 = 1)$, $p(X_2 = 0)$ y $p(X_2 = 1)$. El caso de X_1 , al depender de X_0 y X_2 , requiere de una tabla más compleja:

X_0X_2	00	01	10	11
$p(X_1 = 0 X_0, X_2)$
$p(X_1 = 1 X_0, X_2)$

3.2.2. Algoritmos de Optimización Bayesiana (BOA)

Los BOA son EDAs en los que el modelo probabilístico que se aprende y se muestrea es una BN con variables discretas. La forma de proceder, dentro del bucle del EDA (ver Figura 3.3), se esquematiza en la Figura 3.6: se parte de una población de individuos evaluada de la que se eligen los mejores, a continuación se utiliza un método de aprendizaje de BN para obtener la estructura y las tabla de probabilidad de las distribuciones locales y, finalmente, se generan un conjunto de nuevas soluciones muestreando el modelo almacenado en la BN.

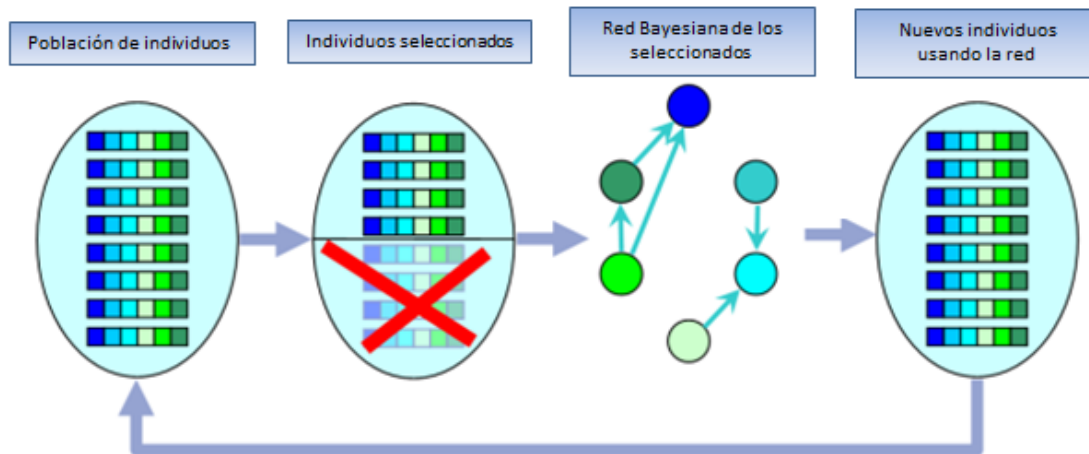


Figura 3. 6 Representación BOA

El proceso de aprendizaje de la BN a partir de las soluciones elegidas es en si mismo un problema de optimización en el que se define una función de utilidad que da una medida de lo bien que se ajusta la estructura y las tablas de probabilidad correspondiente a los datos (que en el caso de los BOAs son los individuos de la población). Es decir, un BOA es un algoritmo de optimización para problemas con variables discretas que explora el espacio de soluciones del problema en cuestión resolviendo otro problema de optimización: el aprendizaje de la red

bayesiana a partir de la información recogida en las mejores soluciones encontradas en la última iteración de BOA.

Existen diferentes funciones de utilidad que permiten aprender la estructura de la BN a partir de la información almacenada en las mejores soluciones de la población [Neapolitan, 2003]. Entre ellas cabe destacar el Criterio de Información Bayesiana (BIC) y la función Bayesiana de Dirichlet uniforme (BDeu). El primero, BIC, es una función que penaliza las BN complejas y maximiza la similitud entre los datos y la BN. El segundo, BDeu, es una función bayesiana que maximiza la probabilidad a posteriori de red para unos datos asumiendo que los valores de las probabilidades locales condicionadas siguen una distribución de Dirichlet con priors uniformes.

Sus propiedades, desde el punto de vista del ajuste de una red a unos datos y priors dados, son diferentes, pero para ajustar la red de un BOA, cuyo objeto es únicamente muestrear el espacio de soluciones de un problema de optimización, no tienen por qué ser, a priori, necesariamente relevantes. Sin embargo, la complejidad computacional asociada al cálculo de BIC y BDeu es significativamente diferente (BIC es más sencillo de evaluar que BDeu), y este hecho sí puede ser determinante a la hora de optimizar el problema original: cuanto más tiempo se necesite para obtener la BN más tiempo se necesitará para llevar a cabo cada bucle de BOA.

Por lo tanto, se ha decidido probar el funcionamiento de dos BOAs diferentes, uno con BDeu y otro con BIC, para ver si alguno resulta ser más ventajoso a la hora de resolver el problema de búsqueda en tiempo mínimo.

Adicionalmente, como un cuello de botella del BOA es el aprendizaje de la estructura de la BN, se comprobará el funcionamiento de la técnica bajo la suposición de que el problema puede resolverse con la red bayesiana más sencilla: aquella en la que no existen conexiones entre los nodos de la red. Es decir, suponiendo que todas las variables son independientes y que, por lo tanto, se pueden muestrear sus valores teniendo en cuenta de forma independiente los valores de cada una de las variables en el subconjunto de muestras mejores.

3.3 Algoritmos Mixtos

Tras estudiar el funcionamiento de las técnicas previamente expuestas (GAs y BOAs), se decidió desarrollar algún algoritmo adicional que aproveche las propiedades de ambos. Aunque la combinación de ambas técnicas ya ha sido realizada en otros trabajos [Thierens, 2011], la propuesta recogida en este TFM sigue una forma de proceder diferente.

En concreto, los algoritmos desarrollados aprenden una BN como el BOA para muestrear las soluciones y realizan una variación de las soluciones muestreadas utilizando elementos propios de los GAs. La forma de proceder se esquematiza en la Figura 3.7, en la que se representa un BOA con una etapa adicional (la segunda) en la que se elige un padre para modificar el valor de los individuos en la población.

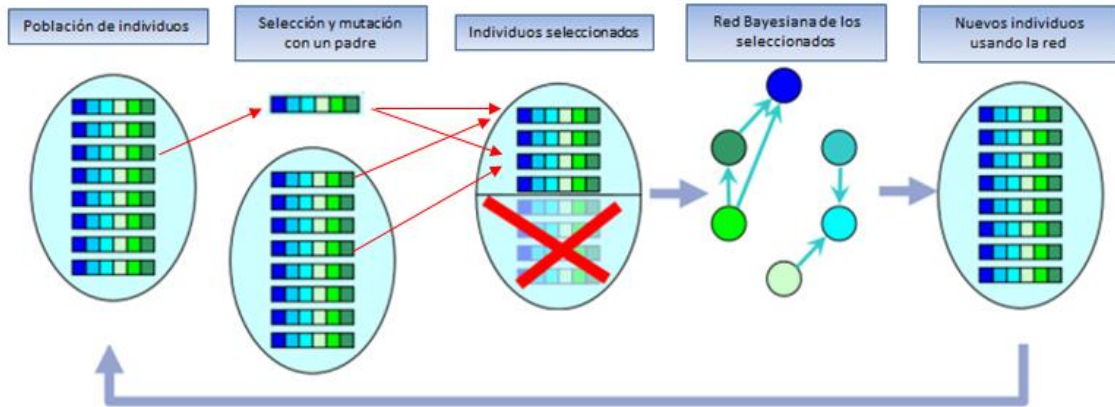


Figura 3. 7 Representación Mixto

La variación introducida en los Algoritmos Mixtos consiste, tal y como se esquematiza en la Figura 3.8, en mutar los valores de unos pocos genes de la población muestreada con la información proporcionada por un individuo padre.

	genes mutados
	↓ ↓
Individuo	7 4 5 6 7 7 1 8 2 6
Padre	6 5 3 8 5 1 1 6 3 4
Individuo mutado	7 4 5 6 5 7 1 8 3 6

Figura 3. 8 Variación Mixta

Los dos algoritmos propuestos son muy similares y difieren en la forma en que eligen el individuo-padre con el que realizan la mutación de las muestras. En un caso, el que hemos denominado ROM, Recombinative Optimal Mixing, se emplea para cada iteración el mismo padre para realizar variaciones con los individuos de la población. En el denominado GOM, Gene-pool Optimal Mixing, se dispone de una piscina genética y en cada iteración del algoritmo se selecciona un nuevo padre para cada individuo con el que se vaya a realizar una mutación. Los individuos padres de ambos métodos se seleccionan de entre los mejores. La forma completa de proceder en cada caso se detalla en las Figuras 3.9 y 3.10 para ROM y GOM respectivamente.

- Inicializar la población
- For i=0 hasta G generaciones
 - Evaluación individuos
 - Selección mejores individuos
 - Selección de Padre
 - Mutación con el Padre
 - Aprender distribución de probabilidad $p(x)$
 - Crear individuos nuevos en base a $p(x)$
 - Crear nueva población tamaño deseado
- Mejor individuo

Figura 3. 9 Pseudocódigo ROM

- Inicializar la población
- For i=0 hasta G generaciones
 - Evaluación individuos
 - Selección mejores individuos
 - For j=0 hasta I individuos
 - Selección de Padre
 - Mutación con el Padre
 - Aprender distribución de probabilidad $p(x)$
 - Crear individuos nuevos en base a $p(x)$
 - Crear nueva población tamaño deseado
- Mejor individuo

Figura 3. 10 Pseudocódigo GOM

4. Desarrollo

En la primera parte de este capítulo se introduce la formulación matemática del problema de búsqueda en tiempo mínimo (Minimum Time Search, MTS), mientras que en la segunda se presenta la codificación de las soluciones y se describe la implementación de los algoritmos que se van a analizar.

4.1 Formulación matemática

El problema de búsqueda en tiempo mínimo en entornos dinámicos inciertos es un proceso de toma de decisiones que involucra dos partes: un grupo de buscadores (agentes con sensores) con dinámica restringida y un objetivo con punto de partida y dinámica desconocidos. El propósito es determinar el mejor conjunto de acciones que hay que realizar sobre el agente para que se mueva siguiendo una trayectoria que simultáneamente 1) maximice la probabilidad de encontrar el objetivo y 2) minimice el tiempo empleado para encontrarlo.

La incertidumbre del problema se modela con una creencia (probabilidad) inicial sobre la localización del objetivo, un modelo probabilístico sobre el movimiento del objetivo y un modelo probabilístico de los sensores de los agentes.

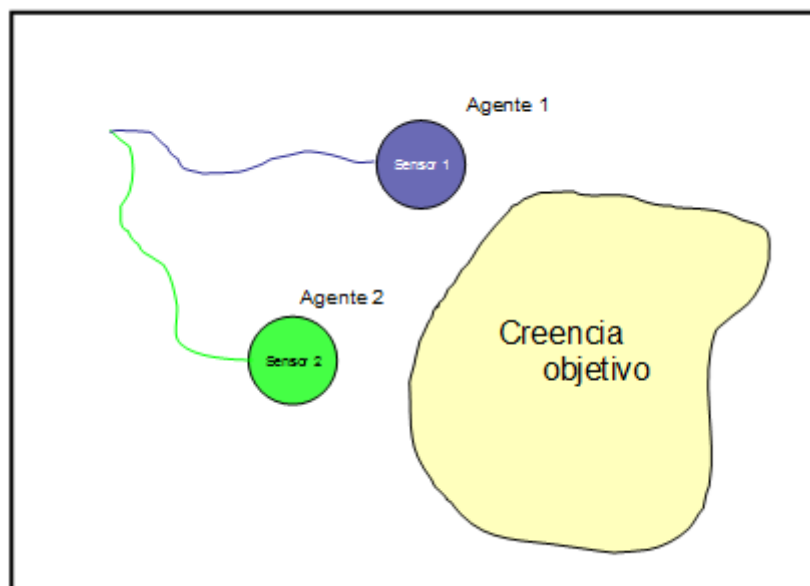


Figura 4. 1 Esquema del problema de búsqueda en tiempo mínima

El objetivo y los agentes se encuentran situados en una cuadrícula discretizada de dos dimensiones de tamaño $\omega_x * \omega_y$. A continuación definiremos matemáticamente los modelos de incertidumbre y las funciones objetivo, siguiendo la terminología empleada en la Tesis de Lanillos (Lanillos 2013) y el artículo donde se recoge el análisis del funcionamiento de un BOA que utiliza el BDeu para aprender la estructura de la BN (Lanillos et. al, 2013).

4.1.1 Variables del problema

Los agentes son M plataformas móviles con sensores que realizan observaciones en los diferentes instantes k . La posición s_i^k de cada agente i está relacionada de forma determinista con la posición anterior del agente s_i^{k-1} por medio de la acción de control u_i^k . Por lo tanto, la secuencia de N acciones de control $u_i^{k+1:k+N} = \{u_i^{k+1}, u_i^{k+2}, \dots, u_i^{k+N}\}$ determina totalmente la trayectoria del agente $s_i^{k:k+N} = \{s_i^k, s_i^{k+1}, \dots, s_i^{k+N}\}$. Las acciones están definidas como las 8 direcciones cardinales que hacen que un agente se mueva de una celda a sus adyacentes y las posiciones como una de las celdas del mapa.

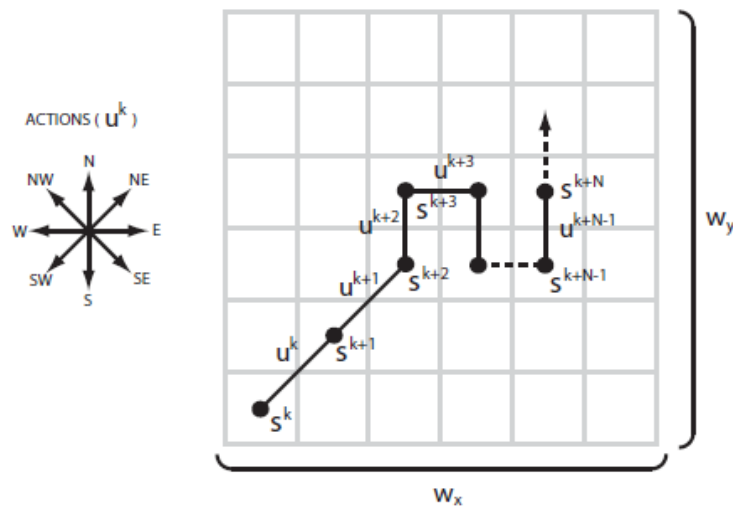


Figura 4. 2 Relación entre las posiciones y acciones de los agentes

El objetivo es el objeto que buscamos. Su posición τ^k en cada instante k y su dinámica son desconocidas, aunque están modeladas por la distribución de probabilidad inicial o creencia $b_\tau^0 = P(\tau^0)$ y el modelo dinámico $P(\tau^k | \tau^{k-1})$, que representa la probabilidad de que el objetivo se mueva a la posición τ^k desde la posición τ^{k-1} .

