

UFRJ — COPPE — PEE — CPE723 — Otimização Natural

Simulated Annealing — Aula 02

1983 — Kirkpatrick, Gelatt e Vecchi — gerador aleatório de estados

"x" é executado em temperaturas progressivamente mais baixas,
com inspiração na termodinâmica estatística (*).

(*) Informações básicas sobre termodinâmica estatística :

https://en.m.wikipedia.org/wiki/statistical_mechanics

Consider:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
J = np.array([0.2, 0.0, 0.6, -1.0, 1.0, 0.4, -0.8])
```

```
plt.plot(J, 'k.')
```

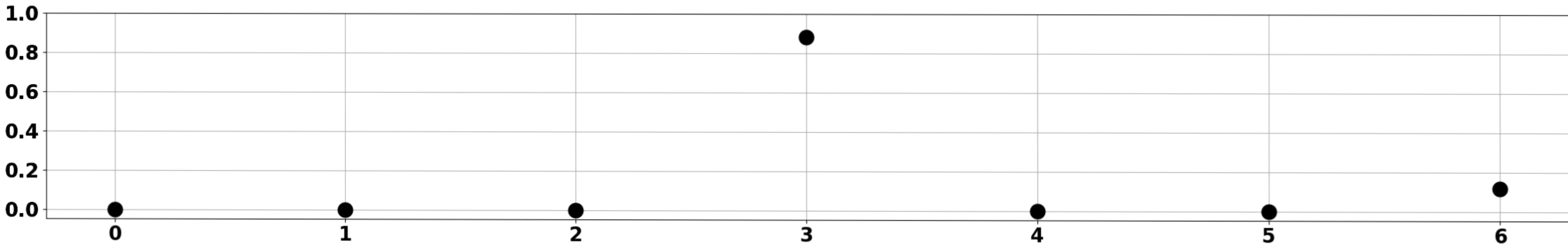
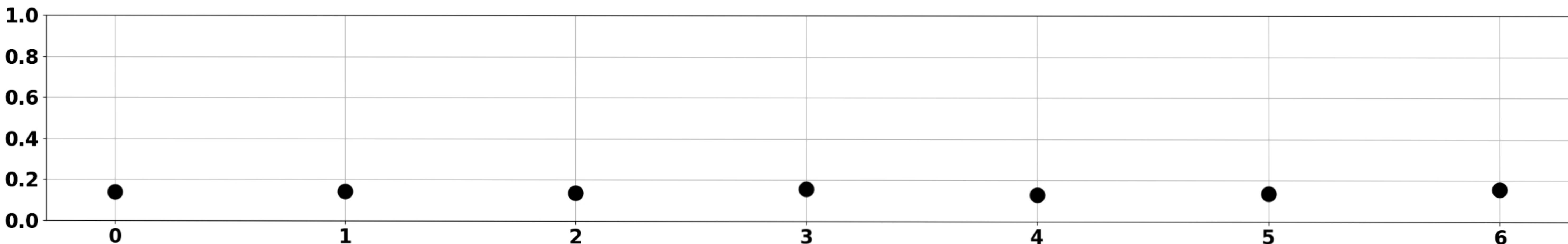
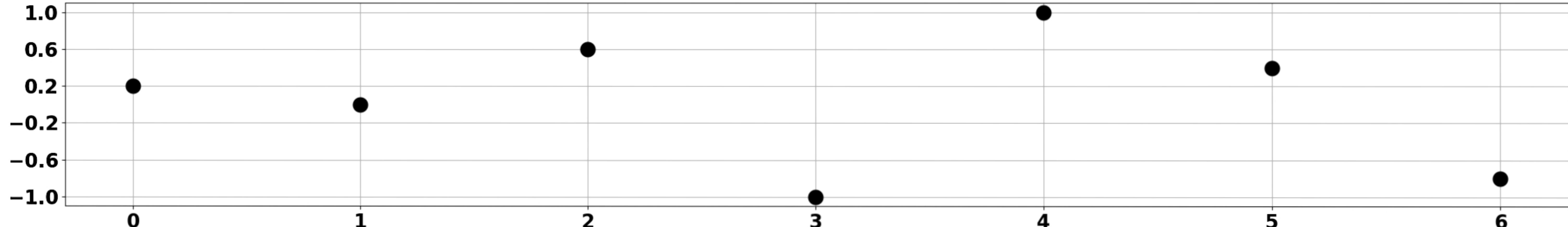
```
plt.plot(np.exp(-J/10)/sum(np.exp(-J/10)), 'k.') ← T = 10
```

