Universidad Nacional Autónoma de México

Facultad de Ciencias

Tarea 02:

Convexidad, vecindarios, búsqueda local: Hill Climbing y Búsqueda Tabú

Pablo A. Trinidad Paz - 419004279

Trabajo presentado como parte del curso de **Cómputo Evolutivo** impartido por el profesor **Mario Iván Jaen Márquez**.

Fecha de entrega: **22 de Febrero de 2019**.

1. Teoría

*Código utilizado en Jupyter Notebook ***Theory.ipynb***

1. Sean $f_2, f_2 : \mathbb{R} \to \mathbb{R}$ dadas por

$$f_1(x) = x^2 - 2ex + e^2 - 2,$$
$$f_2(x) = x^6 - 6ex^5 + 15e^2x^4 - 20e^3x^3 + 15e^4x^2 - 6e^5x + e^6 - 6$$

   a) Demuestre que f_1 y f_2 son funciones convexas

   Solución:
   Una función f es convexa si se cumple que:

$$\forall x, y \in Dom(f) \text{ y } \forall a \in [0,1]$$
$$f(ax + (1-a)y) \leq af(x) + (1-a)f(y).$$

   Además, se cumple que si la función es doblemente derivable (y de una sola variable) es convexa en un intervalo sí y solo sí su segunda derivada no es negativa.
   Para $f_1(x)$:

$$f_1'(x) = 2x - 2e$$
$$f_1''(x) = 2$$
$$\Rightarrow f_1''(x) > 0$$
$$\therefore \; f_1 \text{ es convexa} \quad \blacksquare$$

   Para $f_2(x)$:

$$f_2'(x) = 6x^5 - 30ex^4 + 60e^2x^3 - 60e^3x^2 + 30e^4x - 6e^5$$
$$f_2''(x) = 30x^4 - 120ex^3 + 180e^2x^2 - 120e^3x + 30e^4$$
$$f_2''(x) = 30(e - x)^4$$
$$\Rightarrow f_2''(x) > 0$$
$$\therefore \; f_2 \text{ es convexa} \quad \blacksquare$$

   b) Utilice el algoritmo del descenso por gradiente implementado para minimizarlas. Use $x_0 = 0$ como punto inicial y α arbitrario. ¿Qué valores de α hacen más eficiente el algoritmo para cada función?

   Solución:
   Para responder a la pregunta anterior se corrió el descenso por gradiente con múltiples valores de α y se determinó que el valor de α más eficiente sería aquel que lograra encontrar un óptimo en el menor número de iteraciones. Para saber si el valor óptimo obtenido a partir de dicha α era aceptable, se calculó la desviación estándar, promedio y media de los resultados de cada prueba.

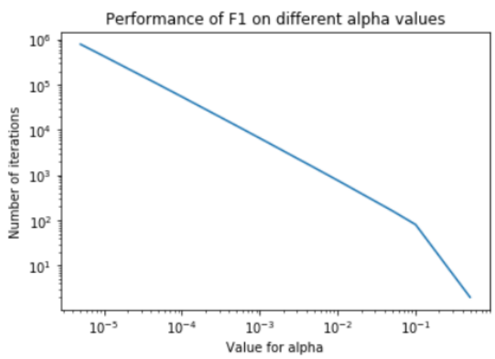
   Para ambas funciones, el gradiente corrió bajo los siguientes parámetros:

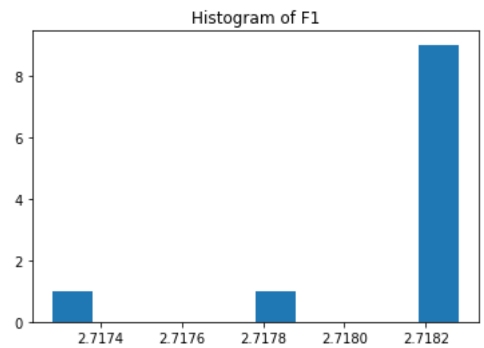
   - Precisión: 1×10^{-8}
   - Número máximo de iteraciones: 10^{10}

Resultados de f_1:

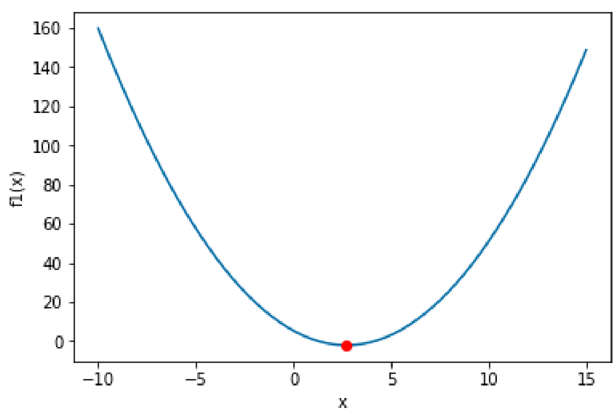
Valores de α para la función f_1

α	Número de iteraciones	Valor mínimo
0.000005	790773	2.71728184
0.00001	430042	2.71778184
0.00005	102100	2.71818185
0.0001	54513	2.71823184
0.0005	12508	2.71827184
0.001	6598	2.71827684
0.005	1476	2.71828085
0.01	769	2.71828134
0.05	164	2.71828174
0.1	81	2.71828179
0.5	2	2.71828183





