Simulación de Sistemas
Práctica 7

Búsqueda local

1455175: Ángel Moreno

24 de septiembre de 2018

Resumen

En esta práctica se simula una optimización heurística llamado búsqueda local variando el factor de pasos en el vecindario del heurístico. Se analiza la distribución de dispersión (varianza) de 15 puntos que converjan a un optimo local teniendo de resultados iteraciones mínimas para que la búsqueda en los puntos converjan.

1. Introducción

Se busca maximizar la función

$$g(x,y) = \frac{(x+0.5)^4 - 30x^2 - 20x + (y+0.5)^4 - 30y^2 - 20y}{100}$$

utilizando una optimización heurística sencilla de búsqueda local en el dominio

$$D = \{(x,y) : x \in [-3,3] \wedge y \in [-3,3]\},$$

se detalla una observación haciendo un análisis de cálculo, se calculan las derivadas parciales de la función g

$$\frac{\partial g}{\partial x} = \frac{4(x+0.5)^3 - 60x - 20}{100} \tag{1}$$

$$\frac{\partial g}{\partial y} = \frac{4(y+0.5)^3 - 60y - 20}{100} \tag{2}$$

se utiliza la herramienta matemática Symbolab[2] para obtener las raíces de las parciales 1 y 2 obteniendo $\{-4.4537, -0.33302, 3.2867\}$. El dominio D incluye solo el punto $(-0.33302, -0.33302)$, por tanto la función g tiene un máximo local en dicho dominio, esto significa que dirección del vector gradiente de g sea hacia el punto $(-0.33302, -0.33302)$ en D.

1.1. Tarea

La tarea se trata de maximizar la función bidimensional g, con dominio D, con la misma técnica del ejemplo unidimensional. La posición actual es un par (x,y) y se ocupan dos movimientos aleatorios, Δx y Δy, cuyas combinaciones posibles proveen ocho posiciones de vecinos, de los cuales aquella que logra el mayor valor para g es seleccionado.

2. Simulación

Tomando en cuenta la parte de la introducción de tener solo un máximo local esto evita atascamientos en los puntos, por tanto se implementa la siguiente simulación: definimos 15 puntos aleatorios simultáneamente y se ejecuta la búsqueda local simultánea en cada iteración, se busca la cantidad mínima de iteraciones nesesarias para lograr una convergencia al máximo local en los puntos. Se utiliza la herramienta wolfram alpha[3] para calcular el máximo local de la función g obteniendo el valor $w = 0.0666822$. Para lograr la convergencia de los puntos se utiliza la varianza de los valores de la función evaluada en dichos puntos tomando como media el valor w,

$$v = \sum_{i=1}^{15} \frac{(g_i - w)^2}{15}$$

donde g_i es el valor de la función evaluada en el punto i con $i \in \{1, 2, ..., 15\}$, y se busca la varianza mínima.

En esta simulación se ejecutan pasos de $0.3^5, 0.3^4, 0.3^3, 0.3^2$ y 0.3 con cantidad de iteraciones de $2^5, 2^6$, $2^7, 2^8, 2^9, 2^{10}$ y 2^{11} para 15 puntos. Se implementaron 30 réplicas.

2.1. Resultados

La figura 1 muestra los resultados de las varianzas con distintas iteraciones en el heuristico de búsqueda local.

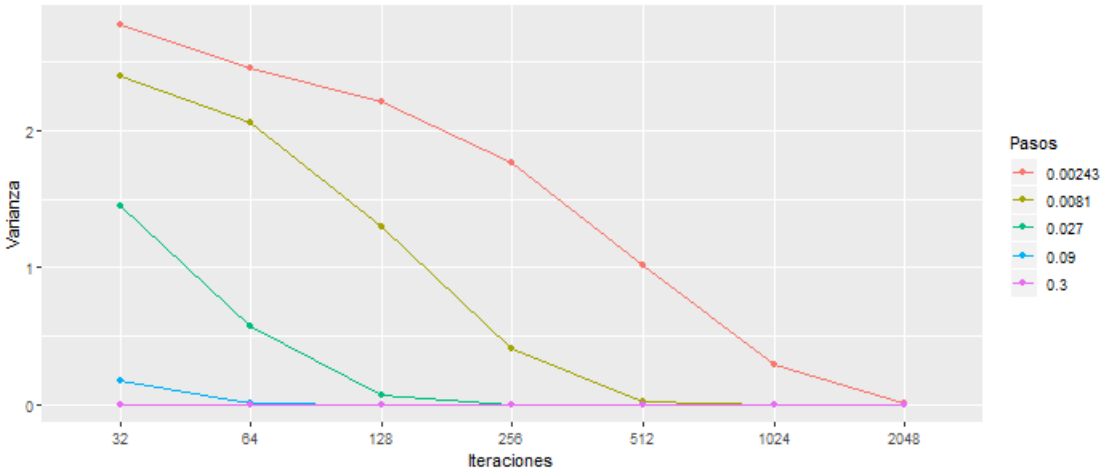

Figura 1: Resultados de 30 réplicas con distintos pasos.

Se observa que el cambio de los pasos en el heuristico afectan los resultados de la varianza de los puntos, esto es, para pasos de 0.00243 en la iteracion 2048 se logra la convergencia de los puntos con un varianza aproximada a 0 pero en cambio para pasos 0.3 es suficiente con un número de 32 iteraciones o talvez menos para lograr la convergencia deseada.

3. Reto 1

El primer reto es crear una visualización animada de cómo proceden 15 réplicas simultáneas de la búsqueda encima de una gráfica de proyección plana.

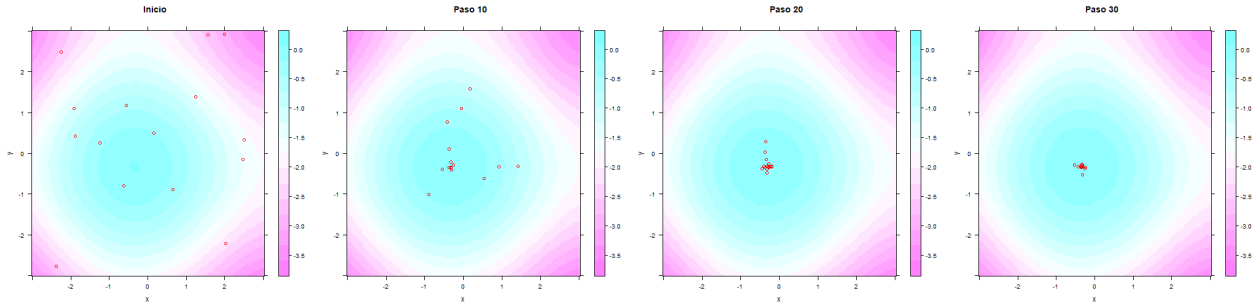

Figura 2: Búsqueda local con pasos de 0.3.

La figura 2 muestra imágenes de la simulación.

Referencias

[1] Schaeffer E. *R paralelo: simulación y análisis de datos, 2018.* `https://elisa.dyndns-web.com/teaching/comp/par/`

[2] Avny M. *Symbolab math solver, 2018.* `https://es.symbolab.com/solver/polynomial-calculator`

[3] *WolframAlpha computational intelligence, 2018.* `http://www.wolframalpha.com`

[4] Valdes E. *Repository of Github, 2017.* `https://github.com/eduardovaldesga/SimulacionSistemas`

[5] Saus L. *Repository of Github, 2018.*
https://github.com/pejli/simulacion