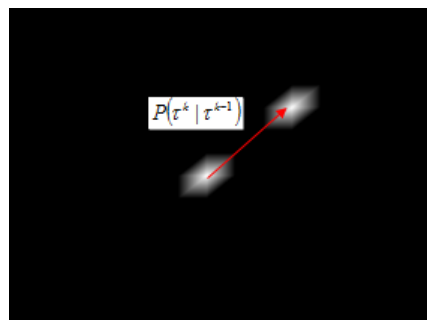


Figura 4. 3 Probabilidad de que el objetivo se desplace

Los sensores de los agentes permiten observar y recolectar información del objetivo. Las observaciones z_i^k realizadas por cada agente i en el instante de tiempo k son usadas por los agentes para actualizar la creencia sobre la localización del objetivo y elegir la mejor secuencia

de acciones. La incertidumbre del sensor es modelada con la función $P(z_i^k | \tau^k, s_i^k)$ que representa la probabilidad de obtener la medida z_i^k cuando el objetivo y el agente se encuentran respectivamente en τ^k y s_i^k . Además, como sólo consideramos dos posibles observaciones, detección y no detección del objetivo (D y \bar{D}), las probabilidades se relacionan como $P(z_i^k = D | \tau^k, s_i^k) = 1 - P(z_i^k = \bar{D} | \tau^k, s_i^k)$. El modelo del sensor más sencillo, que se usa en este TFM durante los experimentos, es el del sensor ideal en el que sólo se detecta el objetivo cuando el agente y el objetivo se encuentran en la misma celda.

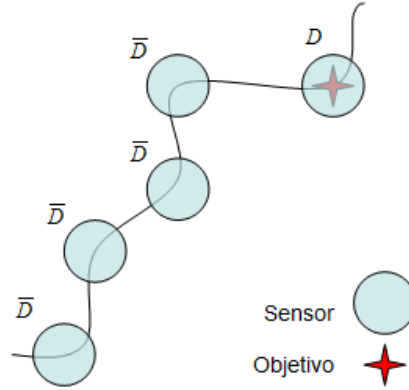


Figura 4. 4 Esquema del funcionamiento del modelo del sensor ideal

La solución al problema de búsqueda en tiempo mínimo es el conjunto de secuencia de acciones $u_{1:M}^{k+1:k+N} = \{u_1^{k+1:k+N}, u_2^{k+1:k+N}, \dots, u_M^{k+1:k+N}\}$ o de trayectorias de los agentes $s_{1:M}^{k+1:k+N} = \{s_1^{k+1:k+N}, s_2^{k+1:k+N}, \dots, s_M^{k+1:k+N}\}$ que minimizan el tiempo necesario para encontrar el objetivo, dados la creencia sobre la localización del objetivo $b_\tau^k = P(\tau^k)$ en el instante k , el modelo dinámico del objetivo $P(\tau^k | \tau^{k-1})$ y los modelos probabilísticos de los sensores $P(z_i^k | \tau^k, s_i^k)$. Finalmente y por simplicidad, en las expresiones usaremos D_i^k y \bar{D}_i^k para representar $z_i^k = D$ y $z_i^k = \bar{D}$.

4.1.2 Funciones de mérito

Los problemas de búsqueda pueden plantearse como un problema de optimización de una función de mérito relacionada con las posiciones iniciales del agente $s_{1:M}^k$, la creencia inicial $b_\tau^k = P(\tau^k)$, y los modelos probabilísticos $P(\tau^k | \tau^{k-1})$ y $P(z_i^k | \tau^k, s_i^k)$. Para el caso de la búsqueda en tiempo mínimo, en la Tesis de Lanillos (Lanillos 2013) se proponen dos funciones de mérito diferentes: el tiempo esperado truncado (Truncated Expected Time, TET) y la probabilidad de detección descontada (Discounted Detection, DD). La primera, TET, calcula el tiempo esperado o medio necesario para detectar el objetivo durante alguna de las medidas realizadas por los sensores en las siguientes N posiciones de los agentes. La segunda, DD, modifica la definición de la probabilidad de detección del objetivo en cualquiera de los N instantes futuros para dar un mayor peso a las detecciones realizadas en los instantes iniciales de la trayectoria. Por lo tanto, realizaremos el análisis del funcionamiento de las heurísticas de optimización utilizando ambas funciones, ya que las dos se encuentran relacionadas (directa o

indirectamente) con el tiempo de búsqueda y las dos sirven para determinar la mejor secuencia de acciones para un escenario dado por $s_{1:M}^k$, $b_\tau^k = P(\tau^k)$, $P(\tau^k | \tau^{k-1})$ y $P(z_i^k | \tau^k, s_i^k)$.

A continuación recogemos sus expresiones matemáticas y las de una probabilidad auxiliar de las que ambas dependen. Su derivación, se detalla en [Lanillos et. al, 2013; Lanillos 2013].

4.1.2.1. Función de probabilidad auxiliar

$f_{\tau, s_{1:M}}^{k+j+1} = P(\tau^{k+j+1}, \bar{D}_{1:M}^{k+1:k+j} | s_{1:M}^{1:k+N}, z_{1:M}^{1:k})$ es una función de probabilidad que aparece de forma natural en las funciones de mérito propuestas y representa la probabilidad de que el objetivo esté en el instante $k+j+1$ en la posición τ^{k+j+1} sin haberlo detectado desde el instante $k+1$ hasta $k+j$, dada las trayectorias de todos los agentes $s_{1:M}^{1:k+N}$ y las medidas pasadas $z_{1:M}^{1:k}$. Sus valores pueden ser calculados de forma iterativa con la ecuación (1) desde $j=1$ hasta $j=N-1$, suponiendo $f_{\tau, s_{1:M}}^{k+0} = b_\tau^k$.

$$f_{\tau, s_{1:M}}^{k+j+1} = \sum_{\tau^{k+j} \in \Omega} P(\tau^{k+j+1} | \tau^{k+j}) \prod_{i=1}^M P(\bar{D}^{k+j} | \tau^{k+j}, s^{k+j}) f_{\tau, s_{1:M}}^{k+j} \quad (1)$$

4.1.2.2. Tiempo esperado truncado

Debido a la naturaleza probabilística del problema, el tiempo de búsqueda puede ser optimizado *minimizando* el Tiempo Esperado Truncado (TET) para buscar el objetivo dadas las trayectorias de los agentes. Se puede calcular con la siguiente expresión, cuyo paso intermedio indica que el *TET* es el sumatorio de las probabilidades de no detección del objetivo hasta cada instante de tiempo $k+j$.

$$TET(s_{1:M}^{k:k+N}) = \sum_{j=1}^N P(\bar{D}_{1:N}^{k+1:k+j} | s_{1:M}^{1:k+j}, z_{1:M}^{1:k}) = \sum_{j=1}^N \sum_{\tau^{k+j} \in \Omega} \prod_{i=1}^M P(\bar{D}^{k+j} | \tau^{k+j}, s^{k+j}) f_{\tau, s_{1:M}}^{k+j} \quad (2)$$

4.1.2.3. Detección descontada

Otra manera de formular el problema de búsqueda consiste en *maximizar* la probabilidad de detectar el objetivo en alguno de los siguientes $j=1:N$ intervalos de tiempo:

$$P\left(\bigcup_{i=1:M, j=1:N} D_i^{k+j} | s_{1:M}^{1:k+j}, z_{1:M}^{1:k}\right) = \sum_{j=1}^N \sum_{\tau^{k+j} \in \Omega} \left(1 - \prod_{i=1}^M P(\bar{D}^{k+j} | \tau^{k+j}, s^{k+j})\right) f_{\tau, s_{1:M}}^{k+j} \quad (3)$$

Sin embargo, con esta función no se optimiza necesariamente el tiempo. Para optimizar simultáneamente la probabilidad y el tiempo de detección, podemos incluir el tiempo en el sumatorio externo como un factor de descuento λ^j (usando $0 < \lambda < 1$), y definir de esta forma la función de mérito de Detección Descontada (*DD*) de la Ec. (4).

$$DD(s_{1:M}^{k:k+N}) = \sum_{j=1}^N \lambda^j \sum_{\tau^{k+j} \in \Omega} \left(1 - \prod_{i=1}^M P(\bar{D}^{k+j} | \tau^{k+j}, s^{k+j})\right) f_{\tau, s_{1:M}}^{k+j} \quad (4)$$

4.2 Algoritmo de optimización

El problema de búsqueda en tiempo mínimo tiene una complejidad NP o superior. Por tanto, para encontrar una buena solución en un tiempo razonable en escenarios con mapas de probabilidad con muchas celdas, el problema debe ser abordado con algoritmos de optimización aproximados o con heurísticas. Además, debido a las restricciones impuestas por los posibles movimientos de los agentes, es más sencillo buscar listados de acciones que sus correspondientes trayectorias, ya que el espacio de acciones (con 8 posibles valores por acción) es más pequeño que el espacio de posiciones (con $\omega_x * \omega_y$ posibles valores por instante de tiempo) y un listado de acciones crea automáticamente trayectorias realizables, sobre las que optimizar el TET o DD. A continuación detallamos la codificación del problema y las características más relevantes de los algoritmos que se analizarán en el próximo capítulo.

4.2.1 Codificación

Como primer paso, debemos definir nuestro espacio de búsqueda. Para ello, dividiremos la región en dónde queremos desarrollar la búsqueda de nuestro objetivo en una serie de casillas y a cada una de ellas le asignaremos un valor entre 0-1 en función de la probabilidad que tenemos de encontrar en esa casilla el objetivo. En la Figura 4.5 se muestra un ejemplo de un mapa de creencia de 20x20 celdas, siendo las casillas más claras aquellas con mayor probabilidad y las casillas más oscuras aquellas con menor probabilidad.

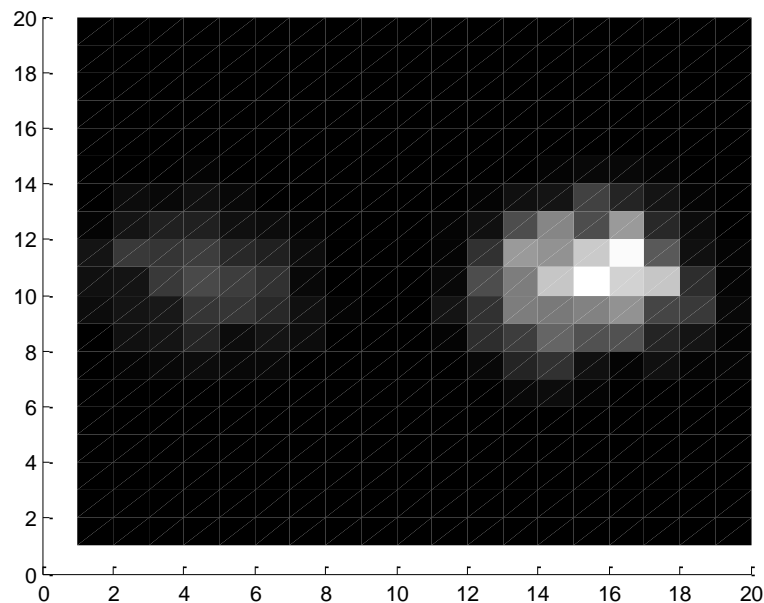


Figura 4. 5 Mapa de creencia

Teniendo en cuenta esta forma de discretizar el espacio de búsqueda, debemos definir la manera en que se desplazarán los agentes por éste. Tal y como acabamos de justificar, codificaremos cada individuo de la población inicial, que será generada de forma aleatoria, como la sucesión de acciones que debe realizar durante el número de pasos N sobre el que se calcula TET o DD. Por la relación directa que existe entre las acciones y las posiciones de los

agentes, la lista de acciones definirá el camino que cada agente recorre en su búsqueda del objetivo.

Cada acción vendrá codificada por medio de un número entero. Cada valor representará cada una de las posibles direcciones que puede tomar el agente en cada paso. Tomando el recuadro azul como la posición actual del agente, la dirección que éste puede tomar viene codificada con un número del 1 al 8, tal y como se muestra en la figura 4.6.

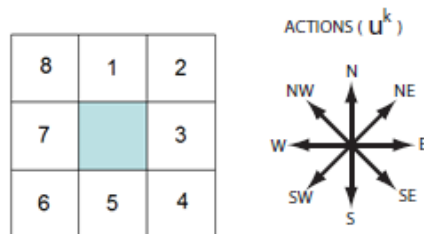


Figura 4. 6 Codificación direcciones de los agentes

En definitiva, cada individuo estará dado por un vector en el que quedará reflejada la dirección que tomó en cada paso. Cada componente del vector será un número aleatorio entre 1 y 8. Además, para poder optimizar la trayectoria de un conjunto de agentes, colocaremos los vectores de cada agente uno a continuación de otro. Este modo de proceder se ilustra en la Figura 4.7 para el caso de dos agentes y trayectorias de N pasos:

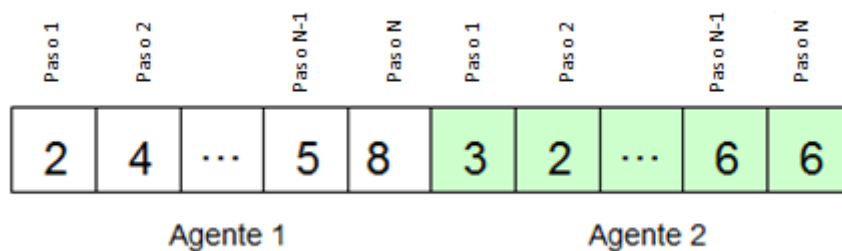


Figura 4. 7 Visualización codificación

4.2.2 Propiedades genéricas de las heurísticas

Una cuestión que se plantea en todas las heurísticas que se van a analizar es la relacionada con el tamaño idóneo de la población: con poblaciones pequeñas los algoritmos corren el riesgo de no cubrir adecuadamente el espacio de búsqueda, mientras que el uso de poblaciones de gran tamaño puede acarrear problemas relacionados con el excesivo costo computacional.

En el artículo (Lanillos et al., 2013) se elige como número de individuos para inicializar la población al producto del número de pasos (N) por el número de agentes (M) por el número de direcciones plausibles (8) por un factor de ajuste (que toma el valor de 10). Así, para un escenario con $N=20$ pasos y $M=2$ agentes se usarán $N \times M \times direcciones \times factor = 20 \times 2 \times 8 \times 10 = 3200$ individuos.

