

# **Multi-armed bandit for selection of binarization scheme in metaheuristics**

**Seminario de Tesis**

Pablo Ábrego - 6 de junio 2022

# Multi-armed bandit for selection of binarization scheme in metaheuristics

## Objetivo

Utilizar un algoritmo de **selección** que proponga **esquemas de binarización** en **algoritmos metaheurísticos** utilizando una técnica de **aprendizaje reforzado**, con el fin de obtener mejores resultados en la optimización de **problemas combinatorios**.

# Objetivo

Utilizar un algoritmo de **selección** que proponga **esquemas de binarización** en **algoritmos metaheurísticos** utilizando una técnica de **aprendizaje reforzado**, con el fin de obtener mejores resultados en la optimización de **problemas combinatorios**.

**algoritmos  
metaheurísticos**

**esquemas de  
binarización**

**aprendizaje reforzado  
(selección)**

**problemas  
combinatorios**

# Temas a tratar

**algoritmos  
metaheurísticos**

**Black Widow Optimization**

**esquemas de  
binarización**

**Función de transferencia  
Función de binarización**

**aprendizaje reforzado  
(selección)**

**Multi-armed bandit**

**problemas  
combinatorios**

**Set Covering  
Problem**

# Temas a tratar

**algoritmos  
metaheurísticos**

**Black Widow Optimization**

**esquemas de  
binarización**

**Función de transferencia  
Función de binarización**

**aprendizaje reforzado  
(selección)**

**Multi-armed bandit**

**problemas  
combinatorios**

**Set Covering  
Problem**

# Black Widow Optimization

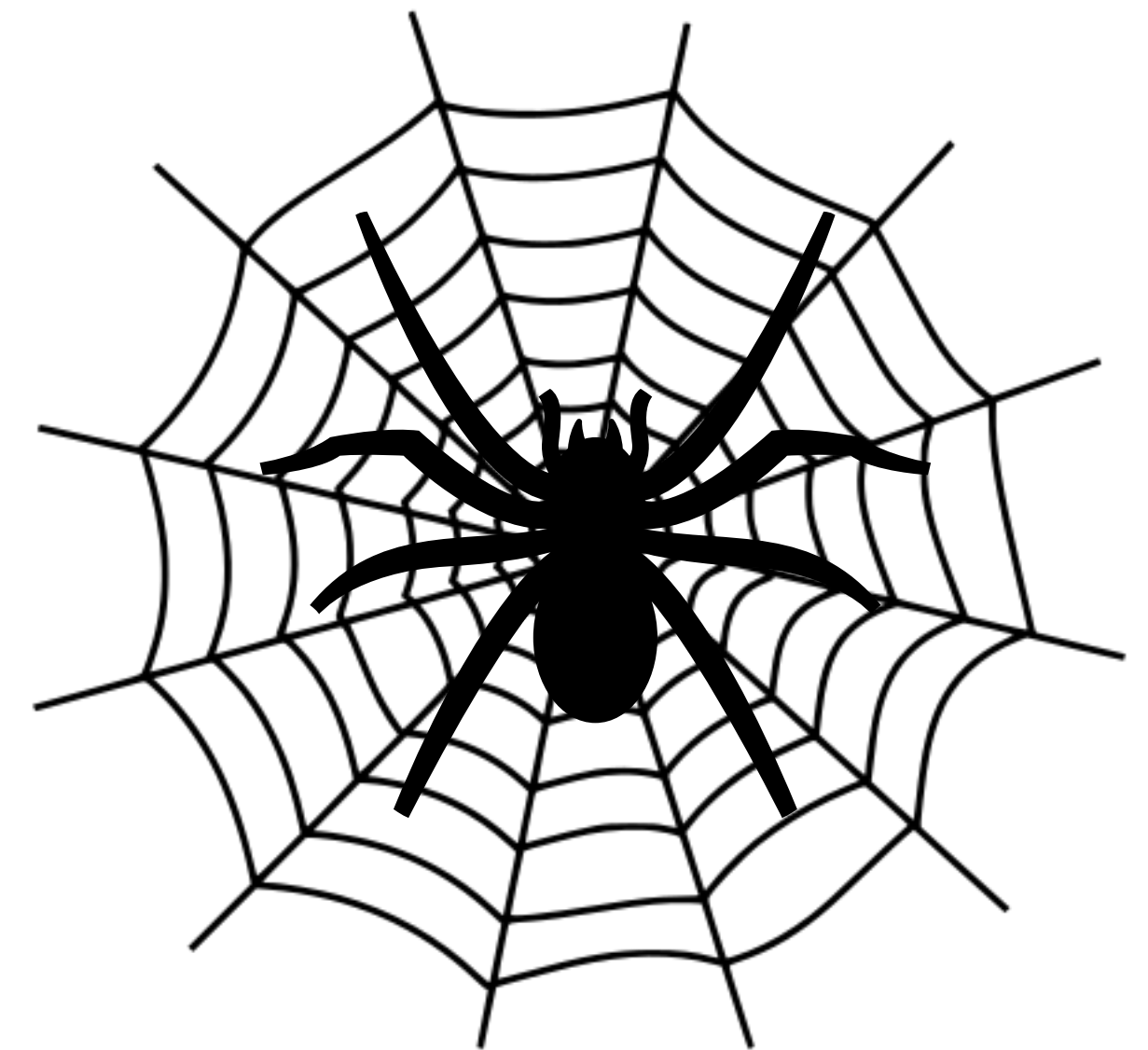
Viuda negra y estilo de vida

## Apareamiento

- Deja marcas de feromonas en su red para atraer al macho.
- Se aparea con el macho mientras se canibaliza.

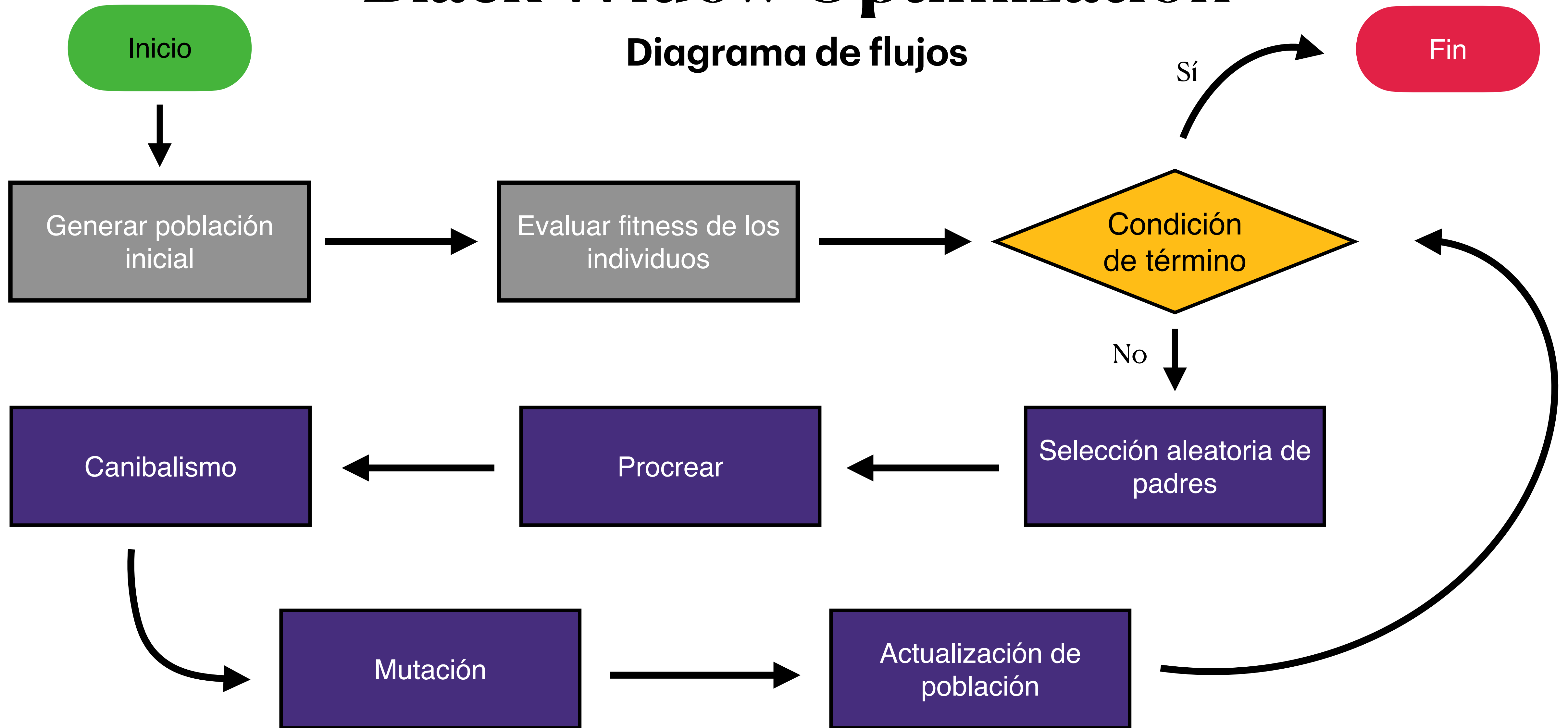
## Canibalismo

- Los hijos nuevos se canibalizan entre ellos.
- Sobreviven los más fuertes (mejor fitness).



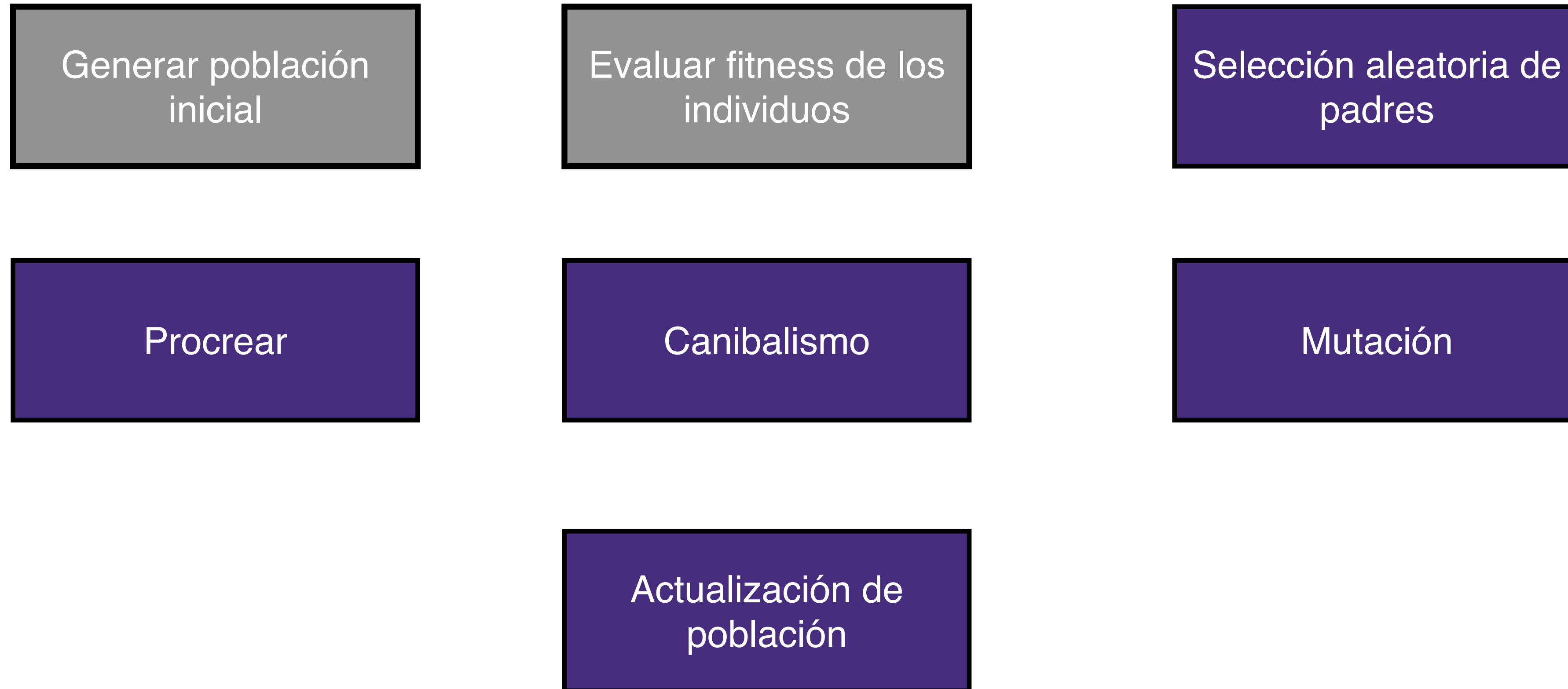
# Black Widow Optimization

## Diagrama de flujos



# Black Widow Optimization

## Diagrama de flujos





# Black Widow Optimization

Generar población inicial

$$Widow = [x_1, x_2, \dots, x_{Nvar-1}, x_{Nvar}]$$

$x_i \rightarrow$  Cromosoma

$$[ x_i = rand(0,1) ]$$

# Black Widow Optimization

Evaluar fitness de los individuos

$$Widow = [x_1, x_2, \dots, x_{Nvar-1}, x_{Nvar}] \quad [ x_i = rand(0,1) ]$$

$$Fitness(Aptitud) = f(widow) = f(x_1, x_2, \dots, x_{Nvar})$$

# Black Widow Optimization

Selección aleatoria de padres

$$\times \rightarrow A \times X = [a_1x_1, a_2x_2, \dots, a_nx_n] \quad \sim \rightarrow \tilde{A} = [1 - a_1, 1 - a_2, \dots, 1 - a_n]$$

$$X_1 = \text{Population} [ \text{rand}(0, N_{\text{pop}}) ]$$

$$X_2 = \text{Population} [ \text{rand}(0, N_{\text{pop}}) ]$$

# Black Widow Optimization

Procrear

$$\times \rightarrow A \times X = [a_1x_1, a_2x_2, \dots, a_nx_n]$$

$$\sim \rightarrow \tilde{A} = [1 - a_1, 1 - a_2, \dots, 1 - a_n]$$

$$X_1 = \text{Population} [ \text{rand}(0, N_{\text{pop}}) ]$$

$$X_2 = \text{Population} [ \text{rand}(0, N_{\text{pop}}) ]$$

$$\alpha = [a_1, a_2, \dots, a_{N_{\text{var}}}]$$

$$0 \leq a_i \leq 1$$

$$Y_1 = \alpha \times X_1 + \tilde{\alpha} \times X_2$$

$$Y_2 = \alpha \times X_2 + \tilde{\alpha} \times X_1$$

# Black Widow Optimization

Canibalismo

$X_1 = \text{Population} [ \text{rand}(0, N_{\text{pop}}) ]$        $X_2 = \text{Population} [ \text{rand}(0, N_{\text{pop}}) ]$

