MIIA

Maestría
en Inteligencia Analítica para
la toma de decisiones

Modelaje y mejora de procesos

Modulo: optimización

Universidad de los Andes
Colombia

Conceptos claves:
Optimización con variables discretas

Universidad de los Andes
Colombia

Conceptos claves: optimización con variables discretas

A) Porque el espacio de soluciones es más pequeño. Lo que dificulta la precisión al momento de hallar una solución. Por esto se recurre a algoritmos de búsqueda local

B) Porque las soluciones de un problema entero son, en general, más difíciles de interpretar que las de un problema continuo. Por esto se debe evitar incluir variables discretas, de no ser absolutamente necesario

C) Porque la formulación de las restricciones y/o función objetivo suele ser más compleja. La intuición requerida para estas formulaciones es muy diferente al caso de variables continuas

D) Porque estos problemas no cumplen los supuestos de linealidad en las variables. Debe recurrirse a *Branch & Bound* para encontrar una solución discreta

¿Por qué los problemas de optimización con variables discretas (i.e., enteras o binarias) son generalmente más difíciles de resolver que aquellos con variables continuas?

Conceptos claves: optimización con variables discretas

Branch & Bound es un método de solución que puede ser utilizado para resolver problemas de optimización con variables continuas, binarias y enteras.

- Verdadero
- Falso

Universidad de los Andes
Colombia

Conceptos claves: optimización con variables discretas

Un excursionista está preparándose para un viaje y debe decidir qué objetos llevar. Ha escrito una lista de objetos que desear llevar y, gracias a su experiencia, ha podido asignar un puntaje (entre 1 y 10, donde 10 es muy importante) de "importancia" de cada objeto para su excursión. Cada objeto tiene asociado un peso (en kg) y el excursionista sabe que, por la distancia del viaje, solo puede llevar hasta 10 kg de equipaje. Adicionalmente, el espacio de su mochila solo le permite llevar hasta 5 objetos. El excursionista quiere saber qué objetos llevar para obtener la mejor combinación posible de objetos en su mochila

Se utiliza la siguiente notación:

J: conjunto de objetos

x_j: 1, si se lleva el objeto $j \in J$. 0, de lo contrario

c_j: importancia de llevar el objeto $j \in J$

a_j: peso del objeto $j \in J$ en kilogramos

¿Cómo se define matemáticamente la naturaleza de las variables de este problema?

$$x_j \in \{0,1\}, \forall j \in J$$

¿Cómo se modelaría la función objetivo de este problema?

$$max \sum_{j \in J} c_j x_j$$

¿Cuál restricción se asegura que el excursionista no exceda el peso total (10 kg) que soporta su mochila?

$$\sum_{j \in J} a_j x_j \leq 10$$

Conceptos claves: optimización con variables discretas

En uno de los restaurantes de Pollería Dantzig se está cambiando el menú de bebidas para que sea más balanceado y tenga más opciones. Entre las opciones se tiene bebidas carbonatadas y jugos. Entre las bebidas carbonatadas se tienen jengibre o piña. Entre los jugos se tienen mandarina, manzana y mora. Además, se puede vender agua y malteada de vainilla. Se desea escoger cuáles opciones incorporar en el menú de bebidas.
Se utiliza la siguiente notación:

$$B = \{Jengibre, Piña, Mandarina, Manzana, Mora, Agua, Malteada\}$$

$$x_i : \begin{cases} 1 & Si\ se\ escoge\ la\ bebida\ i \in B \\ 0 & dlc \end{cases}$$

Por preferencias particulares del restaurante, el jugo de mandarina y el jugo de mora se deben escoger juntos o no se escoge ninguno. ¿Qué restricción modelaría esta condición?