A continuación, en todos los algoritmos se evalúan los individuos generados. Para ello, calcularemos el TET o el DD asociado al camino correspondiente a cada individuo (lista de

acciones). Para calcularlos, pondremos a cero la probabilidad de los puntos ya recorridos, ya que al usar un sensor ideal, al visitar y observar una celda su probabilidad se anula. Si el escenario es estático (el objetivo permanece en la misma posición todo el tiempo), en el cálculo del DD, ilustrado en la Figura 4.8, se acumula, usando un factor de descuento, la probabilidad de las celdas detectadas. En el cálculo de TET para escenarios estáticos, ilustrado en la Figura 4.9, se acumula la probabilidad remanente en el mapa. Finalmente, en los casos dinámicos, antes de realizar cada observación, la probabilidad de cada celda se reparte entre las celdas adyacentes de acuerdo con el modelo de movimiento del objetivo de cada escenario.

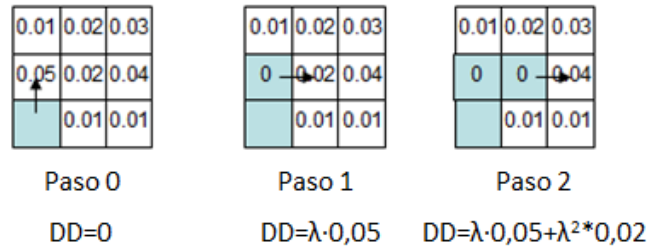


Figura 4. 8 Cálculo del DD para un escenario estático sobre un recorrido de un agente

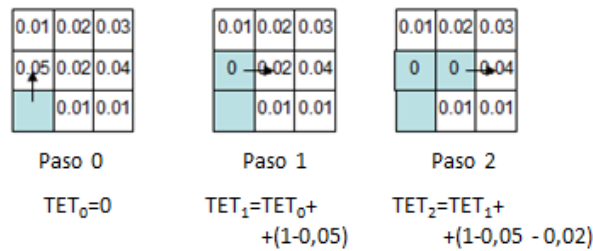


Figura 4. 9 Cálculo del TET para un escenario estático sobre un recorrido de un agente

4.2.3 Algoritmo Genético (GA)

En esta sección se describen los métodos de selección, cruce, mutación y recombinación de poblaciones que se utilizan en el GA implementado en este TFM.

La selección de padres se realiza utilizando el método del torneo de 2 individuos. En este método, esquematizado en la Figura 4.10, se seleccionan al azar dos individuos de la población y de esta pareja se elige, para formar parte de las parejas de padre, al mejor, desechando el peor. Se realiza este método hasta completar el número de parejas necesarias.

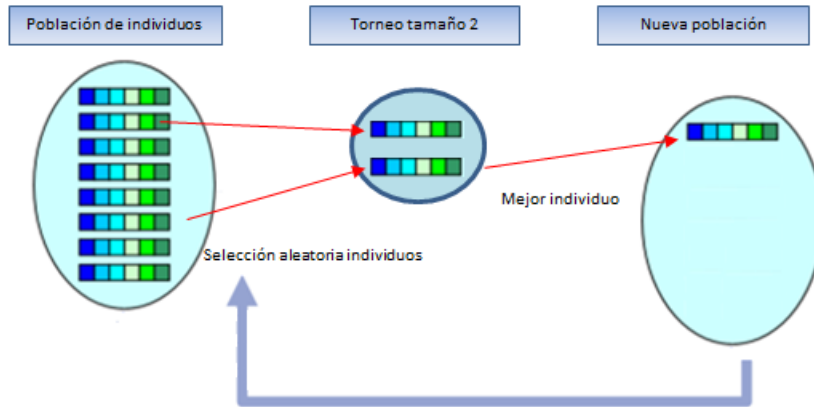


Figura 4. 10 Selección por Torneo

Sobre los individuos seleccionados por el método del torneo haremos actuar el operador de cruce. Hemos implementa dos posibilidades, un cruce con un solo punto de cruce y otro con dos puntos de cruce. La diferencia entre ambos métodos se esquematiza en las Figuras 4.11 y 4.12.

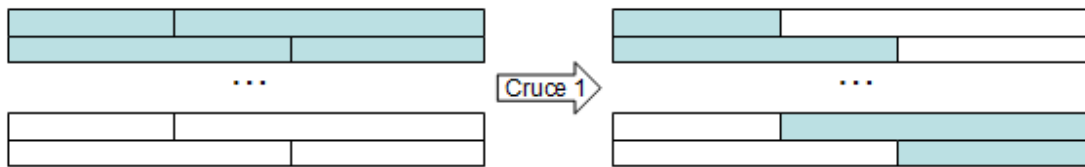


Figura 4. 11 Cruce de 1 punto

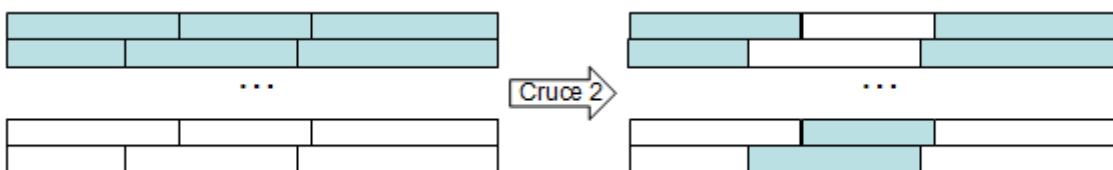


Figura 4. 12 Cruce de 2 puntos

Desde el punto de vista del cruce, es importante destacar que en el caso de tener múltiples agentes, se codifica la solución de un agente después de la de otro. En principio podríamos pensar que deberíamos hacer el corte de manera que el cruce de cada agente fuera independiente. Esto añadiría algo de tiempo de computación, y a priori, no implica que el algoritmo obtenga soluciones mejores. De hecho, el cruce habitual, que no tiene en cuenta la existencia de múltiples listas de acciones en la solución, no hace que se mezclen las acciones de los agentes de forma inapropiada. Por ejemplo, si un corte para dos individuos con $N=10$ une los pasos 8,9 y 10 de un agente con los pasos 1,2 y 3 del segundo, no produce ningún cruce singular en que cambiemos el paso 9 del agente 1 por el paso 2 del agente 2.

A los nuevos individuos, aplicamos el operador de mutación. Lo que hacemos es, modificar la dirección de algunos de los genes (cada uno está asociado a un paso) elegidos aleatoriamente con una probabilidad baja. Si a un gen (paso) le toca mutar, modificamos su valor aleatoriamente entre 1 y 8, sin que se repita el valor que tenía anteriormente.

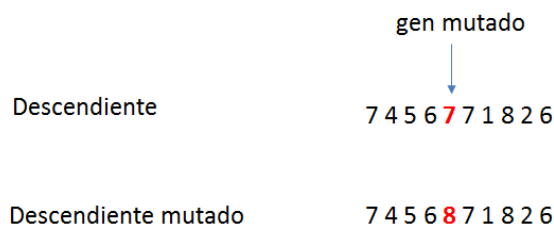


Figura 4. 13 Mutación

Al final, se recombina la población anterior y la nueva, eligiendo los mejores individuos de ambas poblaciones. Esta nueva población, del mismo tamaño que la población original, es en principio mejor que la anterior ya que hemos desechado individuos peores. En la siguiente iteración, la nueva población servirá de nuevo como base al algoritmo, mejorándose sucesivamente las soluciones. Se realizará este proceso un número fijo de iteraciones.

4.2.4 Algoritmos de optimización bayesiana (BOA)

La estructura de las diferentes versiones del BOA que se van a analizar es la que se presenta en el artículo [Lanillos et al., 2013]. En ella, se genera una población de soluciones aleatoriamente (al igual que en GA), se evalúan las soluciones con LET o DD, se organizan las soluciones según los valores de la función de mérito utilizada, se selecciona el 10% de mejores soluciones como datos para aprender la BN que captura la distribución de las mejores soluciones obtenidas en la población actual, y, finalmente, se muestrea la nueva población de soluciones a partir de la BN aprendida.

Recordemos que el proceso de aprendizaje de BN es en si mismo un proceso computacionalmente costoso de optimización de una función de mérito que mide la bondad del ajuste de la estructura y las tablas de probabilidad de una BN a los datos disponibles (soluciones seleccionadas como mejores por BOA). Además, esta función de mérito no es única, y aunque lo habitual en BOAs es usar Bdeu, el uso de otras funciones de mérito es posible. Por lo tanto, en el análisis de este trabajo se compararán los resultados obtenidos con BDeu (que es el método utilizado en [Lanillos et al., 2013]) y con BIC.

El algoritmo voraz utilizado para aprender la red a partir de cada conjunto de datos (soluciones seleccionadas en cada generación) comienza con el cálculo del mérito sobre la BN más sencilla (en la que supone que todas las variables son independientes). A continuación y de forma iterativa, se añade aquel enlace de la red que proporciona la mejora más alta de la función de mérito utilizada, Bdeu o BIC. Además, para limitar el tiempo de cómputo del proceso, se fija el número máximo de padres de un nodo. El proceso termina cuando la inclusión de un nuevo enlace no es posible o ningún nuevo enlace mejora el valor de la función de mérito utilizada.

Alternativamente y con el objeto de eliminar el tiempo de cómputo asociado al aprendizaje de la estructura de la BN, también se comprobará si el uso de una BN predeterminada, la más

sencilla que supone que todas las variables son independientes, permite resolver el problema con éxito. Es decir, si el suponer que los valores de los genes son independientes unos de otros no impide que la heurística alcance una buena solución.

4.2.5 Algoritmos Mixtos (ROM y GOM)

Para la versión de BOA de BIC que aprende la BN, se incluye una función de variación previa al aprendizaje de la misma. La función de mutación es la misma en ambos algoritmos mixtos, y en lo único que difieren, como se ha visto anteriormente, es la forma en que se elige el padre con el que se mutan los genes. En el caso de ROM, elegimos para cada iteración el mismo padre, mientras que en el caso GOM, usamos para cada individuo un padre diferente seleccionado de una piscina genética, diferente para cada iteración.

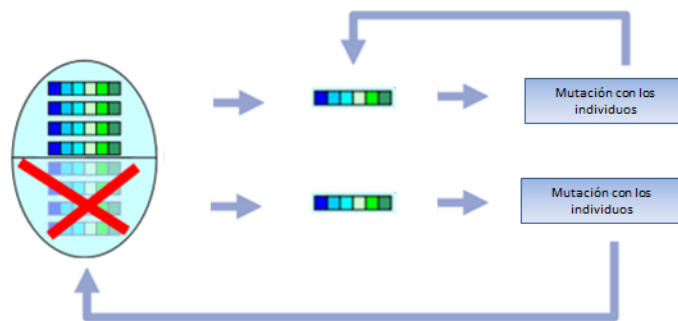


Figura 4. 14 Selección padre ROM y GOM

Las variaciones empleadas son una especie de mezcla entre la mutación y el cruce clásicos de los GA comentadas anteriormente. Se recorren los genes de cada individuo de la población y se modifican algunos (seleccionando los genes a mutar con una probabilidad baja) cambiando su valor por el del padre elegido con la metodología de ROM o de GOM.

	genes mutados
	↓ ↓
Individuo	7 4 5 6 7 7 1 8 2 6
Padre	6 5 3 8 5 1 1 6 3 4
Individuo mutado	7 4 5 6 5 7 1 8 3 6

Figura 4. 15 Variación Mixta

5. Resultados

A continuación, se recogen los resultados obtenidos. Debido a la naturaleza probabilística del problema de MTS y a la naturaleza estocástica de las heurísticas utilizadas para optimizar DD o TET, se realizará un estudio estadístico de los mejores resultados obtenidos al ejecutar 30 veces cada variante de cada heurística y cada función de mérito sobre un conjunto de escenarios.

5.1 Escenarios

Para analizar la buena o mala actuación de los diferentes algoritmos sobre diferentes condiciones iniciales y modelos de probabilidad, usaremos los 5 escenarios que detallamos a continuación.

Estos escenarios, que han sido previamente utilizados en (Lanillos et al, 2013), se diferencian en el número de agentes (M), el número de acciones de control (N), las posiciones iniciales de los agentes, la probabilidad inicial de la posición del objetivo y los modelos probabilísticos dinámicos del objetivo. En todos los casos se emplea el mismo espacio de dimensión 20×20 celdas y el mismo modelo de sensor ideal.

En la Figura 5.1 se esquematizan sus características más relevantes, utilizando estrellas para representar la posición inicial de los agentes, formas amarillas para mostrar las zonas no nulas de probabilidad inicial, y flechas negras para resumir las tendencias dinámicas de los objetivos. Adicionalmente, las flechas azules muestran los pasos iniciales de los caminos que debería seguir cada agente para minimizar el tiempo de búsqueda.

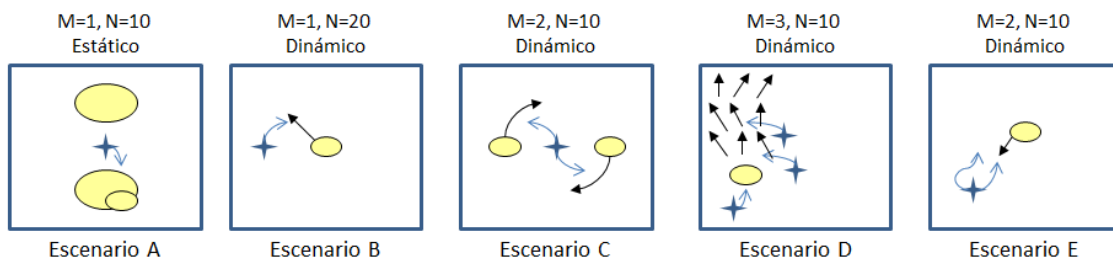


Figura 5. 1 Escenarios

Las características más relevantes de cada escenario son las siguientes:

- En el escenario A tenemos un solo agente y un objetivo estático situado en alguna de las dos regiones elípticas gaussianas. Una de estas regiones presenta una anomalía adicional que hace que el acercarse a ella sea ventajoso. Sobre este escenario, hay que optimizar trayectorias de 10 pasos.
- En el escenario B tenemos un solo agente y un objetivo dinámico situado en una masa de probabilidad pequeña que se desplaza en una dirección determinada. Las trayectorias a optimizar están formadas por 20 acciones.
- En el escenario C tenemos dos agentes y un objetivo localizado en alguna de las dos masas de probabilidad que se desplazan girando alrededor del punto inicial de los agentes que deben seguir trayectorias formadas por 10 acciones.

