Sinencio Granados Dante 316246019
Sandoval Mendoza Antonio 316075725
Reyes Martínez Antonio, 316184931
Cruz Jiménez Alejandro 316008488

Complejidad Computacional
Semestre 2022-1

Práctica Extra

- Descripción y pseudocódigo de la heurística:

  El *hill climbing* intenta maximizar (o minimizar) una función objetivo $f(x)$, donde x es un vector de valores discretos y/o continuos. En cada iteración, el *hill climbing* ajustará un único elemento en x y determinará si el cambio mejora el valor de $f(x)$. Con *hill climbing*, cualquier cambio que mejore $f(x)$ es aceptado, y el proceso continua hasta que no pueda encontrarse un cambio que mejore el valor de $f(x)$, por lo x se dice entonces que es "óptica localmente".

  Por otro lado, en los espacios de vectores discretos, cada valor posible para x puede ser visualizado como un vértice en un grafo. El *hill climbing* seguirá el grafo de vértice en vértice, siempre incrementando (o disminuyendo) localmente el valor de $f(x)$, hasta alcanzar un máximo local (o un mínimo local) x_m.

  Hill climbing estocástico no examina todos los vecinos antes de decidir cómo moverse. En vez de eso, selecciona un vecino aleatorio, y decide (basado en la cantidad de progreso en ese vecino) si moverse a él o examinar otro.

  Descenso coordinado realiza una búsqueda en línea a lo largo de una dirección de coordenadas a partir del punto actual en cada iteración. Algunas versiones del descenso coordinado eligen aleatoriamente una dirección coordinada diferente en cada iteración.

  Pseudocódigo:

```
(1)  |   Hill CLimbing Algorithm
(2)  |       currentNode = startNode;
(3)  |       loop do
(4)  |           L = NEIGHBORS(currentNode);
(5)  |           nextEval = -INF;
(6)  |           nextNode = NULL;
(7)  |           for all x in L
(8)  |               if(EVAL(x) > nextEval)
(9)  |                   nextNode = x;
(10) |                   nextEval = EVAL(x);
(11) |           if nextEval <= EVAL(currentNode)
(12) |               return currentNode;
(13) |           currentNode = nextNode;
```

Sinencio Granados Dante 316246019
Sandoval Mendoza Antonio 316075725
Reyes Martínez Antonio, 316184931
Cruz Jiménez Alejandro 316008488

Complejidad Computacional

Práctica Extra

Semestre 2022-1

- Descripción general de los ejemplares (tamaño, referencias - en caso de que se hayan obtenido o generado de alguna forma en específico, etc), de los recorridos encontrados:

  En el caso de nuestro algoritmo, decidimos usar vectores como nuestros ejemplares, ya que podemos construirlos con valores aleatorios,esto nos servirá para que cada ejemplar sea diferente el uno del otro.

  Donde el vector se construye a partir de un tamaño dado y contiene números los cuales utilizaremos para obtener el máximo costo posible.

- Conclusiones, haciendo énfasis en la importancia del valor del parámetro T:

  El valor T en nuestro algoritmo funciona para calcular la eficiencia de búsqueda de los costos. Aunque, no es ocupado en el algoritmo tal cual.

  Conclusión: Intentamos realizar la mayor parte de los ejercicios de esta práctica. Estuvimos trabajando pero nos surgieron algunas dudas y problemas que nos fueron solucionando los ayudantes, pero aun así el tiempo nos impidió poder terminarla por completo. Aun así enviamos el trabajó que realizamos ya que si teníamos la intención de entregarla.

- Referencias consultadas.

  - Colaboradores de Wikipedia. (2019, 6 octubre). Algoritmo hill climbing. Wikipedia, la enciclopedia libre. Recuperado 12 de enero de 2022, de https://es.wikipedia.org/wiki/Algoritmo_hill_climbing
  - GitHub. (s. f.). Build software better, together. Recuperado 12 de enero de 2022, de https://github.com/clever-algorithms/CleverAlgorithms
  - Global, Z. (s. f.). Brutalk - Escalada estocástica en Python desde cero. Brutalk. Recuperado 12 de enero de 2022, de https://www.brutalk.com/da/brutalk-blog/view/escalada-estocastica-en-python-desde-cero-6046b211cfcc6