



# Algoritmos Genéticos – Aplicación Práctica

## Objetivos

Con esta práctica se pretende que el alumno se plantee y familiarice con los requerimientos y características de la solución de problemas reales empleando técnicas de computación evolutiva; por medio de un caso práctico.

El alumno podrá comprobar los problemas que plantea el estudio e implementación de un problema real con ayuda de una librería de funciones ya facilitada. Se pide también el análisis de la conveniencia de la solución propuesta y la reflexión sobre los propios resultados.

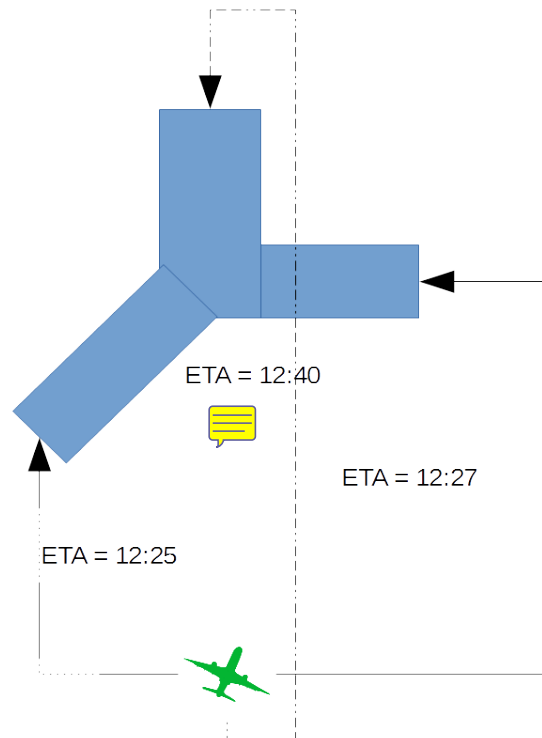
## Descripción

Se pretende obtener un programa que permita al usuario el calculo automático de la distribución más conveniente de vuelos que solicitan aterrizar en un determinado aeropuerto. Debido al número de vuelos, las diferentes horas de llegada, tipos de avión y condiciones de las pistas, el poder encontrar una configuración óptima es de gran importancia puesto que permite disminuir los gastos de las operaciones (tiempo, combustible...etc.)

Se pretende construir una aplicación que pueda adaptarse fácilmente a las diferentes condiciones que se puedan dar en este tipo de problemas:

Se pretende modelar el aeropuerto con una configuración variable que incluye un número de pistas que puede cambiar en cada uno.

Por su parte, los aviones tienen un programa de vuelo que incluye el parámetro de Tiempo Estimado de llegada o *Estimated Time of Arrival* (ETA en inglés) que será diferente dependiendo de la pista a la que se asigne su aterrizaje (orientación, condiciones climatológicas...). Este parámetro se calcula en el momento de la salida del vuelo y en este caso, se facilita como dato del problema. Adicionalmente los aviones se



*Ilustración 1: Un vuelo puede tener un Tiempo de Llegada Estimado diferente para cada pista del aeropuerto*





clasifican en varios tipos. Dependiendo del tipo de avión, se tardará un tiempo diferente en atenderlos para que despejen la pista y dejen paso al siguiente avión en aterrizar. Este tiempo depende de los dos aviones que aterricen de forma consecutiva y es fijo para todas las pistas.

La restricción básica a tener en cuenta en el problema es el tiempo de espera de un vuelo para aterrizar.

Como ejemplo: si el vuelo V1 tiene un ETA en la pista 1 dentro de 120 minutos, pero se ha colocado delante de él otro vuelo (V2) cuyo ETA es de 117 y el tiempo de distancia mínimo entre ambos es de 10 minutos, la hora de aterrizaje del V1 será forzosamente dentro de 127 min. y tendrá que esperar por lo tanto 7 minutos (es la diferencia entre la hora a la que podría haber aterrizado y la hora que se le asigna finalmente).

Se debe tener en cuenta además en este caso el retraso o espera desde el ETA más temprano disponible para ese vuelo.

Tomando el ejemplo anterior, si el V1 tenía un ETA en la pista 1 dentro de 120 min. pero el ETA para la pista 2 es 117, el tiempo de espera será de 10 minutos y no de 7.

Se pretende obtener de forma automática la mejor asignación posible de vuelos a aterrizar en cada pista, de forma que el **tiempo de espera de los vuelos en su conjunto** sea el menor posible.

## Trabajo del Alumno

### *Implementación*

Se solicita que el alumno facilite una implementación para la solución de la **situación planteada** en la descripción del problema. Se realizará la solución en el lenguaje de programación Java y se podrá apoyar en la librería JCLEC para la mayoría de las funciones necesarias en el ciclo del paradigma de computación evolutiva.