$x_{Mandarina}$	x_{Mora}	¿Se quiere permitir?
0	0	☑
0	1	☒
1	0	☒
1	1	☑

$$x_{Mandarina} = x_{Mora}$$

PREGUNTAS

Caso 2: Comentarios Calificación

Priorización de un plan de explotación de pozos petroleros

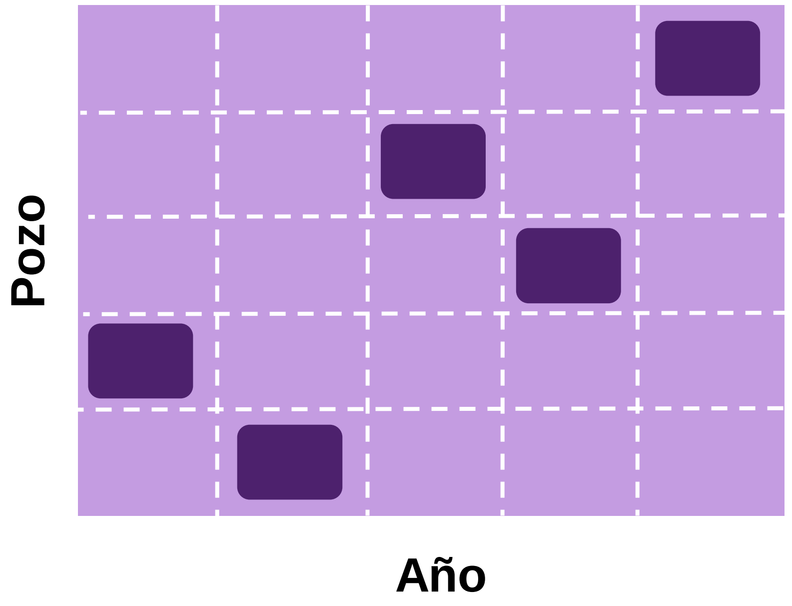
Recapitulación

Selección y programación de proyectos

Contexto

Cuáles pozos se deben explotar

Cuándo se deben explotar



$$x_{it}: \begin{cases} 1, \text{si se explota el pozo } i \in P \text{ en el año } t \in T \\ 0, \text{de lo contrario} \end{cases}$$

Información → **Parámetros**

¿Qué información está disponible para tomar la decisión?

Conjuntos

P: conjunto de pozos
T: conjunto de años

Parámetros

b: presupuesto total de inversión
n: máximo número de pozos a explotar
u: cantidad de generadores eléctricos disponibles
s: cantidad de operarios disponibles
p_i: producción esperada de miles de barriles por día del pozo $i \in P$
g_i: número de generadores requeridos por el pozo $i \in P$
o_i: número de operarios requeridos por el pozo $i \in P$