$\text{Fitness}(X_1) > \text{Fitness}(X_2)$

$\text{Hijos} = [y_1, y_2, y_3, \dots, y_n]$

$\text{NuevosHijos} = \text{maxFitness}(\text{Hijos})$

# Black Widow Optimization

Mutación

$$M_1 = Population [ rand(0, Npop) ]$$

$$M_1 = [ x_1 | x_2 | x_3 | \dots | x_{Nvar-1} | x_{Nvar} ]$$

$$M_1 = [ x_1 | x_{Nvar-1} | x_3 | \dots | x_2 | x_{Nvar} ]$$

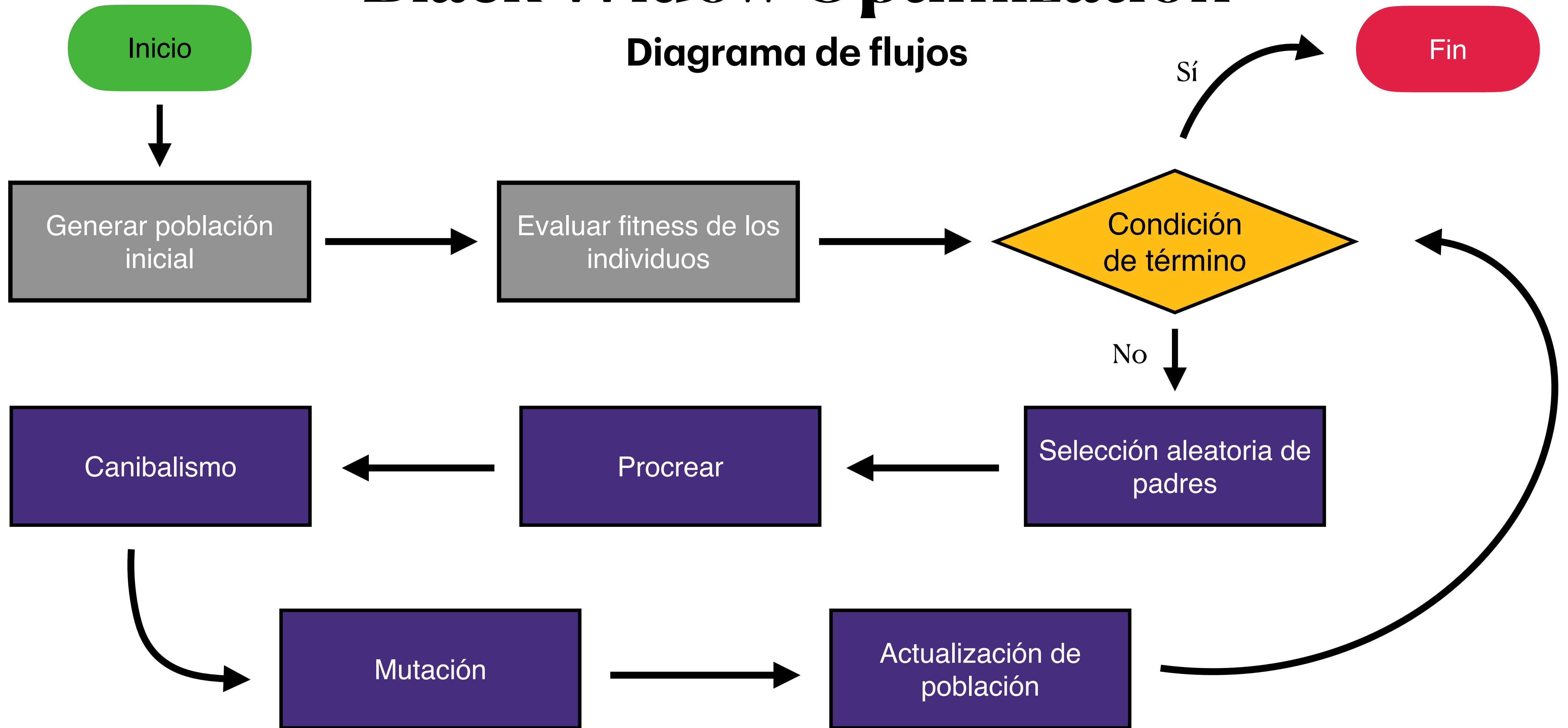
# Black Widow Optimization

Actualización de población

*$NuevaPop = maxFitness(Population)$*

# Black Widow Optimization

## Diagrama de flujos





# Temas a tratar

**algoritmos  
metaheurísticos**

**Black Widow Optimization**

**esquemas de  
binarización**

**Función de transferencia  
Función de binarización**

**aprendizaje reforzado  
(selección)**

**Multi-armed bandit**

**problemas  
combinatorios**

**Set covering  
Problem**

# Temas a tratar

**algoritmos  
metaheurísticos**

**Black Widow Optimization**

**esquemas de  
binarización**

**Función de transferencia  
Función de binarización**

**aprendizaje reforzado  
(selección)**

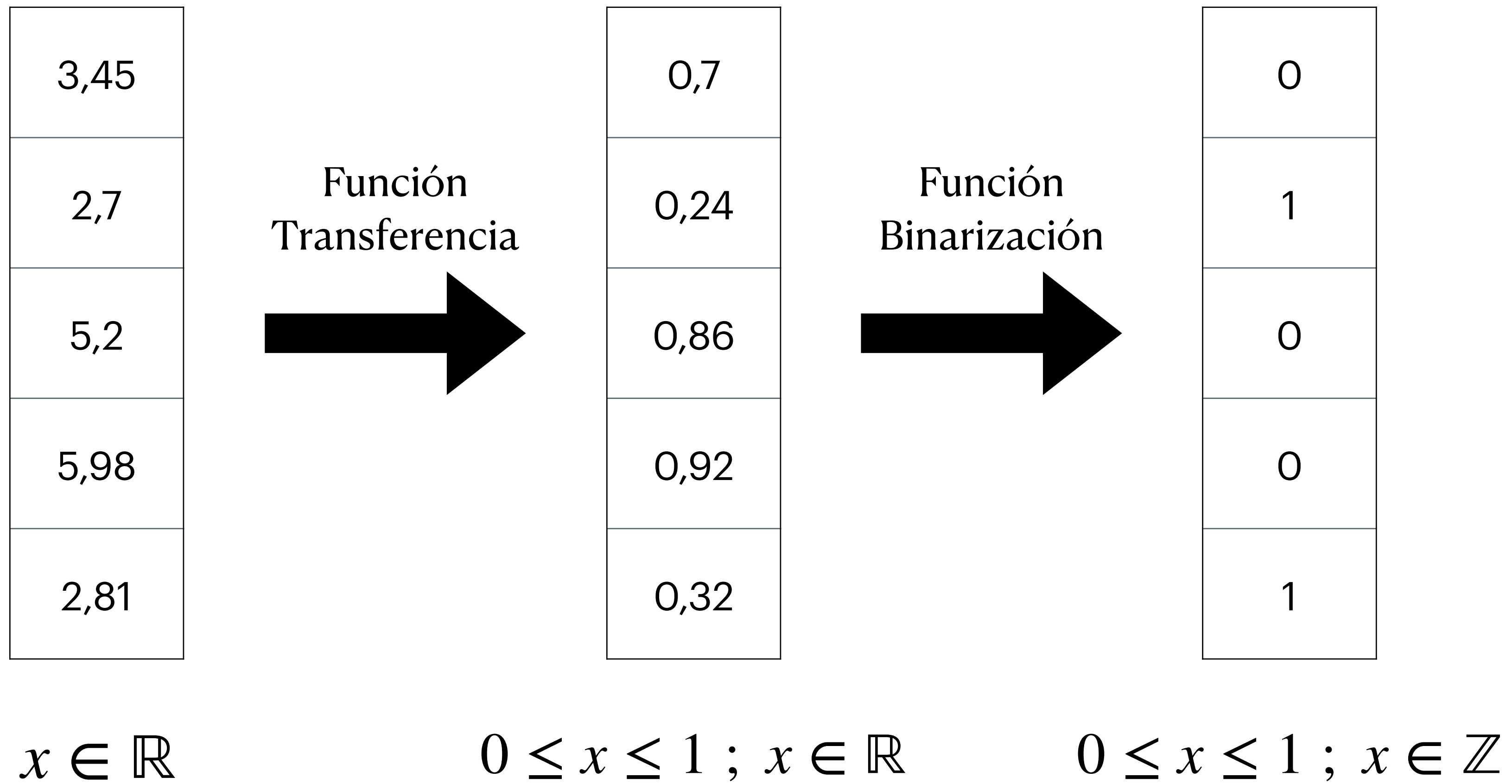
**Multi-armed bandit**

**problemas  
combinatorios**

**Set covering  
Problem**

# Técnicas de binarización

## Técnica de dos pasos



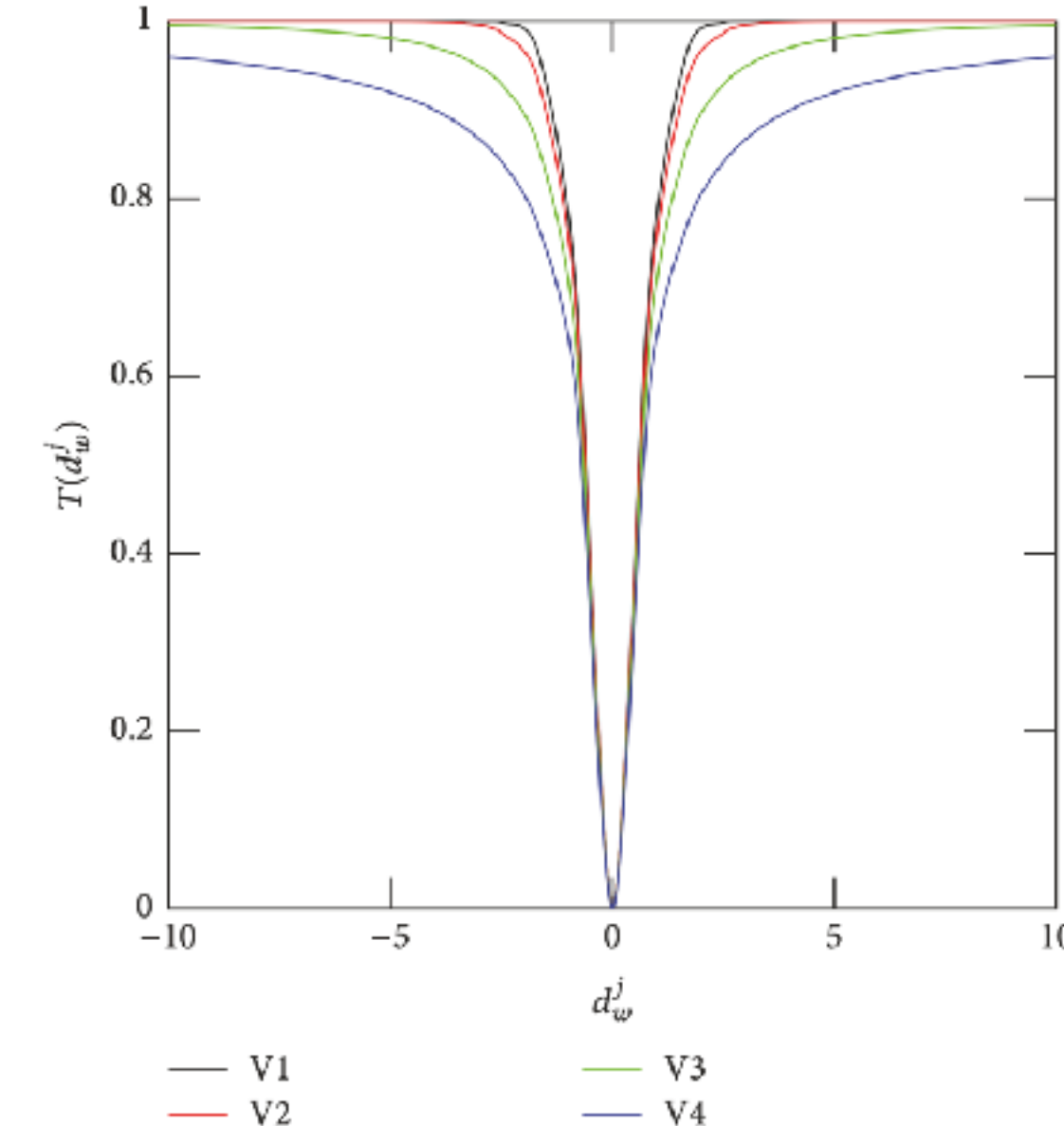
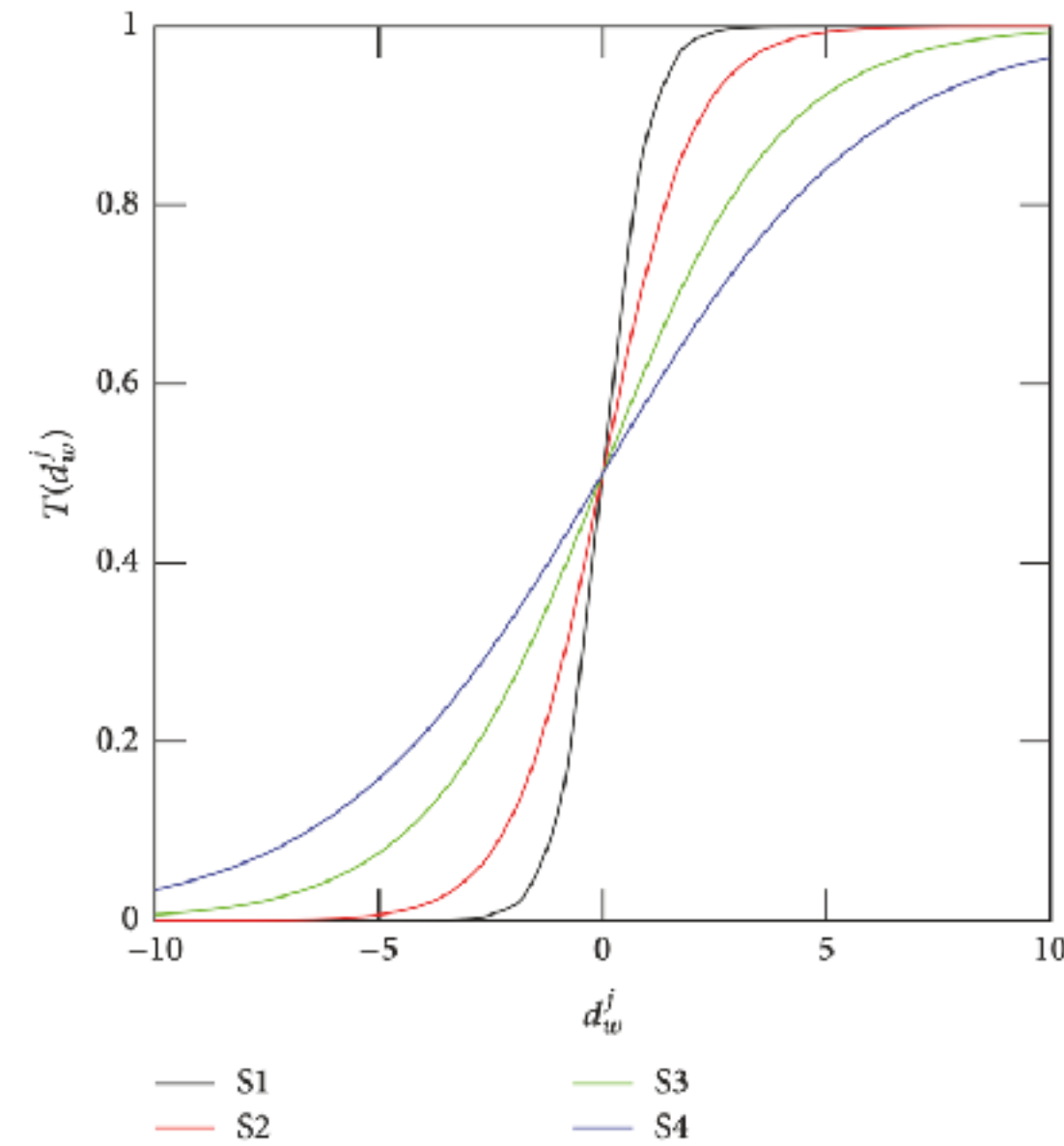
# Técnicas de binarización

## Técnica de dos pasos

### Función Transferencia

3,45
2,7
...
5,98
2,81

$x \in \mathbb{R}$



0,7
0,24
...
0,92
0,32

$0 \leq x \leq 1 ; x \in \mathbb{R}$

# Técnicas de binarización

## Técnica de dos pasos

### Función Binarización

0,7
0,24
...
0,92
0,32

$$0 \leq x \leq 1 ; x \in \mathbb{R}$$

Standard	$X_{new}^j = \begin{cases} 1 & \text{if } rand \leq T(d_w^j) \\ 0 & \text{else.} \end{cases}$
Complement	$X_{new}^j = \begin{cases} X_w^j & \text{if } rand \leq T(d_w^j) \\ 0 & \text{else.} \end{cases}$
Static Probability	$X_{new}^j = \begin{cases} 0 & \text{if } T(d_w^j) \leq \alpha \\ X_w^j & \text{if } \alpha < T(d_w^j) \leq \frac{1}{2}(1 + \alpha) \\ 1 & \text{if } T(d_w^j) \geq \frac{1}{2}(1 + \alpha) \end{cases}$
Elitist	$X_{new}^j = \begin{cases} X_{Best}^j & \text{if } rand < T(d_w^j) \\ 0 & \text{else.} \end{cases}$
Roulette Elitist	$X_{new}^j = \begin{cases} P[X_{new}^j = \zeta_j] = \frac{f(\zeta)}{\sum_{\delta \in Q_g} f(\delta)} & \text{if } rand \leq T(d_w^j) \\ P[X_{new}^j = 0] = 1 & \text{else.} \end{cases}$

