

OPTIMIZACIÓN DE LA ESTRATEGIA EN CARRERAS DE BAJO CONSUMO MEDIANTE ALGORITMOS GENÉTICOS

Francisco Palacios López

Vicente Colomer Romero

Antonio V. Martínez Sanz

Instituto de Diseño y Fabricación

Universitat Politècnica de València

Abstract

This paper deals with the strategy optimization of low fuel consumption races applying genetic algorithms. We study the “stop and go” strategy case, which consist in switching off the engine when drive power is not required, and using it when is most efficient. A genetic algorithm has been developed in Matlab® to determine the optimum start and stop points to achieve the minimal fuel consumption during the race. This algorithm uses a mathematical model of the car implemented in Matlab/Simulink® to estimate the fuel consumption under several race conditions and circuits.

Keywords: *genetic; algorithm; optimization; Shell; Eco-Marathon; race*

Resumen

El presente artículo trata sobre la optimización de la estrategia de carreras de bajo consumo mediante la aplicación de algoritmos genéticos. En él se estudia el caso concreto de la estrategia “stop and go”, que consiste en apagar el motor del vehículo cuando la fuerza motriz no es necesaria, y utilizarlo sólo cuando su uso sea más eficiente. Se ha desarrollado un algoritmo genético en Matlab® para determinar los puntos de arranque y parada óptimos que consigan el mínimo consumo durante la prueba. Este algoritmo utiliza un modelo matemático del vehículo implementado en Matlab/Simulink® para determinar el consumo de combustible bajo diferentes condiciones de la carrera y circuitos.

Palabras clave: *algoritmo; genético; optimización; Shell; Eco-Marathon; carrera*

1. Introducción

El equipo IDF Eco Marathon, constituido por un profesor y varios estudiantes de la Universidad Politécnica de Valencia, participa cada año desde 2006 en la competición Shell Eco-Marathon. A esta competición acuden, año tras año, estudiantes de un amplio rango de niveles: desde estudiantes de secundaria, ciclos formativos, hasta universidades y escuelas de ingeniería. Sea cual sea su nivel, el reto para los estudiantes y sus profesores es siempre el mismo: diseñar un vehículo que cumpla con las normas de la carrera y que consiga el menor consumo de combustible.

Figura 1: Vehículo IDF Eco-Marathon 2011



Uno de los puntos decisivos en la obtención de un buen resultado en la prueba es la aplicación de una estrategia de carrera óptima. Esta estrategia depende principalmente de la correcta elección de los reglajes y el uso que cada equipo hace del sistema motor/transmisión del vehículo. Los dos tipos de estrategias de carrera utilizadas en esta competición dependen en gran medida del motor que utilice el vehículo:

- Estrategia “stop and go”: Es utilizada comúnmente por los vehículos propulsados mediante motor de combustión interna, dado que sólo alcanzan su mayor eficiencia a plena carga (Taylor, 1985). Consiste en encender el motor, haciéndolo funcionar a plena carga, en determinados puntos del circuito donde el uso del mismo es más eficiente, por ejemplo en pendientes de subida y apagarlo en aquéllos donde no sea necesario, por ejemplo en pendientes descendentes.
- Utilización permanente del motor, regulando la intensidad de corriente suministrada: En vehículos propulsados por motor eléctrico, según el estudio de Bernard et al. (2007) ha demostrado ser más eficiente que la estrategia anteriormente expuesta de “stop and go”.

El propulsor usado en nuestro vehículo, el IDF Eco-Marathon 2011, es de combustión interna, lo que nos obliga a utilizar por tanto, una estrategia del tipo “stop and go”. El uso de la misma plantea una serie de cuestiones para los integrantes del equipo:

- ¿Cuál será el compromiso ideal entre el número de arrancadas a utilizar en cada vuelta y su duración?
- ¿Qué puntos serán los óptimos para ejecutar las arrancadas?
- ¿Qué rango de velocidad será el óptimo?

El problema de optimización que estas cuestiones plantean, no tiene una solución directa obtenible mediante resolución analítica. Se pueden plantear una infinidad de combinaciones posibles para las distintas variables.

2. Objetivos

El enfoque que presentamos en este trabajo para la resolución del problema, se basa en el desarrollo de un algoritmo que optimizará tanto el número de arrancadas como los puntos kilométricos de aplicación de cara a minimizar el consumo durante la prueba. Este algoritmo, utiliza el simulador desarrollado previamente en Matlab/Simulink® como función para verificar la bondad de las estrategias generadas.

La validación o no de una estrategia concreta dependerá de los siguientes factores:

- Serán invalidadas aquellas estrategias que produzcan una simulación con una velocidad media en el circuito menor de 30 km/h. El cumplimiento de condición es indispensable ya que así está marcada por la dirección de la carrera.
- Si diferentes estrategias cumplen la primera condición, las mejores serán aquellas que hayan obtenido menor consumo en la simulación.

