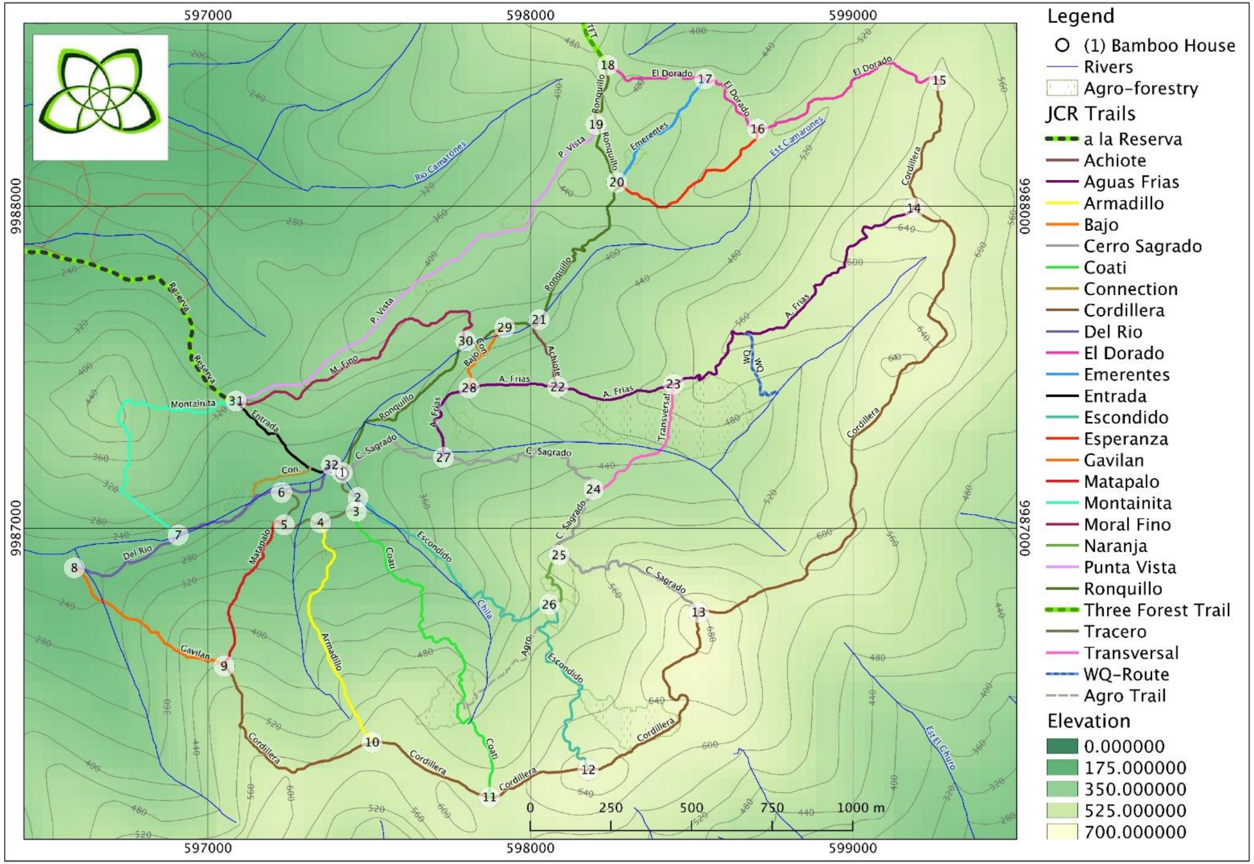
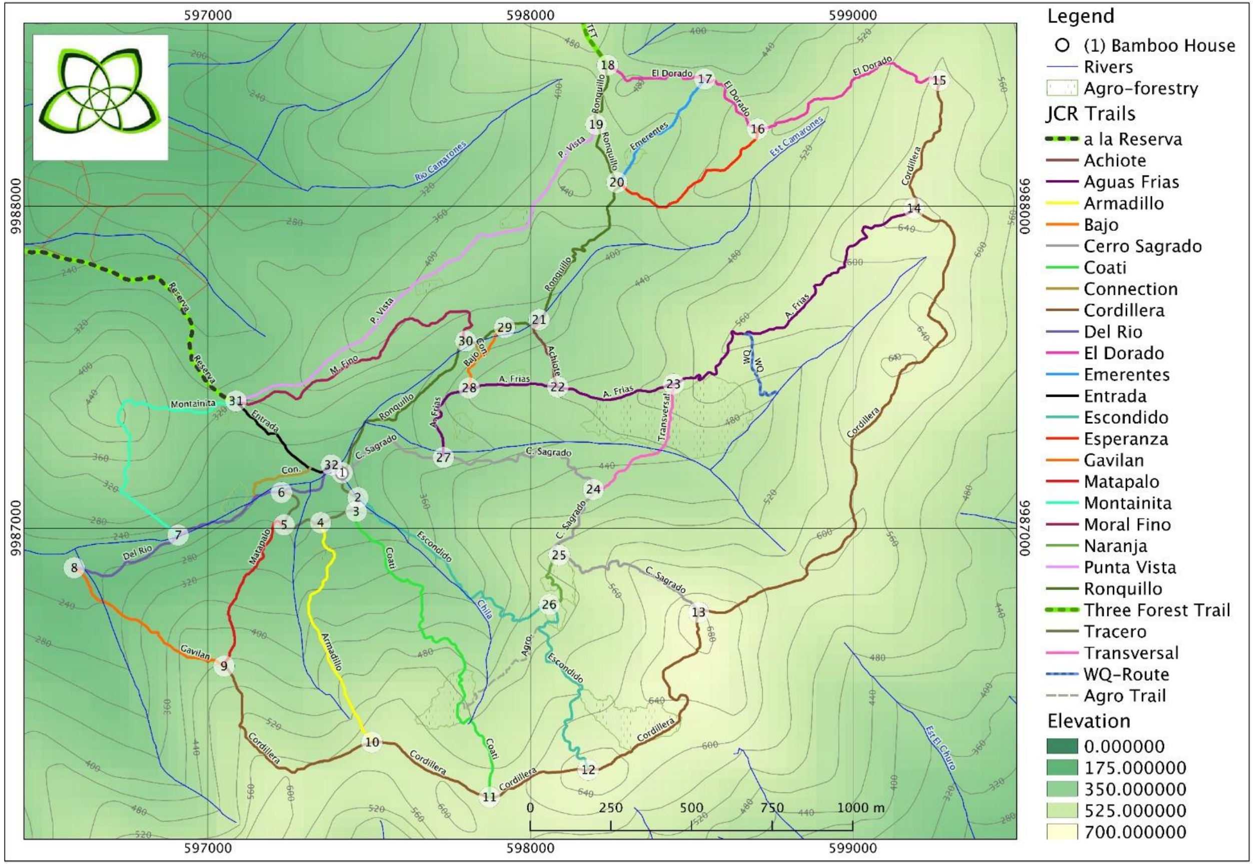
# Optimización para guardabosques