0
1
0
...
0
1

$$0 \leq x \leq 1 ; x \in \mathbb{Z}$$

# Temas a tratar

**algoritmos  
metaheurísticos**

**Black Widow Optimization**

**esquemas de  
binarización**

**Función de transferencia  
Función de binarización**

**aprendizaje reforzado  
(selección)**

**Multi-armed bandit**

**problemas  
combinatorios**

**Set covering  
Problem**

# Temas a tratar

**algoritmos  
metaheurísticos**

**Black Widow Optimization**

**esquemas de  
binarización**

**Función de transferencia  
Función de binarización**

**aprendizaje reforzado  
(selección)**

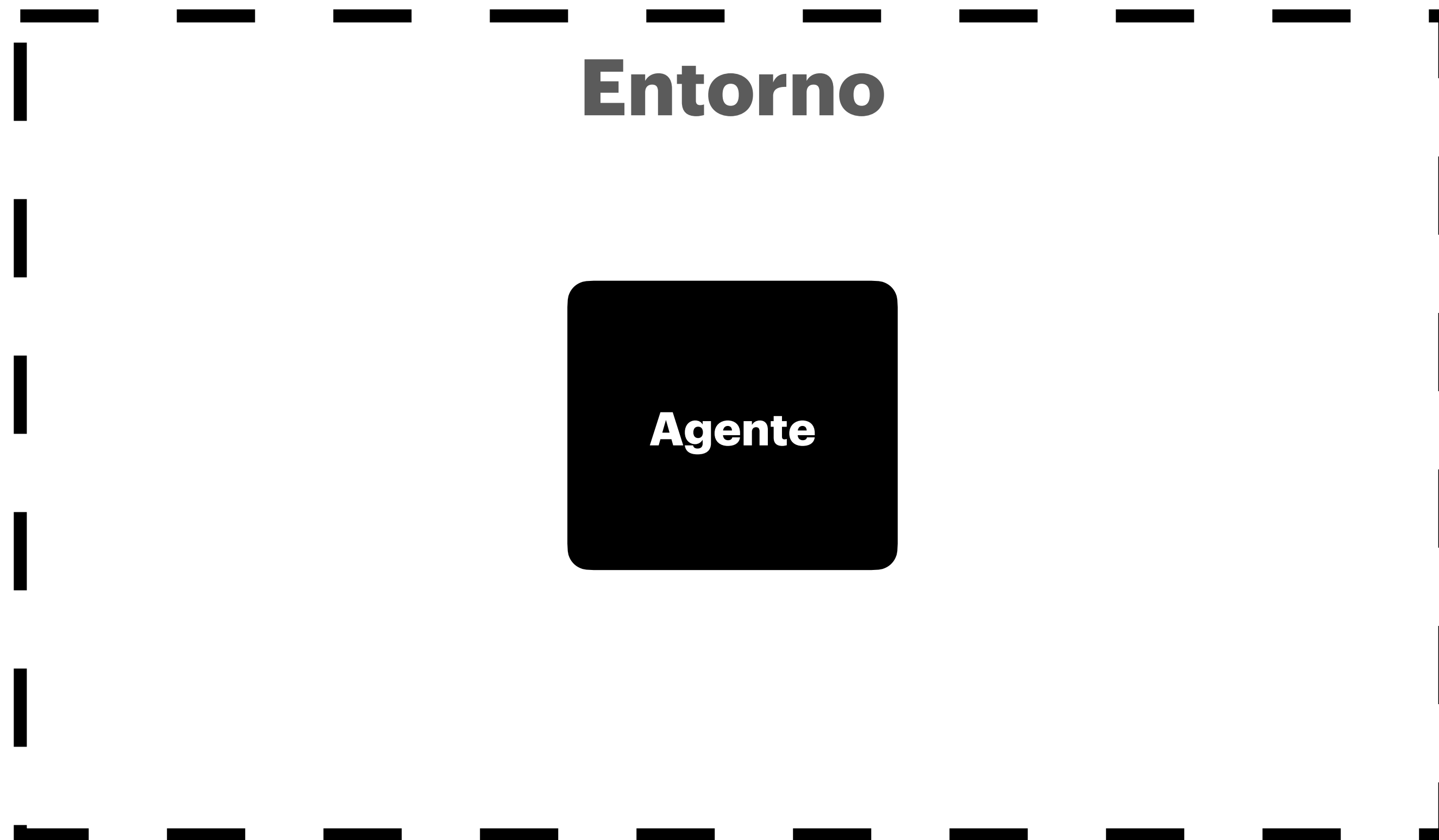
**Multi-armed bandit**

**problemas  
combinatorios**

**Set covering  
Problem**

# Multi-armed bandit

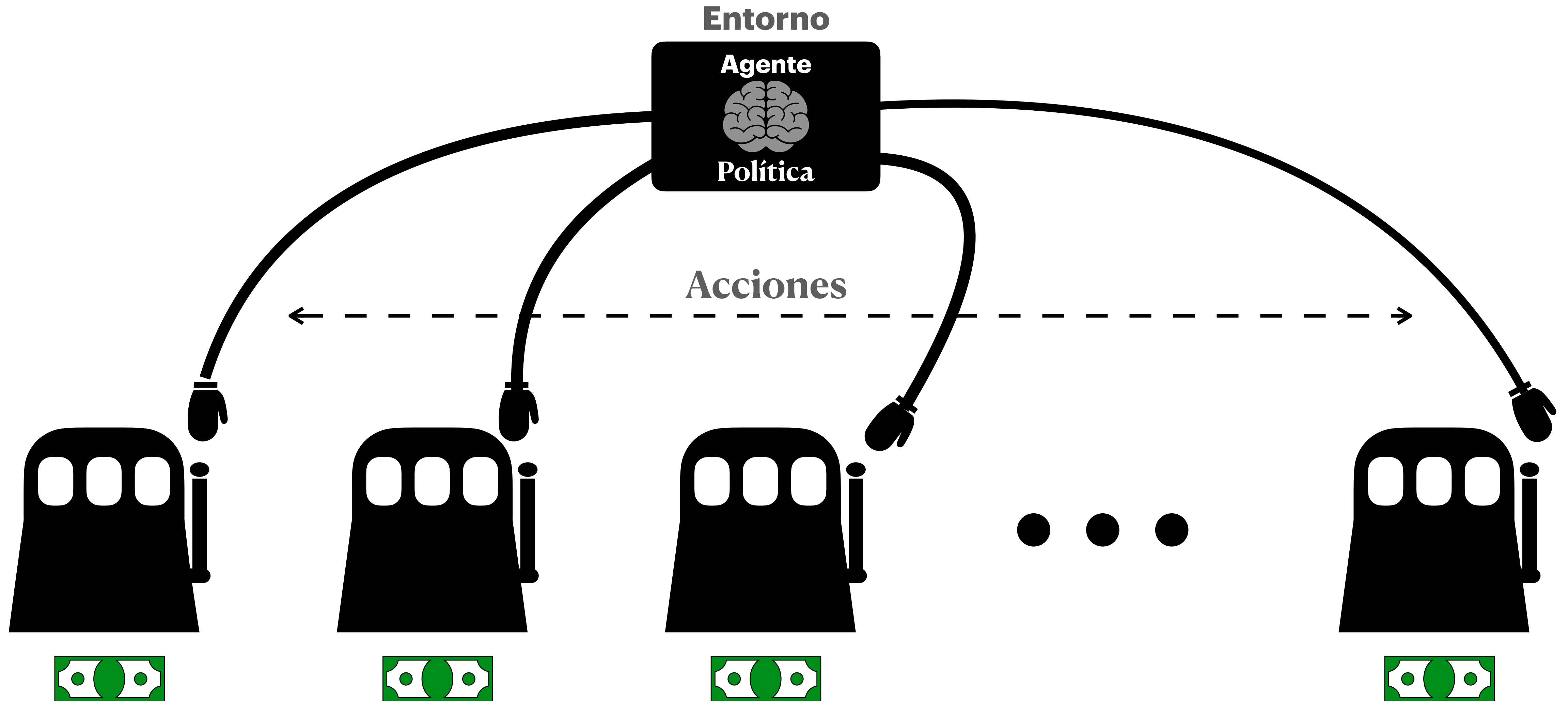
Reinforcement Learning





# Multi-armed bandit

## Reinforcement Learning



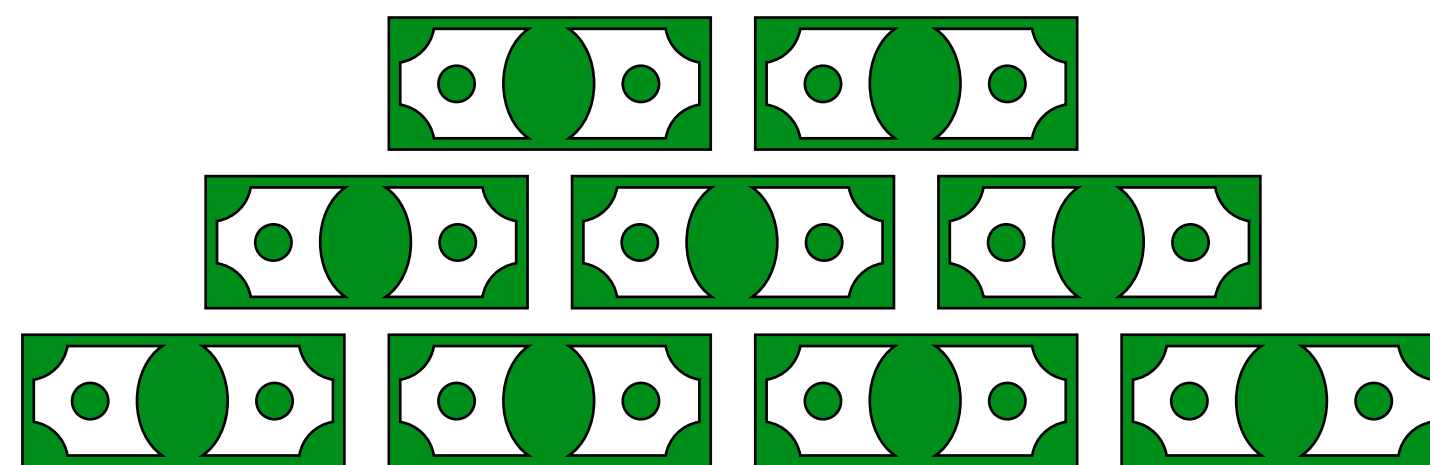
# Multi-armed bandit

## Reinforcement Learning

Entorno



Recompensa



**Política**

**Multi-armed bandit**

**Función de valor**

**Política**

**Función de valor**

**Multi-armed bandit**

**Random**

$$A_t \doteq \mathit{rand}(0, k)$$

**Política**

**Función de valor**

**Multi-armed bandit**

**Random**

**$\mathcal{E}$  - greedy**

$$A_t \doteq \text{rand}(0, k)$$

$$Q_t(a) \doteq \frac{\sum_{i=1}^{t-1} R_i \cdot 1_{A_i=a}}{\sum_{i=1}^{t-1} 1_{A_i=a}}$$

$$A_t \doteq \text{argmax}_a Q_t(a)$$

$$\varepsilon = \mathbb{P}(A_t \doteq \text{rand}(0, k))$$

**Política**

**Función de valor**

**Multi-armed bandit**

**Random**

**Upper Confidence  
Bound**

**$\mathcal{E}$  - greedy**

$$A_t \doteq \text{rand}(0, k)$$

$$A_t \doteq \operatorname{argmax}_a \left[ Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$

$$Q_t(a) \doteq \frac{\sum_{i=1}^{t-1} R_i \cdot 1_{A_i=a}}{\sum_{i=1}^{t-1} 1_{A_i=a}}$$

$$A_t \doteq \operatorname{argmax}_a Q_t(a)$$

$$\varepsilon = \mathbb{P}(A_t \doteq \text{rand}(0, k))$$

**Política**

**Función de valor**

**Multi-armed bandit**

**Random**

**Upper Confidence  
Bound**

**$\mathcal{E}$  - greedy**

$$A_t \doteq \text{rand}(0, k)$$

$$A_t \doteq \operatorname{argmax}_a \left[ Q_t(a) + c \sqrt{\frac{\ln t}{N_t(a)}} \right]$$

$$Q_t(a) \doteq \frac{\sum_{i=1}^{t-1} R_i \cdot 1_{A_i=a}}{\sum_{i=1}^{t-1} 1_{A_i=a}}$$

