

Instrucciones y Reglas

Fecha límite de entrega: viernes 20 de septiembre, 12 m. (medio día) Entrega por Bloque Neón.

- La tarea debe ser realizada en grupos de exactamente 2 personas.
- La solución de cada uno de los problemas que se enuncian a continuación debe contener mínimo:
 - Formulación matemática rigurosa (conjuntos, parámetros, variables de decisión, función objetivo, restricciones).
 - Síntesis de resultados.
 - Conclusiones.
 - Anexos (modelos en Python-Gurobi, salidas y toda la información de soporte que sustente su trabajo) enviados por un solo integrante del grupo al enlace de Bloque Neón. De ser necesario, debe haber una implementación diferente por literal.
- El reporte debe ser auto contenido. Esto quiere decir, que debe analizar reportar los resultados obtenidos de manera clara, concisa y precisa. No debe exceder las 14 páginas.
- Se debe utilizar el formato de entrega de la tarea que se encuentra en Bloque Neón en la primera página de su informe. Si el formato no es usado o se llena incorrectamente, la tarea se calificará sobre 90 puntos.
- Envíe por Bloque Neón su informe en formato PDF y con nombre el *login* del estudiante que realiza el envío (e.g., lj.sanchezd.pdf). Por aparte, envíe todos sus archivos de soporte (modelos de Python-Gurobi, archivos de Excel, etc.) comprimidos en un solo archivo *.zip. Asegúrese que los archivos sean colgados por sólo uno de los integrantes. No se aceptarán archivos enviados por otro medio como correo electrónico.
- Si el informe de la tarea y los archivos de soporte no se entregan en la fecha y hora asignadas la nota de la tarea será 0.0.
- Todo código en Python-Gurobi o cualquier otro lenguaje de programación debe estar debidamente comentado, de lo contrario se penalizará sobre la nota definitiva de la tarea.
- La calificación del reporte se verá afectada en los siguientes casos: no demuestra una comprensión clara del problema que resuelve, el reporte no es claro o está en desorden, los archivos anexos no funcionan o el documento no es entregado según las reglas establecidas.
- Las preguntas acerca del enunciado se responderán a través del foro de MS-Teams: OPTIMIZACIÓN AVANZADA y en el horario de atención.

Cualquier sospecha de fraude será manejada de acuerdo con el reglamento de la Universidad.

Problema 1: Opti-drones (100 puntos)

Recientemente, se han hecho grandes avances tecnológicos en la industria agrícola colombiana. Uno de estos es el uso de drones para realizar la supervisión de distintos tipos de cultivos. El servicio de supervisión es ofrecido por un número reducido de empresas, que reconocen el impacto que podría tener en el desarrollo del sector. Una de estas empresas es Opti-drones.



Para realizar la supervisión de los cultivos de sus clientes, Opti-drones cuenta con drones equipados con cámaras de última tecnología. Estos equipos son utilizados para capturar hasta 300 imágenes de alta resolución que permiten evaluar la presencia de plagas en los cultivos. Los drones tienen una autonomía de 12 horas y pueden recorrer mil metros por minuto. Además, estos drones son guiados por rutas preestablecidas mediante un sistema de GPS que aumenta la automatización del servicio. Estas rutas son diseñadas específicamente para cada cliente, sin embargo, siempre salen de un hangar y deben regresar a este.

El próximo mes, Opti-drones tendrá un nuevo cliente, para el cual dispondrá de cinco drones. Dicho cliente cuenta con 25 cultivos, en donde crecen distintos alimentos. Los cultivos tienen características diferentes, en términos de las pendientes de los cultivos y la calidad del suelo. Esto hace necesario que se deba capturar un número distinto de fotos de cada cultivo para tener información que permita tomar decisiones acertadas acerca de su condición. La Tabla 1 presenta la ubicación de algunos de los cultivos del nuevo cliente, junto con el número de fotos que deberán tomarse para realizar un diagnóstico correcto. La información completa se encuentra en el archivo "Tarea 2-202420.xlsx".

