Algoritmos Genéticos

$\{X^{(1)}, X^{(2)}, \ldots X^{(n)}\}$

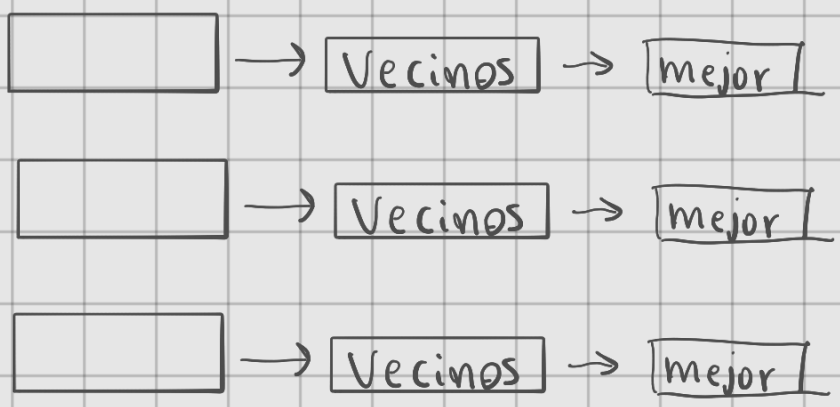
↓

Algoritmo

↓

$\{X'^{(1)}, X'^{(2)}, \ldots X'^{(n)}\}$ Conjunto de nuevas soluciones

Paralelizamos el descenso de colinas



Población

$\{X^{(1)}, \ldots, X^{(n)}\}$ es una generación

Operador de Selección

Prog. 1 $= \{X_1^{(1)}, \ldots, X_1^{(k)}\}$

Prog. 2 $= \{X_2^{(1)}, \ldots, X_2^{(k)}\}$

Operador de Cruza

Cruza $(X_1^{(i)}, X_2^{(i)}) = y^{(i)}$

Operador de mutación

muta $(y^{(i)}) = y'^{(i)}$

Nueva generación $= \{y'^{(1)}, \ldots y'^{(n)}\}$ + elitismo $(\{X^{(1)}, \ldots, X^{(n)}\})$

Operadores de selección

- Método de la ruleta

```
def adaptación (x):
    return (1/(log(costo(x)+1)+1))        1/costo(x)+1
```

↑ Costo mayor, adaptabilidad menor

```
def ruleta (población, adaptación):
    a={ x: adaptación(x) for x en población}
    suma = sum(a.values())
    acc = 0
    ruleta = random.random()
    for x in a
        acc += a[x]/suma
        if ruleta ≤ acc:
            return x
    return x
```

Torneos

$X_1^A \rightarrow X_1^B$ ⋮ ⋮ $X_{n/2}^A \rightarrow X_{n/2}^B$

Gana el de menor costo con prob $1-\varepsilon$

Puede ganar el de mayor costo con prob. ε