$$A_t \doteq \operatorname{argmax}_a Q_t(a)$$

$$\varepsilon = \mathbb{P}(A_t \doteq \text{rand}(0, k))$$

$t$  : Numero de iteración

$R_t$  : Recompensa en iteración  $t$

$k$  : Cantidad de brazos

$Q_t(a)$  : Recompensa media del brazo 'a'

$A_t$  : Brazo seleccionado en  $t$

$c$  : Estimador de confianza

$N_t(a)$  : Veces que se ha accionado 'a'

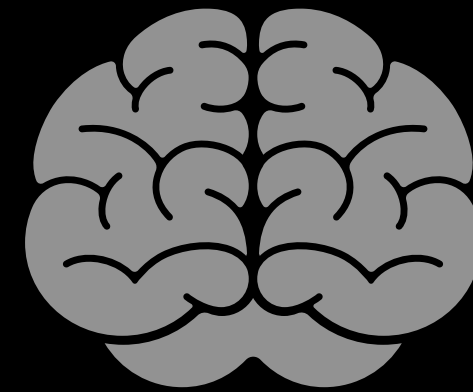
$1_{\text{predicado}}$  : Vale uno si el predicado es verdadero.  
Cero, en caso contrario.

# Multi-armed bandit

Reinforcement Learning

**Entorno**

**Agente**



**Upper Confidence  
Bound**



# **Multi-armed bandit**

**Dynamic Version**

# Multi-armed bandit

## Dynamic Version

Es muy común que un brazo comience a ser menos eficiente a lo largo del experimento.  
MAB puede tardar mucho en comenzar a elegir otros brazos.

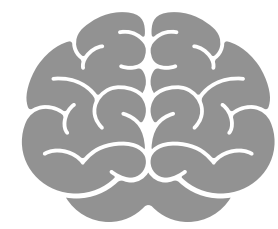
**Page-Hinkley test**  $\longrightarrow$   $\lambda$  : Indica si surgen comportamientos atípicos

$$\mathit{maxDev}_a(t) - \mathit{avgDev}_a(t) > \lambda$$



**[ Reiniciar recompensas y registros de confianza ]**

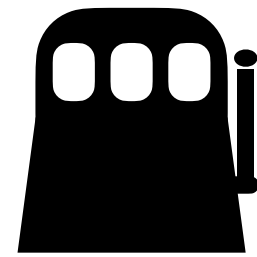
# Dynamic Multi-armed bandit for binarization scheme selection



Política



UCB + PH-test

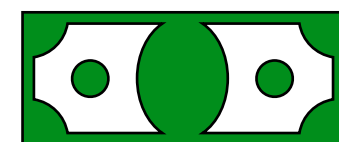


Brazo



Esquema de binarización

Función de transferencia + Función de binarización



Recompensa



Porcentaje de mejora de fitness

$$r = 100 \cdot \left[ \frac{f_{old} - f_{new}}{f_{old}} \right]$$

# Temas a tratar

**algoritmos  
metaheurísticos**

**Black Widow Optimization**

**esquemas de  
binarización**

**Función de transferencia  
Función de binarización**

**aprendizaje reforzado  
(selección)**

**Multi-armed bandit**

**problemas  
combinatorios**

**Set covering  
Problem**

# Temas a tratar

**algoritmos  
metaheurísticos**

**Black Widow Optimization**

**esquemas de  
binarización**

**Función de transferencia  
Función de binarización**

**aprendizaje reforzado  
(selección)**

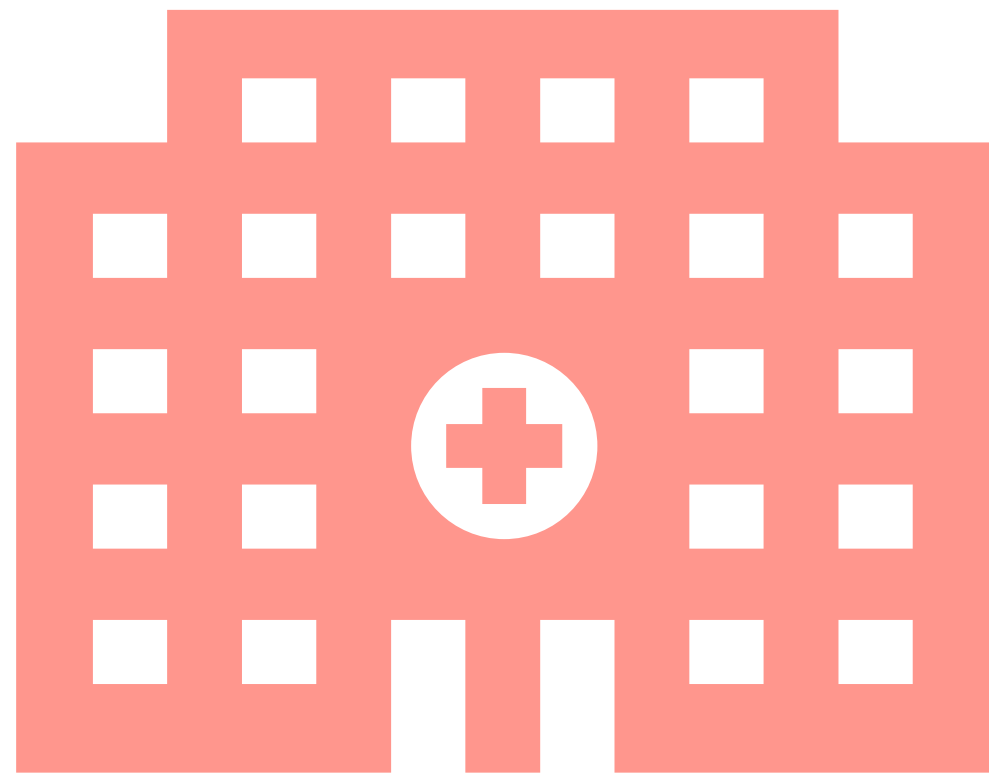
**Multi-armed bandit**

**problemas  
combinatorios**

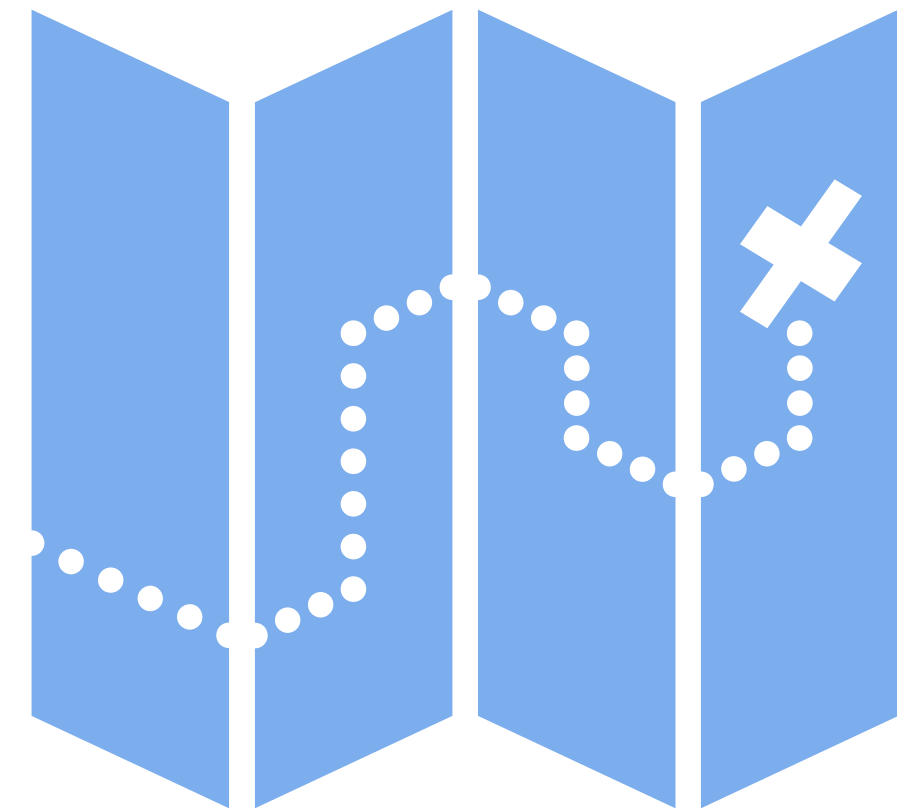
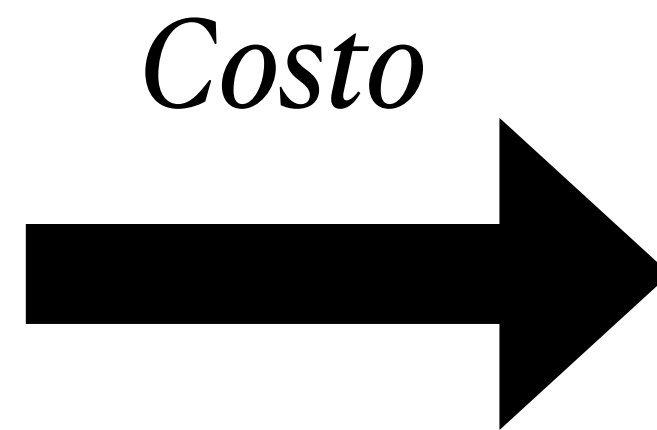
**Set covering  
Problem**