3. Enfoque del problema mediante AG.

El algoritmo utilizado en este simulador, es del tipo algoritmo genético (AG), cuyo funcionamiento está inspirado en la teoría de la evolución de las especies, que enuncia que a lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin.

Por analogía con este principio, un AG genera aleatoriamente como entrada un conjunto de soluciones posibles del problema a resolver y las evalúa mediante la función de adaptación, que mide la aptitud de cada una de ellas a ser la solución óptima del problema. Dependiendo de esta aptitud, a cada solución se le asigna una determinada probabilidad de cruce, que repercutirá en las copias de sus genes que aparecerán en futuras generaciones. De esta forma, se espera que a través de las sucesivas generaciones, la calidad de los individuos o soluciones al problema vayan mejorando paulatinamente, acercándose cada vez más al óptimo global del problema.

Estos algoritmos reúnen características de búsqueda aleatoria con características de búsqueda dirigida que provienen del mecanismo de selección de los individuos más adaptados. La unión de ambas características les permite abordar los problemas de una forma muy particular, ya que tienen capacidad para acceder a cualquier región del espacio de búsqueda, capacidad de la que carecen otros métodos de búsqueda exhaustiva, a la vez que exploran el espacio de soluciones de una forma mucho más eficiente que los métodos puramente aleatorios (Gil, 2006 p. 23)

Aunque el origen de este tipo de algoritmos data de 1975, ha sido principalmente en las dos últimas décadas cuando han sido desarrollados y aplicados con éxito en diversos campos tales como procesamiento de imagen, reconocimiento de patrones, reconocimiento de voz, control, identificación de sistemas, optimización, planificación y programación de tareas, robótica,... (Bäck, 1997)

Para poder abordar cualquier problema mediante el enfoque que proponen los AG, necesitaremos una codificación del mismo. Podemos definir codificación en AG como la acción de representar las variables del problema como genes. De este modo, cada gen representará la codificación de una variable, y la ristra que se forma al unir los distintos genes será el cromosoma, que definirá completamente a cada individuo.

Los dos tipos de codificación más utilizados en AG son la binaria y la real. En las etapas primitivas del estudio se barajaron ambas opciones, descritas con más detenimiento en los siguientes apartados.

3.1 Codificación binaria.

Aunque en este caso concreto, quizá sería más correcto denominarla como “codificación booleana” ya que no se pretende codificar valores reales en binario sino establecer el valor de ciertos genes a “verdadero” y otros a “falso”.

La razón de utilizar esta codificación es la siguiente: El fin de la optimización en este problema no es otro que el obtener en qué puntos del circuito debemos utilizar el motor, y cuáles debemos detener su funcionamiento. Con esta premisa podemos codificar los individuos de la siguiente manera:

- El espacio del circuito quedará dividido en tramos de 10 m, es decir a un circuito de 3300 m de longitud le corresponderán 330 tramos.
- Cada uno de esos tramos será un gen de nuestro individuo que podrá tomar como valor 0 ó 1. La presencia de un 1 indicará que dentro de ese tramo el motor deberá estar encendido y la presencia del 0 indicará que el motor estará apagado.
- De esta forma, tal y como se muestra en la figura 2, el cromosoma del individuo será un vector de ceros y unos.

Figura 2: Codificación binaria del cromosoma

1	1	1	1	0	0	0	0	...	1
---	---	---	---	---	---	---	---	-----	---

Es ésta, una codificación muy directa, y que nos proporciona un valor exacto de puntos de funcionamiento del motor. La ventaja principal de la codificación booleana en este problema, reside en la libertad con la que dotamos al algoritmo de establecer cuantas arrancadas sean necesarias, mientras que sus desventajas principales vienen dadas principalmente por:

1. Necesidad de tratar a los individuos para corregirlos tras las operaciones de cruce y mutación para evitar la presencia de arrancadas o paradas excesivamente cortas, consecuencia de la aparición de 0 ó 1 desagrupados.
2. Gran probabilidad de aparición de individuos defectuosos con grandes lapsos de tiempo en los que el motor quedaría parado, provocando la parada total del vehículo en algún tramo del circuito.

Podríamos pensar que una codificación real de estos puntos kilométricos simplificaría mucho los cálculos debido a la considerable reducción en el tamaño de los vectores que representan a los individuos. La realidad es que, aunque ciertamente su aplicación dejaría el tamaño de la población hasta en un entorno del 15% de la original, los problemas de aplicación de ésta vienen dados por las siguientes razones:

1. Necesidad de establecer previamente el número de arrancadas a utilizar durante la prueba, ya que de él depende la longitud de los cromosomas de individuos, que será el doble de éste, al necesitar una variable para representar el punto de arranque y otra para el punto de parada del motor.
2. Obligatoriedad de tener ordenados en forma creciente todos estos puntos kilométricos para que el simulador pueda ejecutar los arranques de un modo correcto. Este hecho complica seriamente la aplicación de los operadores genéticos de cruce y mutación, ya que requeriría una posterior ordenación de los vectores para que no se dieran paradojas de que el motor necesitara ser arrancado en un punto kilométrico rebasado con anterioridad, o que el punto de parada correspondiente fuera inferior al punto kilométrico de arrancada.

