Laboratorio de Implementación

La **Oncilla (*Leopardus tigrinus*)**, también conocida como tigrillo, es uno de los felinos más pequeños y vulnerables de Colombia. Esta especie es clave para el equilibrio de los ecosistemas de páramo y bosques de niebla, donde cumple un papel importante en el control de poblaciones de pequeños mamíferos y aves. Actualmente, según la Unión Internacional para la Conservación de la Naturaleza (UICN), la oncilla está clasificada en la categoría **“vulnerable”**, debido a la pérdida y fragmentación de su hábitat, la expansión agrícola y minera, así como al tráfico ilegal de individuos como mascotas. Por esta razón, diferentes instituciones académicas y organizaciones de conservación en Colombia desarrollan investigaciones en áreas estratégicas de la cordillera de los Andes con el fin de comprender mejor las amenazas que enfrenta la oncilla y promover acciones para su protección.

El equipo de investigación ha definido un índice que establece, para una coordenada dentro del área de estudio, el nivel de perturbación antrópica que enfrenta el hábitat de la oncilla. Este índice considera factores como la fragmentación del bosque, la cercanía a áreas agrícolas o mineras y la presencia de actividades humanas en el territorio. El índice se define a continuación y puede tomar valores negativos, representando zonas de alta degradación, o positivos, indicando áreas con condiciones favorables para la permanencia de la especie.

El equipo que llevará a cabo la investigación sobre la oncilla ha solicitado apoyo para determinar el mejor lugar para establecer su campamento base dentro del área de estudio. La zona en la que los investigadores están interesados se encuentra delimitada por la región mostrada en la Figura 1.1, la cual corresponde a un sector estratégico de páramo y bosque de niebla donde se han registrado individuos de la especie.

Gráfico

El contenido generado por IA puede ser incorrecto.

**Figura 1.1.** Región de interés.

Los investigadores están interesados en determinar la coordenada dentro del área de estudio que tenga el índice más ***alto*** para allí ubicar el campamento. Para encontrar dicho lugar, usted ayudará a los investigadores implementando:

1. Un método de búsqueda local.
2. Un método de descenso por gradiente.
3. Formule matemáticamente y de forma explícita el problema de optimización planteado. Defina claramente las variables de decisión, la función objetivo y las restricciones. Tenga en cuenta que en este modelo particular **NO necesariamente** se cumplen todos los supuestos de optimización lineal vistos en el curso.

*Búsqueda Local*

Para esto, usted se ha propuesto usar un radar cuadrado de búsqueda. En este radar, a partir de una coordenada actual , se exploran las ocho (8) coordenadas más cercanas a una distancia de unidades, tal como se muestra en la Figura 1.2. Tenga en cuenta que explorar una coordenada significa verificar si está dentro de los límites del parque y calcular su respectivo índice de densidad de personas. Para desarrollar este método de búsqueda local, siga los pasos a continuación.

Diagram, bubble chart

Description automatically generated with medium confidence

**Figura 1.2.** Radar de búsqueda

Para los incisos **b**, **c**, **d** y **e** deberá implementar funciones específicas en Python. Para esto, debe utilizar el espacio indicado en la plantilla. Tenga presente el tipo (entero, booleano, listas, diccionarios, etc.) de los parámetros de entrada y de las salidas de cada una de las funciones.