- En el escenario D tenemos tres agentes y un objetivo situado inicialmente en una masa de probabilidad dinámico. El modelo de desplazamiento del objetivo está asociado a las corrientes de una región del mar, y por lo tanto, la dispersión de la masa de probabilidad es diferente en cada zona del espacio. Cada agente realizará 10 acciones.
- Finalmente, en el escenario E, tenemos dos agentes y un objetivo dinámico situado sobre una masa de probabilidad pequeña, que se desplaza hacia la posición inicial de ambos agentes. De nuevo, los agentes realizarán 10 acciones.

5.2 Heurísticas analizadas

A modo de resumen, las heurísticas que se analizan se recogen en la figura 5.2, agrupadas por familias. Dicho agrupamiento además servirá para comprobar qué algoritmo es mejor dentro de cada familia.

GA	EDA	Mixto
1 punto de cruce	BIC independiente	ROM
2 puntos de cruce	BIC normal	GOM
	BDeu independiente	
	BDeu normal	

Figura 5. 2 Heurísticas empleadas

Además, es importante resaltar que dentro de cada heurística podemos elegir entre dos funciones de mérito diferentes (DD y TET) para evaluar las soluciones. Por lo tanto, el número de heurísticas si duplica, ya que estudiaremos el comportamiento de (GA+1 punto de cruce+DD), (GA+1 punto de cruce+TET), (GA+2 puntos de cruce+DD), ...

Durante las optimizaciones, el número de individuos en las poblaciones de todas las heurísticas se calcula, tal y como se mencionó antes como $N \times M \times 8 \times 10$. El número de individuos seleccionados en EDA para aprender la BN es del 10% ($0.1 \times N \times M \times 8 \times 10$). El número de padres en GA es el mismo que el número de individuos ($N \times M \times 8 \times 10$), mientras que en ROM y GOM se utilizan respectivamente 1 padre o $0.1 \times N \times M \times 8 \times 10$ padres en la etapa de variación. Además, tanto en GA como en los algoritmos Mixtos se utiliza una probabilidad de mutación del 10%.

Finalmente, es importante destacar que dentro de la clasificación de los EDAs, se recogen únicamente BOAs, y que aquellos denominados independientes en la Figura 5.2 se refieren a las versiones en la que se supone que la BN está prefijada a una sin conexiones (que indica la independencia de las variables) y las versiones normales a aquellas en las que también se aprende la estructura de la BN.

5.3 Metodología del análisis estadístico

Analizaremos los resultados obtenidos por los algoritmos de forma estadística debido a las siguientes razones:

- Por un lado, estamos utilizando algoritmos de optimización estocásticos (es decir, no deterministas), que emplean variables aleatorias. Por este motivo, para unos valores iniciales, las diferentes poblaciones pueden evolucionar de distintas maneras,

obteniéndose finalmente soluciones diferentes cada vez que realicemos el mismo experimento u optimización. Este comportamiento se observa tanto en los GAs (debido al no determinismo de sus etapas de selección de padre, cruces y mutación) como en los EDAs (debido a la variabilidad introducida en el muestreo de la BN) y en los algoritmos Mixtos (que combina el no determinismo de GA y EDA)

- Por otro lado, estamos resolviendo el problema de MTS con dos funciones de mérito diferente (TET y DD). Por lo tanto, a la hora de comparar los resultados obtenidos con ambas funciones es necesario utilizar un criterio común, asociado al tiempo medio de búsqueda. Aunque TET permite evaluarlo de forma analítica, el comparar el TET analítico de las soluciones obtenidas con TET con el TET analítico de las soluciones obtenidas con DD, podría favorecer indirectamente a las soluciones obtenidas con TET. Para evitarlo, utilizamos otra forma de evaluación, que no es usada por ninguno de las dos funciones objetivo y que consiste en evaluar el tiempo esperado de búsqueda de forma estadística, evaluando las soluciones obtenidas por cada algoritmo, sobre escenarios muestra de cada escenario y modelo dinámico.

Para poder comparar estadísticamente los resultados obtenidos con todas las versiones de las heurísticas, tanto cuando utilizan TET como DD, se ejecuta cada versión de la heurística con cada función objetivo 30 veces sobre cada escenario y se almacena la solución obtenida en cada una de estas ejecuciones. Los resultados obtenidos se agrupan por escenario, versión de heurística y función de mérito; y a continuación se comparan unos con otros.

A continuación vamos a comparar los datos obtenidos por familias de algoritmos por medio de los criterios utilizados en [Lanillos et al, 2013] y que se presentan en los cuatro apartados siguientes: Tiempo Esperado Truncado Experimental, Función de mérito respecto a Iteraciones, Función de mérito respecto al Tiempo y Probabilidad Acumulada. Finalmente, y en el último apartado de este capítulo se comparan los mejores representantes de cada familia de algoritmos, con el objeto de determinar si una heurística funciona mejor que otra en algún escenario.

5.4 Tiempo Esperado Truncado Experimental

El primer estudio presentado es el que se denomina Tiempo Esperado Truncado Experimental (Experimental Truncated Expected Time, ETET) en [Lanillos et al, 2013]. Este tiempo experimental representa el tiempo necesario para detectar el objetivo con cualquier agente durante las medidas realizadas durante el recorrido final que propone el algoritmo como óptimo.

Para obtener este valor, se generan 10000 trayectorias para cada escenario muestreando la posición inicial del objetivo de su mapa de creencia inicial y desplazándolo de acuerdo con su modelo dinámico. A continuación, se comparan los trayectorias resultados de las 30 optimizaciones de cada versión de heurística & función objetivo, se determina en qué paso del recorrido de los agentes se encuentra el objetivo, y se calcula el valor medio experimental del tiempo esperado truncado (ETET).

Para poder observar la evolución de las heurísticas a lo largo de las iteraciones, el estudio se realiza sobre el mejor resultado obtenido en cada iteración, y en cada una de las siguientes gráficas de las Figuras 5.3 y 5.4 se representa, agrupado por familias, los ETET obtenidos a lo

largo de las diferentes generaciones de cada variante de las heurísticas. La organización de las gráficas en cada figura (y que se mantendrá durante los apartados 5.4-5.7) es la siguiente: en cada fila se presenta un escenario, y en cada columna los resultados asociados a una familia. La Figura 5.3 recoge los resultados obtenidos con DD y la gráfica 5.4 los resultados obtenidos con TET.

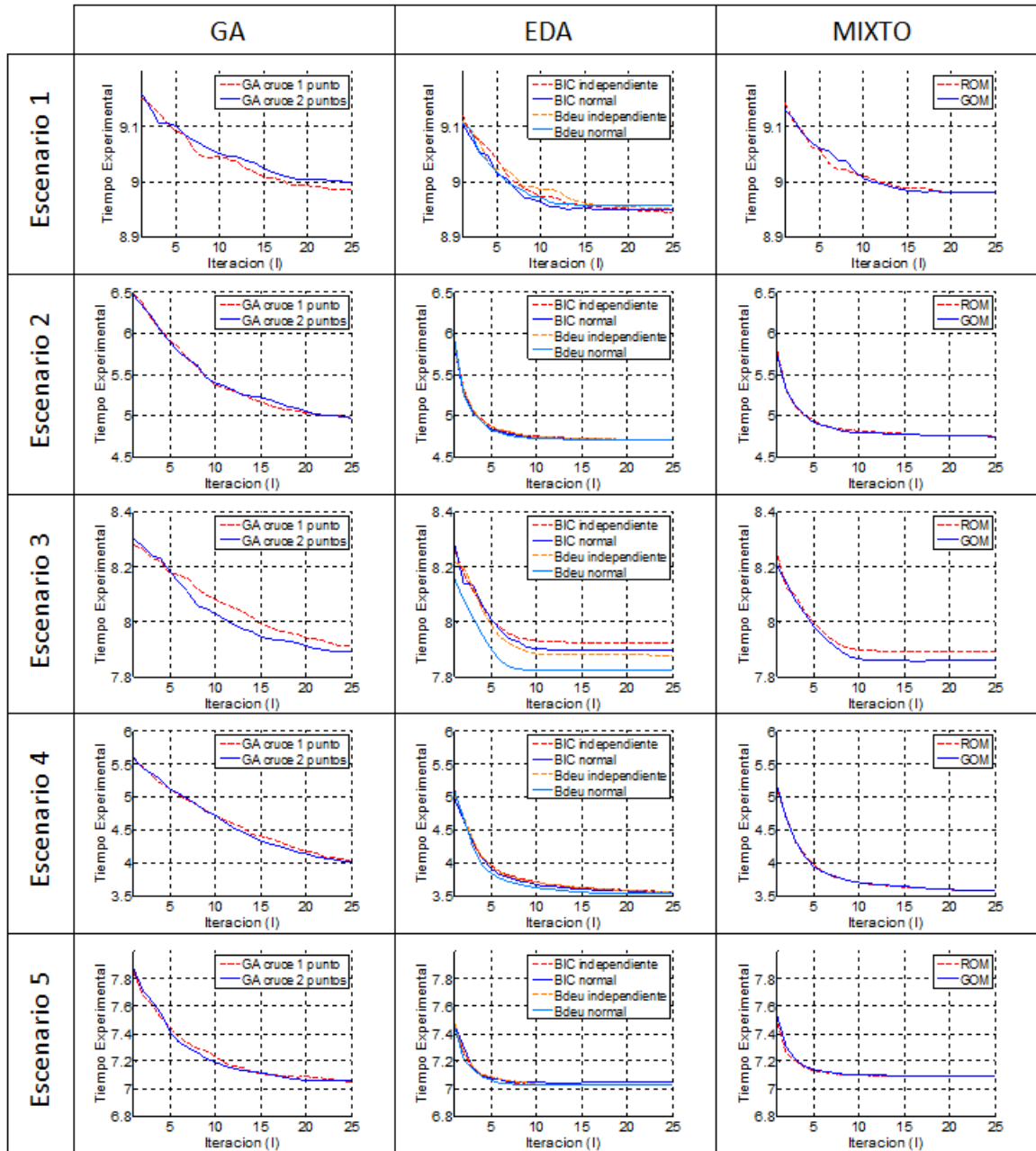


Figura 5. 3 Tiempo Experimental para DD

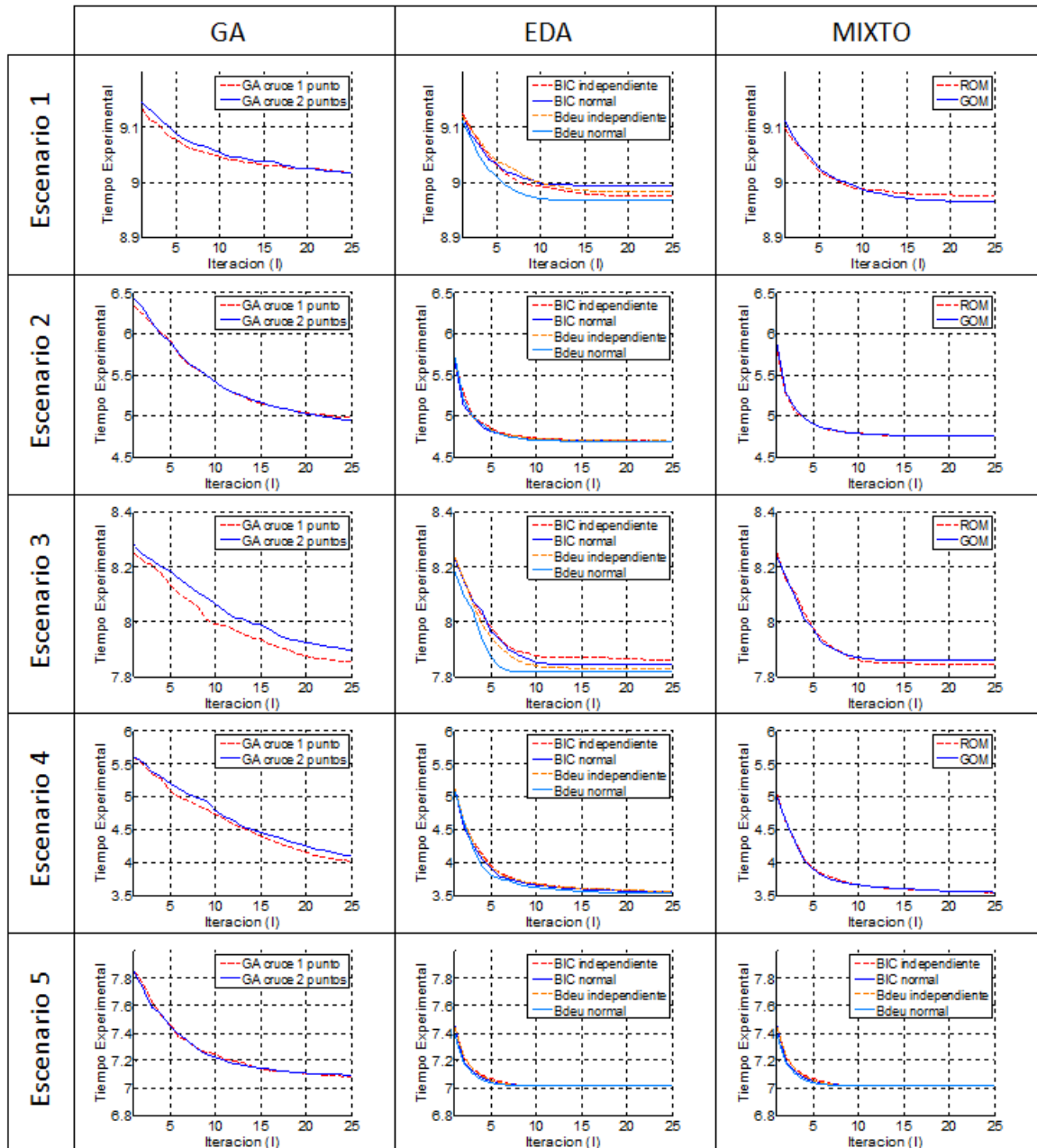


Figura 5. 4 Tiempo Experimental para TET

El ETET que acabamos de representar mide el tiempo que se espera que se va tardar en encontrar el objetivo, por lo que cuanto menor sea el valor, mejor será la solución. En las gráficas vemos que a medida que se realizan iteraciones en el algoritmo, los tiempos son menores, por lo que acabamos obteniendo mejores tiempos a medida que mejora la población en cada iteración.