Selección y programación de proyectos

Modelo base

$$\max \sum_{i \in P} \sum_{t \in T} v_{it} x_{it}$$

s.a,

$$\sum_{t \in T} x_{it} \leq 1 \quad , \quad \forall i \in P$$

$$\sum_{i \in P} \sum_{t \in T} c_{it}\, x_{it} \leq b$$

$$\sum_{i \in P} p_i x_{it} \geq m_t \quad , \quad \forall t \in T$$

$$\sum_{i \in P} o_i x_{it} \leq s \quad , \quad \forall t \in T$$

No exceder el número de generadores disponibles

$$\sum_{i \in P} g_i x_{it} \leq a \quad , \quad \forall t \in T$$

No exceder el número de operarios disponibles

$$\sum_{i \in P} \sum_{t \in T} x_{it} \leq n$$

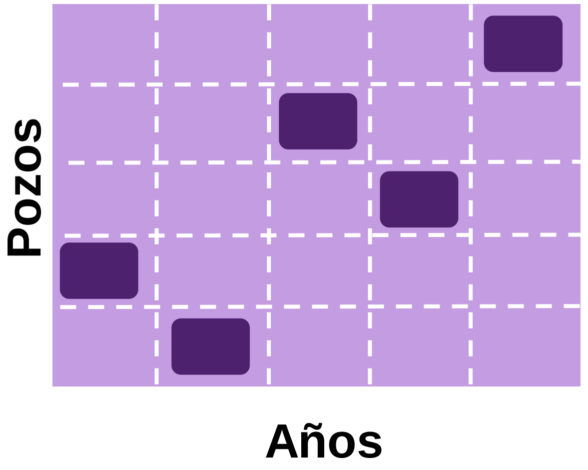
Cumplir legislación ambiental

$$x_{it} \in \{0,1\} ; \quad \forall i \in P, t \in T$$

Caso 3: Consejos Python

Conjunto de Tuplas

Python



Conjuntos

P: *Conjunto de pozos candidatos*

T: *Conjunto de periodos (años) en el horizonte de planeación*

```
# ------------
# Conjuntos
# ------------

# Conjunto de pozos
Pozos = ['DELE B-1','EL MORRO-1','FLORENA A-5','FLORENA C-6','FLORENA N-2','FLORENA N-4 ST','FLORENA-T8',
         'PAUTO J-6','PAUTO M4','PAUTO M-5', 'PAUTO SUR B-1','PAUTO SUR C-2','PAUTO-1','VOLCANERA A-1',
         'VOLCANERA C-2']

# Conjunto de años
Tiempos = range(1,11)  # No incluye 11
```

Conjunto de tuplas (pozo,año)

$Ej: ('DELE\ B-1', 1), ('EL\ MORRO-1', 2), ('FLORENA\ A-5', 3)$

```
# Creación lista Conjunto de tuplas (pozo, año)
Pozo_x_Tiempo = [(i, t) for i in Pozos for t in Tiempos]
```

Variable de decisión

Python

$$x_{it}: \begin{cases} 1, \text{ si el pozo } i \in P \text{ se perfora en el año } t \in T. \\ 0, \text{ de lo contrario} \end{cases}$$

Naturaleza de las variables

$$x_{it} \in \{0,1\} \quad \forall \; i \in P, t \in T$$

Variable Binaria indexada en el conjunto *Pozo_x_Tiempo (pozo, año)*

```
#-------------------------------
# Variables de Decisión
#-------------------------------

x = lp.LpVariable.dicts(name='invertir',
                        indexs = Pozo_x_Tiempo, #Conjunto
                        cat=lp.LpBinary)        #Tipo de variable
```

Doble Sumatoria

Python

Restricción # 6:
No se debe exceder la cantidad de pozos perforados (n) impuesta por el gobierno

$$\sum_{i \in P} \sum_{t \in T} x_{it} \leq n$$

```python
#6. No se debe exceder la cantidad de pozos perforados impuesta por el gobierno
problema += lp.lpSum((x[i,t] for i,t in Pozo_x_Tiempo)) <= n,'R6'
```

```python
#6. No se debe exceder la cantidad de pozos perforados impuesta por el gobierno
problema += lp.lpSum((x[i,t] for i in Pozos for t in Tiempos)) <= n,'R6'
```

Sumatoria y para todo (forall)

Python

Restricción # 3: Meta de Producción
Las metas de producción (en miles de barriles diarios) deben ser cumplidas

$$\sum_{i \in P} x_{it} * p_i \geq m_t \quad \forall\, t \in T$$

```python
#3. Las metas de producción deben ser cumplidas
for t in Tiempos:

    problema += lp.lpSum(((prodMin[i]+prodModa[i]+prodMax[i])/3 * x[i,t] for i in Pozos)) >= metas[t], 'R3_'+str(t)
```

Diccionario de Resultados

Python

Luego de optimizar el modelo, se puede construir un diccionario “resultados” que almacene los valores de las variables optimizadas

Alternativa #1

```
# Construcción del diccionario de resultados (x[i,t].value)
resultados={}

for i, var in x.items():
    resultados = {canal: x[canal].value() for  canal in x}
```

Alternativa #2

```
# Construcción del diccionario de resultados (x[i,t].value)
resultados={}

resultados = {(i, t): x[i, t].value() for i, t in Pozo_x_Tiempo}
```

Taller en equipos

Formulación matemática

Caso 4: estructuras comunes

1. Recursos

Selección y programación de proyectos

Cambio a recursos

Información → **Parámetros**

¿Qué información está disponible para tomar la decisión?