Por:
Mauricio
Velasco
y
Nicolás
Betancourt

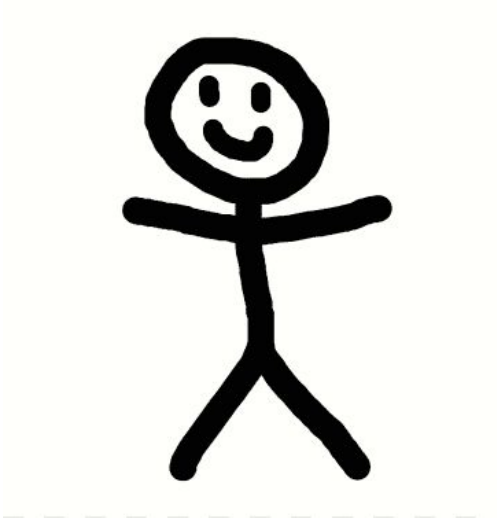
# Optimización para guardabosques




Agenda:

1. Qué es un Security Game
2. Ejemplo (Green Security game)
3. La optimización (y sus problemas)
4. Combinatorial Multi Armed Bandits
5. CMAB en GSG
6. Demostración computacional

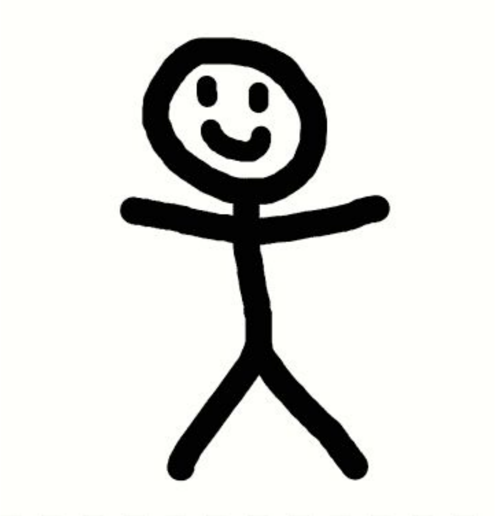
**<u>¿Qué es un Security Game?</u>**



Bueno

Malo

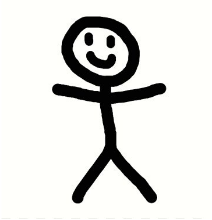
**¿Qué es un Security Game?**

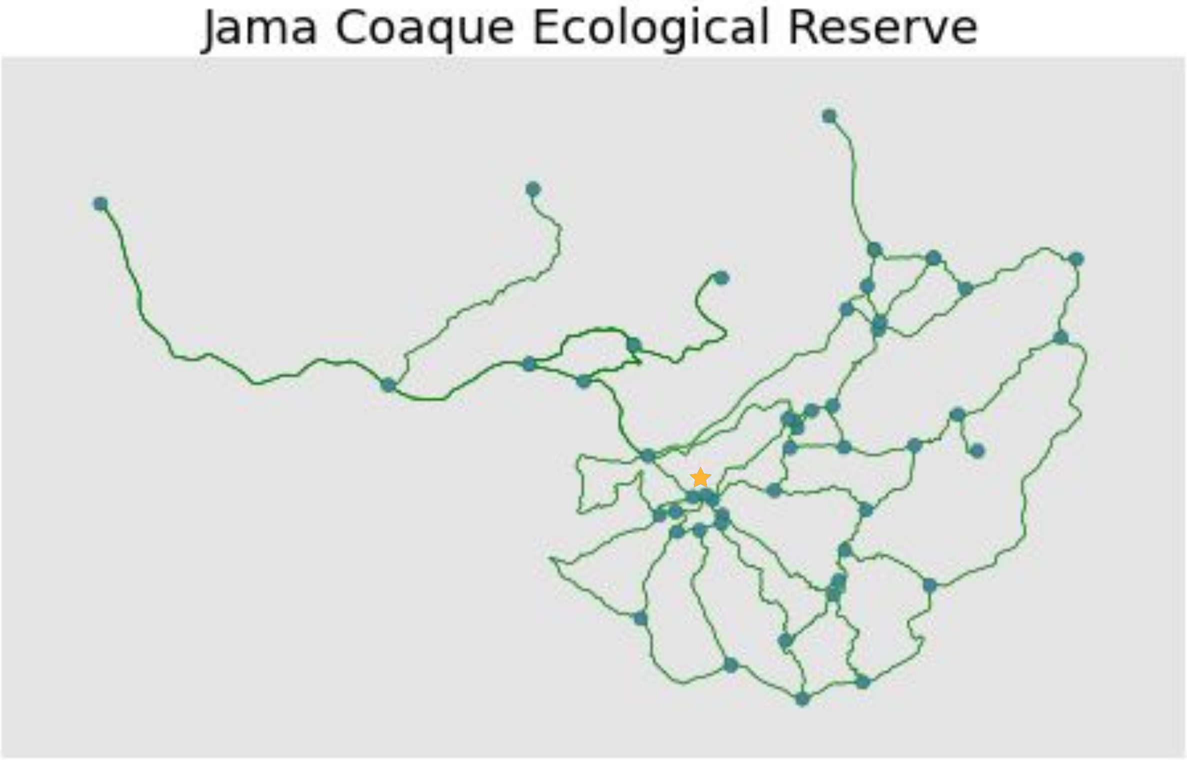


Proteger conjunto de objetivos

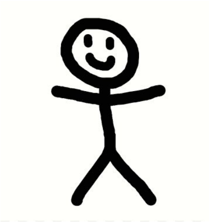
- Atacar
- Ventaja absoluta de información