El problema se plantea desde el punto de vista de la **mejor configuración de asignación de vuelos a pistas** a realizar para una situación de partida determinada. Por eso, se facilitan varias situaciones de partida diferentes del mismo problema en ficheros de texto (formato CSV) que incluyen:

- Listado de Vuelos: Cada línea representa el identificador de cada vuelo, su tipo y la lista de ETA a las diferentes pistas del aeropuerto (siempre en el mismo orden).
- Matriz de tiempos de espera: Indica el tiempo de espera que tiene que dejarse entre 2 tipos de vuelo cualquiera. Los tipos de vuelo son generales para todo el problema.

Se debe tener en cuenta que tanto el número de aviones en cada lista, como el número de pistas disponibles en cada aeropuerto son variables en este problema.

### *Informe*

Junto con el código fuente se pide un informe de la solución al problema propuesto en que se incluya:



- **Descripción de la solución:** ¿como se representan los individuos? ¿como se evalúa la calidad de cada posible solución (pseudocódigo/ejemplos)? ¿qué variables adicionales se emplean (descripción de cada una)? ¿Que ventajas/inconvenientes se aprecian sobre la solución elegida?...etc.
- **Análisis de Resultados:**

Como se detalla en el Tema 8.b, la evaluación de un algoritmo evolutivo debe hacerse teniendo en cuenta todas las variables que influyen en su configuración. En concreto, se necesitará estudiar:

- Resultados sobre las pruebas realizadas sobre cada uno de los ficheros de pruebas facilitados.
- Resultados sobre las pruebas al emplear (al menos dos) variantes en cada una de las partes del ciclo de un algoritmo genético: selección, cruce, mutación, reemplazo (por cada una de las configuraciones del punto anterior).
- Resultados sobre las pruebas realizadas al modificar, los parámetros numéricos de funcionamiento los algoritmos para poder incluir una tabla de resultados completa (tamaño de población, número de iteraciones, probabilidades de mutación, cruce...).

Se espera que el alumno realice las pruebas mencionadas a modo de familiarización con las características del problema. De entre estas pruebas, se seleccionarán aquellos resultados que el alumno considere más interesantes para comentar (3 o 4 resultados) incluyendo gráficas o tablas para ilustrar estos comentarios.

- Se deben incluir conclusiones a cerca de los valores obtenidos en cada una de las pruebas comentadas.

Se sugiere revisar las secciones de pruebas de los diferentes problemas en el manual de JCLEC, para tomar como referencia en cuanto a la realización de diferentes pruebas.

## Recursos Necesarios

Se sugiere consultar con el tutorial de la librería JCLEC y el material de la Sesión 06 para conocer cómo se puede emplear esta librería.

El tutorial que se incluye en el software JCLEC se encuentra en:  
<http://jclec.sourceforge.net/data/JCLEC-tutorial.pdf>

Se encuentran disponibles junto con los ficheros fuentes de JCLEC ficheros de ejemplo de soluciones similares al problema planteado. Se encuentran en el proyecto `jclec4-tutorial`.

La documentación en formato JavaDoc del software incluido en la biblioteca se puede localizar en:  
<http://jclec.sourceforge.net/data/jclec4-classification-doc/> y

<http://jclec.sourceforge.net/mediawiki/index.php/Documentation>



## Consideraciones de Entrega

La entrega se realizará en grupos de 1 o 2 alumnos.

La entrega se realizará solo por medio del correspondiente espacio de subida en plataforma UBUVirtual. No se admitirán entregas pasado el plazo establecido.

La nota correspondiente a este ejercicio se añadirá a la parte de “Ejercicios Prácticos”. Se considera como el Ejercicio 4. Se recuerda que se solicitan entregar 4 ejercicios de los 6 planteados para superar la asignatura. La nota de la práctica se distribuye al 50% entre la implementación de la solución y el informe facilitado.

Para la evaluación de la práctica, se deberán facilitar todos los **ficheros fuentes** generados por el alumno (no incluidos en la librería JCLEC) necesarios para la ejecución y prueba de la solución planteada. Esto incluye ficheros Java, configuraciones en XML y cualquier otro fichero adicional. Se sugiere incluir el código Java en el paquete `es.ubu.gii.cne.freighters_1`

Todos los ficheros fuente deberán estar debidamente comentados (los ficheros Java en formato Javadoc). Se debe adjuntar el **informe** redactado en formato PDF, en el que se describirán los aspectos más generales de los ficheros fuente.

El material de la entrega se debe incluir en un solo fichero comprimido (.zip) para su entrega en el espacio correspondiente en UBUVirtual. El nombre del fichero deberá ser el del alumno (o grupo de 2) que lo entregan.