4 → 6

4 → 6

Selección y programación de proyectos

Modelo base

$$\max \sum_{i \in P} \sum_{t \in T} v_{it} x_{it}$$

La formulación matemática permanece, pero en la implementación se modifican los valores

s.a,

$$\sum_{t \in T} x_{it} \leq 1 \quad , \quad \forall i \in P$$

$$\sum_{i \in P} \sum_{t \in T} c_{it}\, x_{it} \leq b$$

$$\sum_{i \in P} p_i x_{it} \geq m_t \quad , \quad \forall t \in T$$

$$\sum_{i \in P} o_i x_{it} \leq s \quad , \quad \forall t \in T$$

$$\sum_{i \in P} g_i x_{it} \leq u \quad , \quad \forall t \in T$$

$$\sum_{i \in P} x_{it} \leq n$$

$$x_{it} \in \{0,1\}\, ; \quad \forall i \in P, t \in T$$

Caso 4: estructuras comunes

2. Inventarios

Selección y programación de proyectos: inventarios

Nueva información

Información → **Parámetros**

¿Qué información está disponible para tomar la decisión?

Presupuesto de inversión año a año

Presupuesto al inicio de cada año (millones USD)
12

Se cuenta con dos nuevos operarios

Se cuenta con dos nuevos generadores

Selección y programación de proyectos: inventarios

Nuevas condiciones

Condiciones → **Restricciones** ⚠

¿Qué limita las decisiones?

El 5% de las utilidades pueden ser reinvertidas en los años siguientes

Contabilizar el inventario de dinero

Selección y programación de proyectos: inventarios

Nuevas decisiones

Decisiones → **Variables de Decisión**

¿Qué debe decidirse?

Inventario de dinero al final del año

Decisiones

¿Qué debe decidirse?

Variables de Decisión

¿Cuáles pozos se deben seleccionar y cuándo se deben explotar?

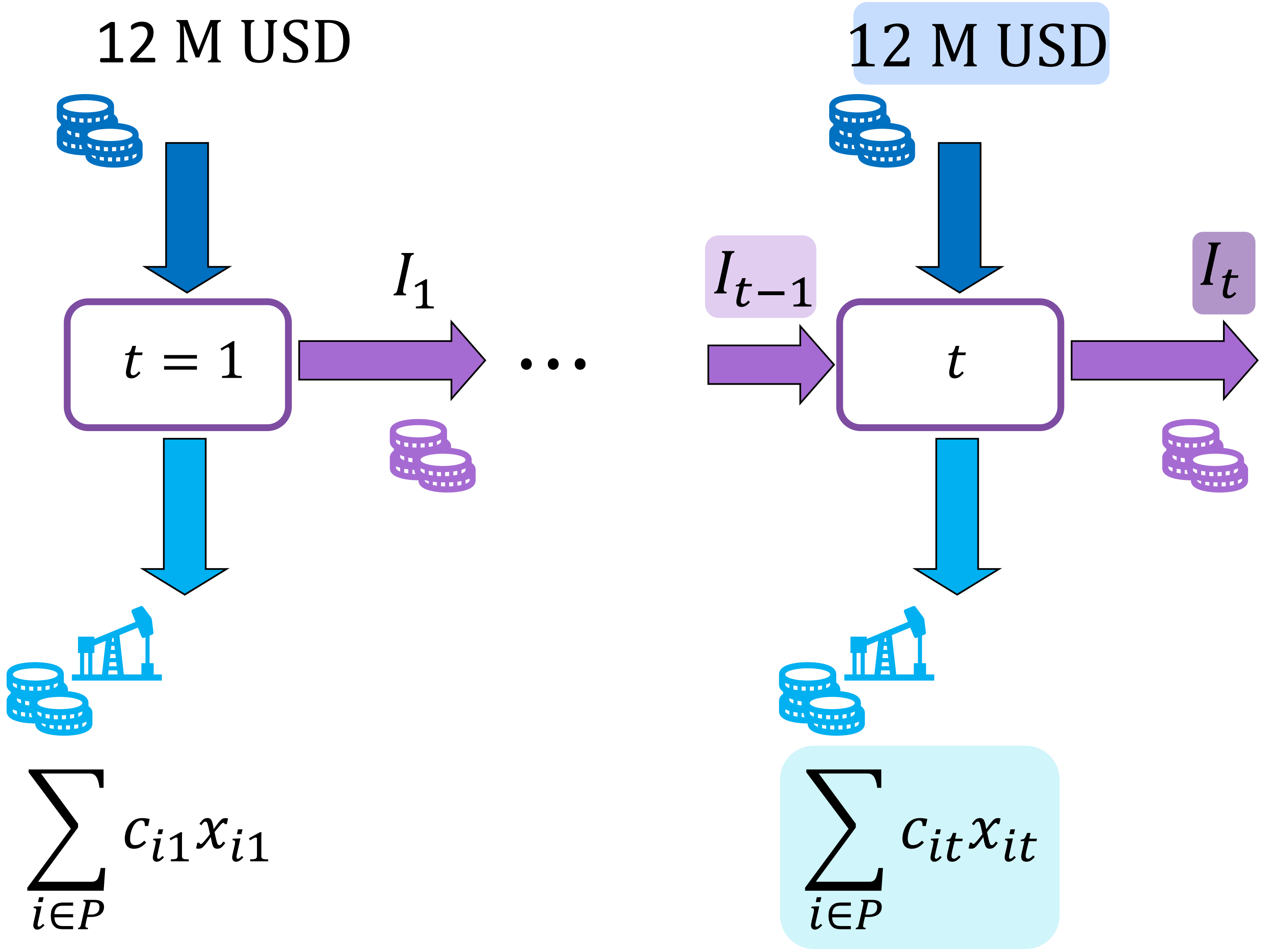
$$x_{it}: \begin{cases} 1, & \text{si el pozo } i \in P \text{ se perfora en el año } t \in T. \\ 0, & \text{de lo contrario} \end{cases}$$

