

## Laboratorio 1: Optimización Combinatoria y Asignación de Recursos

### Problema 1: Asignación de Tareas en un Equipo de Desarrollo Ágil

1. Define las tareas disponibles y sus puntos de historia.

# Tarea	Tarea	Puntos de historia	Prioridad
1	Optimización de la API principal	5	Maxima
2	Revisión de interfaz de usuario	3	Media alta
3	Integración de sistema de pagos	13	Alta
4	Actualización de documentación técnica	1	Media baja
5	Desarrollo de pruebas unitarias extendidas	21	Minima
6	Mejora de la respuesta del servidor	2	Media
7	Implementación de funciones de seguridad	2	Alta
8	Refactorización del código del módulo X	5	Media
9	Optimización de la base de datos	8	baja
10	Creación de un nuevo servicio de microservicios	13	Maxima
11	Implementación de interfaz de administración	21	alta

#### Datos del problema:

4 programadores, 2 semanas.

13 puntos de historia por programador → 52 puntos de historia máximos.

Maximiza la suma de las prioridades de las tareas completadas sin exceder la restricción de los puntos de historia.

#### Modelo matemático:

Conjuntos:  $T = \{\text{tareas}\}$ ,  $H = \{\text{puntos historia}\}$  &  $P = \{\text{prioridades}\}$

Variable de Decisión:

$X_t$ : binaria (1 si la tarea  $t$  se realiza y 0 si no se realiza).

Parámetros: Los puntos de historia  $H_t$  donde  $t \in T$  y las prioridades  $P_t$  donde  $t \in T$

Función objetivo:  $\max \left( \sum_{t \in T} X_t \cdot P_t \right)$

Restricciones:  $\sum_{t \in T} X_t \cdot H_t \leq 52$

## Resultados obtenidos:

```
johanbaq ~ main U:2 ~/Documents/Uniandes Linux/MOS/GroupMOS/Lab1 python task1.py
Task 1 yes
Task 2 yes
Task 3 yes
Task 4 yes
Task 5 no
Task 6 yes
Task 7 yes
Task 8 yes
Task 9 yes
Task 10 yes
Task 11 no
Model unknown

Variables:
x : Size=11, Index=T
Key : Lower : Value : Upper : Fixed : Stale : Domain
1 : 0 : 1.0 : 1 : False : False : Binary
2 : 0 : 1.0 : 1 : False : False : Binary
3 : 0 : 1.0 : 1 : False : False : Binary
4 : 0 : 1.0 : 1 : False : False : Binary
5 : 0 : 0.0 : 1 : False : False : Binary
6 : 0 : 1.0 : 1 : False : False : Binary
7 : 0 : 1.0 : 1 : False : False : Binary
8 : 0 : 1.0 : 1 : False : False : Binary
9 : 0 : 1.0 : 1 : False : False : Binary
10 : 0 : 1.0 : 1 : False : False : Binary
11 : 0 : 0.0 : 1 : False : False : Binary

Objectives:
objective : Size=1, Index=None, Active=True
Key : Active : Value
None : True : 44.0

Constraints:
constraint : Size=1
Key : Lower : Body : Upper
None : None : 52.0 : 52.0
```

Obtenemos una F.O de 52, y se aceptan todas las tareas excluyendo la 5 y 11. Lo cual es consistente con una primera visión del problema, ya que la tarea 5 presenta una prioridad mínima, y por otro lado, la tarea 11, si bien tiene una prioridad alta, consume el mayor puntos de historia de todas las tareas.

## Problema 2: Asignación de Trabajos a Trabajadores

5. Define los trabajadores disponibles y su tiempo de trabajo.
6. Define los trabajos y su ganancia.

# Trabajador	Horas
1	8
2	10
3	6

# Trabajo	Trabajo	Ganancia (USD)	Tiempo (Horas)
1	Consultoría	50	4
2	Análisis mod	60	5
3	Informes	40	3
4	Investigación	70	6
5	Deuda técnica	30	2

### Modelo matemático:

Índices:

t = tareas

w = trabajador

Conjuntos:

Workers = {trabajadores}

HorasWorker =  $\{HW_{w-1}, HW_w, HW_{w+1}\}$

T = {tareas}

TareasGanancia =  $\{TG_{t-1}, TG_t, TG_{t+1}\}$

TareasCosto =  $\{TC_{t-1}, TC_t, TC_{t+1}\}$

Variable de decisión

$X_{wt}$ : binaria, donde  $X_{wt} = 1$  si la tarea  $t$  se asigna al trabajador  $w$ , y  $X_{wt} = 0$  de lo contrario.

