Comienza a programar o generar con IA.

Trabajo: Caso de Aplicación

Objetivos:
A través de esta actividad podrás articular una gran parte de los conocimientos adquiridos en el curso: construcción de modelos de programación lineal, programación lineal entera y uso de cuadernos de Google Colaboratory y el lenguaje de programación Python.

Descripción de la actividad y pautas de elaboración:
A continuación, se presenta el caso de aplicación a desarrollar (Anderson et al., 2011).

Un sistema de programación matemática, llamado *SilverScreener*, utiliza un modelo de programación entera 0-1 para ayudar a los gerentes de salas de cine a decidir cuáles películas exhibir semanalmente en una sala con múltiples pantallas. Suponga que a la gerencia de *Valley Cinemas* le gustaría investigar el potencial de utilizar un sistema de programación similar para su cadena de salas de cine con múltiples pantallas. *Valley* seleccionó una sala pequeña con dos pantallas para las pruebas piloto y le gustaría elaborar un modelo de programación lineal entera para ayudar a programar películas para las cuatro semanas siguientes. Hay seis cintas disponibles. La primera semana cada película está disponible, la última semana se exhibe cada película y el número máximo de semanas que se puede exhibir se muestra aquí:

Película	Primera semana disponible	Última semana disponible	Exhibicion maxima (semanas)
1	1	2	2
2	1	3	2
3	1	1	2
4	2	4	2
5	3	6	3
6	3	5	3

El programa de exhibición general para el cine se compone de programas individuales para cada una de las seis películas. Cada programa debe elaborarse especificando la semana que inicia la exhibición de la película y el número de semanas consecutivas que se exhibirá. Por ejemplo, un programa posible para la película 2 es que inicie en la semana 1 y se proyecte durante dos semanas. La política del cine requiere que una vez iniciada una película debe exhibirse en semanas consecutivas; su proyección no puede suspenderse y reiniciarse de nuevo. Para representar las posibilidades de programación para cada película, se elaboraron las variables de decisión siguientes:

x_ijk {(1 si la película i se programa para iniciar en la semana j y se exhibe por w semanas@0 en caso contrario)}

Por ejemplo, x_532=1 significa que el programa seleccionado para la película 5 comenzará en la semana 3 y se exhibirá durante dos semanas. Para cada película se proporciona una variable separada para cada programa posible.

a. Tres programas se asocian con la película 1. Liste las variables que representan estos programas.
b. Escriba una restricción que requiera que se seleccione solo un programa para la película 1.
c. Escriba una restricción que requiera que se seleccione solo un programa para la película 5.
d. ¿Qué restringe el número de películas que se pueden exhibir en la semana 1? Escriba una restricción que limite el número de películas seleccionadas para ver en la semana 1.
e. Escriba una restricción que limite el número de películas seleccionadas para ver en la semana 3.

Criterios de Evaluación

La evaluación se realizará de acuerdo con las siguientes especificaciones:

Aspecto a Evaluar	Porcentaje de Evaluación
Construcción del modelo	25%
Solución utilizando Python	65%
Conclusiones y presentación del informe en general (ecuaciones, tablas, gráficas, redacción y ortografía)	10%

Extensión máxima de la actividad:
Un documento con una extensión máxima de 15 páginas.

Entrega:
Como resultado final, se entregará un documento grupal con los siguientes ítems: construcción del modelo, implementación en Google Colaboratory usando Python e interpretación de resultados.

Solucion

a. Tres programas se asocian con la película 1. Liste las variables que representan estos programas.

Para poder resolver este punto hay que tener en cuenta que cada película en el cine debe decidir el programa de exhibición, con esto quiereo decir o buscar, en qué semana comienza la película y cuántas semanas consecutivas se proyectará, ademas de esto debemos tener en cuenta que la política del cine exige que una vez iniciada la proyección, esta debe continuar sin interrupciones.

En este caso, la película 1 puede exhibirse como máximo durante 2 semanas consecutivas dentro del período total de 4 semanas. Por tanto, las combinaciones posibles de inicio y duración que no exceden las 4 semanas son:

- Iniciar en la semana 1 y durar 1 semana.
- Iniciar en la semana 1 y durar 2 semanas.
- Iniciar en la semana 2 y durar 1 semana.