# Set Covering Problem

Problema combinatorial



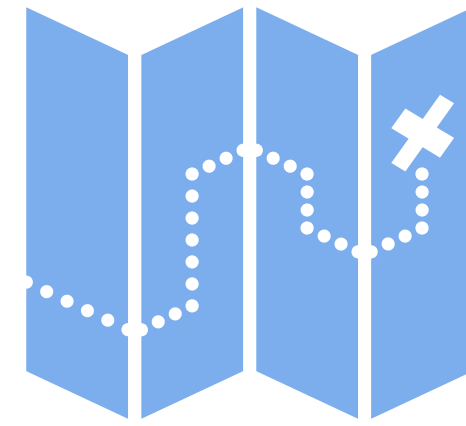
Hospitales  
*Covertura*



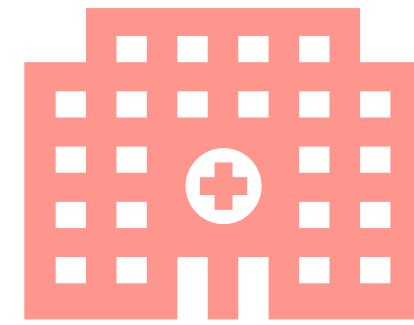
Zonas  
*Vecinas*

# Set Covering Problem

Problema combinatorial



Zonas =  $\{1, 2, 3, \dots, n-1, n\}$



Hospitales =  $\{1, 2, 3, \dots, m-1, m\}$

*Cobertura: Ubicación del hospital y sus zonas vecinas*

# Set Covering Problem

## Modelo matemático

### *Instancia*

$$A = \left\{ \begin{array}{ccc} a_{1,1} & a_{1,2} & \dots \\ \vdots & a_{i,j} & \\ a_{n,1} & & a_{n,m} \end{array} \right\} \quad a_{i,j} : 1, \text{ si el hospital } j \text{ cubre la zona } i$$
$$C = \{c_1, c_2, \dots, c_m\} \quad c_i : \text{ Costo de seleccionar hospital } i$$

### *Variable de decisión*

$$X = \{x_1, x_2, \dots, x_m\} \quad x_i : 1, \text{ Si se selecciona hospital } i$$



# Set Covering Problem

## Modelo matemático

$$A = \left\{ \begin{array}{ccc} a_{1,1} & a_{1,2} & \dots \\ \vdots & a_{i,j} & \\ a_{n,1} & & a_{n,m} \end{array} \right\}$$

$$C = \{c_1, c_2, \dots, c_m\}$$

$$X = \{x_1, x_2, \dots, x_m\}$$

$$\text{Minimize } \textit{Fitness} = \sum_{j=1}^N c_j \cdot x_j$$

$$\textit{s . t} \quad \sum_{j=1}^N a_{i,j} \cdot x_j \geq 1 \quad x_j \in \{0,1\}$$

# Temas a tratar

**algoritmos  
metaheurísticos**

**Black Widow Optimization**

**esquemas de  
binarización**

**Función de transferencia  
Función de binarización**

**aprendizaje reforzado  
(selección)**

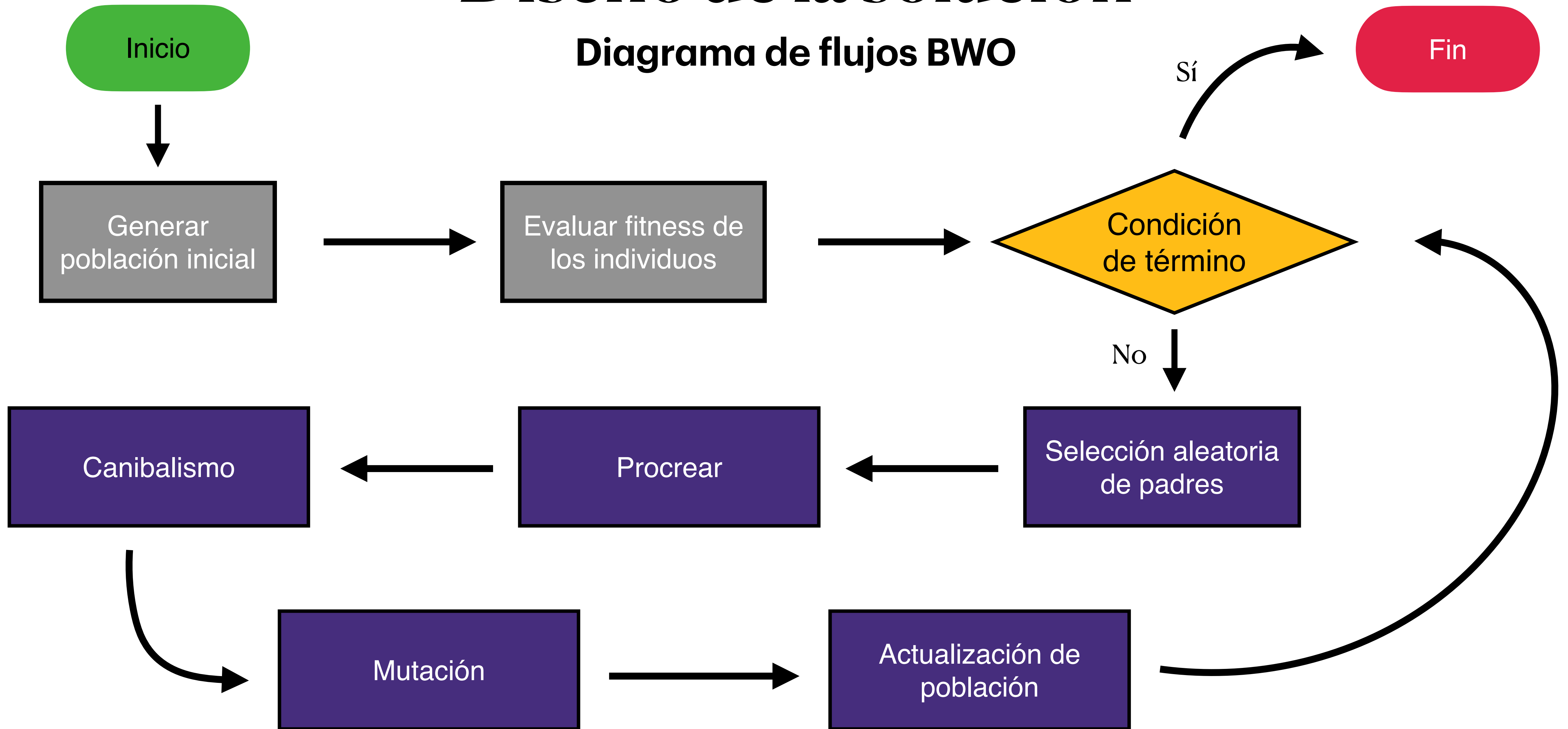
**Multi-armed bandit**

**problemas  
combinatorios**

**Set covering  
Problem**

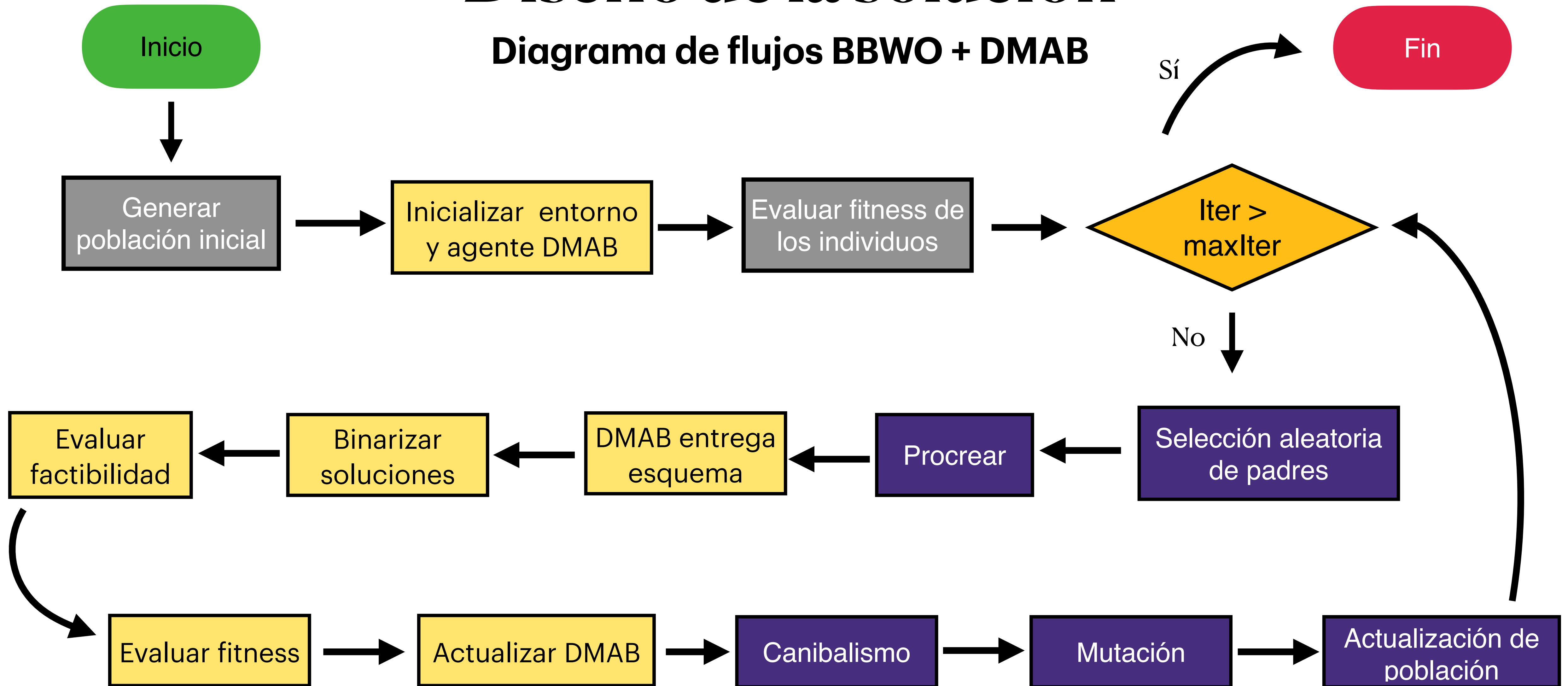
# Diseño de la solución

## Diagrama de flujos BWO



# Diseño de la solución

## Diagrama de flujos BBWO + DMAB



# Diseño de la solución

## BBWO + DMAB

---

### Algorithm 1 Binary Black Widow Optimizer

---

**Input:** Population  $X = \{x_1, x_2, \dots, x_{n_{pop}}\}$ , params  $\{pp, cr, pm, maxIter\}$

**Output:** Updated population  $X' = \{x'_1, x'_2, \dots, x'_{n_{pop}}\}$  and  $X_{best}$

Set de parámetros	{	1: Set procreation number $Pn \rightarrow n_{pop} * pp$ 2: Set mutation number $Mn \rightarrow n_{pop} * pm$ 3: Set survivors number $Sn \rightarrow n_{pop} * cr$ 4: <b>for</b> $t$ to $maxIter$ <b>do</b>
Actualización de la mejor solución	{	5:     Sort the population according to their fitness 6:     Update best solution $X_{best}$ 7:     Select best $Pn$ individuals and set at $pop1$ 8: <b>for</b> $i = 1$ to $Pn$ <b>do</b>
Crossover y filtrado	{	9: <b>Procreation</b> (Algorithm 2) 10: <b>Sexual cannibalism:</b> Delete parent with worst fitness from $pop1$ 11: <b>Siblings cannibalism:</b> Keep best $Sn$ new individuals at $pop2$ 12: <b>end for</b>
Mutación y actualización de la población	{	13: <b>for</b> $i = 1$ to $Mn$ <b>do</b> 14: <b>Mutation</b> (Algorithm 3) 15: <b>end for</b> 16: $pop = pop2 + pop3$ : Keep best $n_{pop}$ individuals from $pop$ 17: <b>end for</b>