Función objetivo:  $\max \left( \sum_{w \in W, t \in T} X_{wt} \cdot TG_t \right)$

Restricciones:

$$\checkmark \sum_{w \in W} X_{wt} \leq 1 \text{ para todo } t \text{ en } T$$

$$\checkmark \sum_{w \in W, t \in T} X_{wt} \cdot TC_t \leq HW_w$$

## Resultados obtenidos:

```
johanbaq  main U:2  ~/Documents/Uniandes Linux/MOS/GroupMOS/Lab1  python task2.py
Tarea 1:
Seleccionada por trabajador 1
Tarea 2:
Seleccionada por trabajador 2
Tarea 3:
Seleccionada por trabajador 2
Tarea 4:
Seleccionada por trabajador 3
Tarea 5:
Seleccionada por trabajador 1
Model unknown

Variables:
  x : Size=15, Index=T*W
      Key : Lower : Value : Upper : Fixed : Stale : Domain
      (1, 1) : 0 : 1.0 : 1 : False : False : Binary
      (1, 2) : 0 : 0.0 : 1 : False : False : Binary
      (1, 3) : 0 : 0.0 : 1 : False : False : Binary
      (2, 1) : 0 : 0.0 : 1 : False : False : Binary
      (2, 2) : 0 : 1.0 : 1 : False : False : Binary
      (2, 3) : 0 : 0.0 : 1 : False : False : Binary
      (3, 1) : 0 : 0.0 : 1 : False : False : Binary
      (3, 2) : 0 : 1.0 : 1 : False : False : Binary
      (3, 3) : 0 : 0.0 : 1 : False : False : Binary
      (4, 1) : 0 : 0.0 : 1 : False : False : Binary
      (4, 2) : 0 : 0.0 : 1 : False : False : Binary
      (4, 3) : 0 : 1.0 : 1 : False : False : Binary
      (5, 1) : 0 : 1.0 : 1 : False : False : Binary
      (5, 2) : 0 : 0.0 : 1 : False : False : Binary
      (5, 3) : 0 : 0.0 : 1 : False : False : Binary

Objectives:
  objective : Size=1, Index=None, Active=True
      Key : Active : Value
      None : True : 250.0

Constraints:
  constraint1 : Size=5
      Key : Lower : Body : Upper
      1 : None : 1.0 : 1.0
      2 : None : 1.0 : 1.0
      3 : None : 1.0 : 1.0
      4 : None : 1.0 : 1.0
      5 : None : 1.0 : 1.0
  constraint2 : Size=3
      Key : Lower : Body : Upper
      1 : None : 6.0 : 8.0
      2 : None : 8.0 : 10.0
      3 : None : 6.0 : 6.0
```

Obtenemos una F.O. de 250, seleccionando todas las tareas disponibles, por los trabajadores 1, 2 y 3. Lo cual es consistente con que incluir las 5 tareas no sobrepasa el límite por trabajador, pero adicionalmente se cumple que una tarea se asigna una única vez. Quizá en un escenario con una variedad más grande de tareas se podría ver una distinción mejor de las tareas y de la optimización realizada. Por el momento, podemos decir que es correcto al no incumplirse ninguna restricción.

## Problema 3: Problema Complejo de Misión Humanitaria en Zambia

### Modelo matemático:

Índices:

a = aviones

r = recursos

Conjuntos:

Recursos = {recursos}

Valor =  $\{V_{r-1}, V_r, V_{r+1}\}$

Peso =  $\{P_{r-1}, P_r, P_{r+1}\}$

Volumen =  $\{VO_{r-1}, VO_r, VO_{r+1}\}$

A = {aviones}

Capacidad Peso =  $\{CP_{a-1}, CP_a, CP_{a+1}\}$

Capacidad Volumen =  $\{CV_{a-1}, CV_a, CV_{a+1}\}$

### Variable de decisión

$X_{ar}$ : binaria, donde  $X_{ar} = 1$  si avion  $a$  se le asigna el recurso  $r$ , y  $X_{ar} = 0$  de lo contrario.