Esto genera las siguientes variables de decisión binarias asociadas a la película 1:

- x_{111} = 1 esto significa que la película 1 comienza en la semana 1 y se proyecta durante 1 semana.
- x_{112} = 1 esto significa que la película 1 comienza en la semana 1 y se proyecta durante 2 semanas.
- x_{121} = 1 esto significa que la película 1 comienza en la semana 2 y se proyecta durante 1 semana.

Con esto puedo decir que las variables que representan los 3 programas de la pelicula 1 son:

$x_{111}, x_{112}, x_{121}$

b. Escriba una restricción que requiera que se seleccione solo un programa para la película 1.

Para este punto y teniendo en cuenta que cada película solo puede ser proyectada una vez dentro del periodo de planificación, es necesario garantizar que solo una de las combinaciones posibles de inicio y duración sea seleccionaday esto se logra sumando todas las variables binarias asociadas a la película 1 y obligando a que exactamente una de ellas tenga valor 1. Y para daar solucion a esto voy a tomar como referencia las tres variables listadas en el literal a:

$x_{111} + x_{112} + x_{121} = 1$

Esto se da debido a que esta ecuación representa una restricción de unicidad: la cual hace que el modelo a elegir un único programa de exhibición para la película 1 entre las opciones posibles.

c. Escriba una restricción que requiera que se seleccione solo un programa para la película 5.

Al igual que en el caso de la película 1, para este punto se debe asegurar que solo un programa de exhibición sea elegido para la película 5 como lo pide el enunciado. Esto implica que, de todas las combinaciones válidas de semana de inicio y duración para esta película 5, una y solo una puede ser seleccionada.

Ahora teniendo en cuenta la información de la tabla tenemos que la primera semana disponible de esta pelicula es la semana 3, la ultima semana disponible es la 6 y la exhibicion maxima es de 3 semanas, pero ademas de esto le sumamos la restriccion del enunciado el cual esl ". Valley seleccionó una sala pequeña con dos pantallas para las pruebas piloto y le gustaría elaborar un modelo de programación lineal entera para ayudar a programar películas para las **cuatro** semanas siguientes" tenemos que

- x_{531}: película 5, semana 3, duración 1
- x_{532}: película 5, semana 3, duración 2
- x_{541}: película 5, semana 4, duración 1

Entonces, la restricción correspondiente sería:

$x_{531} + x_{532} + x_{541} = 1$

Ya que esto nos asegura que solo uno de los tres programas válidos y dentro del horizonte de planificación sea seleccionado para la película 5.

d. ¿Qué restringe el número de películas que se pueden exhibir en la semana 1? Escriba una restricción que limite el número de películas seleccionadas para ver en la semana 1.

Para poder resolver este punto hay que tener en cuenta que el modelo cubre solo las semanas 1 a 4 ademas de esto solo se pueden proyectar 2 películas por semanase debe de tener en cuenta las películas activas en la semana 1.

Película	Semana inicio (j)	Duración (k)	Semana final (j + k - 1)	Última semana disponible	Incluye semana 1	¿Válida?	Variable
1	1	1	1	2	✅	✅	x_{111}
1	1	2	2	2	✅	✅	x_{112}
2	1	1	1	3	✅	✅	x_{211}
2	1	2	2	3	✅	✅	x_{212}
3	1	1	1	1	✅	✅	x_{311}

Nota: Esta tabla es de mi autoría y ha sido elaborada de acuerdo con las normas APA

Ya con la tabla y teniendo una perspectiva visual mas clara encontramos que las variables que se proyectan en la smeana 1 son: x_{111}, x_{112}, x_{211}, x_{212} y x_{311}

Asi que la restriccion de este punto d seria de

$x_{111} + x_{112} + x_{211} + x_{212} + x_{311} \leq 2$

e. Escriba una restricción que limite el número de películas seleccionadas para ver en la semana 3.