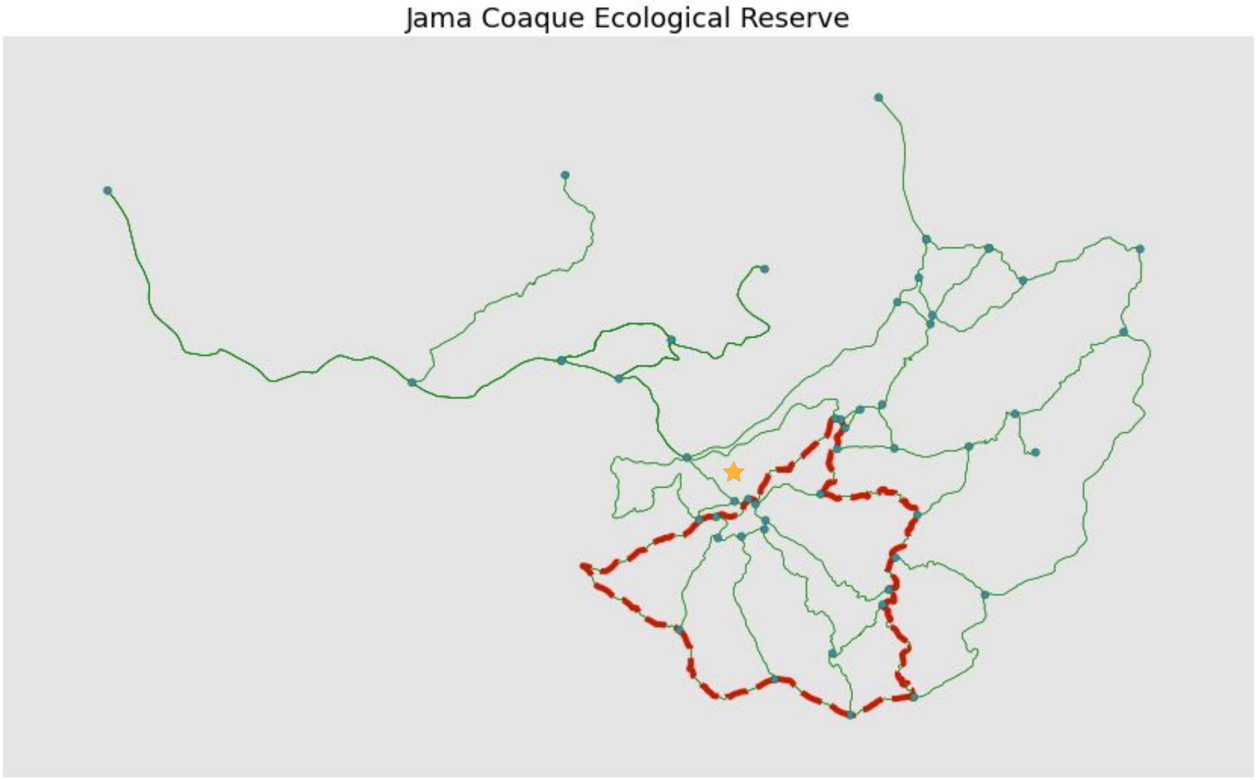
# Ejemplo 1





# Ejemplo 1





## Ejemplo 1: Utilidad guardabosques

$$C_G = \left( C^1 \middle| \quad C^2 \middle| \quad \cdots \quad \middle| C^M \right) \quad \text{Matriz } n \times M \quad n = \#\text{senderos y } M = \#\text{ciclos}$$

## Ejemplo 1: Utilidad guardabosques

$$C_G = \left(C^1 \middle| \quad C^2 \middle| \quad \cdots \quad \middle| C^M\right) \quad \text{Matriz } n \times M \quad n = \#\text{senderos y } M = \#\text{ciclos}$$

$$C_i^j = \begin{cases} 1 & \text{Ciclo } j \text{ pasa por sendero } i \\ 0 & \text{En cualquier otro caso} \end{cases}$$

## Ejemplo 1: Utilidad guardabosques

$$C_G = \left( C^1 \middle| \quad C^2 \middle| \quad \cdots \quad \middle| C^M \right) \quad \text{Matriz } n \times M \quad n = \#\text{senderos y } M = \#\text{ciclos}$$

$$C_i^j = \begin{cases} 1 & \text{Ciclo } j \text{ pasa por sendero } i \\ 0 & \text{En cualquier otro caso} \end{cases}$$

$$U_g(t, C^j) = \sum_{i=1}^{n} t_i C_i^j - t_i(1 - C_i^j)$$

## Ejemplo 1: Utilidad guardabosques

$$C_G = \left( C^1 \middle| \quad C^2 \middle| \quad \cdots \quad \middle| C^M \right) \quad \text{Matriz } n \times M \quad n = \#\text{senderos y } M = \#\text{ciclos}$$

$$C_i^j = \begin{cases} 1 & \text{Ciclo } j \text{ pasa por sendero } i \\ 0 & \text{En cualquier otro caso} \end{cases}$$

$$U_g(t, C^j) = \sum_{i=1}^{n} t_i C_i^j - t_i(1 - C_i^j)$$

$$\beta \in [0,1]^M, \sum_{j=1}^{M} \beta_j = 1$$

**Ejemplo 1: Utilidad guardabosques**

$$C_G = \left( C^1 \middle| \quad C^2 \middle| \quad \cdots \quad \middle| C^M \right) \quad \text{Matriz } n \times M \quad n = \#\text{senderos y } M = \#\text{ciclos}$$

$$C_i^j = \begin{cases} 1 & \text{Ciclo } j \text{ pasa por sendero } i \\ 0 & \text{En cualquier otro caso} \end{cases}$$

$$U_g(t, C^j) = \sum_{i=1}^{n} t_i C_i^j - t_i(1 - C_i^j)$$

$$\beta \in [0,1]^M, \sum_{j=1}^{M} \beta_j = 1$$

$$\Rightarrow \quad U_g(t, \beta) := \mathbb{E}\left[U_g(t, C^j)\right] = 2t^T C_G \beta - t^T \mathbb{1}_n = 2t^T C_G \beta - \|t\|_1$$

## Ejemplo 1: Un traidor puede más que unos cuantos

$$U_c(t, \beta) = -U_g(t, \beta)$$

## Ejemplo 1: Un traidor puede más que unos cuantos

$$U_c(t,\beta) = -U_g(t,\beta)$$

$$(PC)_\beta \begin{cases} \max\limits_{\alpha \in \mathbb{R}^M} & \mathbb{E}\left[t^T \mathbb{1}_n - 2t^T C_G \beta\right] \\ & \text{s.t.:} \\ & \sum_{j=1}^M \alpha_j = 1 \\ & \alpha \geq 0 \end{cases}$$

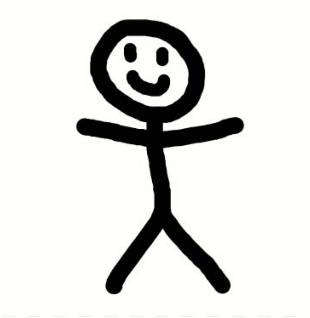
## Ejemplo 1: Un traidor puede más que unos cuantos

$$U_c(t,\beta) = -U_g(t,\beta)$$

$$(PC)_\beta \begin{cases} \max\limits_{\alpha \in \mathbb{R}^M} & \mathbb{E}\left[t^T \mathbb{1}_n - 2t^T C_G \beta\right] \\ & \text{s.t.:} \\ & \sum_{j=1}^M \alpha_j = 1 \\ & \alpha \geq 0 \end{cases}$$

El cazador toma una decisión determinística ante cualquier estrategia del guardabosques

## Ejemplo 1: Y que esos cuantos no lo olviden fácilmente



$$\max_{t,\beta \in \mathbb{R}^n} U_g(t, \beta)$$

s.t.:

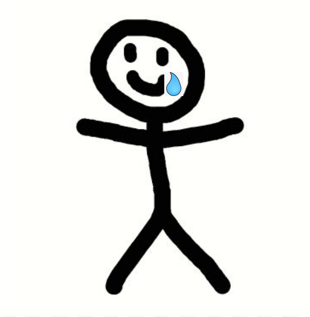
$$\beta \in P_G, t \in \mathcal{C}_G$$

$$t = \arg\max U_c(\beta, t)$$

$\mathcal{C}_G$ = conjunto de ciclos

$P_G$ = Distribuciones de probabilidades sobre $\mathcal{C}_G$

# Ejemplo 1: Implementación costosa, hipótesis fuertes



$$\max_{t,\beta\in\mathbb{R}^n} U_g(t,\beta)$$

s.t.:

$$\beta \in P_G, t \in \mathcal{C}_G$$

$$t = \arg\max U_c(\beta, t)$$

- Problema cuadrático con restricciones enteras
- Muchas restricciones generalmente difíciles de conocer
- La última condición requiere una restricción por cada ciclo
- Es posible que el cazador no invierta tantos recursos en vigilar ni en evitar al guardabosques

# Multi-Armed Bandits

## Multi-Armed Bandits

1

2

...