Lugar	Coordenada X	Coordenada Y	Fotos
Hangar	40	80	0
Cultivo 1	39	59	50
Cultivo 2	8	82	30
Cultivo 3	8	53	55
...

Tabla 1. Ubicación y número de fotos para algunos cultivos.

Independientemente del número de fotos que deban tomarse, el tiempo que deberá permanecer un dron en un cultivo es de exactamente 90 minutos. Adicionalmente, la distancia (en cientos de metros) entre los cultivos se puede calcular como la distancia Manhattan¹.

¹ Para dos ubicaciones (x_1, y_1) y (x_2, y_2) , la distancia Manhattan se calcula como $|x_1 - x_2| + |y_1 - y_2|$ e indica los cientos de metros entre ellas.

Opti-drones lo ha contactado a usted para determinar **cuál debe ser la ruta de cada uno de los cinco drones**, de tal forma que se realice la supervisión del nuevo cliente **minimizando la distancia total recorrida**. Para resolver este problema, siga la estrategia de solución que se presenta a continuación.

a. Considere una estrategia de solución que garantice únicamente las siguientes condiciones:

- Todo cultivo debe ser visitado por un dron.
- Se deben utilizar todos los drones disponibles.
- No deben existir ciclos entre un par de ubicaciones. La Figura 1 muestra un ciclo entre un par de nodos i y j . Note que esta condición incluye al hangar, lo que significa que no es posible que un dron sólo visite un cultivo.

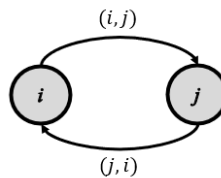


Figura 1. Ejemplo de ciclo.

- Sin incluir ninguna otra restricción**, plantee una formulación matemática rigurosa. Describa los conjuntos, parámetros, variables de decisión, función objetivo y restricciones.
- Resuelva el problema implementándolo en Python-Gurobi. Grafique la solución obtenida utilizando librerías como *NetworkX* o *Matplotlib* en Python. También, presente los resultados utilizando el formato que muestra la Tabla 2.
- ¿Qué puede concluir al observar la solución obtenida? ¿Es esta solución factible en términos de las capacidades de los drones y de las secuencias (estructuras) de las rutas formadas?

# Ruta	Secuencia	Tiempo (horas)	Fotos

Tabla 2. Formato de solución.

b. Plantee una estrategia de solución basada en *cortes* (es decir, incluyendo iterativamente restricciones a su formulación) que le permita resolver el problema original, teniendo en cuenta la **capacidad de cada dron y el tiempo máximo de operación**. Iterativamente, su implementación debe ser capaz de incluir restricciones que permitan:

- Eliminar rutas que excedan la capacidad de los drones (i.e., *capacity-cut-constraints*);
- Eliminar rutas que excedan el tiempo máximo de operación diaria de los drones (i.e., *time-cut-constraints*);
- Eliminar rutas (ciclos) que no pasen por el hangar.

Una vez usted genera un corte o un grupo de estos, la implementación debe resolver de nuevo el problema y verificar que se satisfagan todas las restricciones. En caso de incumplir alguna de las restricciones, se deben buscar y agregar nuevos cortes hasta que se obtenga un ruteo que cumpla con todas las especificaciones mencionadas. En particular, siga los pasos a continuación.

- i) Plantee una formulación matemática rigurosa que describa los conjuntos, parámetros, variables de decisión, función objetivo y restricciones.
- ii) Grafique la solución obtenida cada vez que genere un nuevo corte o un grupo de cortes utilizando una librería (es decir *Matplotlib* o *NetworkX*). Reporte algunos de estos gráficos.
- iii) Reporte la función objetivo de la solución óptima del problema y grafique las rutas de los drones. También presente la solución en el formato de la Tabla 2.
- iv) Presente el pseudocódigo de su algoritmo de solución en el informe.
- v) ¿Cómo se ve afectado el valor de la función objetivo cada vez que se agrega un corte o un conjunto de cortes? ¿Por qué? Elabore una gráfica que muestre el valor de la función objetivo en cada una de las iteraciones.