Película	Inicio (j)	Duración (k)	Final (j+k-1)	Incluye semana 3	Variable
2	2	2	3	✅	x_{223}
2	3	1	3	✅	x_{231}
4	2	2	3	✅	x_{423}
4	3	1	3	✅	x_{431}
5	3	1	3	✅	x_{531}
5	3	2	4	✅	x_{532}
6	3	1	3	✅	x_{631}
6	3	2	4	✅	x_{632}

Nota: Esta tabla es de mi autoría y ha sido elaborada de acuerdo con las normas APA

Con esta tabla definida encontramos que no se proyecten más de dos películas simultáneamente y que la restriccion para la semana 3 como lo solicita el enunciado es de:

$x_{223} + x_{231} + x_{423} + x_{431} + x_{531} + x_{532} + x_{631} + x_{632} \leq 2$

Solución utilizando Python

```
pip install pyomo
```

```
Requirement already satisfied: pyomo in /usr/local/lib/python3.11/dist-packages (6.8.2)
Requirement already satisfied: ply in /usr/local/lib/python3.11/dist-packages (from pyomo) (3.11)
```

```
!apt-get install -y -qq glpk-utils
```

```
Selecting previously unselected package libsuitesparseconfig5:amd64.
(Reading database ... 126213 files and directories currently installed.)
Preparing to unpack .../libsuitesparseconfig5_1%3a5.10.1+dfsg-4build1_amd64.deb ...
Unpacking libsuitesparseconfig5:amd64 (1:5.10.1+dfsg-4build1) ...
Selecting previously unselected package libamd2:amd64.
Preparing to unpack .../libamd2_1%3a5.10.1+dfsg-4build1_amd64.deb ...
Unpacking libamd2:amd64 (1:5.10.1+dfsg-4build1) ...
Selecting previously unselected package libcolamd2:amd64.
Preparing to unpack .../libcolamd2_1%3a5.10.1+dfsg-4build1_amd64.deb ...
Unpacking libcolamd2:amd64 (1:5.10.1+dfsg-4build1) ...
Selecting previously unselected package libglpk40:amd64.
Preparing to unpack .../libglpk40_5.0-1_amd64.deb ...
Unpacking libglpk40:amd64 (5.0-1) ...
Selecting previously unselected package glpk-utils.
Preparing to unpack .../glpk-utils_5.0-1_amd64.deb ...
Unpacking glpk-utils (5.0-1) ...
Setting up libsuitesparseconfig5:amd64 (1:5.10.1+dfsg-4build1) ...
Setting up libamd2:amd64 (1:5.10.1+dfsg-4build1) ...
Setting up libcolamd2:amd64 (1:5.10.1+dfsg-4build1) ...
Setting up libglpk40:amd64 (5.0-1) ...
Setting up glpk-utils (5.0-1) ...
Processing triggers for man-db (2.10.2-1) ...
```

```
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
/sbin/ldconfig.real: /usr/local/lib/libtcm.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libhwloc.so.15 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtcm_debug.so.1 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libumf.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_level_zero.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_adapter_opencl.so.0 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libur_loader.so.0 is not a symbolic link
```

```
from pyomo.environ import *


# 1. Crear modelo
model = ConcreteModel()


# 2. Datos de entrada
peliculas = {
    1: {"inicio": 1, "fin": 2, "max_duracion": 2},
    2: {"inicio": 1, "fin": 3, "max_duracion": 2},
    3: {"inicio": 1, "fin": 1, "max_duracion": 2},
    4: {"inicio": 2, "fin": 4, "max_duracion": 2},
    5: {"inicio": 3, "fin": 6, "max_duracion": 3},
    6: {"inicio": 3, "fin": 5, "max_duracion": 3},
}

semanas_modelo = range(1, 5)  # semanas consideradas en el modelo (1-4)


# 3. Generar índices válidos (i, j, k)
def generar_variables_validas():
    idx = []
    for i, info in peliculas.items():
        for j in range(info["inicio"], info["fin"] + 1):
            for k in range(1, info["max_duracion"] + 1):
                if j + k - 1 <= info["fin"] and j + k - 1 <= 4:
                    idx.append((i, j, k))
    return idx

model.IDX = Set(initialize=generar_variables_validas(), dimen=3)


# 4. Variables de decisión binarias
model.x = Var(model.IDX, domain=Binary)

# 5. Restricción: Solo un programa por película (literales b y c)
def un_programa_por_pelicula(model, i):
    return sum(model.x[i, j, k] for (ii, j, k) in model.IDX if ii == i) == 1

model.restriccion_unica = Constraint(peliculas.keys(), rule=un_programa_por_pelicula)