3. Existe un riesgo muy alto de arruinar buenos individuos mediante el operador de cruce, debido al desequilibrio que produciríamos en la longitud de ciertas arrancadas, incurriendo en duraciones muy largas o largos lapsos en los que el motor estaría parado pudiendo provocar la parada total del vehículo.

3.2 Codificación real.

Avanzando en nuestro estudio, una de las variantes de codificación que tomó más fuerza y finalmente fue escogida por las ventajas que presentaba fue la codificación real según velocidades de arranque/parada del motor.

Nuestro vehículo cuenta con sólo una relación de transmisión, fijada previamente a la realización de la prueba. También es conocido que los motores de combustión interna tienen un rango en el cual el consumo específico es mínimo, coincidiendo éste casi con exactitud en el régimen de par máximo. Con estas limitaciones, es lógico pensar que existirá un rango de velocidades en las cuales el consumo del vehículo se minimice, también habrá otros valores de velocidad imposibles de alcanzar para nuestro vehículo o bien valores a evitar como velocidades cercanas a 0. La codificación de los individuos según estos principios será la siguiente:

- Cada gen representará el valor real de una velocidad, ya sea de parada del motor o de arranque.
- El cromosoma del individuo quedará dividido en dos grupos de genes (figura 3). La primera mitad del cromosoma estará formada por los genes que representan las velocidades mínimas en m/s a las cuales deberemos poner el motor en marcha, y la segunda mitad del cromosoma será la correspondiente a las velocidades, que una vez rebasadas, indicarán la parada del motor.

Figura 3: Codificación real del cromosoma

7.5	7.2	6.5	...	7.2	10.1	11.1	10.5	...	11.3
-----	-----	-----	-----	-----	------	------	------	-----	------

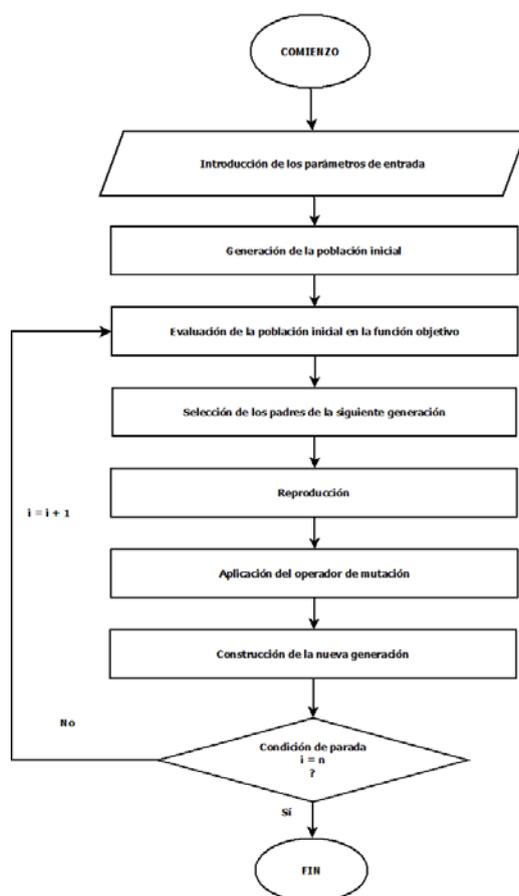
A pesar de que la interpretación de los puntos kilométricos de arrancada/parada del motor con esta codificación no es directa como en los casos expuestos en el apartado anterior, esto es un cálculo que puede realizar fácilmente el computador a la hora de presentarnos los resultados de la optimización. La aplicación de este tipo de codificación presenta las siguientes ventajas:

1. Posibilita la acotación del rango de velocidades durante la prueba a los valores factibles, evitando tanto velocidades imposibles de alcanzar para el vehículo, como velocidades cercanas o iguales a 0 m/s.
2. Facilita en gran medida la generación aleatoria de los individuos, los cuales contarán con una calidad media mucho más alta que con la codificación booleana o real de puntos kilométricos, al evitar individuos con valores absurdos.
3. El número de arrancadas no queda condicionado previamente a la optimización. Si la elección de la longitud del cromosoma del individuo es escogida de manera que posibilite más arrancadas de las ideales para completar una vuelta al trazado, el algoritmo genético, por su construcción, será capaz de encontrar un mínimo global al problema con menos arrancadas de las previamente establecidas. Podemos establecer un número de 12 arrancadas por vuelta al trazado, pero si solamente con las 8 primeras conseguimos completar la vuelta, no se ejecutarán las 4 últimas.