Inventario al final del año

I_t: Inventario al final del año $t \in T$

Condiciones → **Restricciones**

¿Qué limita la decisión?



Inventario del primer periodo

$$I_1 = 12 - \sum_{i \in P} c_{i1} x_{i1}$$

Inventario del segundo periodo en adelante

$$I_t = I_{t-1} + 12 + 0.05 \cdot \sum_{i \in P} v_{i,t-1} x_{i,t-1} - \sum_{i \in P} c_{it} x_{it}, \forall t \in T | t \geq 2$$

Naturaleza de la variable de inventario

$$I_t \geq 0, \qquad \forall t \in T$$

Selección y programación de proyectos

Modelo con inventarios

$$\max \sum_{i\in P}\sum_{t\in T} v_{it}x_{it}$$

s.a,

$$\sum_{t\in T} x_{it} \leq 1 \quad , \quad \forall i \in P$$

$$\sum_{i\in P} p_i x_{it} \geq m_t \quad , \quad \forall t \in T$$

$$\sum_{i\in P} o_i x_{it} \leq s \quad , \quad \forall t \in T$$

$$\sum_{i\in P} g_i x_{it} \leq u \quad , \quad \forall t \in T$$

$$\sum_{i\in P} x_{it} \leq n$$

$$I_1 = 12 - \sum_{i\in P} c_{i1}x_{i1} \quad , \quad \forall t \in T$$

$$I_t = I_{t-1} + 12 + 0.05 \cdot \sum_{i\in P} v_{i,t-1}x_{i,t-1} - \sum_{i\in P} c_{it}x_{it} \,, \quad \forall t \in T | t \geq 2$$

$$x_{it} \in \{0,1\} \quad , \quad \forall i \in P, t \in T$$

$$I_t \geq 0 \quad ; \quad \forall t \in T$$

Caso 4: estructuras comunes

3. Restricciones suaves

Selección y programación de proyectos: suavización

Nueva información

Información → **Parámetros**

¿Qué información está disponible para tomar la decisión?