Función objetivo:  $\max \left( \sum_{a \in A, r \in R} X_{ar} \cdot V_r \right)$

Restricciones:

- ✓  $\sum_{a \in A} X_{ar} \leq 1$  para todo  $r$  en  $R$
- ✓  $\sum_{a \in A, r \in R} X_{ar} \cdot P_r \leq CP_a$
- ✓  $\sum_{a \in A, r \in R} X_{ar} \cdot VO_r \leq CV_a$
- ✓  $X_{a2} = 0$  para todo  $a=1$  (considerando  $r = 2$  como el índice de medicinas y  $a = 1$  como el avión 1, debido a que el avión 1 no puede transportar medicinas)
- ✓  $X_{a3} + X_{a4} \leq 1$  para todo  $a \in A$  (considerando  $r = 3$  como equipos médicos y  $r = 4$  como agua potable)

### Casos para analizar:

- ¿Que impacto tiene las restricciones de almacenamiento de recursos?

**Restricción Medicinas en Avión 1:** Impide que se use el avión con la menor capacidad para transportar un recurso de alta prioridad y bajo peso/volumen, lo que podría haber permitido maximizar el uso de otros aviones para recursos más voluminosos y pesados. Esto puede reducir la eficiencia global al tener que redistribuir cargas de manera menos óptima entre los aviones restantes.

**Incompatibilidad entre Equipos Médicos y Agua Potable:** Esta restricción limita la flexibilidad para cargar aviones basándose en la maximización del espacio y peso disponible. Al no poder enviar estos recursos juntos, puede resultar en la necesidad de realizar vuelos adicionales.

- ¿Que pasa si no tengo en cuenta la restricción volumetrica?

Si se ignoran las restricciones volumétricas y solo se consideran las de peso, se pueden presentar varios problemas:

**Sobrecarga Volumétrica:** Aunque el peso total de los recursos cargados en un avión pueda estar dentro del límite, el exceso de volumen puede hacer físicamente imposible cerrar las puertas del avión o comprometer la seguridad del vuelo debido a una distribución inadecuada del espacio, ya que no se tiene en cuenta el volumen.

**Ineficiencia en la Distribución de Cargas:** El volumen es un factor crítico en la logística de transporte. Ignorarlo puede resultar en una distribución donde los recursos más livianos pero voluminosos puedan ocupar un espacio que podría haber sido utilizado de manera más eficiente por recursos más densos y con mayor prioridad.