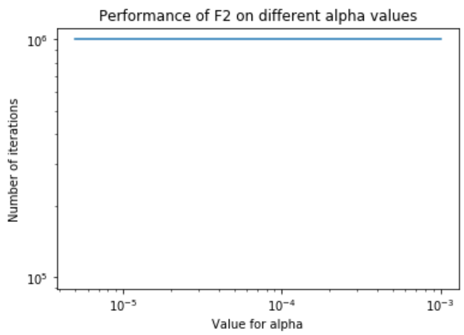
Con una desviación estándar $\sigma \approx 0.000303$, promedio $\mu \approx 2.7181303273$ y media ≈ 2.7182768400 se puede concluir de forma segura que el valor óptimo de α para f_1 es **0.5** y que el mínimo se encuentra en $x = 2.718281828459045$.

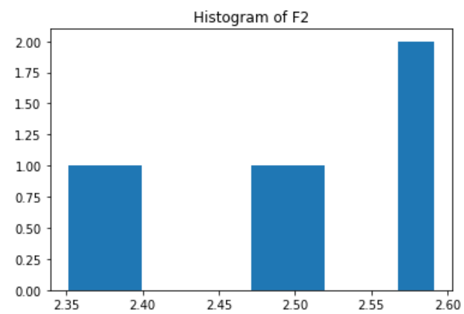


Resultados de f_2:

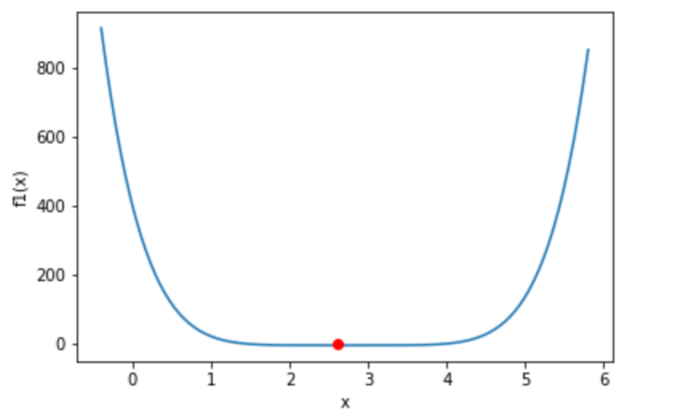
Valores de α para la función f_2

α	**Número de iteraciones**	**Valor mínimo**
5e-06	5266300	2.45496429845281
1e-05	4691757	2.4836925848909828
5e-05	3587912	2.538885692619941
0.0001	3196467	2.5584580445544773
0.0005	2444410	2.59606066860143
0.001	2163480	2.60939515596596





Con una desviación estándar $\sigma \approx 0.0558$, promedio $\mu \approx 2.5402427408$ y media ≈ 2.5486718686 Se puede utilizer **el valor más óptimo de** $\alpha = 0.001$ y su mínimo hallado $x = 2.60939515596596$ ya que se encuentra dentro de $[\sigma - \mu, \sigma + \mu]$.



2. Lea el capítulo 1 del siguiente libro para resolver las preguntas de esta sección:
   Z. Michalewicz et al., How to Solve It: Modern Heuristics. Springer Berlin Heidelberg 2004

   a) Mencione al menos 2 razones por las cuales un problema del mundo real puede no ser resuelto fácilmente. NO USE ninguna de las razones enumeradas en la página 11 del capítulo 1.

   - Algunos problemas son tan complejos que obtener un modelo que contemple todos sus aspectos puede ser muy complicado. Por otro lado, dado un modelo detallado del problema, derivar una solución óptima puede ser extremadamente complicado. Usualmente se reduce a dos opciones: 1. Modelar el problema contemplando todos los aspectos y obtener una solución aproximada usando métodos no convencionales o 2. Aproximar un segundo modelo que imite las características del problema y obtener una solución exacta a ese modelo aproximado por métodos convencionales.

   $$\begin{aligned}1)\ \text{Problem} &\implies \text{Model}_{\text{a}} \implies \text{Solution}_{\text{p}}(\text{Model}_{\text{a}})\\ 2)\ \text{Problem} &\implies \text{Model}_{\text{p}} \implies \text{Solution}_{\text{a}}(\text{Model}_{\text{p}})\end{aligned}$$