1. Implemente una función en Python llamada evaluar\_indice que reciba como parámetro una tupla con una coordenada y retorne el valor del índice para dicha coordenada.
2. Implemente una función en Python llamada generar\_radar que reciba dos parámetros: una tupla con las coordenadas actuales , y la longitud . La función debe retornar las 8 coordenadas vecinas en una lista.
3. Implemente una función en Python llamada evaluar\_factibilidad que reciba como parámetro una tupla representando una coordenada y retorne **True** en caso de que dicha coordenada se encuentre dentro de la región de estudio (cumpla todas las restricciones) y **False** de lo contrario.
4. Implemente una función en Python llamada encontrar\_mejor\_coor que reciba como parámetro un diccionario donde las llaves son coordenadas y los valores el índicecorrespondiente. Esta función debe retornar una tupla con la coordenada de ***MAYOR*** índice y, además, el valor de dicho índice. En esta función, debe evaluar si el diccionario que entra por parámetro está vacío, en cuyo caso deberá retornar **False.**
5. A partir de estas funciones, se puede implementar un método de búsqueda local. Dicha implementación se encuentra en el archivo entregado y corresponde a la función busqueda\_local. Modifíquela si lo considera necesario: además, explique en sus propias palabras, ¿qué hace la función busqueda\_local?
6. Utilizando la función busqueda\_local, encuentre la mejor ubicación para el campamento del equipo de investigación. Defina tanto como Use código ‘plot\_’ que encuentra en la plantilla para hacer un gráfico de calor (colores oscuros indican un índice mayor y colores más claros indican un índice menor) con la trayectoria seguida durante la búsqueda local.

*Descenso por Gradiente*

1. Implemente una función en Python llamada grad\_evaluar\_phi que reciba como parámetro un diccionario donde las llaves son coordenadas y retorne el gradiente en dicho punto. A continuación, cuenta con las derivadas parciales y el respectivo gradiente.

**Función**

**φ(x, y) = [ -100·(x² + y − 60)² − 100·(x + y² − 25)² ] / 100**

**Simplificando:**

**φ(x, y) = −(x² + y − 60)² − (x + y² − 25)²**

**Derivada parcial respecto de x**

**∂/∂x [ −(x² + y − 60)² ]**

**= −2·(x² + y − 60) · ∂/∂x(x² + y − 60)**

**= −2·(x² + y − 60) · (2x)**

**= −4x·(x² + y − 60)**

**∂/∂x [ −(x + y² − 25)² ]**

**= −2·(x + y² − 25) · ∂/∂x(x + y² − 25)**

**= −2·(x + y² − 25) · (1)**

**= −2·(x + y² − 25)**

**∂φ/∂x = −4x·(x² + y − 60) − 2·(x + y² − 25)**

**Derivada parcial respecto de y**

**∂/∂y [ −(x² + y − 60)² ]**

**= −2·(x² + y − 60) · ∂/∂y(x² + y − 60)**

**= −2·(x² + y − 60) · (1)**

**= −2·(x² + y − 60)**

**∂/∂y [ −(x + y² − 25)² ]**

**= −2·(x + y² − 25) · ∂/∂y(x + y² − 25)**

**= −2·(x + y² − 25) · (2y)**

**= −4y·(x + y² − 25)**

**∂φ/∂y = −2·(x² + y − 60) − 4y·(x + y² − 25)**

**Gradiente**

**∇φ(x, y) = ( ∂φ/∂x , ∂φ/∂y )**

**= ( −4x·(x² + y − 60) − 2·(x + y² − 25) , −2·(x² + y − 60) − 4y·(x + y² − 25) )**

1. Implemente en Python la función **gradient\_descent**, que reciba como parámetros una tupla inicial (x, y), la función objetivo, la función que calcula el gradiente, la tasa de aprendizaje (**alpha**), el número máximo de iteraciones y un valor de tolerancia (**epsilon**); asegúrese de que cada nuevo punto permanezca dentro de la región factible y retorne el recorrido completo de puntos generados durante el proceso, mostrando la trayectoria seguida hasta aproximarse al ***máximo*** de la función.
2. Utilizando la función **gradient\_descent**, encuentre la mejor ubicación para el campamento del equipo de investigación. Defina tanto como Use código ‘plot\_’ que encuentra en la plantilla para hacer un gráfico de calor (colores oscuros indican un índice mayor y colores más claros indican un índice menor) con la trayectoria seguida durante la búsqueda local.