100

$\mu_1$ $\mu_2$ $\mu_{100}$

Objetivo: maximizar la plata esperada

Presupuesto: 500 monedas

# Multi-Armed Bandits

Solución:
Meta todas sus 500 monedas en la máquina que mayor retorno tiene en promedio y obtenga 500 veces $\mu^*$

# Multi-Armed Bandits

Solución:

Meta todas sus 500 monedas en la máquina que mayor retorno tiene en promedio y obtenga 500 veces $\mu^*$

Problema:

Uno nunca conoce los retornos promedio de nada en la vida

# Multi-Armed Bandits

Solución:
Meta todas sus 500 monedas en la máquina que mayor retorno tiene en promedio y obtenga 500 veces $\mu^*$

Problema:
Uno nunca conoce los retornos promedios de nada en la vida

**Regret:** $$500\mu^* - \sum_{t=1}^{500} r_t$$

# Multi-Armed Bandits

Solución:
Meta todas sus 500 monedas en la máquina que mayor retorno tiene en promedio y obtenga 500 veces $\mu^*$

Problema:
Uno nunca conoce los retornos promedios de nada en la vida

**Regret:**

$$500\mu^* - \sum_{t=1}^{500} r_t$$

Crece logarítmicamente

## Upper Confidence Bound

En la k-ésima etapa del problema sean

$T_i(k)$ = Número de veces que he jugado la i-ésima máquina

$\overline{X}_{i,n} := \frac{1}{n}\sum_{t=1}^{n} x_t$ = Promedio muestral de la i-ésima máquina tras haberla jugado n veces

Elija la máquina que maximice una cantidad que crece con $\overline{X}_{i,n}$ y el recíproco de $T_i(k)$

$$i_k = \arg\max_{i=1,\cdots,K}\left\{\overline{X}_{i,T_i(k-1)} + c\sqrt{\frac{\ln k}{T_i(k-1)}}\right\}$$

# Combinatorial Multi-Armed Bandits

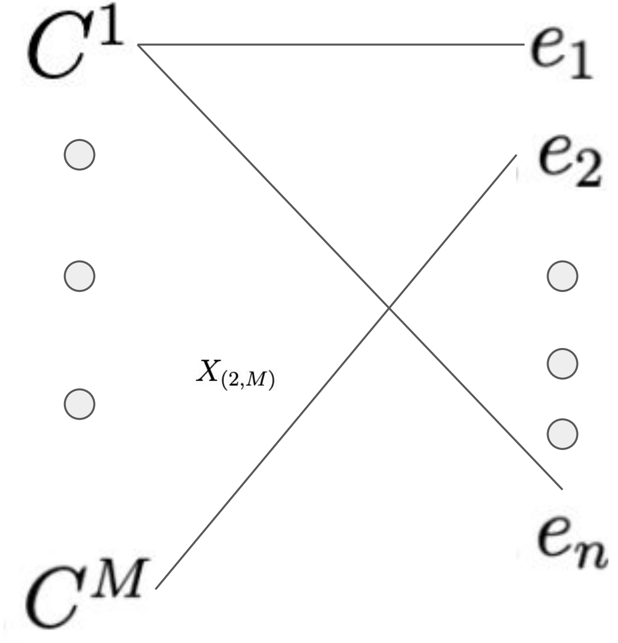



Elegir más de una máquina en cada turno

## Ejemplo 2: Maximum coverage bandits

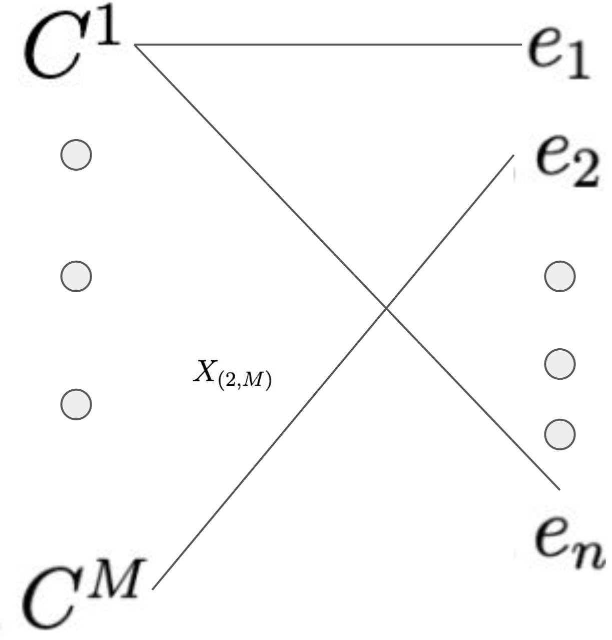
Considere un conjunto $E = \{e_1, e_2, \cdots, e_n\}$, y $\mathcal{C} = \{C^1, \cdots, C^M\}$ una colección de subconjuntos de $E$. En este problema cada par $(i, j)$ tal que $e_i \in C^j$ tiene asociada una variable Bernoulli $X_{(i,j)}$ con probabilidad de éxito $p(i, j)$ desconocida.

Restricción de presupuesto: $B$



# Ejemplo 2: Maximum coverage bandits

Cuando las probabilidades de éxito son conocidas basta resolver un MILP

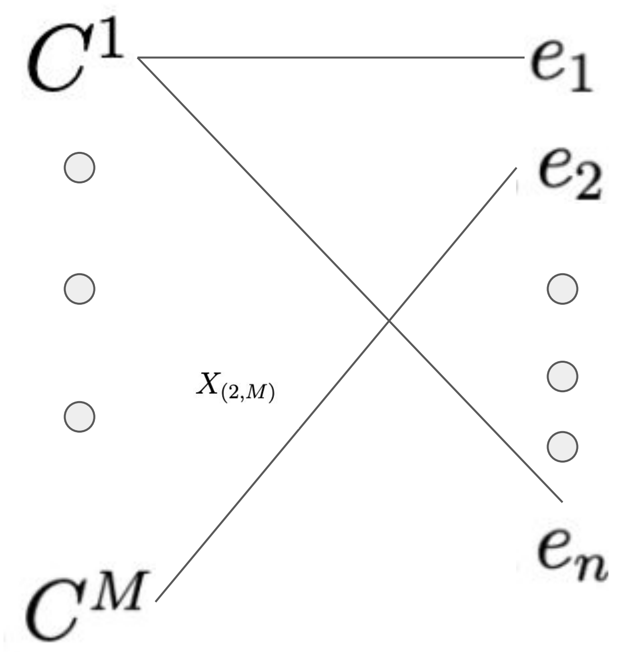



$$
(\mathcal{I})_\mu \begin{cases} \max\limits_{x\in\mathbb{R}^M \; y\in\mathbb{R}^n} \quad \sum_{i=1}^{n} \mu_i y_i \\ \text{s.t.:} \\ \sum_{j=1}^{m} x_j = B \\ \sum_{i\in C^j} x_j \geq \quad y_i \forall\, i \\ x_j, y_i \in \{0,1\} \end{cases}
$$