Si nos fijamos en los valores finales, es decir, para la última iteración, vemos que los valores agrupados por familias de algoritmos, son similares. Es decir, dentro de los GAs, se obtienen valores similares para 1 y 2 puntos de cruce, y ocurre lo mismo para los cuatro EDAs y para los dos Mixtos.

Si bien es verdad que se obtienen valores similares, en el escenario 3 para DD, parece que es donde se aprecian variaciones más significativas, obteniendo mejores resultados el GA de 2 puntos, el BDeu con conexiones y el GOM, en cada una de sus respectivas familias. Para la función TET, obtenemos un comportamiento similar que para DD, dentro de las familias se obtienen valores similares, salvo en los escenarios 1 y 3, donde parece que Bdeu con conexiones y GOM para el escenario 1 y BDeu normal y ROM para el caso 3 funcionan. En el escenario 4, el GA de 1 punto de cruce parece proporcionar mejores resultados.

Cabe destacar también el comportamiento de GAs frente a EDAs y Mixtos, mientras que para los primeros, las soluciones evolucionan de una forma más lenta hacia valores mejores, parece que en EDAs y Mixtos, las soluciones convergen antes a valores óptimos. Los algoritmos estaban fijados en 25 iteraciones para compararlos en iguales condiciones. A la vista de la evolución de los GAs, es posible que hagan falta más de 25 iteraciones para asegurar la convergencia a ciertos valores. Este hecho además podría reducir los tiempos de cómputo en los EDAs, ya que vemos que convergen rápidamente, y podríamos reducir el número de iteraciones de los algoritmos.

En general, la familia de EDAs proporciona mejores tiempos que las otras dos familias, siendo los resultados obtenidos por medio de GAs peores en todos los casos.

5.4 Función de mérito respecto Iteraciones

El segundo estudio muestra la evolución del individuo con mejor función de mérito, DD o TET según corresponda, respecto a la iteración del algoritmo. Para ello se representan los valores de la media y de la desviación estándar de los valores de la función de mérito de las 30 ejecuciones de cada heurística.

En este conjunto de gráficos analizamos cómo evoluciona la función de mérito en el algoritmo a lo largo de las iteraciones y por tanto podemos ver si realmente la población, o al menos el mejor individuo de ella, evoluciona hacia mejores soluciones. Cabe destacar que para el caso de DD mejores valores significa valores más altos, mientras que para TET mejores valores significa valores más bajos.

A continuación, en las figuras 5.5 y 5.6 se recogen los resultados obtenidos, de nuevo agrupados por familias. Como en el caso de estudio anterior, en la figura 5.5 se recogen los resultados obtenidos para la función DD y en la figura 5.6 se hace lo propio para la función TET.

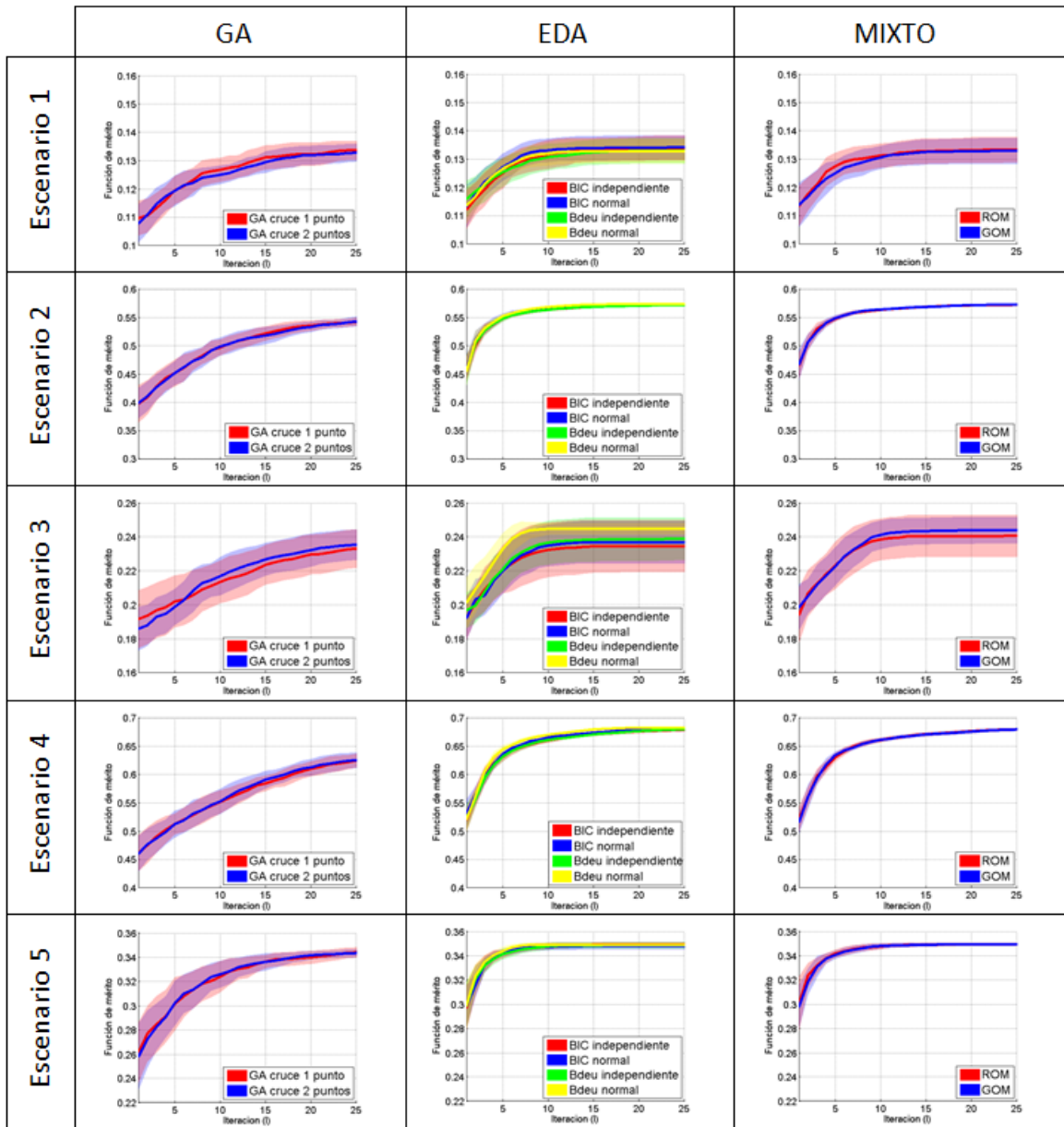


Figura 5. 5 Función de mérito respecto Iteraciones para DD

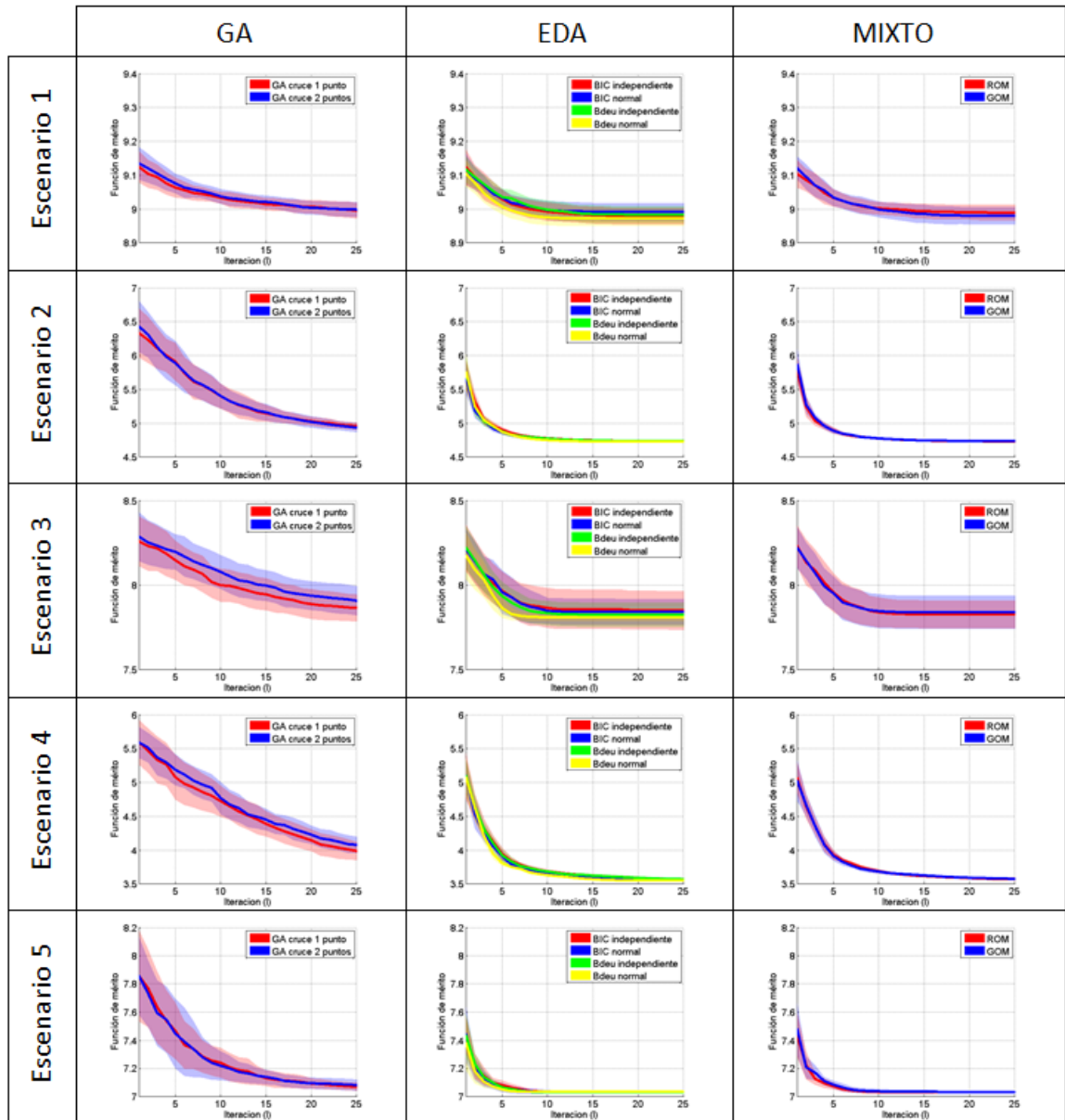


Figura 5. 6 Función de mérito respecto Iteraciones para TET

Los resultados obtenidos presentan un comportamiento muy similar al observado para el tiempo experimental: dentro de familias de algoritmos se obtienen comportamientos similares y resultados parecidos, con leves variaciones.

De nuevo, el escenario 3 parece ser el único en que destacan algunos algoritmos: El GA de 2 puntos de cruce, el BDeu con conexiones y el GOM.

Así mismo, de nuevo, EDAs y Mixtos parecen proporcionar valores mejores y GAs se quedan un poco atrás en este sentido. Los valores obtenidos por EDAs y Mixtos son prácticamente iguales.

Como en el estudio anterior, GAs obtienen peores resultados que EDAs y Mixtos y su convergencia es más lenta.

5.5 Función de mérito respecto Tiempo

El tercer estudio muestra la evolución del individuo con mejor función de mérito, DD o TET según corresponda, esta vez respecto al tiempo real de computación de los algoritmos ejecutados sobre un ordenador procesador Intel(R) Core(TM) i5-2450M a 2.50GHz con 8Gb de RAM. En este caso sólo se representan los valores medios de las funciones de mérito de las 30 ejecuciones de cada versión de la heurística.

De nuevo analizamos cómo evoluciona la función de mérito en el algoritmo comprobando si evoluciona hacia mejores soluciones, en el caso de DD mejores valores son valores más altos, y en el de TET mejores valores son valores más bajos. La diferencia con el experimento anterior radica en que en vez de realizar la comparativa respecto a la iteración del algoritmo, en este caso la realizaremos respecto el tiempo utilizado por el algoritmo para ejecutarse.