```
plt.plot(np.exp(-J/0.1)/sum(np.exp(-J/0.1)), 'k.') ← T = 0.1
```

```
plt.show()
```

x	J(x)	$e^{-\frac{J(x)}{\tau}}$	$\frac{e^{-\frac{J(x)}{\tau}}}{\sum}$
0	0.2		$p(x=0)$
1	0.0		$p(x=1)$
2	0.6		:
3	-1.0		
4	1.0		
5	0.4		
6	-0.8		
		Σ	$\frac{p(x=6)}{\Sigma = 1}$

$$z = \sum e^{-\frac{J(x)}{\tau}}$$



% Simulated Annealing Básico — Pseudo-Código

x_0 aleatório; $\boxed{J = J(x_0)}$; $x_{atual} = x_0$; $J_{atual} = J_0$;

$N = 10000$; $k_{max} = 8$; $T_0 = 5e-3$; $\epsilon = 5e-2$;

$fim = 0$; $n = 0$; $\kappa = 1$; $J_{min} = J_{atual}$; $x_{min} = x_{atual}$; $T = T_0$;

while not(fim),

$n = n + 1$;

$x = x_{atual} + \epsilon * \text{randn}(\text{size}(x))$;

($\omega^{x_{atual}}$)

$\boxed{J = J(x)}$;

if $\text{rand}(1) < \exp(-(|J_{atual} - J|)/T)$, } $x_{atual} = x$; $J_{atual} = J$;

if $J < J_{min}$, } $J_{min} = J$; $x_{min} = x$;

if $\text{rem}(n, N) = 0$, } $\kappa = \kappa + 1$; $T = T_0 / \log_2(1 + \kappa)$; if $\kappa = k_{max}$, } $fim = 1$;

(Os dois retângulos)

indicam as chamadas

à função custo $J(x)$)

- X_0 aleatório; $J_0 = J(X_0)$; $X_{atual} = X_0$; $J_{atual} = J_0$;