4. Características del AG desarrollado.

El esquema general del algoritmo genético desarrollado para la resolución de nuestro problema puede verse en la figura 4. Para el comienzo de la optimización requiere una serie de parámetros de entrada tales como el tamaño de la población, número de generaciones a evaluar, y rango de valores de las variables, en este caso velocidad mínima admisible y velocidad media mínima de la prueba. Con estos datos se genera la población inicial que será evaluada posteriormente en la función objetivo. Dependiendo del valor de adaptación conseguido por cada individuo, se le asignará una probabilidad para participar en las operaciones de cruce para formar la siguiente generación, tras aplicar también el operador de mutación aleatoriamente sobre algunos individuos. Esta nueva generación será evaluada siempre que i (número de la generación actual) sea menor a n (número de generaciones a evaluar).

Figura 4: Esquema del algoritmo genético desarrollado



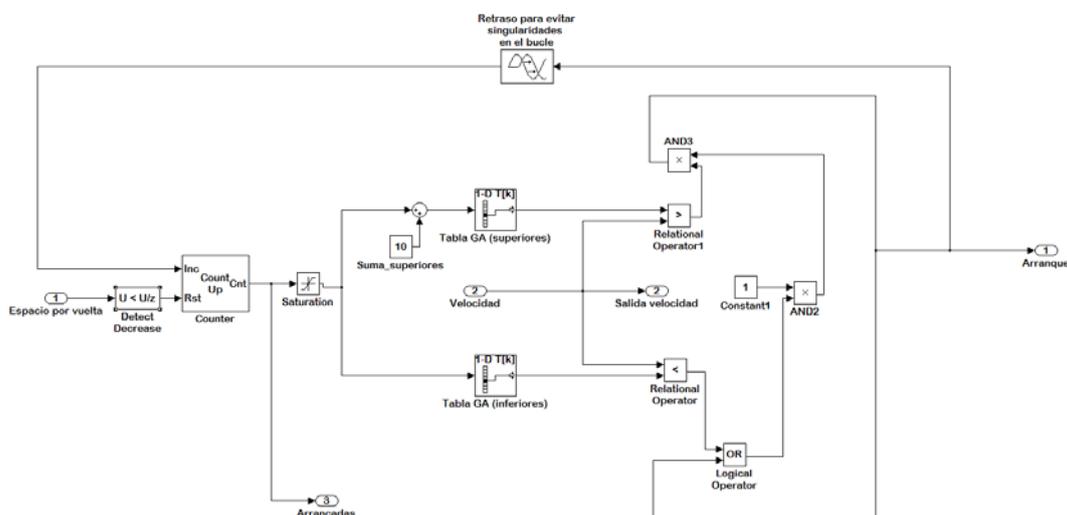
Las características principales de este algoritmo son:

- El operador de selección utilizado es el muestreo estocástico universal.
- Como operadores de reproducción cuenta con el cruce de dos puntos y cruce aritmético.
- Los operadores de mutación serán el reemplazo aleatorio e intercambio repetido.
- Se utiliza un elitismo en la población del 2%.
- Sustitución incondicional de los padres por los individuos hijos.
- El criterio de terminación se conseguirá al alcanzar el número máximo de generaciones previamente fijado por el usuario.

4.1 Función objetivo.

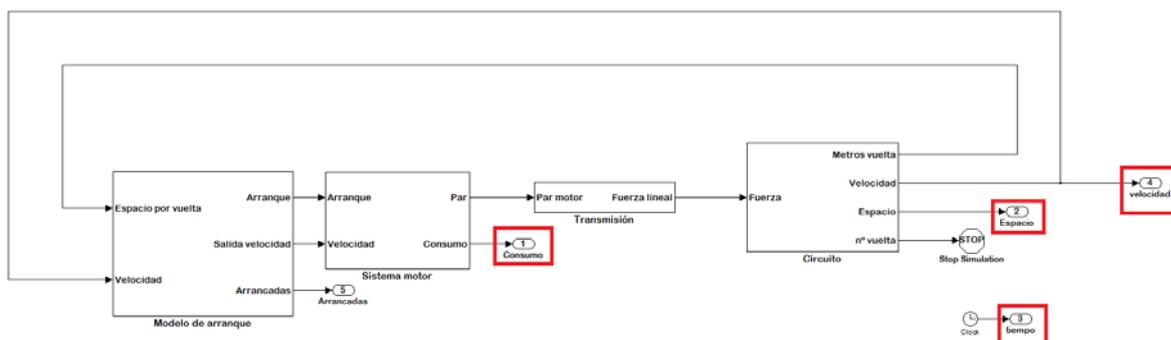
La función objetivo que utilizaremos para evaluar a la población será el kilometraje obtenido por cada individuo mediante su simulación dentro del modelo Simulink® del vehículo. Para llevar a cabo correctamente esta simulación necesitamos introducir el cromosoma de cada individuo dentro de nuestro modelo y lanzar la simulación. Cada individuo es introducido en el bloque llamado “Modelo de arranque” de la figura 5. Para su funcionamiento se vale de diversos bloques comparadores, que verifican si la velocidad instantánea del vehículo está dentro del rango de velocidad mínima y máxima para determinar si el motor necesita ser encendido o no. Las velocidades mínimas y máximas están codificadas en el cromosoma del individuo y son introducidas en cada uno de los bloques llamados “TablaGA”, de forma que la primera mitad del cromosoma establece las velocidades mínimas a las que hay que encender el motor y la segunda mitad establece las velocidades máximas, las cuales, una vez rebasadas indicarán la parada incondicional del motor.