Donde $\mu_i$ es la probabilidad de que cubramos el elemento **i**

# Ejemplo 2: Maximum coverage bandits

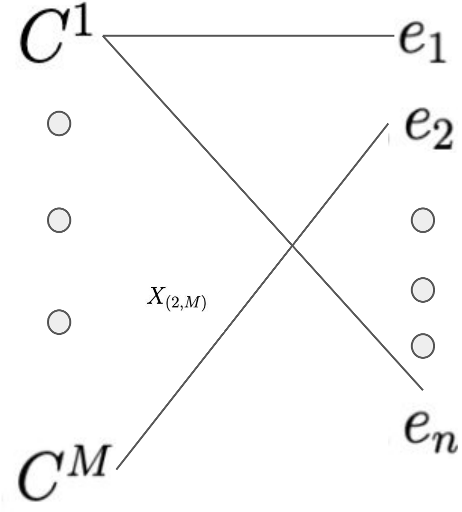
Considere un conjunto $E = \{e_1, e_2, \cdots, e_n\}$, y $\mathcal{C} = \{C^1, \cdots, C^M\}$ una colección de subconjuntos de $E$. En este problema cada par $(i, j)$ tal que $e_i \in C^j$ tiene asociada una variable Bernoulli $X_{(i,j)}$ con probabilidad de éxito $p(i, j)$ desconocida.



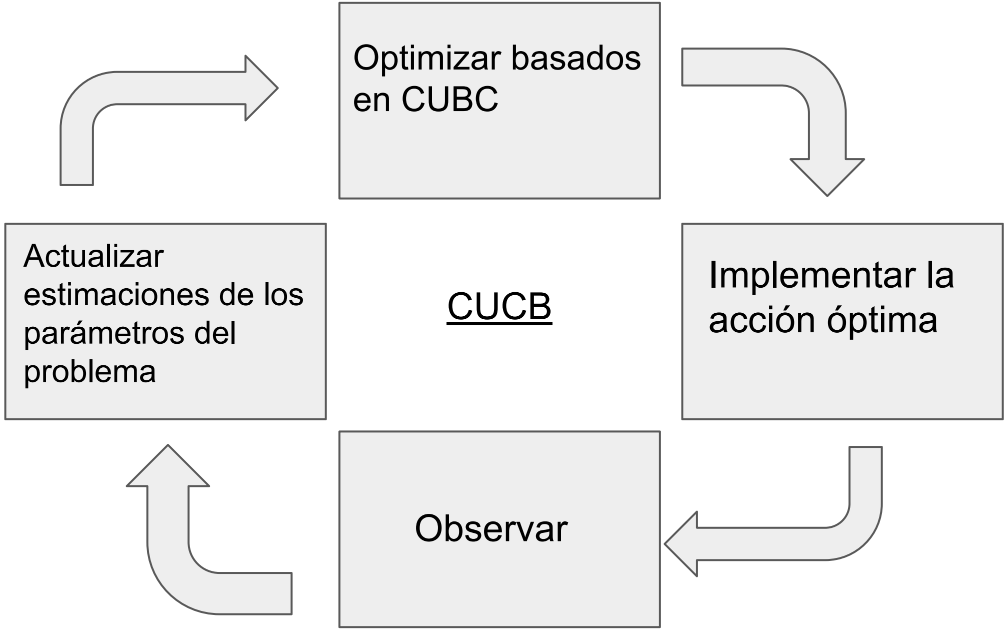

- Las máquinas (brazos) son las $X_{(i,j)}$
- Los conjuntos de máquinas admisibles (superbrazos) son las colecciones

  resultantes de elegir $B$ conjuntos a la izquierda.
- La idea es maximizar el valor esperado de elementos cubiertos

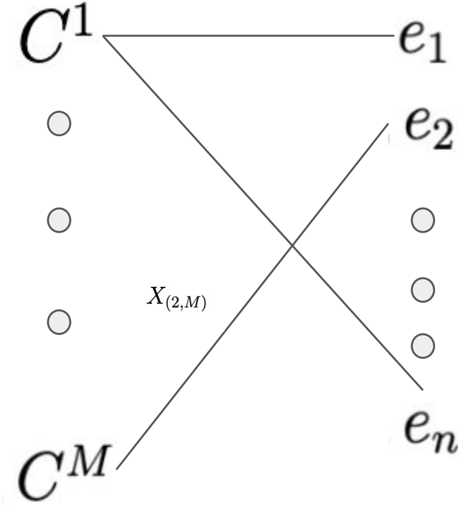
# Ejemplo 2: Combinatorial Upper Confidence Bound




$$S_k = \underset{i=1,\cdots,K}{\arg\max} \left\{ \overline{X}_{(i,j),T_{(i,j)}(k-1)} + c\sqrt{\frac{\ln k}{T_{(i,j)}(k-1)}} \right\}$$



# Ejemplo 2: Combinatorial Upper Confidence Bound




$$S_k = \underset{i=1,\cdots,K}{\arg\max} \left\{ \overline{X}_{(i,j),T_{(i,j)}(k-1)} + c\sqrt{\frac{\ln k}{T_{(i,j)}(k-1)}} \right\}$$

- Es igual que el UCB regular salvo que el argmax implica resolver una instancia del en en $(\mathcal{I})_\mu$ etapa.

- Es decir, en cada paso se actualiza el $\mu$ con la expresión de arriba y se resuelve $(\mathcal{I})_\mu$

¿Resolver iterativamente un problema NP- Hard?

# <u>Ejemplo 2: Oracle de aproximación</u> $\alpha - \beta$

La solución al problema

$$S_k = \underset{i=1,\cdots,K}{\arg\max} \left\{ \overline{X}_{(i,j),T_{(i,j)}(k-1)} + c\sqrt{\frac{\ln k}{T_{(i,j)}(k-1)}} \right\}$$

La reemplazamos con una aproximación

$$S_k = \mathrm{Oracle}(\mu)$$

Tal que la probabilidad de que el valor esperado de elementos cubiertos por $S_k$ sea menor que valor óptimo de $(\mathcal{I})_\mu$ menos $\alpha$ sea menor a $\beta$

## Ejemplo 2: Oracle de aproximación $\alpha - \beta$

Un oracle es una aproximación eficiente tal que la probabilidad de que sea mala es bajita