---

# Diseño de la solución

## Procreación

---

### Algorithm 2 Black Widow Procreation

---

**Input:**  $pop1$   $X = \{x_1, x_2, \dots, x_{Pn}\}$

**Output:** New individuals.

- |   |   |   |
|---|---|---|
| Selección de padres                             | → | 1: Select two random parents from $pop1$                  |
|   |   | 2: <b>for</b> $j$ to $nvar/2$ <b>do</b>                   |
| Crossover                                       | { | 3: Generate random value $\alpha$                         |
|   |   | 4: Generate 2 new individuals with Eq. (7) and (8)        |
| Se pasa la solución por el esquema seleccionado | → | 5: Apply binarization scheme selected by DMAB (Table 1-3) |
| Se reparan las soluciones no factibles          | → | 6: Repair solutions if the are not feasible               |
|   |   | 7: Calculate fitness                                      |
| Actualización de DMAB                           | { | 8: <b>Update DMAB</b> (Algorithm 4)                       |
|   |   | 9: Save new binary individuals                            |
|   |   | 10: <b>end for</b>  |
- 

$$0 \leq a_i \leq 1$$

$$Y_1 = \alpha \times X_1 + \tilde{\alpha} \times X_2$$

$$Y_2 = \alpha \times X_2 + \tilde{\alpha} \times X_1$$

# Diseño de la solución

## DMAB

---

### Algorithm 4 Dynamic Multi-armed Bandit

---

**Input:** Environment (Bin. schemes)  $BS = \{bs_1, bs_2, \dots, bs_{40}\}$  and new fitness.

- Se calcula la recompensa obtenida y actualiza agente {
- Se ejecuta el PH-test para buscar comportamiento atípico {
- 1: Generate reward with Eq. (20) from fitness (BBWO output)
  - 2: Update reward mean history
  - 3: Run PH-test with Eq. (13) and (14)
  - 4: **if** Eq. (15) is True **then**
  - 5:     Restart agent learning. Start from first line
  - 6: **else**
  - 7:     Continue
  - 8: **end if**
- 

$$r = 100 \cdot \left[ \frac{f_{old} - f_{new}}{f_{old}} \right]$$

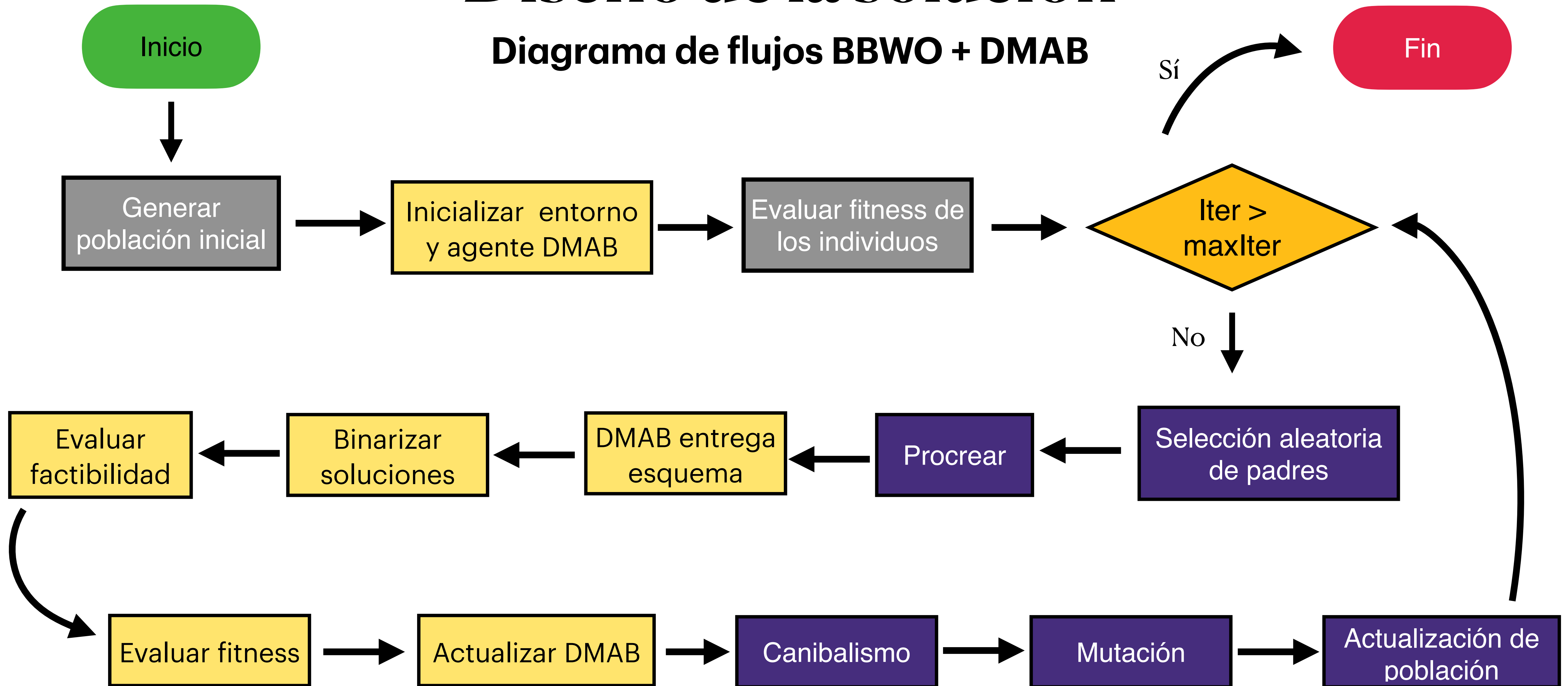
$$avgDev_a(t) = \sum_{i=1}^t (r_{a,i} - \bar{r}_{a,i} + \delta)$$

$$maxDev_a(t) = max_a \{ avgDev_a(i), i = 1 \dots t \}$$

$$maxDev_a(t) - avgDev_a(t) > \lambda$$

# Diseño de la solución

## Diagrama de flujos BBWO + DMAB





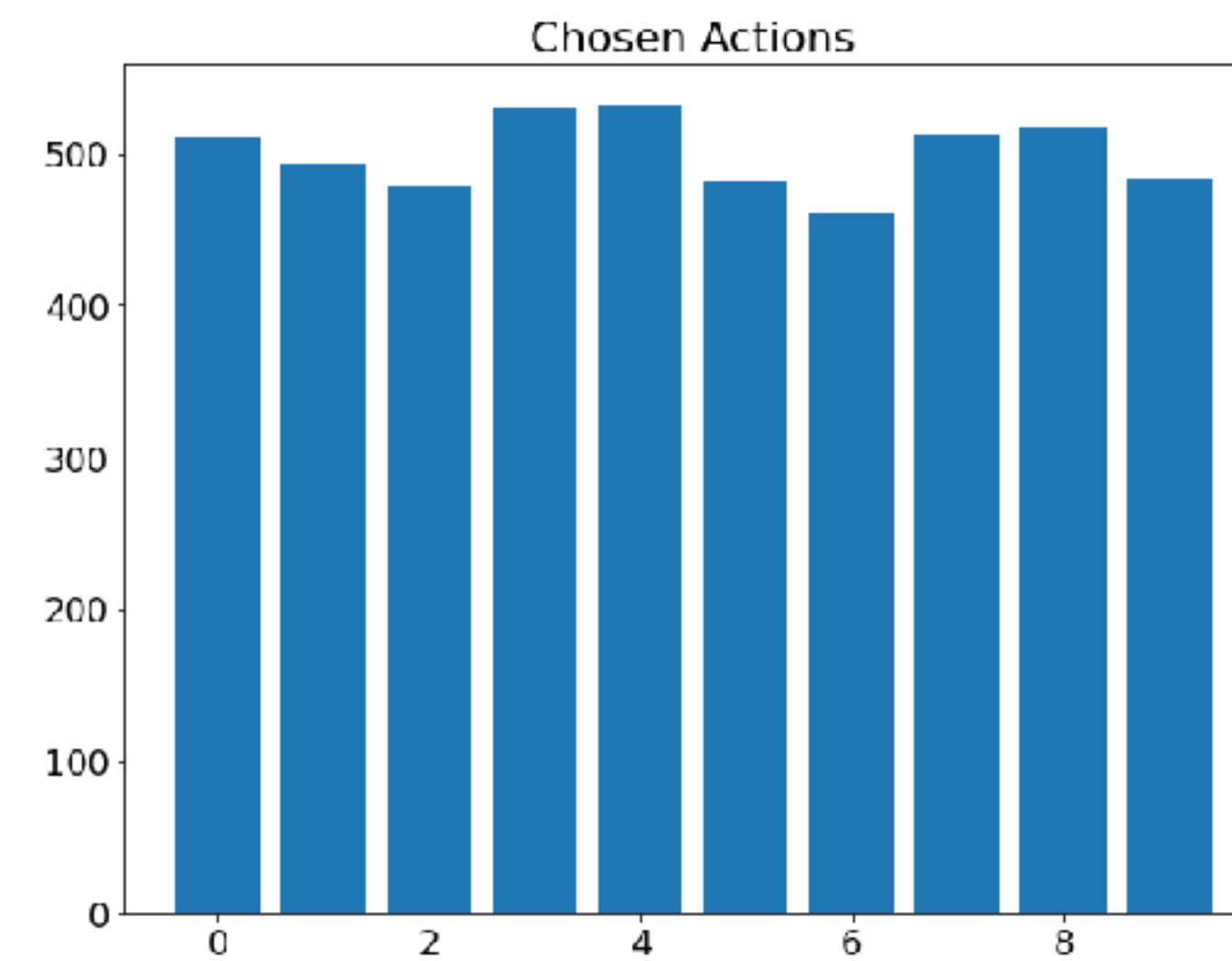
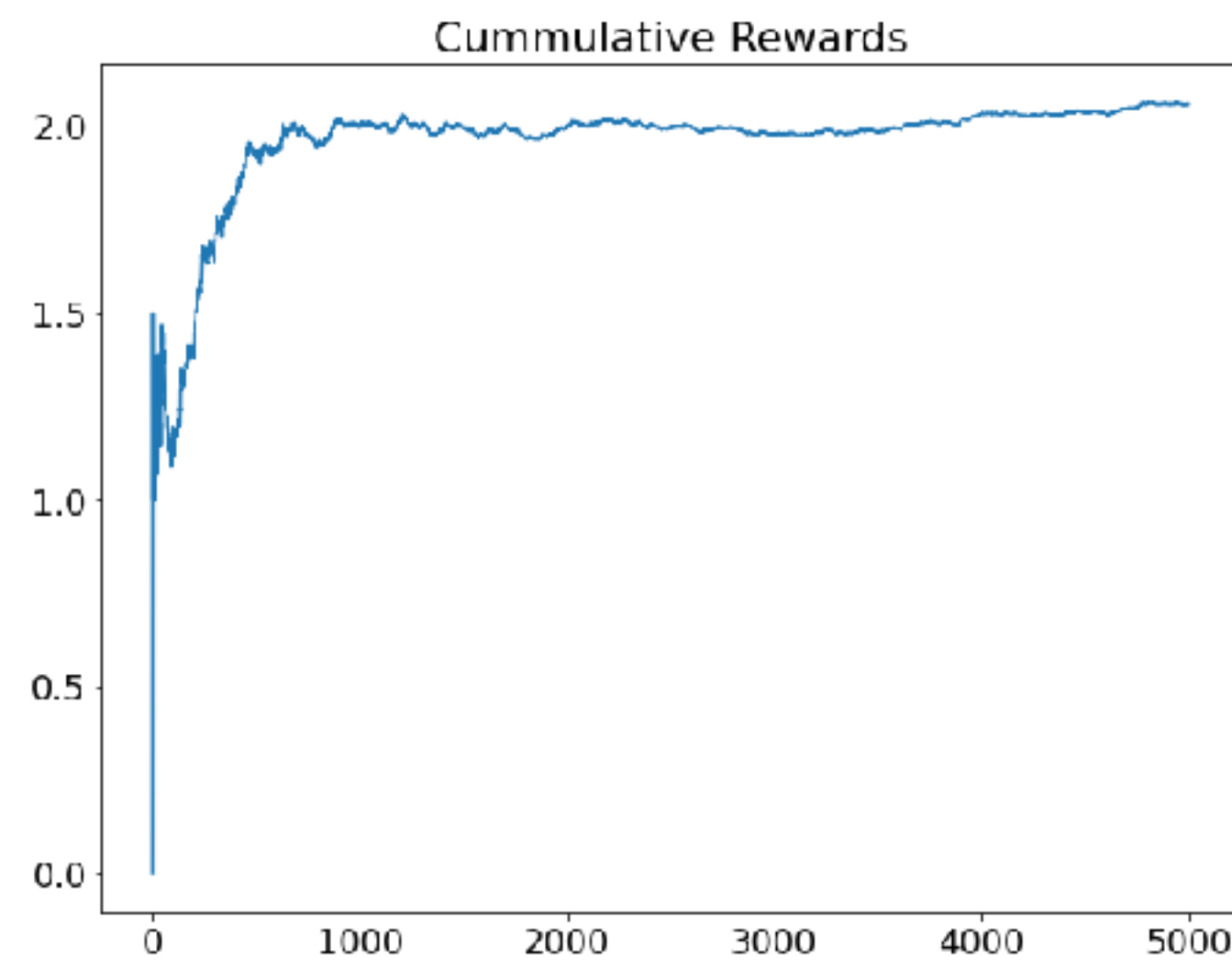
# Primeros experimentos

## MAB

### *Instancia*

<b>Id. Brazo</b>	0	1	2	3	4	5	6	7	8	9
<b>Prob. de asierto</b>	0,1	0,3	0,5	0,8	0,9	0,2	0,4	0,5	0,7	0,95
<b>Recompensa</b>	9	7	4	2	1	8	7	9	2	3