# 6. Restricción: Máximo 2 películas activas por semana (literales d y e)
def max_dos_por_semana(model, s):
    return sum(model.x[i, j, k] for (i, j, k) in model.IDX if j <= s <= j + k - 1) <= 2

model.capacidad_semanal = Constraint(semanas_modelo, rule=max_dos_por_semana)


# 7. Función objetivo (no requerida, pero necesaria)
model.obj = Objective(expr=0, sense=minimize)
```

```
# 8. Resolución
solver = SolverFactory('glpk')  # Asegúrate de tener GLPK instalado
result = solver.solve(model, tee=True)


# 9. Mostrar solución de acuerdo al ejercicio resuelto
print("\nProgramación seleccionada:")
for (i, j, k) in model.IDX:
    if model.x[i, j, k]() == 1:
        print(f"Película {i} se exhibe desde la semana {j} por {k} semana(s)")
```

```
WARNING:pyomo.core:Implicitly replacing the Component attribute obj (type=<class 'pyomo.core.base.objective.ScalarObjective'>) on block
This is usually indicative of a modelling error.
To avoid this warning, use block.del_component() and block.add_component().
GLPSOL--GLPK LP/MIP Solver 5.0
Parameter(s) specified in the command line:
 --write /tmp/tmpdyo6ylpy.glpk.raw --wglp /tmp/tmpc5afuf3n.glpk.glp --cpxlp
 /tmp/tmpy0_7l5j3.pyomo.lp
Reading problem data from '/tmp/tmpy0_7l5j3.pyomo.lp'...
/tmp/tmpy0_7l5j3.pyomo.lp:109: warning: lower bound of variable 'x4' redefined
/tmp/tmpy0_7l5j3.pyomo.lp:109: warning: upper bound of variable 'x4' redefined
10 rows, 21 columns, 47 non-zeros
20 integer variables, all of which are binary
129 lines were read
Writing problem data to '/tmp/tmpc5afuf3n.glpk.glp'...
92 lines were written
GLPK Integer Optimizer 5.0
10 rows, 21 columns, 47 non-zeros
20 integer variables, all of which are binary
Preprocessing...
9 rows, 19 columns, 45 non-zeros
19 integer variables, all of which are binary
Scaling...
 A: min|aij| =  1.000e+00  max|aij| =  1.000e+00  ratio =  1.000e+00
Problem data seem to be well scaled
Constructing initial basis...
Size of triangular part is 9
Solving LP relaxation...
GLPK Simplex Optimizer 5.0
9 rows, 19 columns, 45 non-zeros
      0: obj =   0.000000000e+00 inf =   1.000e+00 (1)
      1: obj =   0.000000000e+00 inf =   0.000e+00 (0)
OPTIMAL LP SOLUTION FOUND
Integer optimization begins...
Long-step dual simplex will be used
+     1: mip =     not found yet >=              -inf        (1; 0)
+     1: >>>>>   0.000000000e+00 >=   0.000000000e+00   0.0% (1; 0)
+     1: mip =   0.000000000e+00 >=     tree is empty   0.0% (0; 1)
INTEGER OPTIMAL SOLUTION FOUND
Time used:   0.0 secs
Memory used: 0.1 Mb (66683 bytes)
Writing MIP solution to '/tmp/tmpdyo6ylpy.glpk.raw'...
40 lines were written

Programación seleccionada:
Película 1 se exhibe desde la semana 2 por 1 semana(s)
Película 2 se exhibe desde la semana 3 por 1 semana(s)
Película 3 se exhibe desde la semana 1 por 1 semana(s)
Película 4 se exhibe desde la semana 2 por 1 semana(s)
Película 5 se exhibe desde la semana 4 por 1 semana(s)
Película 6 se exhibe desde la semana 4 por 1 semana(s)
```

Conclusion

- Por medio de esta actividad se logró desarrollar un modelo de programación entera binaria 0-1 como se solicitaba en la actividad para representar la planificación de películas en una sala de cine con dos pantallas para la prueba, basado en el caso SilverScreener.
- El objetivo principal fue formular matemáticamente un conjunto de restricciones que permitieran seleccionar un único programa de exhibición por película con los iterales solicitados en la actividad, cumpliendo con limitaciones de disponibilidad, duración máxima y capacidad semanal de las pantallas.
- El modelo fue implementado utilizando el lenguaje de programación Python junto con el paquete Pyomo el cual es una herramienta especializada para la modelación algebraica de problemas de optimización y en este se definieron únicamente aquellas variables de

decisión correspondientes a combinaciones de película, semana de inicio y duración que fueran válidas según la información que se nos brindo en la actividad.