

# Estrategias evolutivas para la obtención de la curva braquistócrona

Facultad de Ciencias, UNAM - Pablo A. Trinidad - Mayo 2019

## 1. Resumen

El problema de la curva de descenso más rápido plantea encontrar aquel camino en el plano vertical que lleve a un objeto deslizándose o girando sin fricción sobre él, de un punto  $A$  a un punto  $B$  que se encuentra estrictamente a una altura menor a la de  $A$  y con una traslación horizontal, en la menor cantidad de tiempo posible. Aunque la solución analítica a ésta curva es conocida (curva braquistócrona), el objetivo de éste reporte es mostrar los resultados obtenidos al aplicar una estrategia evolutiva  $(\mu + \lambda)$  a curvas discretizadas para resolver el problema mencionado anteriormente. El reporte incluye los detalles de representación de cada individuo dentro de la estrategia evolutiva, así como la función de costo que evalúa el desempeño de cada curva. La decisión de discretizar las curvas como colecciones de puntos surge del trabajo realizado por **Boschback** y **Dreckmann**[1] donde prueban su hipótesis de que una curva braquistócrona discretizada aproximada mediante estrategias evolutivas tendrá mejor desempeño que aquella, también discretizada, obtenida analíticamente.

## 2. Introducción

El problema de la curva de descenso más rápido fue originalmente propuesto en 1696 por **Johann Bernoulli** aunque sus primeros planteamientos datan de fechas anteriores a esa[2]. El problema propuesto por Bernoulli dice lo siguiente:

Dados 2 puntos  $A$  y  $B$  en un plano vertical ¿cuál es la curva trazada por un punto que es afectado únicamente por la gravedad, el cuál inicia en  $A$  y alcanza  $B$  en la menor cantidad de tiempo?

Y la solución encontrada por **Johann Bernoulli** se basa en uno de los principios de Fermat que enuncia que:

La ruta tomada por un rayo de luz entre dos puntos en el mismo medio siempre es la ruta que toma menor tiempo.

Es así que a **Johann Bernoulli** se le ocurrió que podría modelar el problema de manera diferente: en lugar de pensar en él como un punto deslizándose a través de una curva, podría pensar en el problema como un rayo de luz viajando a través de diferentes medios con múltiples índices de refracción, implicando que el rayo de luz iría a diferentes velocidades conforme viajaba a través de los diferentes medios. Usando éste modelo Bernoulli notó que existía una relación constante entre la velocidad del rayo de luz y la distancia relativa a la altura del punto de origen  $A$  (Ley de Snell) y que podría obtener una solución final aproximando el problema a infinitas capas de diferentes materiales. Finalmente, J Bernoulli reconoció la relación de la ley de Snell con la ecuación diferencial de la curva de un Cicloide, encontrando así la solución analítica.

De la analogía que Johan Bernoulli encuentra surge la inspiración de plantear el problema como una secuencia discreta de diferentes segmentos de recta en las cuales un punto se deslizaría con diferentes velocidades. La inspiración fue posteriormente reforzada por el trabajo realizado por **Boschback** y **Dreckmann** donde demuestran por simulación que dada una definición discreta de una curva **braquistócrona aproximada (AB)**, las soluciones encontradas por una estrategia evolutiva resultaban más eficientes que la aproximación analítica. Finalmente, y aunque la representación también era discretizada, se decidió no utilizar la técnica propuesta por **van Koot K.A.A.M. (KK)**[3] debido a que su solución dependía de la propuesta inicial de funciones a optimizar junto con análisis previo de dichas funciones alrededor del punto  $A$ .

A continuación se muestra la similitud entre la función de tiempo dada por el análisis analítico del problema:

$$t = \int_{x_A}^{x_B} \sqrt{\frac{1 + \left(\frac{dy}{dx}\right)^2}{2gy}} dx, \quad (1)$$

y la función de tiempo dada por la solución discretizada en  $n$  puntos:

$$t = \sum_{i=0}^{n-1} \sqrt{\frac{1 + \left(\frac{y_{i+1} - y_i}{\Delta x}\right)^2}{g(y_{i+1} + y_i)}} \Delta x \quad (2)$$

Debido al interés personal de poder visualizar como se comporta una estrategia evolutiva se optó por utilizar la representación de la curva braquistócrona aproximada (AB). Dicho lo anterior, podemos definir AB como:

**Definición 2.1.** Approximated Brachistochrone (AB) Sea  $X$  la distancia horizontal entre los puntos  $A$  y  $B$  dividido en  $n$  distancias iguales tal que:

$$x_0 = 0, x_1 = \frac{X}{n}, x_2 = \frac{2X}{n}, \dots, x_n = \frac{nX}{n} = X \quad (3)$$

La curva braquistócrona aproximada (AB) para dos puntos  $A$  y  $B$  y un número  $n \in \mathbb{N}, n \geq 1$ , está definida como el conjunto de  $n - 1$  puntos con una altura aproximada  $y_i$  en cada punto  $x_i$  para  $i \in \{1, \dots, n - 1\}$   $P_i(x_i, y_i)$ , con la posición de los puntos  $y_0 := A_{vert}$  y  $y_n = 0$ . Finalmente definimos (AB) aproximando la curva generada por la unión mediante líneas rectas de los puntos vecinos.