### *Random*



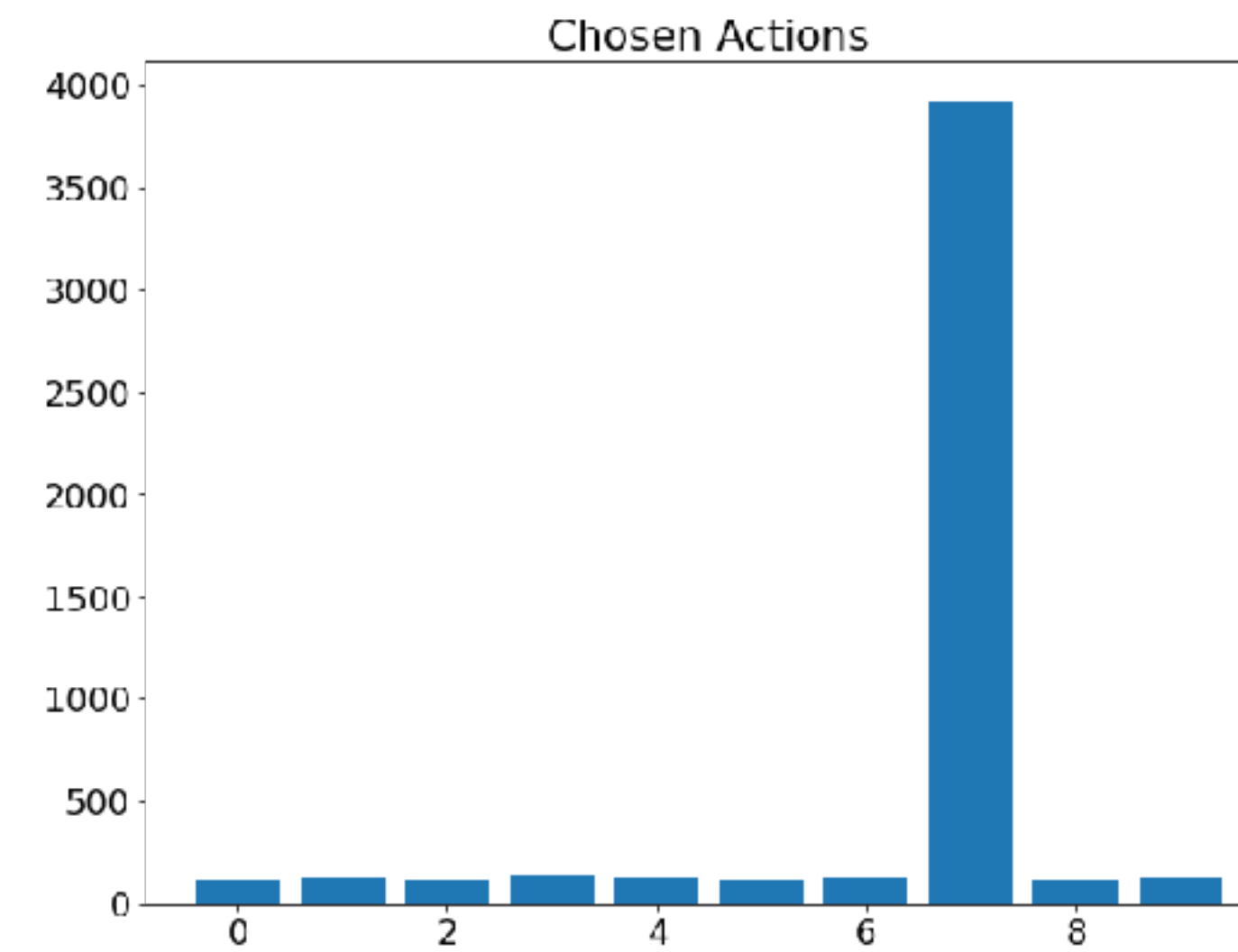
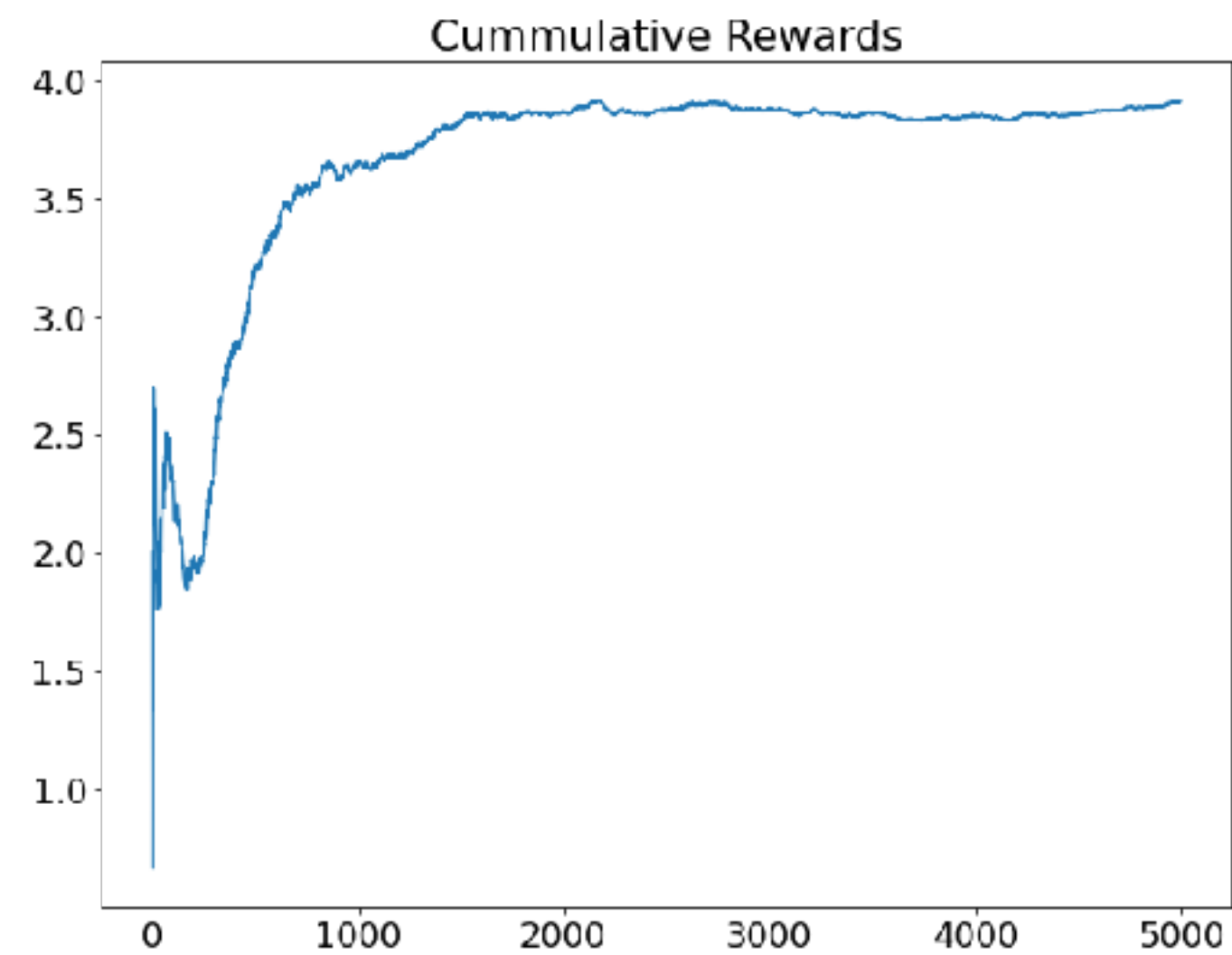
# Primeros experimentos

## MAB

### *Instancia*

<b>Id. Brazo</b>	0	1	2	3	4	5	6	7	8	9
<b>Prob. de asierto</b>	0,1	0,3	0,5	0,8	0,9	0,2	0,4	0,5	0,7	0,95
<b>Recompensa</b>	9	7	4	2	1	8	7	9	2	3

### *Epsilon - greedy*



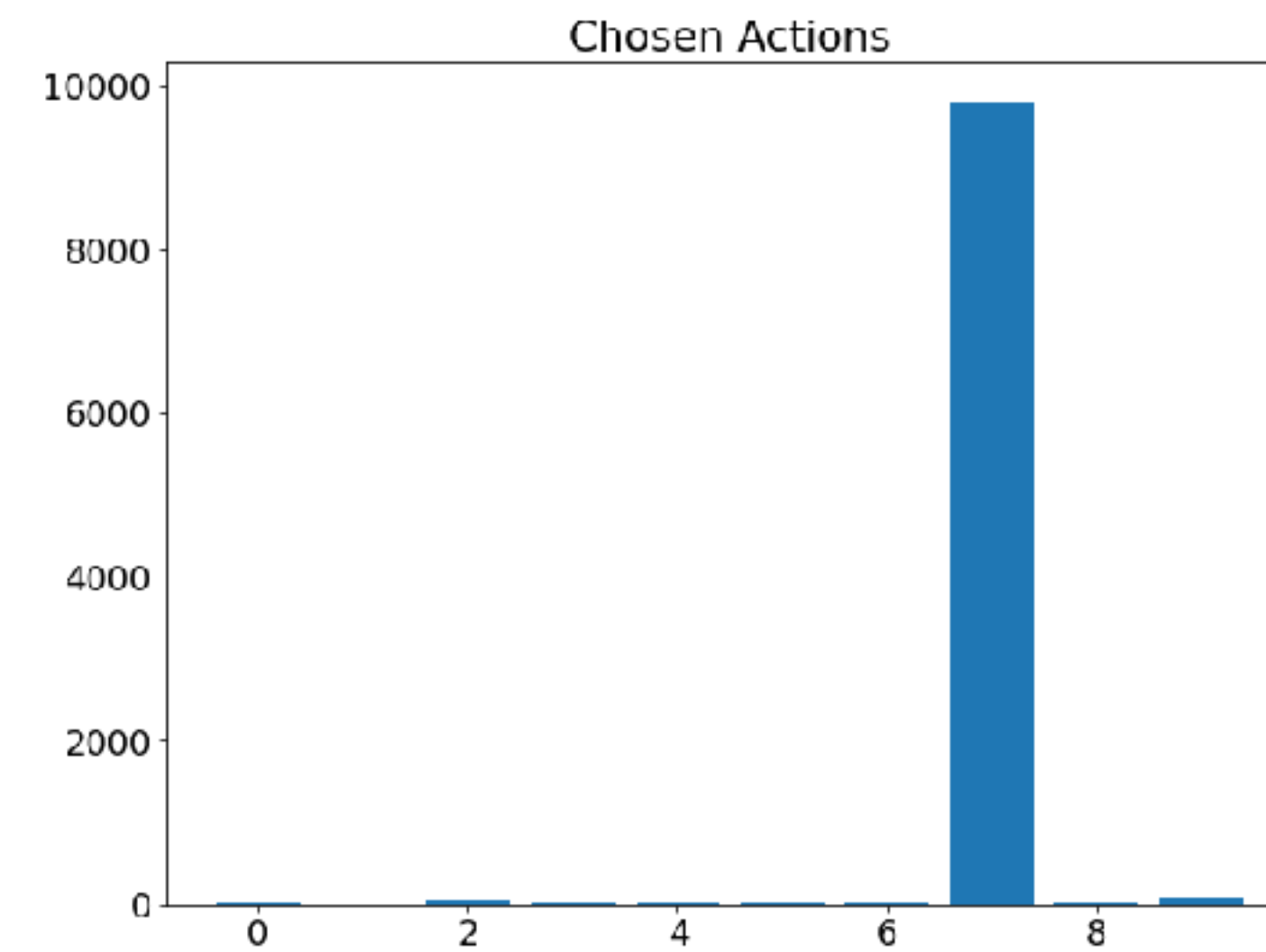
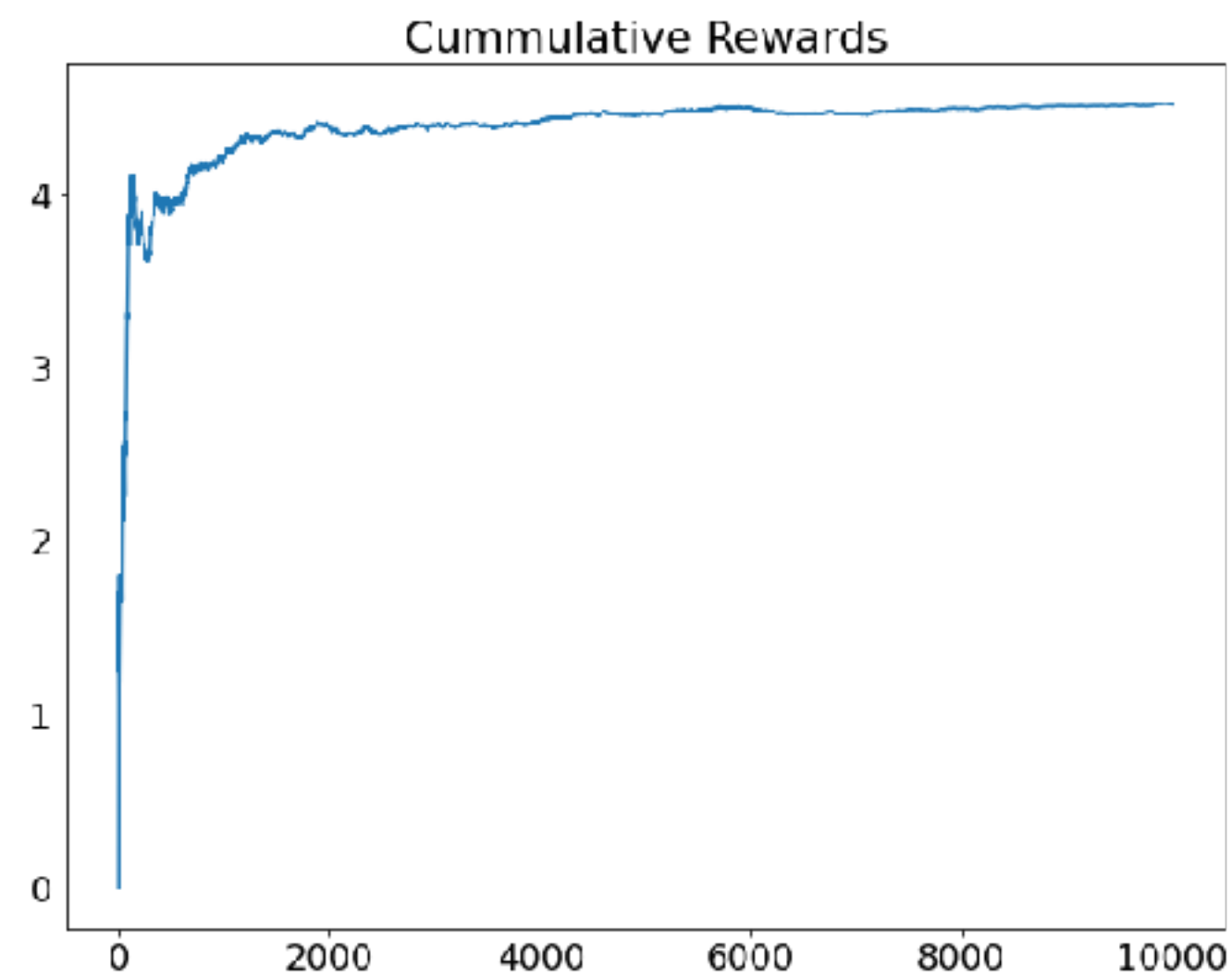
# Primeros experimentos