   - **Los problemas cambian con el tiempo**. Si derivar un modelo (de cualquier tipo) al problema fue costoso y no contempla que la naturaleza del problema cambió mientras se derivaba, éste puede terminar siendo obsoleto. Además, existe otra complicación: **las limitantes del problema**. Las limitantes, aunque pueden reducir el tamaño del espacio de búsqueda, suelen terminar agregando más complejidad al modelo ya que no todas las posibles soluciones cumplirán necesariamente con las limitantes.

   b) Supongamos que la optimización de una función f requiere de 10 variables de decisión $x_i (i = 1, ..., 10)$, cada una de las cuales está en el dominio $-50 \leq x_i \leq 50$.

   - Si x_i puede tomar solo valores enteros ¿cuál es el tamaño del espacio de búsqueda de este problema?

     Soluución: Existen 101 enteros en el rango $[-50, 50]$, por lo tanto hay 101^{10} combinaciones posibles.

   - Si x_i puede tomar valores reales y se usa una precisión de ocho lugares decimales ¿cuál es el tamaño del espacio de búsqueda del problema? **Soluución:**
     Soluución: Existen 10^8 números decimales entre cada entero (incluyéndolo) y hay 101 enteros disponibles en el rango propuesto, es decir, hay 101×10^8 valores disponibles, por lo tanto hay $(101 \times 10^8)^{10} \approx 1.1 \times 10^{100}$ combinaciones.

3. Investiga a qué se refieren los términos: **exploración** y **explotación** dentro de las fases de un algoritmo evolutivo.

   a) **Exploración** (exploration): Suele ser usado para referirse a buscar en todo el espacio de búsqueda una solución candidata al problema.

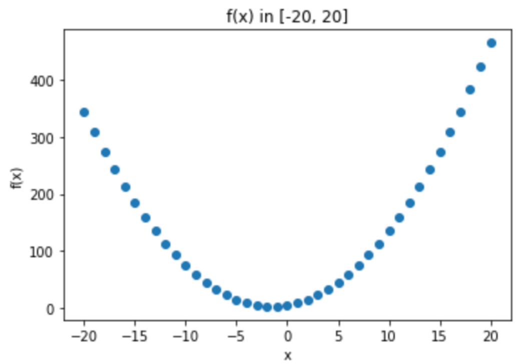
   b) **Explotación** (exploitation): Suele ser usado para referirse a buscar una nueva solución en una región limitada del espacio de búsqueda al rededor de una solución ya existente.

4. ¿Considerarías al algoritmo *hill climbing* un *greedy algorithm*? ¿Por qué?

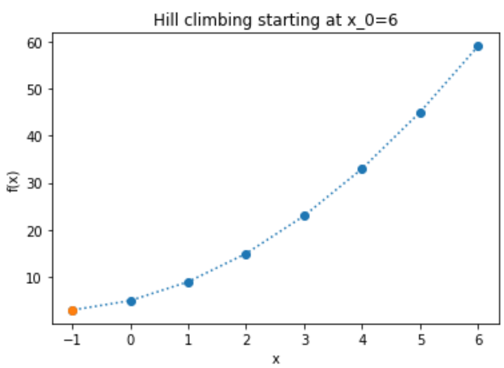
   Solución: Sí. Los algoritmos greedy son conocidos por encontrar soluciones "parcialmente buenas", en otras palabras, óptimos locales, y aunque *hill climbing* pueda encontrar óptimos globales en funciones convexas, para otro tipo de funciones terminará convergiendo en un óptimo local, es decir, las soluciones ya no podrán mejorar dentro de su vecindario local.

5. Considera la función $f(x) = x^2 + 3x + 5$ definida sobre los enteros $\mathbb{Z}$ en el intervalo $[-20, 20]$. Vamos a usar el algoritmo *hill climbing* para encontrar el valor de x que minimiza la función en ese intervalo.

   a) Grafica la función



   b) Considera el vecindario definido por $N(x) = x+1, x-1\ \forall x \in [-19, 19]$. Usa como solución inicial $x_0 = 6$ e indica los valores que toma x_t y $f(x_t)$ en cada iteración t del algoritmo.



t	x_t	$f(x_t)$
1	6	59
2	5	45
3	4	33
4	3	23
5	2	15
6	1	9
7	0	5
8	-1	3

2. Práctica

Solución en Jupyter Notebook ***PSet.ipynb***

1. Con V_1 se econtró $x^* = (0.0, -5.0)$ el cual resulta en $F(x^*) = -54.6875$
2. Con V_∞ se econtró $x^* \approx (5.6, 5.7)$ el cual resulta en $F(x^*) \approx -191.995608$