La solución al problema

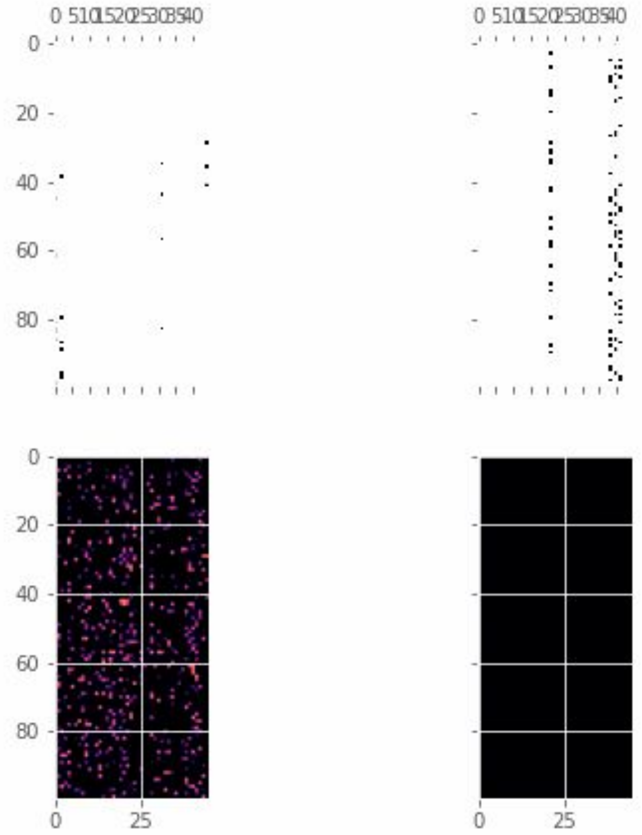
$$S_k = \underset{i=1,\cdots,K}{\arg\text{máx}} \left\{ \overline{X}_{(i,j),T_{(i,j)}(k-1)} + c\sqrt{\frac{\ln k}{T_{(i,j)}(k-1)}} \right\}$$

La reemplazamos con una aproximación

$$S_k = \text{Oracle}(\mu)$$

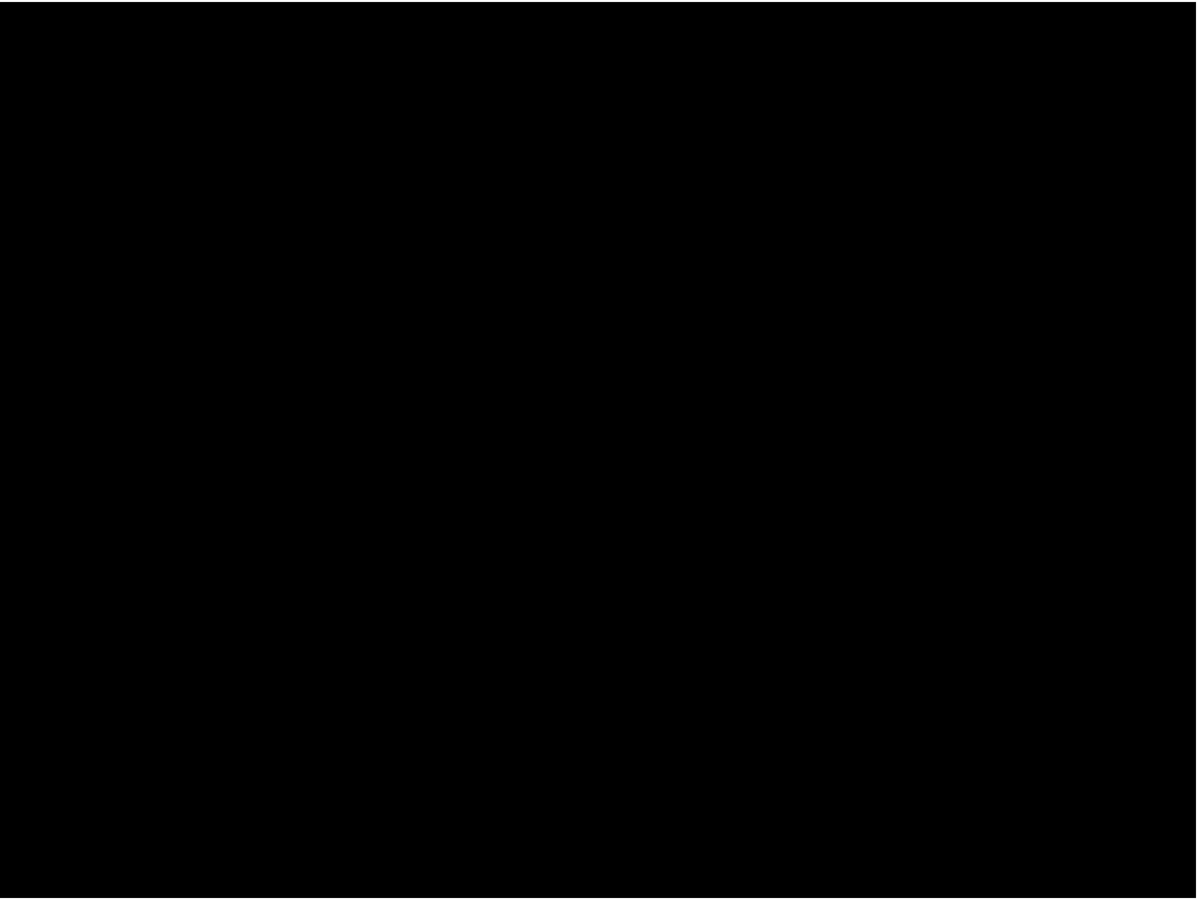
Tal que la probabilidad de que el valor esperado de elementos cubiertos por $S_k$ sea menor que valor óptimo de $(\mathcal{I})_\mu$ menos $\alpha$ sea menor a $\beta$

# Ejemplo 2: Demostración computacional



- Para 100 elementos se eligieron 45 conjuntos al azar (de tamaño pequeño) y se resolviò un CUCB de presupuesto B=4
- En el largo plazo, el error de la estimación de la matriz de probabilidades se acerca a cero
- En el largo plazo CUCB eligió muy frecuentemente el superbrazo óptimo

# Ejemplo 2: Demostración computacional



- Para 100 elementos se eligieron 45 conjuntos al azar (de tamaño pequeño) y se resolviò un CUCB de presupuesto B=4
- En el largo plazo, el error de la estimación de la matriz de probabilidades se acerca a cero
- En el largo plazo CUCB eligió muy frecuentemente el superbrazo óptimo

# CMAB en GSG: Nuestro problema comparte muchas cosas con un Probabilistic Maximum Coverage problem

Asuma que el cazador patrulla la reserva con una probabilidad fija e invariable $\alpha$
Con esto se puede calcular para cada sendero **i** la probabilidad de encontrar el cazador $\mu_i$

**Theorem 1.1.** *Si los guardabosques conocen el vector $\mu$ de probabilidades entonces elegir un $S$ que maximice $r_\mu(S)$ es equivalente a resolver el siguiente problema de optimización*

$$(PG)_\mu \begin{cases} \max\limits_{\eta \in \mathbb{R}^M, y \in \mathbb{R}^n} & \sum \mu_i (y_i - 1) \\ & s.t.: \\ & \sum_{j=1}^{M} \eta_j = B \\ & y_i = \sum_{j=1}^{M} C_i^j \eta_j \\ & \eta_j \in \{0,1\}, y \in \{0,1,\cdots,B\} \end{cases} \tag{1}$$