- Definições: $N = (10000)$; $K = (8)$; $T_0 = (5e-3)$; $\varepsilon = (5e-2)$;
- $fim = 0$; $n = 0$; $k = 1$; $J_{min} = J_{atual}$; $X_{min} = X_{atual}$;
- while not(fim)
 - {
 - $n = n+1$;
 - $X = X_{atual} + \varepsilon \times \text{randn}(\text{size}(X))$;
 - $J = J(X)$;
 - if $\text{rand}(1) < \exp((J_{atual} - J)/T)$ { $X_{atual} = X$; $J_{atual} = J$ };
 - if $J < J_{min}$ { $J_{min} = J$; $X_{min} = X$ };
 - if $\text{rem}(n, N) = 0$ { $k = k+1$; $T = T_0 / \log_2(1+k)$; if $k = K$, { $fim = 1$ } };

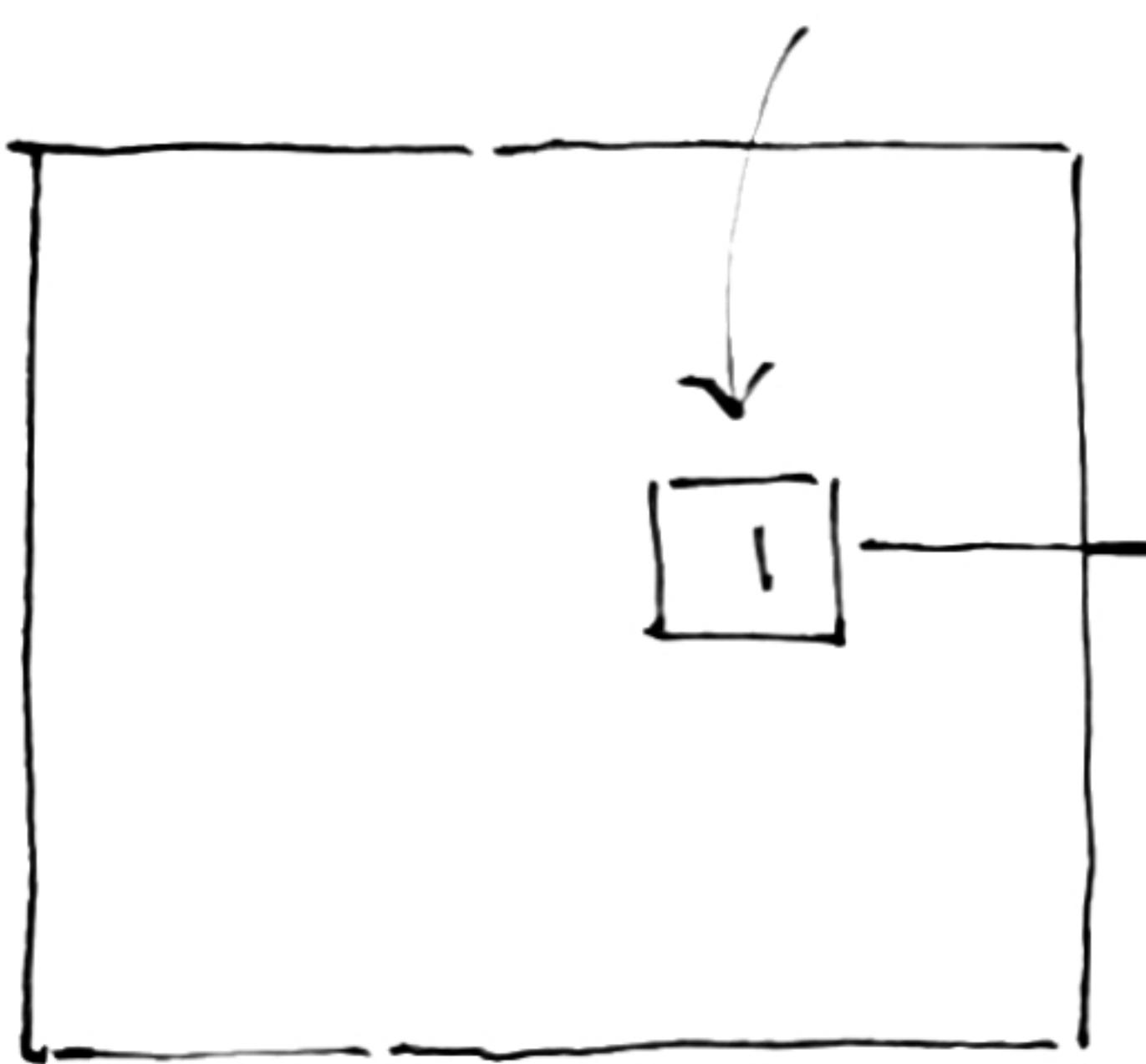
Alguns comentários sobre o pseudo-código:

- alcançar temperatura T baixa não é suficiente. O resfriamento deve ser "cuidadoso" (lento), com muito tempo gasto em cada faixa de temperaturas. Caso contrário, o algoritmo converge para mínimos locais ruins.
- T alto: soluções em "maior escala"; T baixo: "ajustes finos" da solução.
- $T = 0 \rightarrow$ algoritmo só admite descidas (e converge para mínimos locais).
- Repare na substituição da "função energia" $J(x)$ do Algoritmo de Metropolis pela "função custo" $J(x)$ do Simulated Annealing.