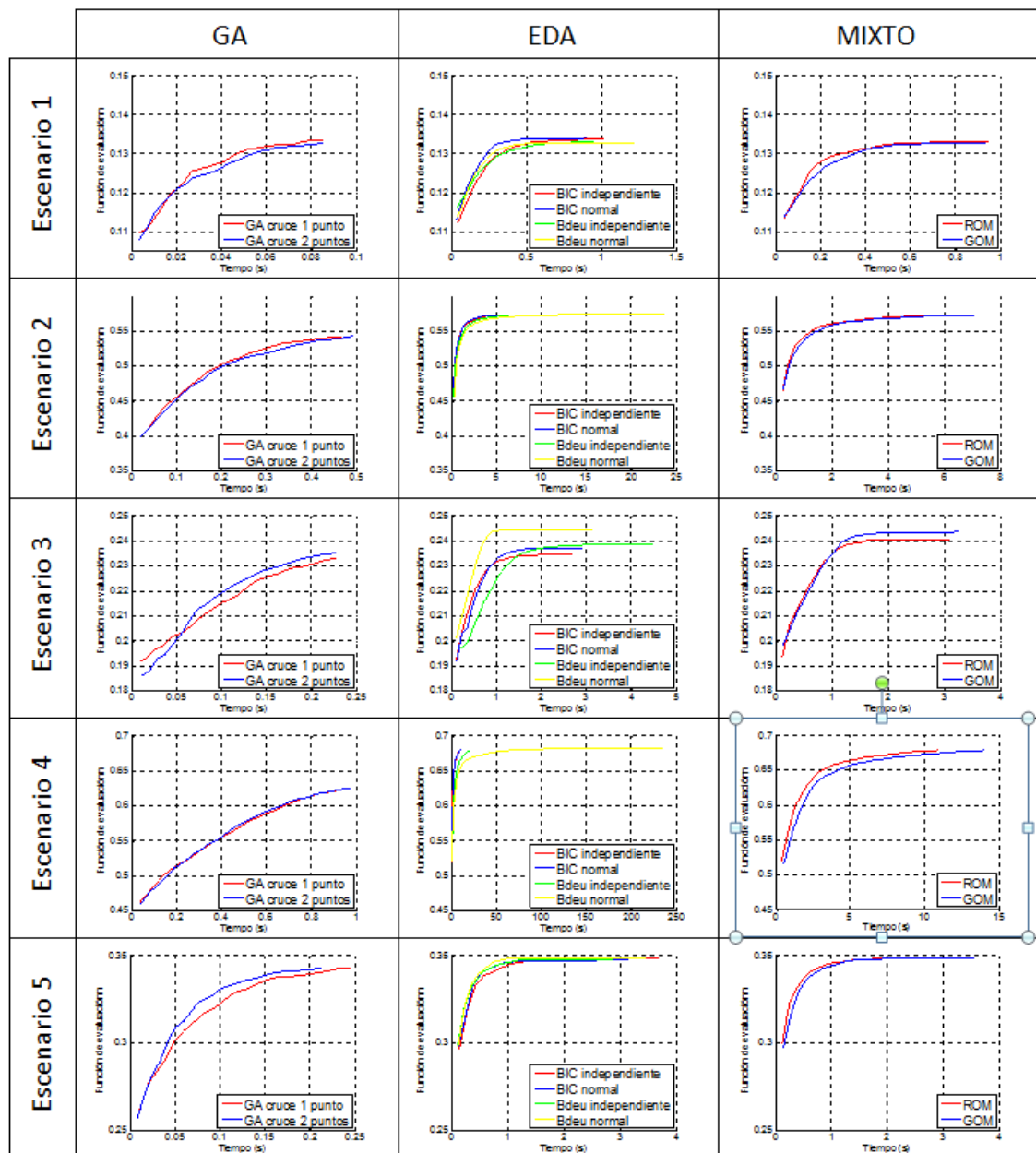


Figura 5. 7 Función de mérito respecto Tiempo para DD

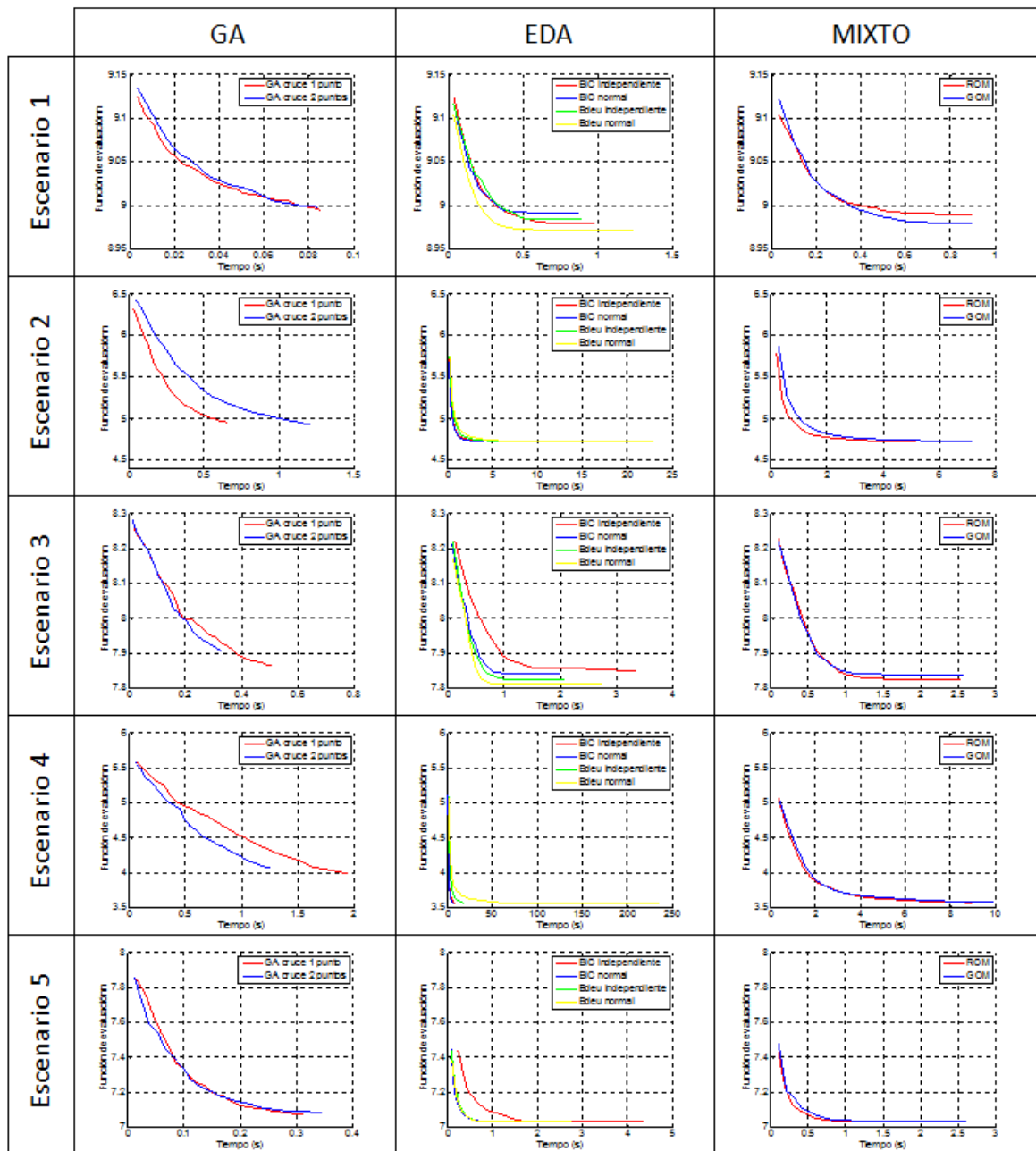


Figura 5. 8 Función de mérito respecto Tiempo para TET

Para comparar las gráficas es importante tener en cuenta que la escala del eje temporal (abcisas) no se mantiene constante a lo largo de las gráficas.

Este caso de estudio arroja diferencias más significativas. Como en los casos anteriores, vemos que EDAs y Mixtos obtienen mejores valores de la función de mérito que GAs, pero vemos que lo hacen a costa de emplear tiempos mucho mayores de computación, hablando incluso de 2 órdenes de magnitud para las optimizaciones con BDEu normal en el escenario 4.

En todos los escenarios, los GAs emplean aproximadamente la décima parte de tiempo de computación que las otras dos familias. Además, para DD, el GA de cruce de 2 puntos obtienen mejores resultados que el de 1 punto en menor tiempo (en algunos casos la mitad). En el caso de TET, el GA de cruce de 2 puntos no se impone tan claramente al de 1 punto de cruce: obtiene mejores tiempos para soluciones similares en el escenario 4, iguales en los escenarios 1 y 5 pero peores en los escenarios 2 y 3.

Los algoritmos Mixtos y todos los EDAs salvo el EDA con BDeu normal, utilizan tiempos similares dentro del mismo escenario. Los motivos de este comportamiento son los siguientes. Por una parte, los algoritmos Mixtos utilizan el criterio BIC para realizar la optimización, y por lo tanto, su tiempo de computo únicamente se ve incrementado por el uso de la etapa de variación adicional. Por otra, y tras realizar un análisis de las estructuras de la red que construyen los EDAs normales, observamos que únicamente el EDA con BDeu normal aprende estructuras con enlaces. Es decir, aunque a BIC normal se le permite aprender estructuras con enlaces, el algoritmo voraz utilizado para crearlas finaliza rápidamente, decidiendo no añadir ningún enlace, porque su término de penalización no permite que la mejora asociada a ajustarse a los datos mejor valga la pena.

Por lo tanto, aunque en los casos de estudio anteriores, hemos visto que, por lo general los EDAs eran los que mejores resultados arrojaban, con estas gráficas podemos ver que lo hacen utilizando un mayor tiempo de computo. Esto es especialmente negativo en el caso de BDeu normal, ya que aprender la BN es otro problema de optimización NP-duro, y hace que el tiempo sea 10 veces superior que para los GAs.

5.6 Probabilidad Acumulada

En el último estudio, se analiza la probabilidad acumulada de detectar al objetivo antes del paso k . Este valor resulta interesante, ya que el tiempo esperado de detección se encuentra directamente ligada con ella. De hecho, el tiempo esperado de detección se puede calcular sumando el remanente de la probabilidad acumulada de detección, y por lo tanto valores altos de la probabilidad acumulada en los instantes previos implican que los agentes encuentran el objetivo antes.

Al igual que en los ejemplos anteriores, se presentan las gráficas para DD y TET organizadas por algoritmos y escenarios. Además, en este caso, únicamente se presentan los resultados obtenidos en la iteración final, ya que en el eje de ordenadas se muestra el valor medio (para las 30 ejecuciones) de la probabilidad acumulada por la mejor solución obtenida frente al eje de ordenadas que recoge los N pasos para los que se ha optimizado la trayectoria.

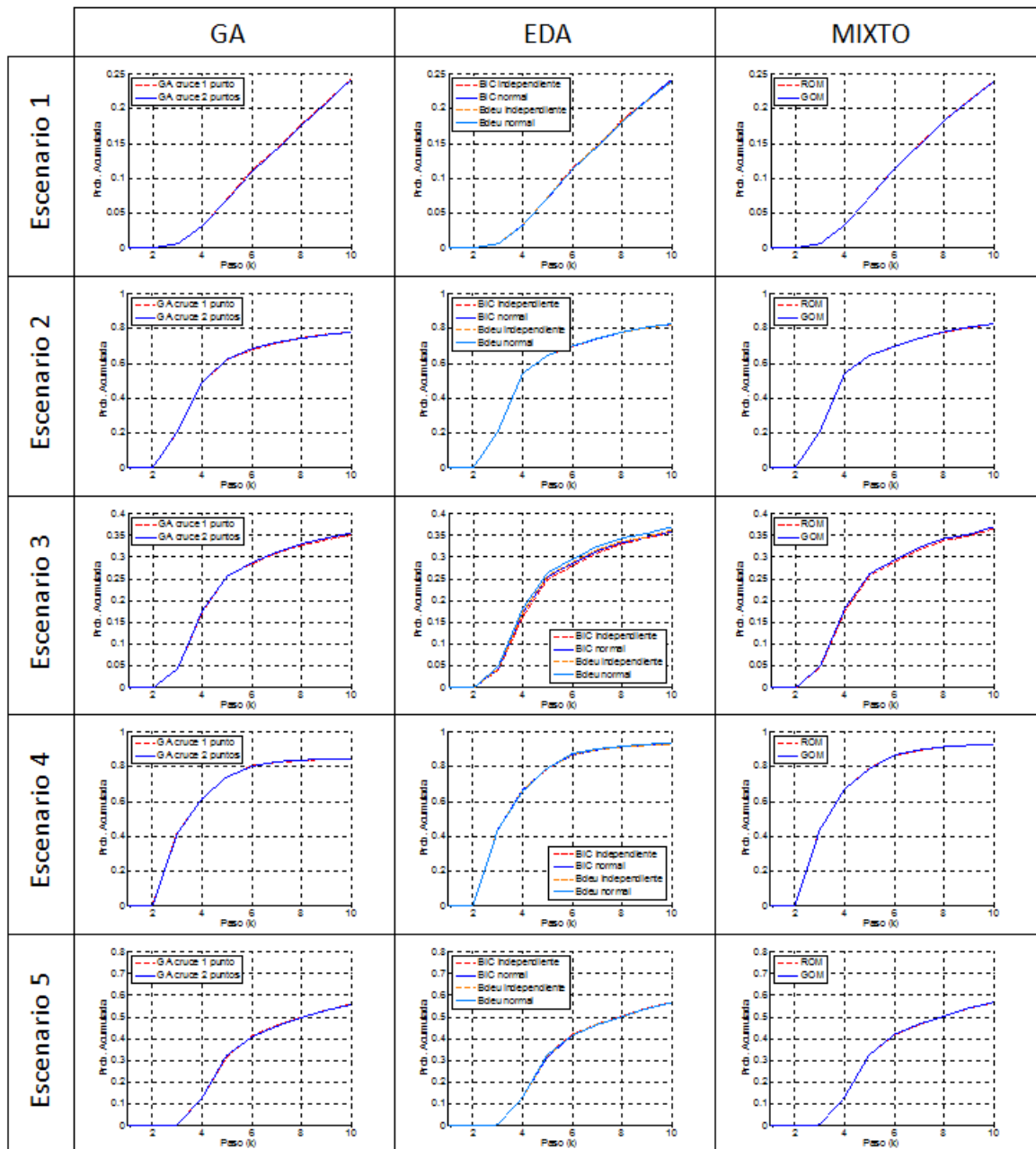


Figura 5. 9 Probabilidad Acumulada para DD

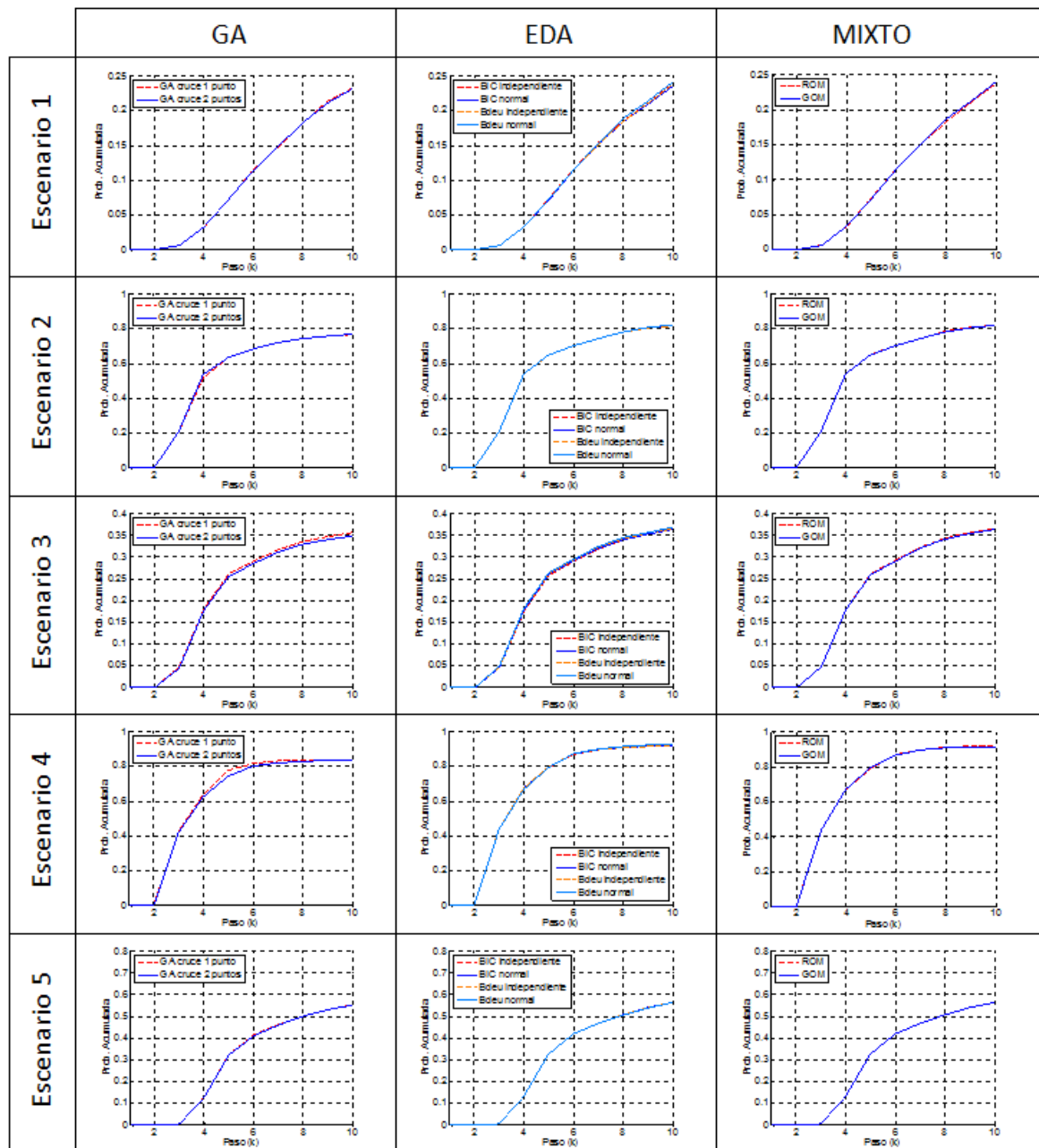


Figura 5. 10 Probabilidad Acumulada para TET

En este estudio podemos ver, tal y como cabría esperar, que las tres familias de algoritmos presentan comportamientos similares a los de los dos primeros apartados, y que el empeoramiento de los GAs es especialmente relevante en los escenarios 2 y 4.