## MAB

### *Instancia*

<b>Id. Brazo</b>	0	1	2	3	4	5	6	7	8	9
<b>Prob. de asierto</b>	0,1	0,3	0,5	0,8	0,9	0,2	0,4	0,5	0,7	0,95
<b>Recompensa</b>	9	7	4	2	1	8	7	9	2	3

## **UCB**



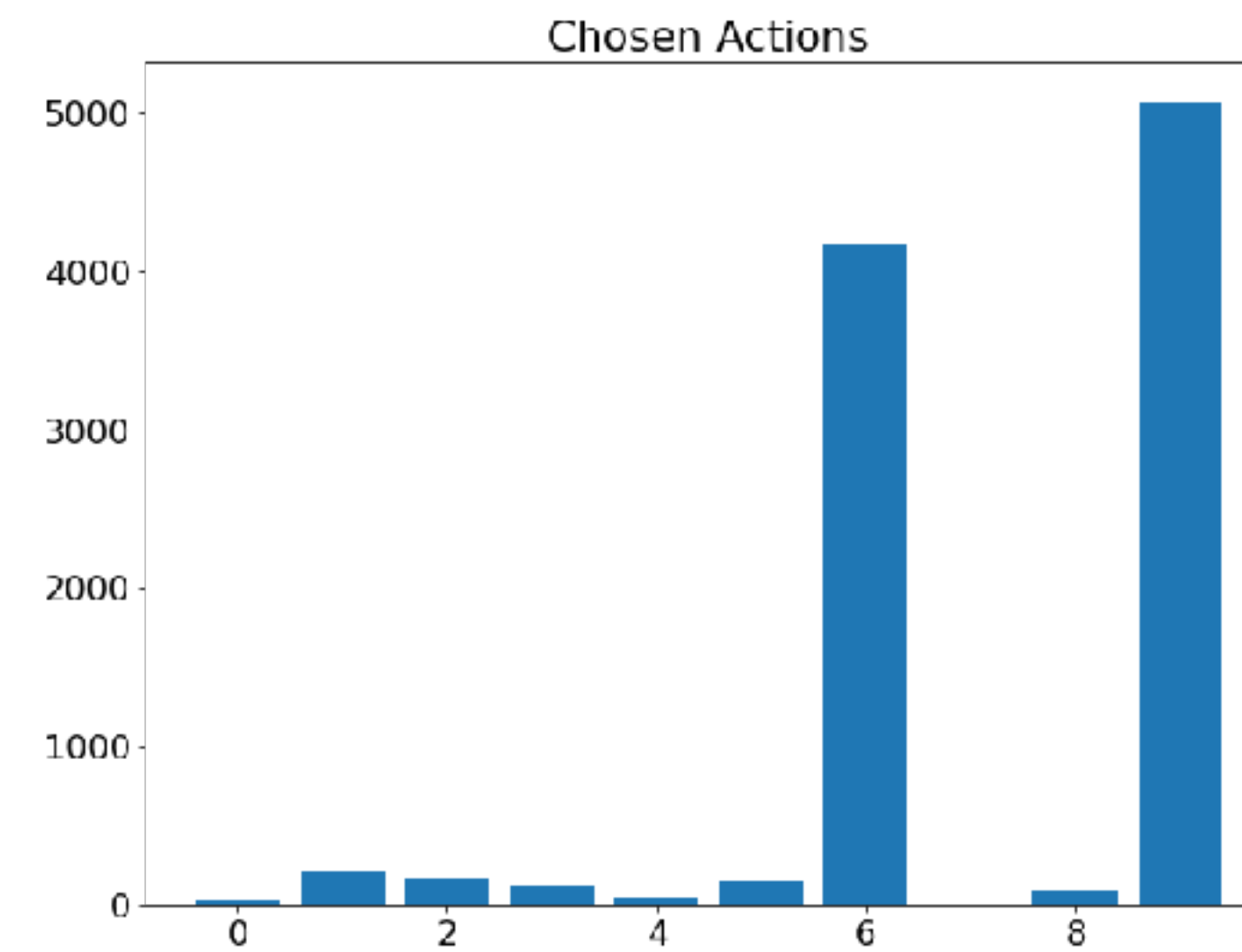
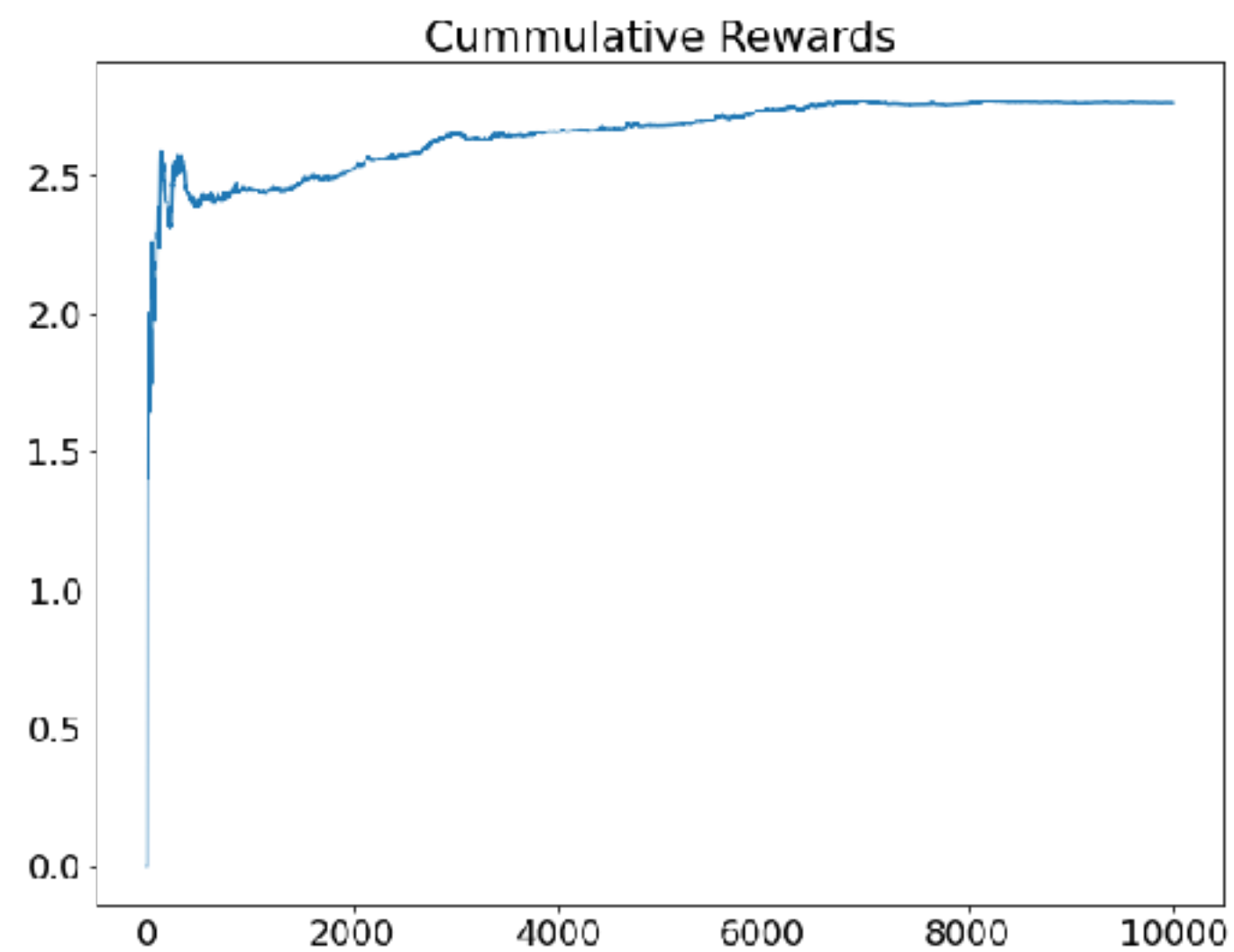
# Primeros experimentos

## MAB

### *Instancia*

<b>Id. Brazo</b>	0	1	2	3	4	5	6	7	8	9
<b>Prob. de asierto</b>	0,1	0,3	0,5	0,8	0,9	0,2	0,4	0,5	0,7	0,95
<b>Recompensa</b>	9	7	4	2	1	8	7	9	2	3

### *UCB cuando necesita PH-test*



# Primeros experimentos

## BBWO

### Instancia

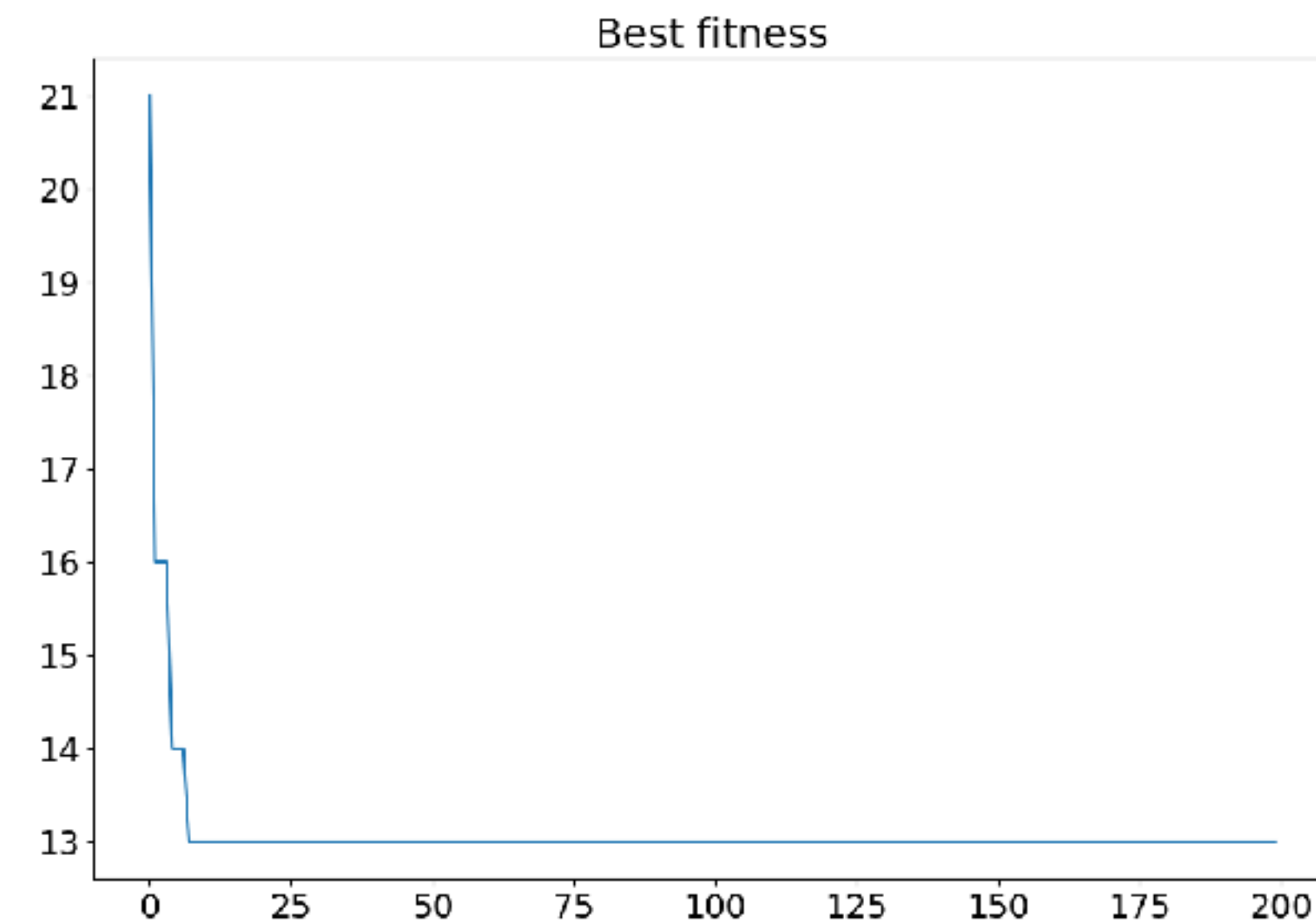
$$A = \begin{pmatrix} 1. & 0. & 0. & 1. & 0. & 0. & 1. & 1. \\ 0. & 0. & 1. & 1. & 0. & 0. & 1. & 1. \\ 1. & 0. & 0. & 0. & 1. & 0. & 0. & 0. \\ 1. & 1. & 0. & 0. & 0. & 0. & 0. & 1. \\ 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. \\ 1. & 0. & 0. & 0. & 1. & 0. & 1. & 1. \\ 1. & 1. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 1. & 0. & 0. & 0. & 1. & 0. & 1. \\ 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. & 0. & 0. & 1. & 0. \\ 0. & 1. & 0. & 0. & 1. & 0. & 0. & 1. \\ 0. & 0. & 1. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 1. & 1. & 1. & 0. & 0. \\ 0. & 0. & 0. & 1. & 0. & 1. & 0. & 0. \end{pmatrix}$$

<b>Costo</b>	4	2	3	4	4	3	4	4
--------------	---	---	---	---	---	---	---	---

### Parámetros

maxIter: 200    npop: 15    pp: 0,6    cr: 0,44    pm: 0,4

### Resultado



# Conclusión

## Desafíos

Implementar PH-Test para DMAB

Optimizar reparación de soluciones

Combinar BBWO y DMAB

Dynamic Multi-armed bandit



# **Multi-armed bandit for selection of binarization scheme in metaheuristics**

**Seminario de Tesis**

Pablo Ábrego - 6 de junio 2022