Exemplo 5 — Projeto de Computadores — Particionamento

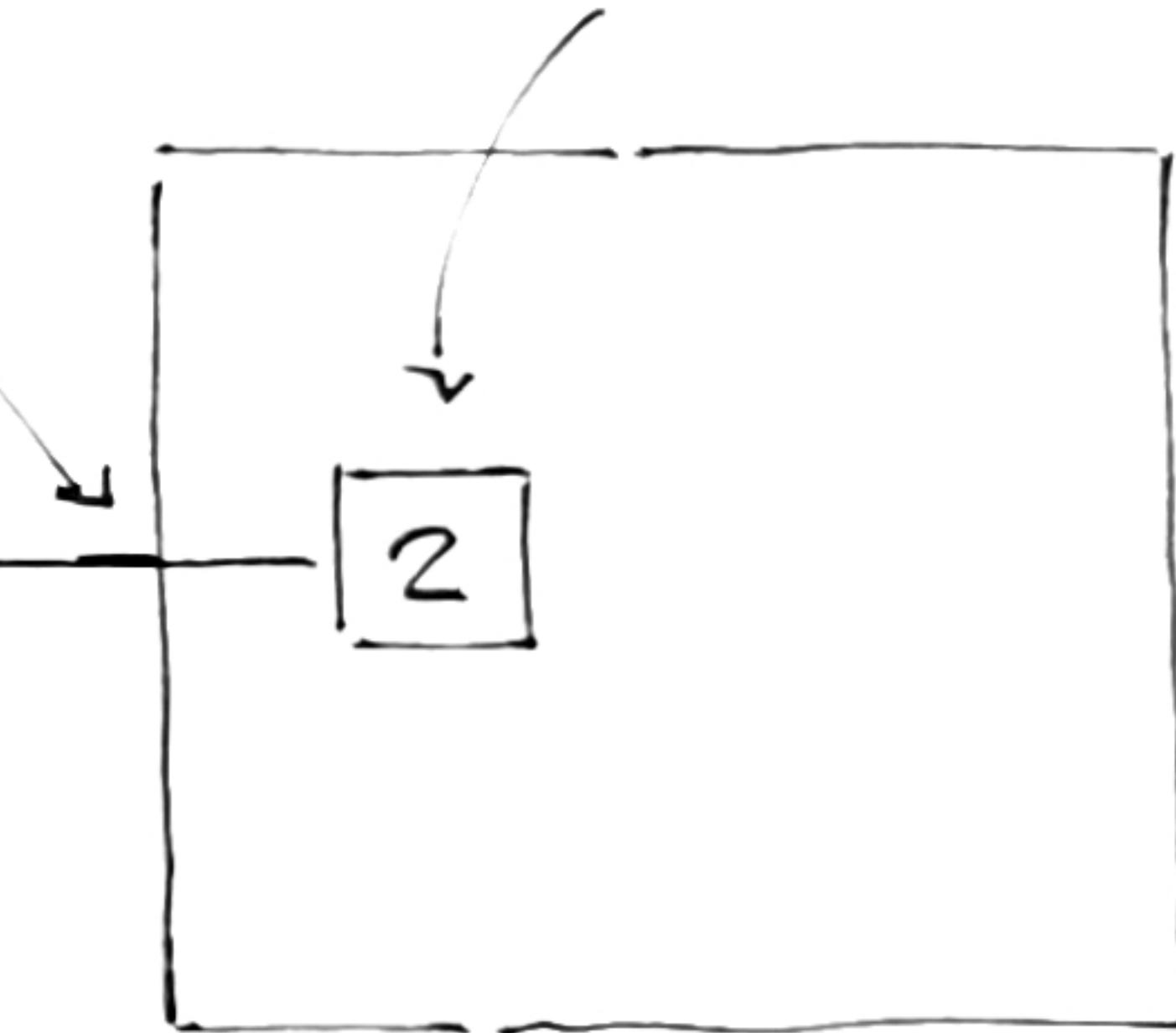
Decidir quais blocos básicos (elementos) de circuitos entram em cada chip.

Elemento 1 no chip 1



Chip 1 (+1)

Elemento 2 no chip 2



Chip 2 (-1)

(A matriz de
conectividade
é dada)

Matriz de Conectividade: $A =$

$$A = \begin{bmatrix} 1 & 2 & \dots & N \\ 1 & +1 & & \\ 2 & +1 & & \\ \vdots & & & \\ N & & & \end{bmatrix}$$

$$\text{Ex.: } A =$$

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{bmatrix}$$

Particfó (colocacfo dos N elementos aos 2 chips do enunciado):

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_N \end{bmatrix}$$

; Ex.: $\mu = \begin{bmatrix} +1 \\ +1 \\ -1 \end{bmatrix}$

(No sistema de simbolos do simulated annealing básico (pseudo-código), o vetor μ é o estado x .)

Uma particfó "equilibrada" seria tal que $\sum \mu_i \approx 0$.

Desbalanceamento: $N_D = (\sum \mu_i)^2$

Número de conexões (pinos): $N_C = \sum_{i>j} \frac{a_{ij}}{4} (\mu_i - \mu_j)^2$ (elemento (i,j) da matriz A)

Funçfó custo: $J = \lambda N_D + N_C$, onde o multiplicador de lagrange λ regula o compromisso entre N_D e N_C .