Penalización por incumplir la meta de producción

Penalización por cada 1,000 barriles al día por debajo de la meta (millones de USD)
17

Selección y programación de proyectos: suavización

Nuevas condiciones

Condiciones → **Restricciones** ⚠

¿Qué limita las decisiones?

Cumplir las metas de producción o contabilizar el número de barriles al día por debajo de la meta

Selección y programación de proyectos: suavización

Nuevas decisiones

Decisiones → **Variables de Decisión**

¿Qué debe decidirse?

número de barriles al día por debajo de la meta cada año

Decisiones

Variables de Decisión

¿Qué debe decidirse?

Número de barriles al día debajo de la meta

m_t^-: Número de barriles al día debajo de la meta en el año $t \in T$

Condiciones → Restricciones

¿Qué limita la decisión?

Se registra el incumplimiento en miles de barriles por día

$$\sum_{i \in P} p_i x_{it} + m_t^- \geq m_t, \qquad \forall t \in T$$

Naturaleza de la variable de miles de barriles al día por debajo de la meta

$$m_t^- \geq 0, \qquad \forall t \in T$$

Objetivo

Función objetivo

Se incluye la penalización por los miles de barriles al día por debajo de la meta

$$\max \sum_{i \in P} \sum_{t \in T} v_{it} x_{it} - \sum_{t \in T} 17 m_t^-$$

Selección y programación de proyectos: suavización

Modelo con suavización

$$\max \sum_{i \in P} \sum_{t \in T} v_{it} x_{it} - \sum_{t \in T} 17 m_t^-$$

s.a,

$$\sum_{t \in T} x_{it} \leq 1 \quad , \quad \forall i \in P$$

$$\sum_{i \in P} \sum_{t \in T} c_{it}\, x_{it} \leq b$$

$$\sum_{i \in P} p_i x_{it} + m_t^- \geq m_t \quad , \quad \forall t \in T$$

$$\sum_{i \in P} o_i x_{it} \leq s \quad , \quad \forall t \in T$$

$$\sum_{i \in P} g_i x_{it} \leq u \quad , \quad \forall t \in T$$

$$\sum_{i \in P} x_{it} \leq n$$

$$x_{it} \in \{0,1\} \quad , \quad \forall i \in P, t \in T$$

$$m_t^- \geq 0 \quad ; \quad \forall t \in T$$

Consejos de programación

¿Cómo traducimos las expresiones a programación?

Ejemplo con condicional en la sumatoria

Python

$$\min \sum_{i \in P} \sum_{j \in D | i<j} x_{ij} c_{ij}$$

$$i \quad \begin{array}{c|ccc} & 1 & 2 & 3 \\ \hline 1 & 11 & \boxed{12} & \boxed{13} \\ 2 & 21 & 22 & \boxed{23} \\ 3 & 31 & 32 & 33 \end{array}$$

```
#------------------------------
# Función objetivo
#------------------------------
#Crea la expresión de minimización de costos
problema+=lp.lpSum(x[(i,j)]*c[(i,j)] for i in P for j in D if i < j), "Costos Totales"
```

En los condicionales solo podemos poner índices o parámetros.
No se pueden colocar variables, esto haría el problema no lineal.