En este contexto $B$ es el número de guardabosques disponibles

# CMAB en GSG: Nuestro problema comparte muchas cosas con un Probabilistic Maximum Coverage problem

Asuma que el cazador patrulla la reserva con una probabilidad fija e invariable $\alpha$
Con esto se puede calcular para cada sendero **i** la probabilidad de encontrar el cazador $\mu_i$

$$(PG)_\mu \begin{cases} \max\limits_{\eta \in \mathbb{R}^M, y \in \mathbb{R}^n} & \sum \mu_i (y_i - 1) \\ & s.t.: \\ & \sum_{j=1}^{M} \eta_j = B \\ & y_i = \sum_{j=1}^{M} C_i^j \eta_j \\ & \eta_j \in \{0,1\}, y \in \{0, 1, \cdots, B\} \end{cases}$$

$$(\mathcal{I})_\mu \begin{cases} \max\limits_{x \in \mathbb{R}^M \; y \in \mathbb{R}^n} & \sum_{i=1}^{n} \mu_i y_i \\ & \text{s.t.:} \\ & \sum_{j=1}^{m} x_j = B \\ & \sum_{i \in C^j} x_j \geq \quad y_i \forall\, i \\ & x_j, y_i \in \{0,1\} \end{cases}$$

En este contexto $B$ es el número de guardabosques disponibles

# CMAB en GSG: Nuestro oracle le gana al oracle de MCP

**Theorem 1.2.** *Es equivalente resolver* $(PG)_\mu$ *que resolver su versión relajada sin las restricciones enteras*

# CMAB en GSG: Nuestro oracle le gana al oracle de MCP

**Theorem 1.2.** *Es equivalente resolver $(PG)_\mu$ que resolver su versión relajada sin las restricciones enteras*

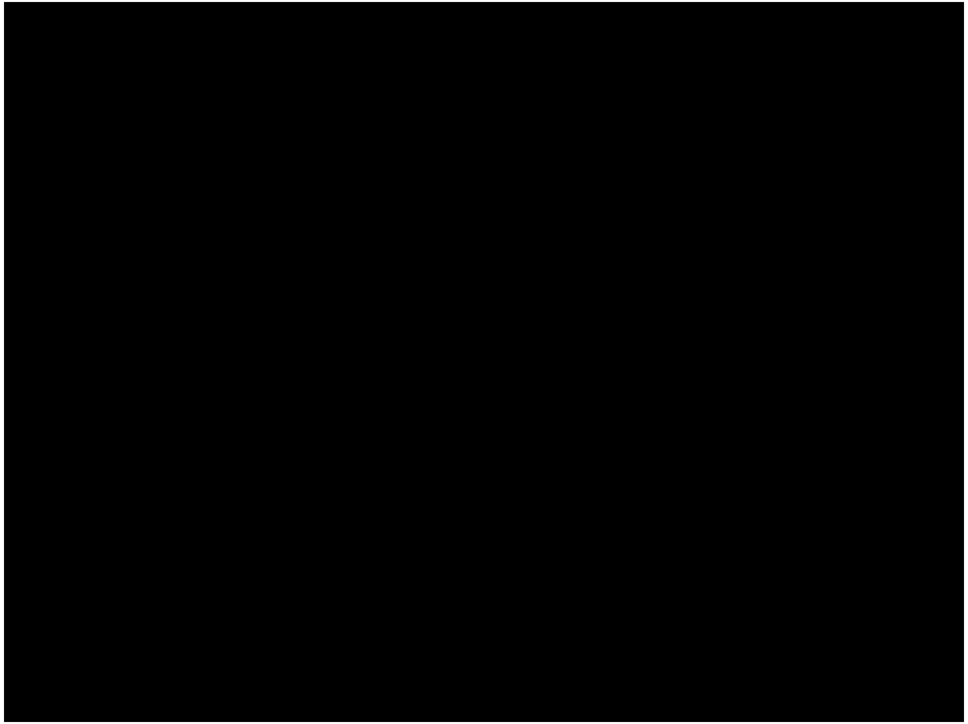
$$
(PG)_\mu \begin{cases} \displaystyle\max_{\eta\in\mathbb{R}^M, y\in\mathbb{R}^n} & \sum \mu_i(y_i - 1) \\ & s.t.: \\ & \sum_{j=1}^{M} \eta_j = B \\ & y_i = \sum_{j=1}^{M} C_i^j \eta_j \\ & \eta_j \in \cancel{\{}0, 1\cancel{\}}, y \in \cancel{\{}0, 1, \cdots, B\cancel{\}} \end{cases}
$$

## CMAB en GSG: Nuestro oracle le gana al oracle de MCP

**Theorem 1.2.** *Es equivalente resolver* $(PG)_\mu$ *que resolver su versión relajada sin las restricciones enteras*

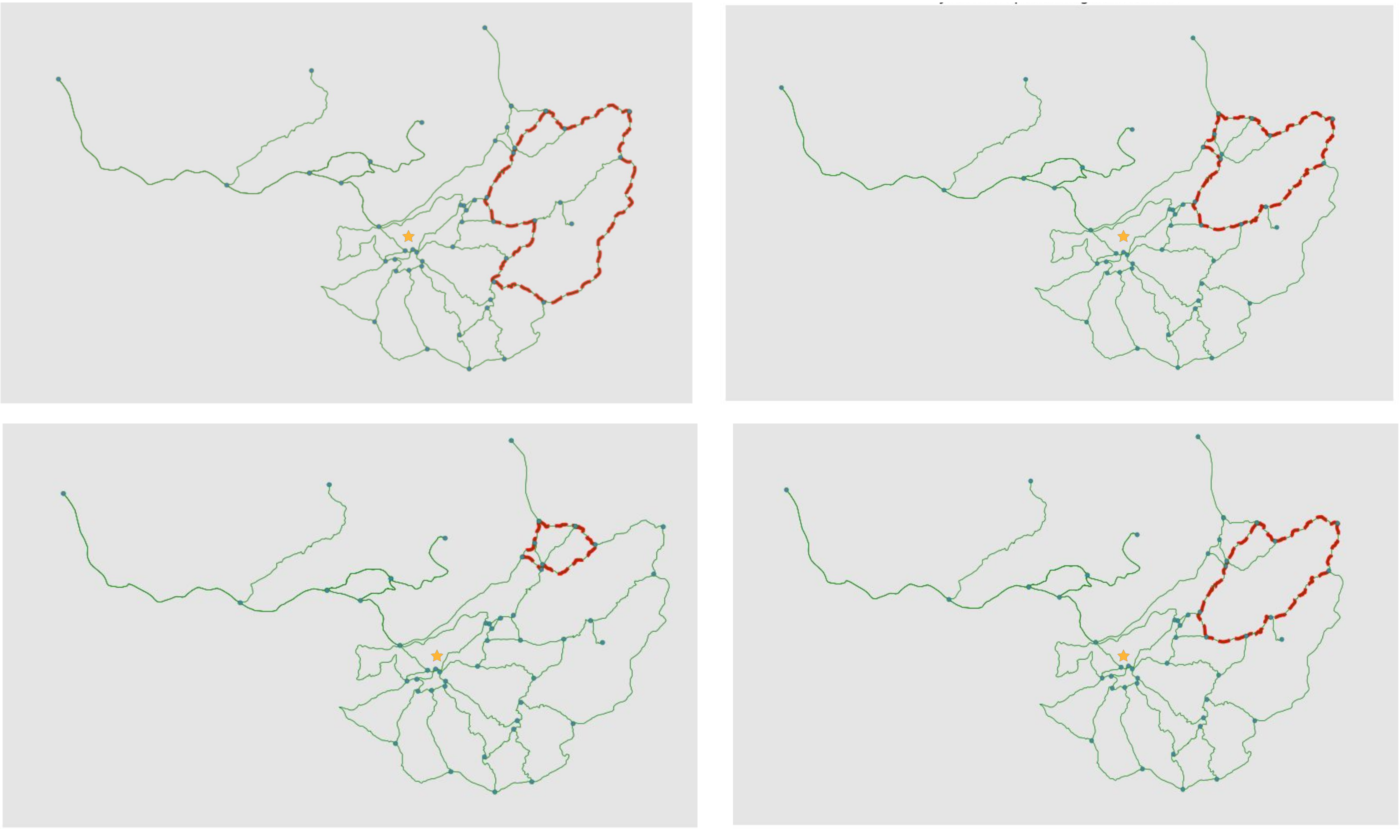
⇒ Podemos utilizar CUBC para generar sugerencias de rutas para uno o más guardabosques