Además, es importante destacar que estas gráficas representan la probabilidad acumulada por los agentes durante su recorrido, así que a la vista de estos datos, todos los algoritmos presentan valores muy similares, lo que significa que la probabilidad de encontrar nuestro objetivo empleando cualquiera de los algoritmos será muy similar. Sin embargo, el tiempo esperado es diferente, ya que para calcularlo hay que ir acumulando la probabilidad remanente, y por lo tanto, las mejoras de probabilidad acumulada en instantes previos, repercuten en el cómputo final del tiempo de búsqueda.

5.7 Comparativa Mejores

A continuación se realiza una comparativa de los mejores algoritmos de cada familia. Para facilitar el análisis se representan sobre la misma gráfica los resultados obtenidos por los 3 algoritmos seleccionados (GA cruce 2 puntos, GOM, BDeu normal) para cada uno de los estudios realizados en los cuatro apartados anteriores, para cada una de las funciones objetivo, y para uno de los escenarios.

Las 4 tipo de gráficas (una por tipo de estudio) de cada función objetivo y escenario se recogen en una figura diferente, donde la gráfica superior izquierda es la del Tiempo Experimental, la gráfica superior derecha la de la función objetivo respecto a las iteraciones, la gráfica inferior izquierda la de la función objetivo frente al tiempo de computo, y la gráfica inferior derecha la de la probabilidad acumulada de detección de la última solución de cada algoritmo. En todas ellas se representan los valores medios, y en la de la función objetivo respecto a las iteraciones la desviación estándar. Las Figuras 5.11-5.15 recogen los resultados asociados a DD y las Figuras 5.16-5.20 los resultados asociados a TET.

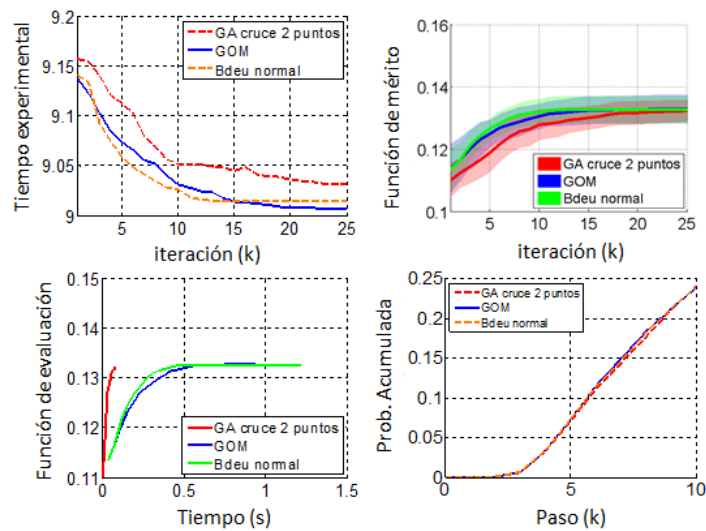


Figura 5. 11 Escenario 1 DD

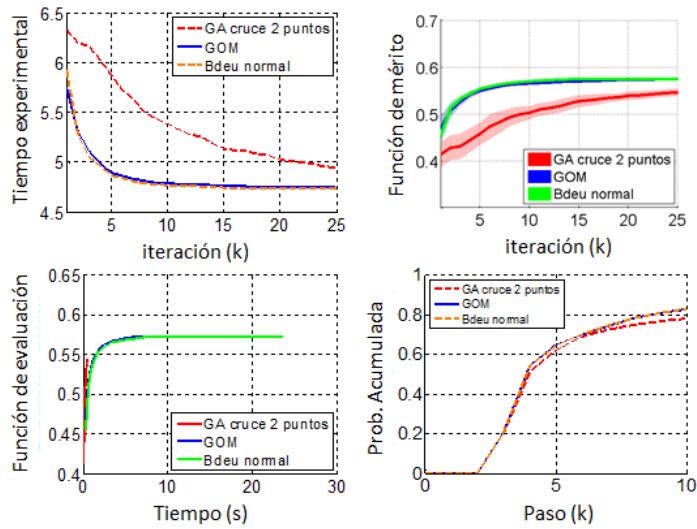


Figura 5. 12 Escenario 2 DD

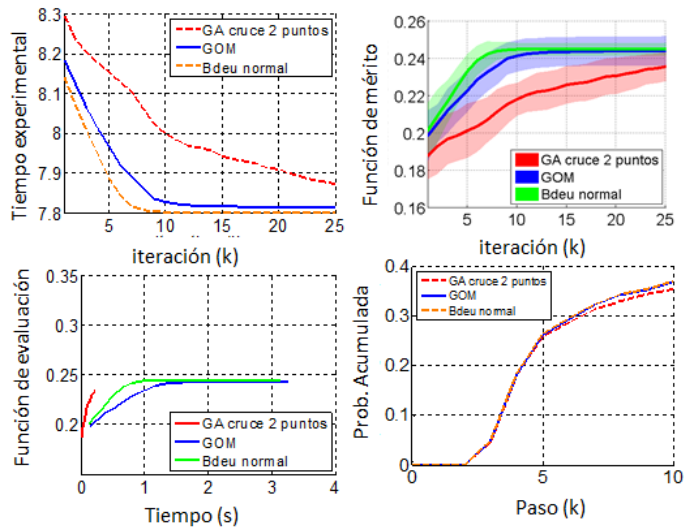


Figura 5. 13 Escenario 3 DD

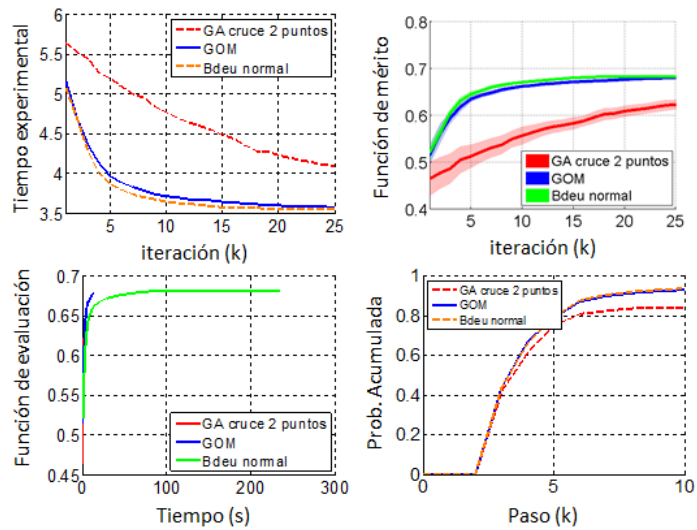


Figura 5. 14 Escenario 4 DD

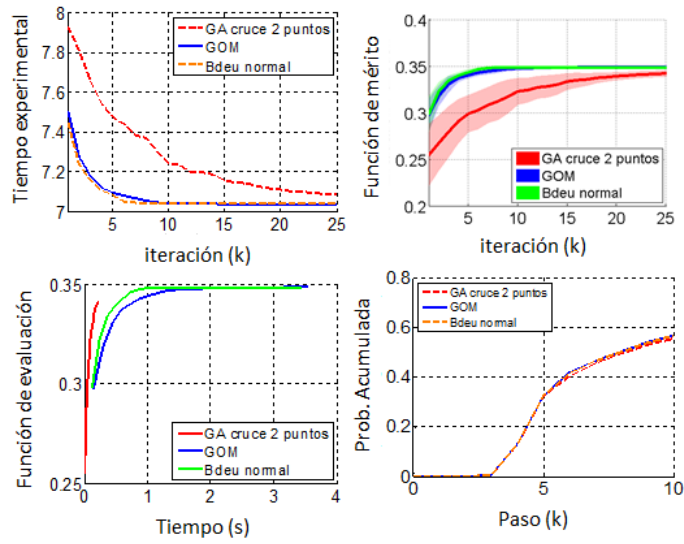


Figura 5. 15 Escenario 5 DD

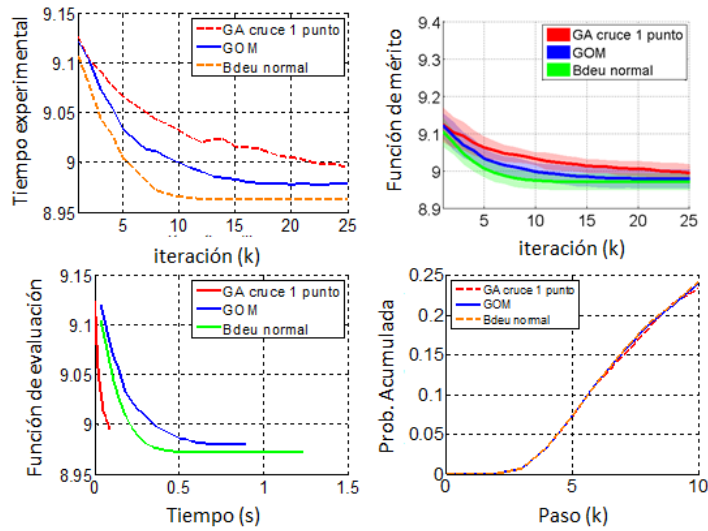


Figura 5. 16 Escenario 1 TET

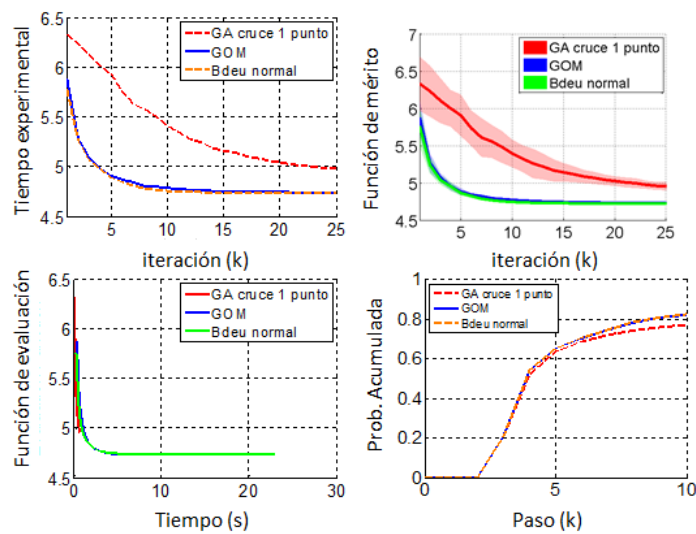


Figura 5. 17 Escenario 2 TET

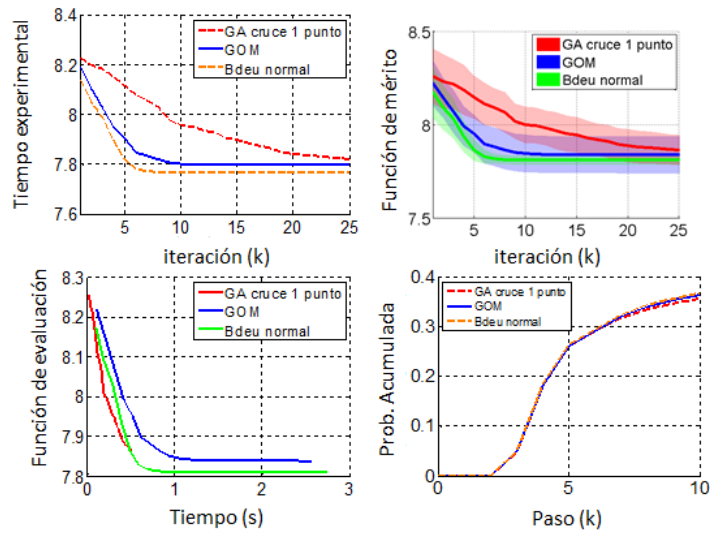


Figura 5. 18 Escenario 3 TET

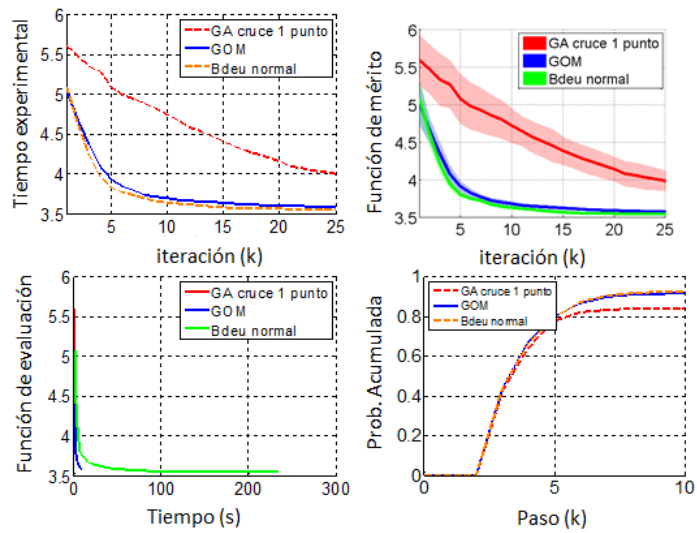


Figura 5. 19 Escenario 4 TET

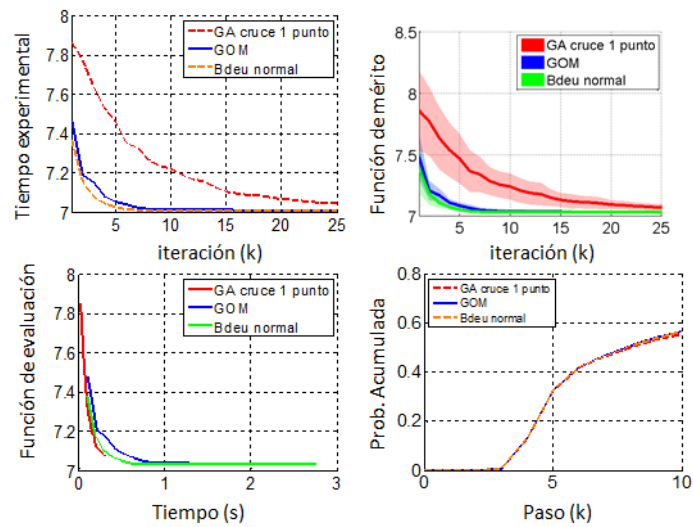


Figura 5. 20 Escenario 5 TET

De estas gráficas, y a modo de resumen podemos concluir:

- Analizando el tiempo experimental, GAs obtienen peores resultados que las otras dos familias de algoritmos.
- Si analizamos la evolución respecto a las iteraciones de los valores obtenidos para las funciones de mérito empleadas para evaluar los individuos en los algoritmos, todos alcanzan valores muy similares, si bien es verdad, EDAs y Mixtos lo logran en menos iteraciones que los GAs.
- Respecto el tiempo de computación, los GAs son más rápido, y aunque los valores de su función de mérito en algunos casos son ligeramente inferiores, los obtienen en tiempos 10 veces menores que el EDA BDeu normal.
- En cuanto a la probabilidad acumulada, vemos que de nuevo GAs tienen en algunas veces comportamientos un poco peores que EDAs y Mixtos. Estas variaciones en algún caso son más llamativas, como en el escenario 4, pero en el resto la diferencia no es significativa.
- Los resultados asociados a EDA con BDeu normal son ligeramente mejores (salvo por el tiempo de cómputo) en varios escenarios. De hecho, mejoran algunas veces en las iteraciones intermedias al algoritmo Mixtos. Esto es debido a que esta versión de BOA es la única capaz de utilizar las dependencias que aparecen entre las acciones de una solución, mientras que el resto de los algoritmos (GAs y Mixtos) no explotan estas dependencias. Sin embargo, los resultados en las iteraciones finales son semejantes en el EDA y en el Mixto.
- Sin embargo, es importante destacar que los resultados de los algoritmos Mixtos son similares a los del EDA con BIC (ver gráficas en apartados anteriores). Por lo tanto, las operaciones adicionales del algoritmo Mixto parece que no introducen mejoras significativas a la hora de realizar los algoritmos.

A la vista de estos datos, podemos afirmar que pese a que EDAs y Mixtos hayan proporcionado en general mejores valores de tiempo experimental y probabilidad acumulada, que al final son los que nos permiten encontrar antes al objetivo, emplean tiempos de computación mucho mayores (sobre todo la versión de EDA con BDeu normal). Además como el tiempo de evaluación de las funciones de mérito crece con el tamaño del espacio de búsqueda, número de agentes y número de pasos de la trayectoria, el tiempo de cómputo crecerá en escenarios más grandes. Esto hecho es relevante a la hora de elegir los algoritmos, ya que aunque el objetivo es minimizar el tiempo de búsqueda, si hay vidas humanas en juego, el tiempo de computación a la hora de calcular las rutas puede ser fundamental también.

Por tanto, aunque GAs generen soluciones peores, y algunos EDAs no consigan explotar las dependencias, las soluciones no difieren mucho de las generadas con el EDA con BDeu normal. Además, al emplear tiempos de cómputo mucho menores, es posible que sean útiles a la hora de generar las primeras soluciones y luego afinar dichas soluciones por medio del EDA.

En definitiva, las tres familias aportan soluciones muy satisfactorias. GAs en tiempos de computación muy bajos y algunos variantes de EDAs una solución compromiso entre el tiempo de cómputo y la calidad de la solución.

Finalmente, es importante destacar que los algoritmos Mixtos propuestos, no han mejorado los resultados de los EDAs, y por lo tanto, es mejor utilizar el EDA que les da soporte: BOA con BIC normal.

6 Conclusiones

En este apartado se analizará la consecución o no de los objetivos proyectados y las posibles mejoras y estudios futuros.

6.1 Consecución de objetivos

El objetivo principal de este TFM era analizar las capacidades de optimización de diferentes técnicas estocásticas para resolver el problema de MTS. Para alcanzarlo, se definieron los siguientes hitos:

- Estudiar del problema de MTS y de una familia de técnicas, denominadas EDAS, que estaban dando resultados satisfactorios en el trabajo de Lanillos [Lanillos et al., 2013]. La consecución de este objetivo se ha logrado, junto con la familiarización de todas las herramientas desarrolladas en dicho trabajo, para su posterior reutilización, con las adaptaciones necesarias, al análisis de técnicas realizado en este TFM.
- Implementar un GA para resolver el problema de MTS. Dicho GA ha sido implementado satisfactoriamente, ya que a la vista de los resultados obtenidos, resuelve el problema y, aunque sus resultados son un poco peores a los obtenidos por parte de los EDAs, obtienen valores buenos en tiempos de computación menores, debido también a su mayor simplicidad computacional.
- Opcionalmente, implementar alguna técnica adicional (Mixta), que combine las características de los EDAs y los GAs. En el trabajo se proponen dos nuevos algoritmos, fundamentados principalmente sobre los EDAs, que incorporan una etapa variacional inspirada en los GAs en la que se mutan las soluciones con la información de padres. Los resultados muestran que obtienen resultados similares al EDA sobre el que sustenta (BOA con BIC normal), y que por lo tanto, aunque funciona correctamente, no aporta ninguna ventaja sobre él.
- Analizar, mediante un estudio estadístico, si alguna de las tres familias de técnicas (GAs, EDAs y Mixtos) es mejor que las anteriores. El TFM recoge un estudio estadístico basado en cuatro características diferentes que determina que la variante de EDA recogida en [Lanillos et al., 2013] es la que obtiene habitualmente resultados más satisfactorios, a costa de un incremento computacional que puede resultar muy elevado para problemas con muchos agentes en los que se optimice la trayectoria durante muchos pasos.

6.2 Trabajo futuro

El futuro abre distintas posibles líneas de trabajo. Por un lado, se plantea la opción de utilizar nuevas formas de implementar los GAs, aplicando nuevas técnicas de cruce y mutación y analizando si mejoran los resultados de probabilidad acumulada y función de mérito, igualándose en este sentido con los EDAs.

Por otro lado, debido a que las discrepancias en tiempos de computación entre EDAs y GAs se deben a la hora de aprender la BN, una forma de intentar mejorar los tiempos puede ser buscar técnicas en que estas dependencias se aprendan de formas más eficientes. Una posible solución podría ser usar GAs para realizar las primeras iteraciones de forma más rápida y a partir de estas aplicar el EDA. Además, como los resultados de los EDAs convergen antes de alcanzar el número máximo de iteraciones, se puede reducir el tiempo de computación

afinando los criterios de parada, y como los resultados de los GA todavía no han convergido utilizar más generaciones en esos algoritmos.

Así mismo, y aplicable tanto a GAs como a EDAs, se abre la puerta a emplear técnicas heurísticas que ayuden a mejorar resultados, haciendo que los agentes busquen en regiones del espacio que por alguna razón han quedado fuera de su espacio de búsqueda.

7 Siglas y abreviaturas

BDeu: Bayesiana de Dirichlet uniforme (Bayesian Dirichlet equivalent uniform)

BIC: Criterio de Información Bayesiana (Bayesian Information Criterion)

BN: Red Bayesiana (Bayesian Network)

BOA: Algoritmos de Optimización Bayesiana (Bayesian Optimization Algorithms)

DD: Detección Descontada (Discounted Detection)

EDA: Algoritmos de Estimación de la Distribución (Estimation Distribution Algorithms)

ETET: Tiempo Esperado Truncado Experimental (Experimental Truncated Expected Time)

GA: Algoritmos Genéticos (Genetic Algorithms)

GOM: Gene-pool Optimal Mixing

MTS: Búsqueda en Tiempo Mínimo (Minimum Time Search)

POMDP: Procesos de Decisión Parcialmente Observables de Markov (Partially Observable Markov Decision Process)

ROM: Recombinative Optimal Mixing

TET: Tiempo Esperado Truncado (Truncated Expected Time)

8 Bibliografía

- Bernstein D. S., Givan R., Immerman N., Zilberstein S.,(2002). The complexity of decentralized control of markov decision processes, *Math. Oper. Res.*, vol. 27, pp. 819–840, November 2002.
- Blum, A., Chawla, S., Karger, D. R., Lane, T., Meyerson, A., and Minko, M. (2003). Approximation algorithms for orienteering and discounted-reward TSP. In Proc. 44th Annual IEEE Symp. Foundations of Computer Science, pages 46-55.
- Bourgault F., Furukawa T., Durrant-Whyte H. F.,(2003). Optimal search for a lost target in a bayesian world, in *FSR*, 2003, pp. 209–222.
- Bourgault F., Furukawa T., Whyte H. F. D.,(2004). Decentralized bayesian negotiation for cooperative search, in *IEEE Conf. Intelligent robots and Systems*.
- Clark J., Fierro R.,(2005). Cooperative hybrid control of robotic sensors for perimeter detection and tracking, in *Proc. American Control Conference the 2005*, pp. 3500–3505 vol. 5.
- Eagle J. N. , Yee J. R.,(1990). An optimal branch-and-bound procedure for the constrained path, moving target search problem, *Oper. Res.*, vol. 38, pp. 110–114.
- Eagle J. N.,(1984). The optimal search for a moving target when the search path is constrained, *Operations Research*, vol. 32, no. 5, pp. 1107–1115.
- Gan S. K., Sukkarieh S.,(2010). Multi-uav target search using explicit decentralized gradient-based negotiation, in *Proc. IEEE International Conference on Robotics and Automation*.
- Goldberg D.E., (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.
- Hsu, D., Lee, W. S., and Rong, N. (2008). A point-based POMDP planner for target tracking. In Proc. IEEE Int. Conf. Robotics and Automation ICRA 2008, pages 2644-2650.
- ISBA (2009). *Statistics in the real world: The search for the scorpion*. International Society for Bayesian Analysis.
- Lanillos P., Besada-Portas E., Pajares G., Ruz J. J.,(2012). Minimum time search for lost targets using cross entropy optimization. In Proc. IEEE/RSJ Int. Conf. Intell. Robots and Systems, pages 602 {609.
- Lanillos, P. (2013). *Búsqueda de objetivos móviles en tiempo mínimo en entornos con incertidumbre*. Tesis doctoral. Departamento de Arquitectura de Computadores y Automática, Facultad de Informática, Universidad Complutense de Madrid.
- Lanillos, P., Besada-Portas, E., Pajares, G., and Ruz, J. J. (2012). Minimum time search for lost targets using cross entropy optimization. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 602-609.

Lanillos, P., Yañez-Zuluaga, J., Ruz, J.J., Besada-Portas, E.,(2013). A bayesian approach for constrained multi-agent minimum time search in uncertain dynamic domains. ;In GECCO,391-398.

Larrañaga, P., Lozano, J. A.,(2002), (Eds.). Estimation of distribution algorithms: A new tool for evolutionary computation. Kluwer Academic Publishers, Boston.

Lavis B., Furukawa T., Durrant Whyte H. F.,(2008). Dynamic space reconfiguration for bayesian search and tracking with moving targets, *Auton. Robots*, vol. 24, no. 4, pp. 387–399.

Ljunggren, D., (2009). "Update 2-No more survivors in Canada chopper crash, 17 dead". Reuters. Retrieved 13 March 2009.

Mathews G.,(2008), Asynchronous decision making for decentralised autonomous systems, Ph.D. dissertation, The University of Sydney.

Melson L. B., CAPT USN (1967). Contact 261. United States Naval Institute Proceedings.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press

Neapolitan R. E.,(2003) Learning Bayesian Networks. Prentice Hall.

Pelikan, M. (2005), Hierarchical Bayesian optimization algorithm: Toward a new generation of evolutionary algorithms, Springer.

Pelikan M., Goldberg D. E., Cantú-Paz E., (1999). BOA: The Bayesian optimization algorithm. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), I, 525-532. Also IlliGAL Report No. 99003.

Richardson H.R., Stone L.D., (1971). Operations Analysis During the Underwater Search for Scorpion, Naval Research Logistics Quarterly, Vol. 18.

Ross P. E., Top 11 technologies of the decade,(2011). *IEEE Spectrum Magazine*, vol. 48, no. 1.

Sarmiento A., Murrieta-Cid R., Hutchinson S.,(2009). An efficient motion strategy to compute expected-time locally optimal continuous search paths in known environments, *Advanced Robotics*, vol. 23, no. 12-13, pp. 1533–1560.

Stone L. D.,(1975). Theory of optimal search, *Academic Press*, New York.

Thierens D., Bosman P. A. N., (2011), Optimal Mixing Evolutionary Algorithms, *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2011*, pages 617-624, ACM Press, New York, New York.

Trummel K. E., Weisinger J. R.,(1986). The complexity of the optimal searcher path problem, *Operations Research*, vol. 34, no. 2, pp. 324–327.

Vose, M. (1999). *The Simple Genetic Algorithm: Foundations and Theory*. Cambridge, MA: MIT Press.

Washburn A. R.,(1998). Branch and bound methods for a search problem, *Naval Research Logistics (NRL)*, vol. 45, no. 3, pp. 243–257.

Yang Y., Minai A., Polycarpou M.,(2002). Decentralized cooperative search in uav's using opportunistic learning, *AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Yang Y., Minai A., Polycarpou M.,(2004). Decentralized cooperative search by networked uavs in an uncertain environment, in *Proc.American Control Conference the 2004*, vol. 6, 30 June–2 July 2004,pp. 5558–5563.