### Resultados obtenidos:

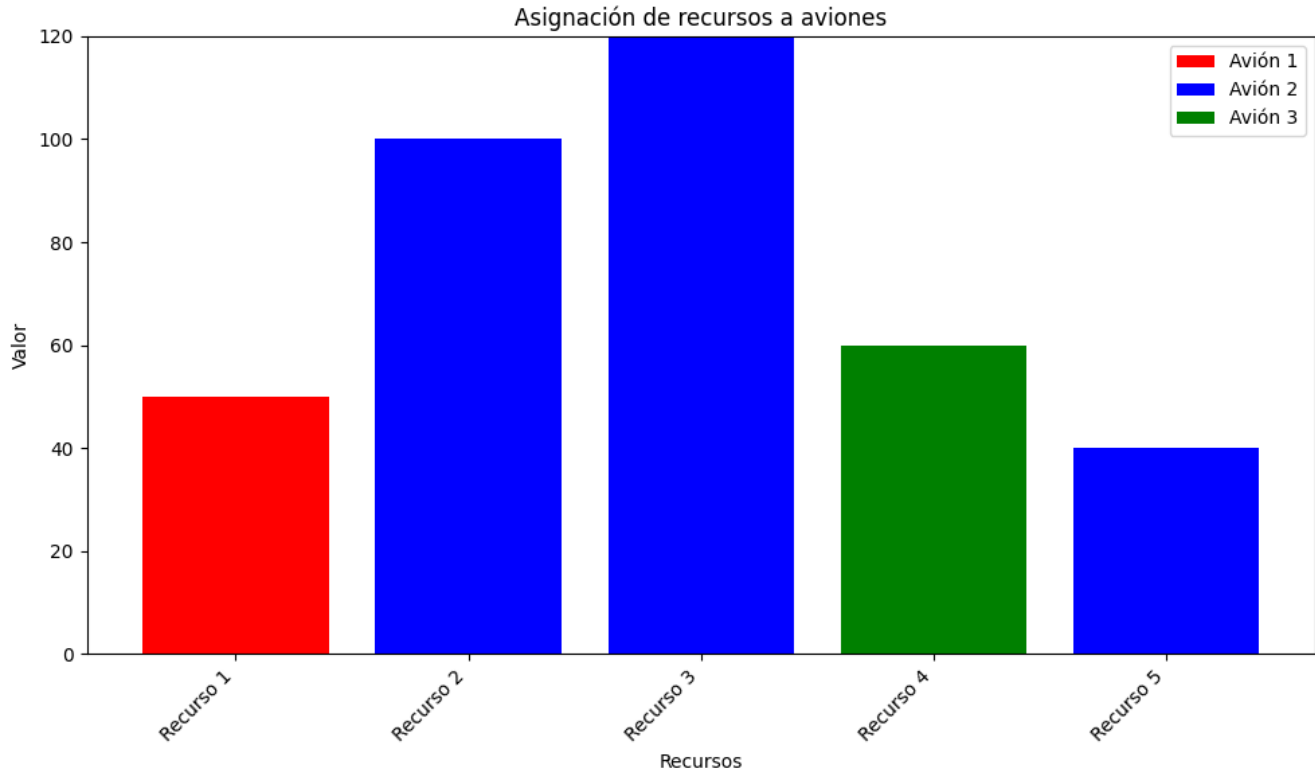
```
johanbaq 2 main U:2 ~/Documents/Uniandes Linux/MOS/GroupMOS/Lab1 python task3.py
Model unknown
Variables:
x : Size=15, Index=A*R
Key : Lower : Value : Upper : Fixed : Stale : Domain
(1, 1) : 0 : 1.0 : 1 : False : False : Binary
(1, 2) : 0 : 0.0 : 1 : False : False : Binary
(1, 3) : 0 : 0.0 : 1 : False : False : Binary
(1, 4) : 0 : 0.0 : 1 : False : False : Binary
(1, 5) : 0 : 0.0 : 1 : False : False : Binary
(2, 1) : 0 : 0.0 : 1 : False : False : Binary
(2, 2) : 0 : 1.0 : 1 : False : False : Binary
(2, 3) : 0 : 1.0 : 1 : False : False : Binary
(2, 4) : 0 : 0.0 : 1 : False : False : Binary
(2, 5) : 0 : 1.0 : 1 : False : False : Binary
(3, 1) : 0 : 0.0 : 1 : False : False : Binary
(3, 2) : 0 : 0.0 : 1 : False : False : Binary
(3, 3) : 0 : 0.0 : 1 : False : False : Binary
(3, 4) : 0 : 1.0 : 1 : False : False : Binary
(3, 5) : 0 : 0.0 : 1 : False : False : Binary

Objectives:
objective : Size=1, Index=None, Active=True
Key : Active : Value
None : True : 370.0

Constraints:
constraint1 : Size=3
Key : Lower : Body : Upper
1 : None : 15.0 : 30.0
2 : None : 35.0 : 40.0
3 : None : 18.0 : 50.0
constraint2 : Size=3
Key : Lower : Body : Upper
1 : None : 8.0 : 25.0
2 : None : 18.0 : 30.0
3 : None : 12.0 : 35.0
constraint3 : Size=1
Key : Lower : Body : Upper
None : None : 0.0 : 0.0
constraint4 : Size=3
Key : Lower : Body : Upper
1 : None : 0.0 : 1.0
2 : None : 1.0 : 1.0
3 : None : 1.0 : 1.0
constraint5 : Size=5
Key : Lower : Body : Upper
1 : None : 1.0 : 1.0
2 : None : 1.0 : 1.0
3 : None : 1.0 : 1.0
4 : None : 1.0 : 1.0
5 : None : 1.0 : 1.0
```

Obtenemos una F.O. de 370. Adicionalmente no se incumple ninguna restricción, ya que podemos comprobar que en ningún caso se rompen las restricciones volumétricas o de peso, así como se asigna una única vez cada recurso. De igual forma se cumple que el avión 1 no transporte medicina y que los equipos médicos no van en el mismo transporte que el agua potable.

A continuación una representación gráfica de los resultados obtenidos:



Nota: considerando la siguiente notación:

Recursos	# Recurso	Valor	Peso (TON)	Volumen (m <sup>3</sup> )
<b>Alimentos Básicos</b>	1	50	15	8
<b>Medicinas</b>	2	100	5	2
<b>Equipos Médicos</b>	3	120	20	10
<b>Agua Potable</b>	4	60	18	12
<b>Mantas</b>	5	40	10	6