Figura 5: Diagrama Simulink de arranque



El algoritmo genético, previo a la evaluación de cada individuo, cambiará los valores de los vectores correspondientes a los genes de cada individuo en los bloques “TablaGA (inferiores)” y “TablaGA (superiores)” presentes en el modelo de la figura 5 por la primera y segunda mitad del cromosoma del individuo respectivamente. Una vez la simulación del individuo ha sido terminada, se leen las salidas proporcionadas por el modelo general representado en la figura 6:

Figura 6: Diagrama Simulink general



La función objetivo será por tanto maximizar el kilometraje, que es obtenido al dividir el espacio en Km entre el volumen de combustible en l.

$$Kilometraje = \frac{S}{V_{combustible}} \quad (1)$$

La aceptación o no del valor obtenido por cada individuo está sujeto a las siguientes restricciones:

$$V_{media} \geq 30 \text{ km/h} \quad (2)$$

$$V_{final} \geq V_{inicial} \quad (3)$$

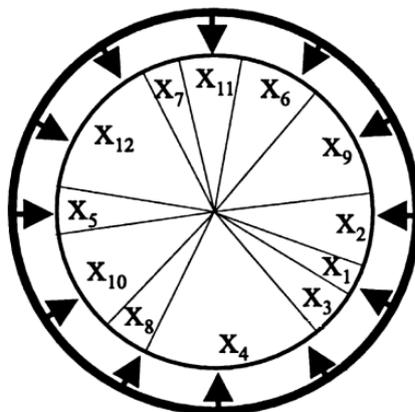
El no cumplimiento de alguna de estas dos restricciones, invalidaría el kilometraje obtenido por el individuo pasando su valor a ser de 100 km/l, un valor lo suficientemente bajo como para descartarlo casi por completo de las operaciones de reproducción mediante el operador de selección que daría una probabilidad muy baja de ser escogidos a estos individuos. Utilizamos por tanto una técnica de penalización para el tratamiento de restricciones en este problema. Dentro de las técnicas de penalización se probó a realizar una penalización parabólica en función de la desviación con respecto a las restricciones obtenida por el individuo, pero hacía muy difícil la convergencia del algoritmo al convertir la función objetivo original en otra del tipo multiobjetivo ponderado.

Las técnicas de reparación han sido descartadas en este problema, porque conllevarían un costo computacional importante, ya que implicarían volver a comprobar varias veces cada individuo no válido hasta que cumpliera ambas restricciones y pudiera ser reintroducido en la población.

4.2 Operador de selección.

Como operador de selección utilizamos el muestreo estocástico universal. Es un procedimiento similar al de selección por ruleta, pero en este caso se genera un sólo número aleatorio y a partir de él se generan los K números que se necesitan, para generar K individuos espaciados de igual forma. Podemos imaginar que estos números generados son como punteros igualmente espaciados que señalan los individuos a seleccionar dentro de la ruleta formada por los sectores proporcionales al valor de adaptación obtenido por cada uno de ellos (figura 7). Este método es mucho más eficiente que el de selección por ruleta.

Figura 7: Muestreo estocástico universal (Blasco, 1998, p. 4)



Los números se calculan mediante la expresión 4 (Araujo, 2009, p. 31) :

$$a_j = \frac{a+j-1}{k} \quad (\forall j = 1, \dots, k) \quad (4)$$

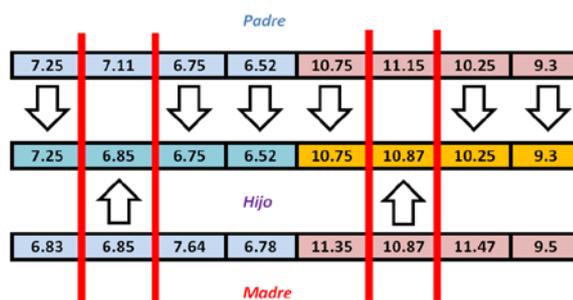
Siendo a el número aleatorio generado, a_j los sucesivos números generados a partir de a y K el número de individuos a seleccionar.