Esta definición será utilizada posteriormente para representar y evaluar las soluciones que la estrategia evolutiva vaya generando.

### 3. Estrategia evolutiva $(\mu + \lambda)$

El algoritmo utilizado para la solución de problema es la estrategia evolutiva  $(\mu + \lambda)$ , ES- $(\mu + \lambda)$ . En éste caso particular se utilizó una variable de control  $\sigma$  inicializada en 1.0 por cada individuo que formará parte de su cromosoma y consecuentemente será modificada durante cada iteración del ES- $(\mu + \lambda)$ .

Los criterios de paro utilizados fueron un máximo de iteraciones  $I$  y un valor  $\epsilon$  que representa la precisión mínima que algoritmo aceptará como diferencia entre el valor promedio del fitness de la población anterior y el fitness promedio de la población actual.

El algoritmo comienza con la generación aleatoria de individuos y tras cada iteración sucederá lo siguiente:

1. **Selección de padres:** Si  $\lambda > \mu$ , después de  $\mu$  padres seleccionados secuencialmente se seleccionarán nuevos padres de manera aleatoria, en el caso contrario, se seleccionarán  $\lambda$  hijos de manera secuencial.
2. **Creación de hijos:** Todos los hijos heredan las mismas características que sus padres, en otras palabras, un hijo tiene un sólo padre y es una réplica exacta del padre antes de la mutación.
3. **Mutación de los hijos:** Por cada gen  $g_t$  en los cromosomas se escoge un valor aleatorio  $z$  de una distribución normal utilizando  $\mu = 0$  y el valor de  $\sigma$  del padre, y el gen resultante  $g_{t+1}$  resulta en  $g_{t+1} = g_t + z$  siempre y cuando no exceda el espacio de búsqueda del algoritmo (definido en la siguiente sección). Al finalizar la mutación de los genes el valor  $\sigma$  es alterado utilizando la expresión

$$\sigma_{t+1} = \sigma_t \exp\left(\frac{z_1^*}{\sqrt{2N}} + \frac{z_2^*}{\sqrt{2\sqrt{N}}}\right) \quad (4)$$

Donde  $N$  es el número de cromosomas y  $z_1^*$  y  $z_2^*$  son valores aleatorios extraídos de una distribución normal con  $\mu = 0, \sigma = 1$ .

4. **Selección de individuos:** Los  $\mu$  individuos con mejor fitness son escogidos.

### 3.1. Codificación

La codificación de cada individuo resulta sencilla debido a la discretización mencionada en la definición de (AB). Para una AB particular en los puntos  $A$  y  $B$  y un  $n$  específico, el cromosoma de un individuo es el vector  $a = \{a_1, a_2, \dots, a_{n-1}, \sigma_a\}$  donde  $\sigma_a$  es la desviación estándar utilizada como parámetro de control y cada  $a_i, i = \{1, \dots, n-1\}$ , representa un valor  $y_i$  de la definición de  $AB$  para  $P_i(x_i, y_i)$ . Es importante mencionar que  $a_i \in [S_i, A_{vert}]$  para un  $S_i$  arbitrario  $\leq 0$ . Los valores  $x_i$  del punto  $P_i$  son los descritos en la **Definición 2.1**.

### 3.2. Función objetivo

La función objetivo es producto de un análisis sencillo de movimiento uniformemente acelerado sobre una recta y es el resultado de la suma de los tiempos calculados en cada segmento de recta, la expresión final resulta en:

$$F(\vec{y}) = \sqrt{\frac{2}{g}} \sum_{i=0}^{n-1} \frac{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}{\sqrt{y_0 - y_i} + \sqrt{y_0 - y_{i+1}}} \quad (5)$$

Donde  $\vec{y}$  es el conjunto de puntos  $P_i(x_i, y_i)$  descrito en la **Definición 2.1**.