# CMAB en GSG: Ejemplo computacional



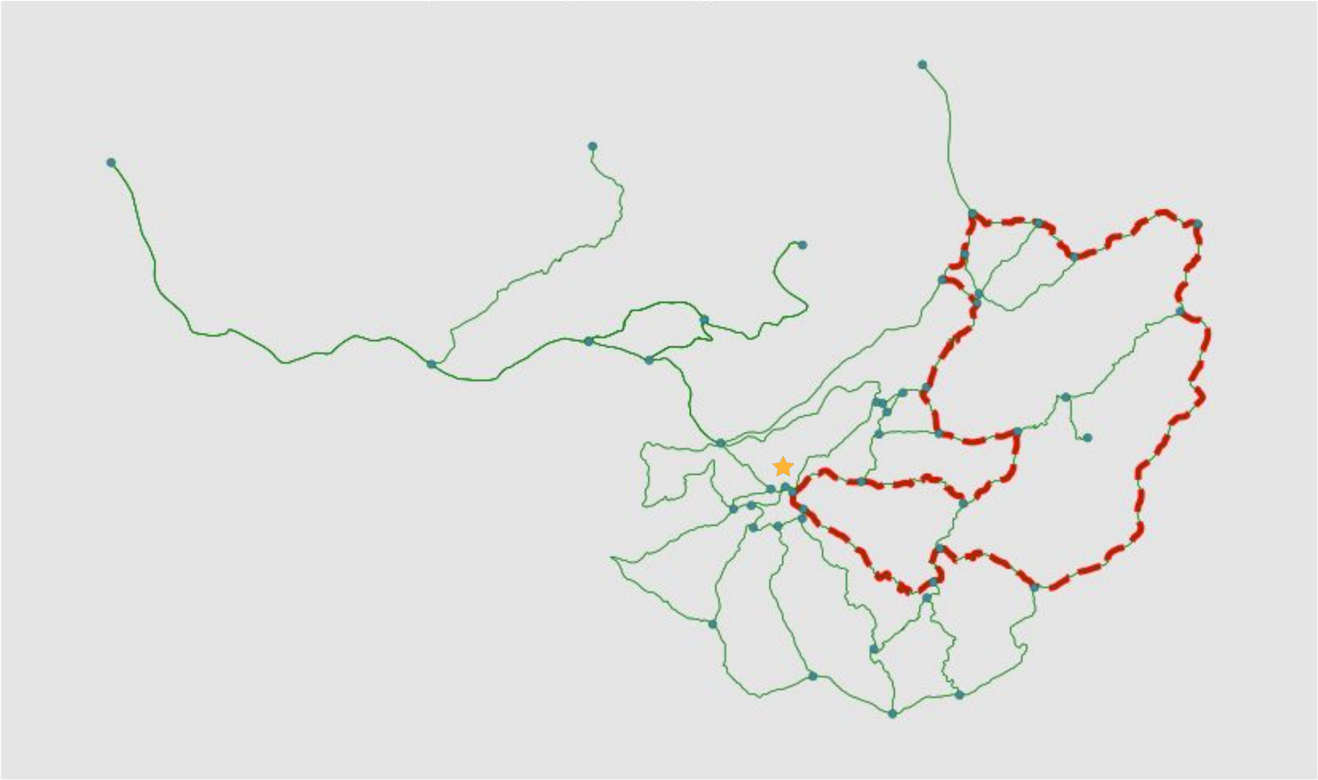

Gracias

# CMAB en GSG: Demostración computacional



El cazador visita estas rutas con alguna frecuencia desconocida para el guardabosques. Estos ciclos son también desconocidos para él

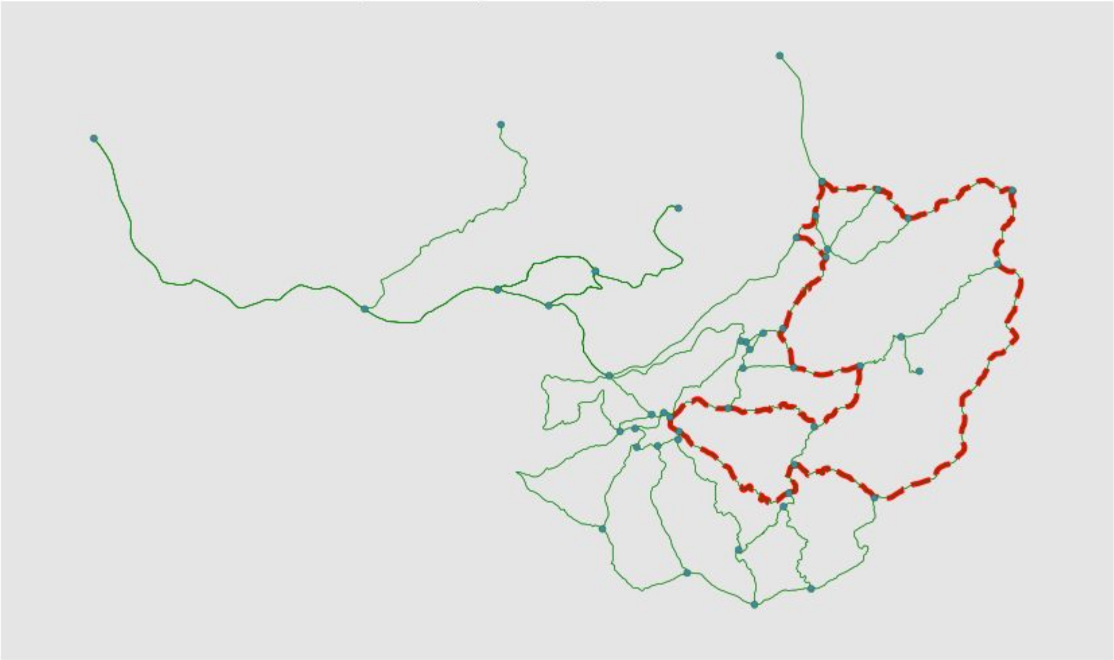
# CMAB en GSG: Demostración computacional



De conocer las probabilidades, esta sería la respuesta óptima del cazador

# CMAB en GSG: Demostración computacional

Jama Coaque Ecological Reserve



De conocer las probabilidades, esta sería la respuesta óptima del cazador