Ejemplo con condicional en el para todo

Python

$$I_i = I_0 + x_1 - d_1$$

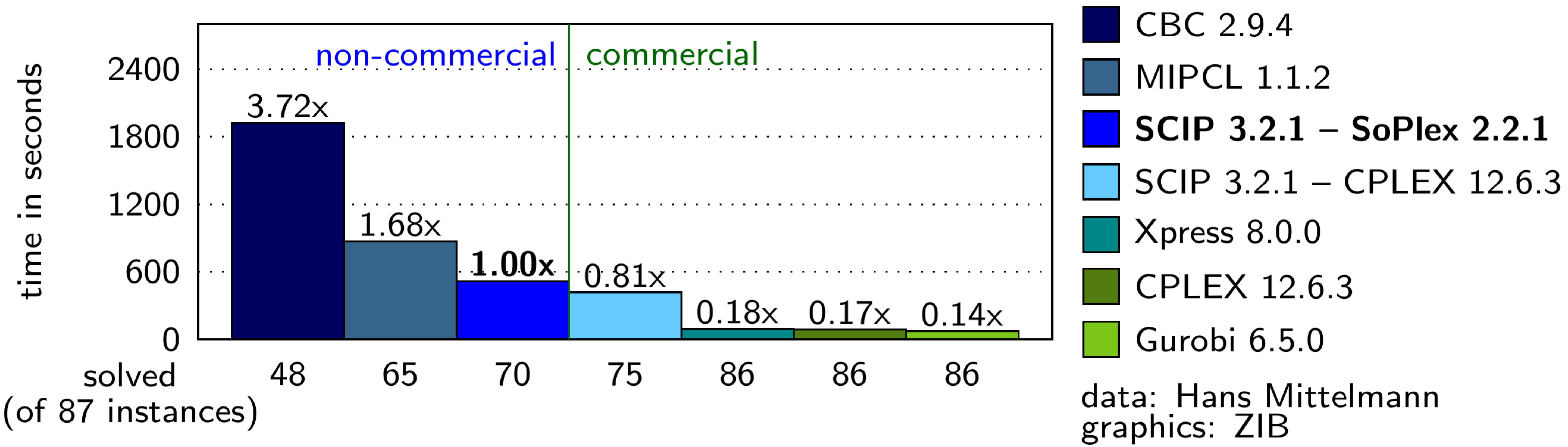
$$I_i = I_{i-1} + x_i - d_i, \quad \forall i \in S | i > 1$$

```python
#Contabilizar los inventarios de cada semana
problema+=I[1]==I0+x[1]-d[1], "Inventario en la semana 1"
for i in S:
    if i > 1:
        problema+=I[i]==I[i-1]+x[i]-d[i], "Inventario en la semana "+ str(i)
```

En los condicionales solo podemos poner índices o parámetros.
No se pueden colocar variables, esto haría el problema no lineal.

Solución de modelos

Optimizadores



MIP solver benchmark (1 thread): Shifted geometric mean of results taken from the homepage of Hans Mittelmann (13/Jun/2016). Unsolved or failed instances are accounted for with the time limit of 2 hours.

Repositorio de optimización

Python


copa-uniandes.github.io/optimizacion/intro.html

Universidad de los Andes

Repositorio de Optimización

Search this book...

PRÁCTICAS DE FORMULACIÓN

Chocolatería
Planeación de menús
Producción de jugos
Transporte de productos lácteos
Producción de cemento
Asignación de tareas
Extracción minera
Salsa al Parque
Búsqueda Local

GUÍAS

Repositorio de Optimización

Bienvenidos al repositorio de prácticas de formulación de Optimización. El objetivo de este repositorio es el de acompañar las clases del curso con ejercicios prácticos que los estudiantes puedan revisar como material complementario. Para resolver los ejercicios, usaremos *Python* como lenguaje de programación y el paquete *PuLP* para los modelos de optimización.

En la parte izquierda te encontrarás con algunas secciones que corresponden a practicas de formulación. Cada sección tiene una pestaña de enunciado y una pestaña de solución. La idea de esta división es que puedas intentar desarrollar el ejercicio por tu cuenta, sin distracciones, y que luego puedas comparar tu desarrollo con nuestra solución. Los códigos que encuentres en este compilado, los puedes copiar y pegar en un script de *Python* y modificarlos con un ambiente de desarrollo como *Spyder* o *Jupyter*. Alternativamente, puedes interactuar directamente con el código a través de tu explorador por medio de Binder para *Jupyter*.

Las prácticas están organizadas por temas, empezando desde las más sencillas e incrementando poco a poco el nivel de complejidad. Si no estás seguro de cuáles son las prácticas que puedes resolver hasta el momento puedes consultarle a un asistente o instructor del curso correspondiente al respecto.

Sin más, ¡a optimizar!

https://copa-uniandes.github.io/optimizacion/intro.html

PREGUNTAS