4.3 Operadores de cruce.

Este algoritmo utiliza dos operadores de cruce:

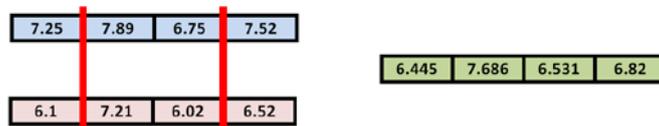
1. Operador de cruce de dos puntos: Se trata de la generalización del cruce de 1 punto. En lugar de cortar por un único punto los cromosomas de los padres como en el caso anterior se realizan dos cortes. Deberá tenerse en cuenta que ninguno de estos puntos de corte coincida con el extremo de los cromosomas para garantizar que se originen tres segmentos. Para generar la descendencia se escoge el segmento central de uno de los padres y los segmentos laterales del otro padre. La particularidad de la aplicación de este operador a nuestro problema es que se aplicarán de igual manera a cada mitad del cromosoma, realizando la transferencia genética de arrancadas completas de padres a hijos en lugar de alguna de las velocidades máximas o mínimas (figura 8).

Figura 8: Cruce de dos puntos



2. Operador de cruce aritmético de dos puntos: En este caso utilizamos un sistema similar al anterior, aunque los hijos ahora, en lugar de heredar genes completos de sus padres, éstos se obtienen como consecuencia de la suma de los mismos multiplicados por un número aleatorio entre 0 y 1. En la figura 9 estudiamos el caso para $\alpha = 0.3$.

Figura 9: Cruce aritmético de dos puntos

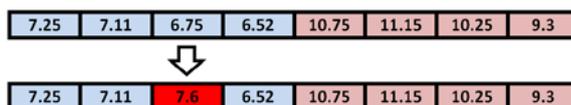


4.4 Operadores de mutación.

El operador de mutación, tendrá una tasa de aplicación del 10%. El funcionamiento es el siguiente: una vez los individuos han sido seleccionados y cruzados, se escoge aleatoriamente el 10% de la población. El AG desarrollado incluye dos tipos de mutación que se escogen de manera aleatoria al finalizar la reproducción. Cada uno tiene la misma probabilidad de ser escogido que el otro.

1. Mutación por reemplazo aleatorio: De cada individuo escogido se selecciona uno de sus genes y se sustituye por un número aleatorio dentro del rango establecido (figura 10):

Figura 10: Mutación por reemplazo aleatorio



2. Mutación por intercambio repetido: En este caso se escogen dos genes correspondientes del individuo y se intercambian sus posiciones. Hay que tener en cuenta que el intercambio afectará a los pares de velocidades que forman una arrancada. Se intercambiarán por igual velocidades mínimas y máximas (figura 11):

Figura 11: Mutación por intercambio repetido



4.5 Reemplazo de la población.

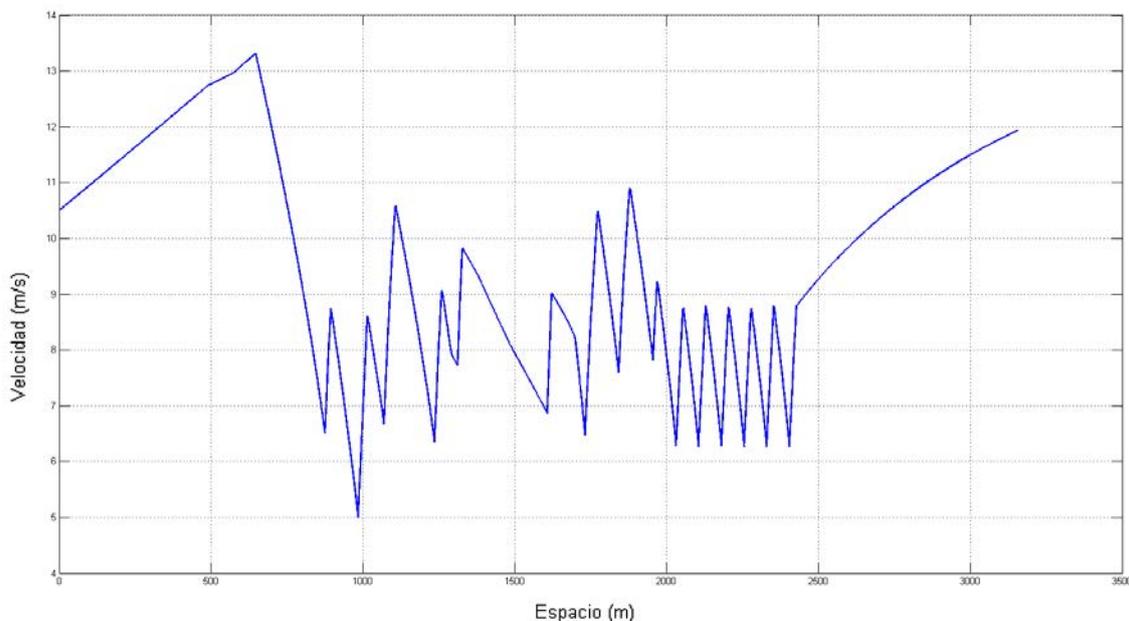
El algoritmo genético que hemos diseñado es un AG con estado estacionario lo que implica que ciertos individuos de la población pueden pasar a la siguiente sin sufrir cambios y otra parte de la población es renovada y sustituida por la anterior. Para el reemplazo tendremos en cuenta:

- La élite de la población será del 2%: Estará formada por los mejores individuos que pasaran duplicados a la siguiente generación para garantizar la permanencia de los mejores genes encontrados.
- Los hijos sustituirán a sus padres aunque tengan un valor de adaptación menor que el de ellos: De esta forma garantizaremos la diversidad en la población.
- El 20% de la población será renovada por individuos creados aleatoriamente: Así garantizamos una vez más la diversidad en la población, al mismo tiempo que continuamos con la búsqueda aleatoria por el espacio de soluciones, un factor que no debe faltar en ningún AG. Los individuos sustituidos por los nuevos serán los que obtuvieron peores valores de adaptación en la generación anterior.

5. Resultados.

Durante unas jornadas de pruebas que nuestro equipo realizó en el mes de Mayo del 2011 en el circuito Ricardo Tormo de Cheste, tuvimos ocasión de establecer las primeras comparativas entre el simulador y la realidad. Las primeras pruebas se encaminaron a verificar la precisión del simulador sin utilizar la optimización. El mejor kilometraje obtenido durante las mismas se estableció en 846 km/l de gasolina. La estrategia seguida para la consecución de esta marca fue la mostrada en la figura 12.

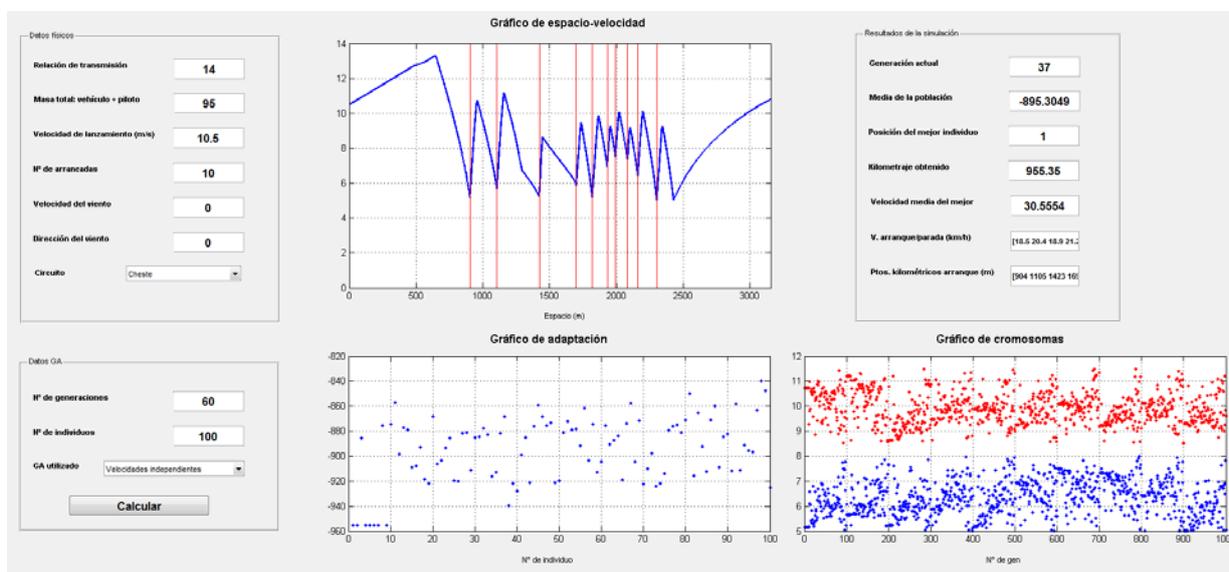
Figura 12: Estrategia seguida sin optimización



Mediante la ayuda del algoritmo genético, tal y como podemos apreciar en la figura 13, pudimos encontrar otra estrategia diferente que arrojó un kilometraje estimado de 955 km/l., que en teoría es un 12.9 % mayor que el obtenido con la estrategia de la figura 12. La ejecución por parte del piloto de esta estrategia óptima encontrada por el algoritmo genético demostró obtener un kilometraje real de 976 km/l que viene a demostrar la eficacia de la optimización en este caso. Una vez más nuestro simulador peca de obtener unas predicciones algo pesimistas con respecto a las obtenidas en pruebas reales, lo que indica que será necesario un reajuste de sus parámetros cuando tengamos disponibles más datos procedentes de telemetrías en diferentes trazados.