Debido a que la expresión  $\sqrt{\frac{2}{g}}$  únicamente escala el resultado final y no nos interesa conocer los tiempo exactos de la curva en la gravedad terrestre, dicho término fue ignorado resultando en:

$$F(\vec{y}) = \sum_{i=0}^{n-1} \frac{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}}{\sqrt{y_0 - y_i} + \sqrt{y_0 - y_{i+1}}} \quad (6)$$

## 4. Resultados

Utilizando los valores propuestos por **Boschback** y **Dreckmann** de  $\mu = 10$  y  $\lambda = 100$  junto con una precisión  $\epsilon = 10^{-4}$ , número máximo de iteraciones  $I = 10^5$ , los puntos  $A = (0, 10)$  y  $B = (30, 0)$  y  $S_i = -15$  se obtuvieron los siguientes resultados:

$n - 1$	generaciones	fitness promedio
10	138	7.2053
30	287	7.3715
50	630	7.3230
100	3090	8.2103

#### Evolución de las curvas:

Notamos que las curvas con  $n \geq 30$  comienzan a estancarse en óptimos locales mientras que para  $n$  pequeñas las curvas convergen relativamente rápido.

## 5. Conclusiones

El algoritmo converge relativamente rápido (en cuestión de segundos) y las curvas generadas son extremadamente similares a aquellas generadas por un cicloide.

### 5.1. Mejoras

**Boschback** y **Dreckmann** proponen la definición analítica de la curva braquistócrona aproximada (CS, Cycloid Solution) de la serie de puntos  $P_i(x_i, y_i)$  para  $i = 0, 1, \dots, n$ . En su trabajo ellos demuestran a través de múltiples simulaciones de Monte Carlo que las soluciones encontradas con ES- $(\mu + \lambda)$  superan en rendimiento a aquellas soluciones obtenidas analíticamente (CS). Me gustaría poder replicar esa demostración.

Referente al trabajo de **van Koot K.A.A.M. (KK)**, debido a que la representación del problema que propone depende de funciones analizadas alrededor del punto  $A$  es posible obtener una serie de puntos de la forma  $P_i(x_i, y_i)$  de su respectiva AB. Por lo tanto, me gustaría mucho poder implementar la representación de van Koot, transformar las soluciones obtenidas al tipo AB y finalmente comparar los resultados obtenidos.

Por último, me quedé con la duda de cómo implementar ES- $(\mu + \lambda)$  si el sistema estuviera corriendo en una simulación física, es decir, si existiera una representación gráfica que mostrara las soluciones y un motor físico que pudiera calcular los costos de cada una de las curvas, así como el impacto que dicha simulación tendría sobre el desempeño del algoritmo, aunque es claro, que ya no estaríamos hablando del mismo problema debido a las leyes del motor físico y también tendríamos que pensar en alguna otra forma de representar las soluciones que sea más placentera visualmente.

## 6. Apéndice A: Ejecución del código

El código está compuesto de dos archivos principales:

1. `algorithms.py`: Contiene la implementación del algoritmo ES- $(\mu + \lambda)$  en una clase llamada `ESMuPLambda`. Para iniciar la ejecución del algoritmo se deben proveer los valores iniciales que el constructor solicita y después ejecutar el método `run`. Al finalizar la ejecución, la instancia de la clase contará con la lista `results` como propiedad la cuál está compuesta de tuplas de dos elementos; el primero siendo el fitness promedio de la  $i$ -ésima población y el segundo siendo una lista de  $\mu$  individuos referentes a la  $i$ -ésima generación.
2. `individuals.py`: Contiene la clase `AB` con el la representación de una curva AB. Ésta clase no tiene uso directo, sin embargo, es usada por `ESMuPLambda` durante su ejecución. Ésta cuenta con las propiedades `chromosome` y `fitness`.

Adicionalmente se incluye un notebook de Jupyter con ejemplos de uso de la clase `ESMuPLambda` y dos métodos para generar un conjunto de imagenes para después ser convertidos en GIFs, para lo último se necesitan los folders `output/curves` y `output/mean` y ejecutar el archivo `./build_gifs.sh`.

## Referencias

- [1] Borschbach, M., Dreckmann, W. (2007). *On the Role of an Evolutionary Solution for the Brachistochrone-Problem*. University of Münster, Germany. IEEE - 1-4244-1340-0
- [2] 3Blue1Brown (2006), Canal de YouTube.com *The Brachistochrone, with Steven Strogatz*  
<https://www.youtube.com/watch?v=Cld0p3a43fU>
- [3] van Koot, K.A.A.M. (2014) *Particle swarm optimization algorithms, analysis and computing - Master thesis*  
Eindhoven University of Technology, Netherlands.