Comparando ambas estrategias, y a la vista de los resultados obtenidos, se puede concluir que para este circuito es preferible utilizar un número de 10 arrancadas (líneas verticales rojas en figura 13) en lugar de las 15 arrancadas utilizadas en la estrategia de la figura 12. También se puede apreciar una cierta tendencia a utilizar arrancadas con una mayor diferencia entre velocidad de encendido y parada, en tramos llanos, mientras que en las pendientes de subida el optimizador parece indicar como ideal utilizar un mayor número de arrancadas más cortas, estableciéndose la velocidad mínima de encendido del motor 2 m/s por encima de lo que se hacía en los tramos llanos.

Figura 13: Optimización en el circuito de Cheste



6. Conclusiones.

Durante el desarrollo del algoritmo se probaron varias codificaciones de los individuos. Podemos concluir que la codificación según velocidades mínimas y máximas fue la escogida presentando las siguientes ventajas principales con respecto a la anterior:

- Imposibilidad de que el vehículo quede parado durante la simulación: Al tener siempre fijada una velocidad mínima según el rango establecido, ésta nunca podrá ser rebasada inferiormente.
- Mayor robustez para evitar arrancadas en pendientes descendentes: Si la pendiente es lo suficientemente fuerte como para hacer acelerar el vehículo, durante ella nunca se viajará a velocidad menor que la mínima.
- Relativa independencia del nº de arrancadas con respecto al fijado al iniciar la simulación: Podemos establecer por ejemplo un nº de 12 arrancadas, pero el vehículo, al cambiar de una a otra sólo por el criterio de nº de veces que se rebasen inferiormente las velocidades mínimas, si durante la simulación sólo se ha bajado de las mínimas 10 veces, el vehículo sólo utilizará 10 arrancadas de las 12 previstas si con ello obtiene un kilometraje mayor.

También los operadores de cruce fueron objeto de diversos cambios, concluyendo que la combinación de los expuestos en el apartado 4.3 fueron los que lograron converger a un valor de kilometraje más alto, evitando la caída en mínimos locales. Algunos de los mecanismos de reproducción que se desecharon son:

- Cruce monopunto: Debido a la complejidad del problema, el cruce monopunto, al cambiar de una vez prácticamente la mitad del cromosoma del individuo, en la mayoría de los casos conseguía desequilibrarlo, bien reduciendo mucho su kilometraje, o bien haciendo que violara alguna de las restricciones, siendo descartado de la evolución.

- Cruce uniforme: Este operador de cruce presenta un problema parecido al anterior y es que normalmente cambia muchos valores de arrancadas de una sola vez en los hijos, lo que provoca, que si los padres eran individuos que estaban muy cerca del óptimo, den lugar a hijos con mucho peor valor de adaptación.

Por ello finalmente se puede concluir con respecto a los operadores de cruce para este problema que los que mejor funcionamiento han obtenido han sido aquellos en los que alguno de los hijos hereda la mayor parte del cromosoma de uno de los padres y otra pequeña porción del otro progenitor. Aparte de esto, han demostrado mayor eficacia los cruces cuando para ello se intercambian conjuntamente velocidad mínima y máxima de cada arrancada, en lugar intercambiar independientemente sólo las máximas o las mínimas. Como medida para evitar la falta de diversidad, la alternancia cada 10 generaciones de operador de reproducción demostró su éxito.

7. Referencias

- Araujo L., & Cervigón C. (2009). *Algoritmos evolutivos. Un enfoque práctico* (pág. 31). Editorial Ra-Ma, Paracuellos de Jarama.
- Bäck T., Hammel U., Hans-Paul, & Schwefel (1997). *Evolutionary computation: comments on the history and current state*. IEEE Trans. Evolutionary Comput. 1 3–17.
- Bernard J., Guzella L., Isler D., Kobler P., Kolb F., Onder C.H., Santin J.J., & Weidmann N. (2007). *The world's most efficient vehicle. Design and development of PAC Car II* (págs. 315, 324). Vdf Hochschulverlag AG and der ETH Zürich.
- Blasco X., Senent J. S., & Martínez M. (1998). *Generalized predictive control using genetic algorithms (GAGPC)* (pág. 4). Engineering Applications of Artificial Intelligence 11 (1998) 355-367. Pergamon
- Gil N. (2006). *Algoritmos genéticos* (pág.23). Universidad Nacional de Colombia. Escuela de Estadística, Medellín.
- Taylor C. F. (1985). *The internal combustion engine in theory and practice vol. 1* (2ª ed. pág. 131). The M.I.T. Press, Cambridge Massachusetts.

Correspondencia (Para más información contacte con):

Vicente Colomer Romero
Camí de Vera, 46022 Valencia, Spain.
Phone: + 34 96 652 85 75
E-mail: vicoro@mcm.upv.es
URL: <http://www.institutoidf.com>