Considerando $A \in \mathbb{R}^{3 \times 3}$ e $\mu \in \mathbb{R}^{3 \times 1}$, só para facilitar os cálculos:

$$N_D = (\mu_1 + \mu_2 + \mu_3)^2 = \mu_1^2 + \mu_2^2 + \mu_3^2 + 2 \sum_{i>j} \mu_i \mu_j = \text{constante}_1 + \boxed{2 \sum_{i>j} \mu_i \mu_j}$$

$$N_C = \frac{a_{12}}{4} (\mu_1^2 - 2\mu_1 \mu_2 + \mu_2^2) + \frac{a_{13}}{4} (\mu_1^2 - 2\mu_1 \mu_3 + \mu_3^2) + \frac{a_{23}}{4} (\mu_2^2 - 2\mu_2 \mu_3 + \mu_3^2)$$

$$= (a_{12} + a_{13}) \mu_1^2 + (a_{12} + a_{23}) \mu_2^2 + (a_{13} + a_{23}) \mu_3^2 - \frac{1}{2} \sum_{i>j} a_{ij} \mu_i \mu_j$$

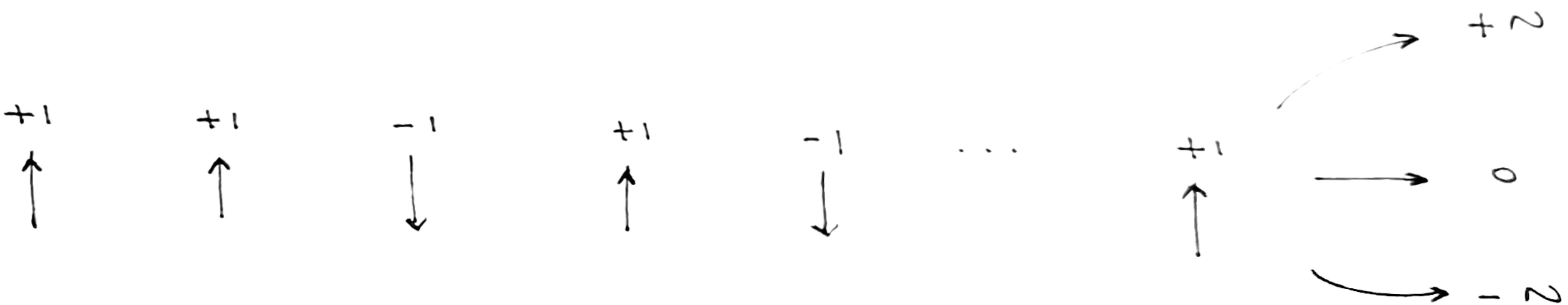
$$= \text{constante}_2 - \frac{1}{2} \sum_{i>j} a_{ij} \mu_i \mu_j$$

$$\mathcal{J} = 2 \lambda \sum_{i>j} \mu_i \mu_j - \frac{1}{2} \sum_{i>j} a_{ij} \mu_i \mu_j = \boxed{\sum_{i>j} \left(\lambda - \frac{a_{ij}}{2} \right) \mu_i \mu_j}$$

(↑ por não serem otimizáveis, as constantes 1 e 2 são ignoradas)

Resumindo, o algoritmo consiste em começar de um estado μ (ou seja, um estado x) e executar o "simulated annealing" básico (do pseudo-código)

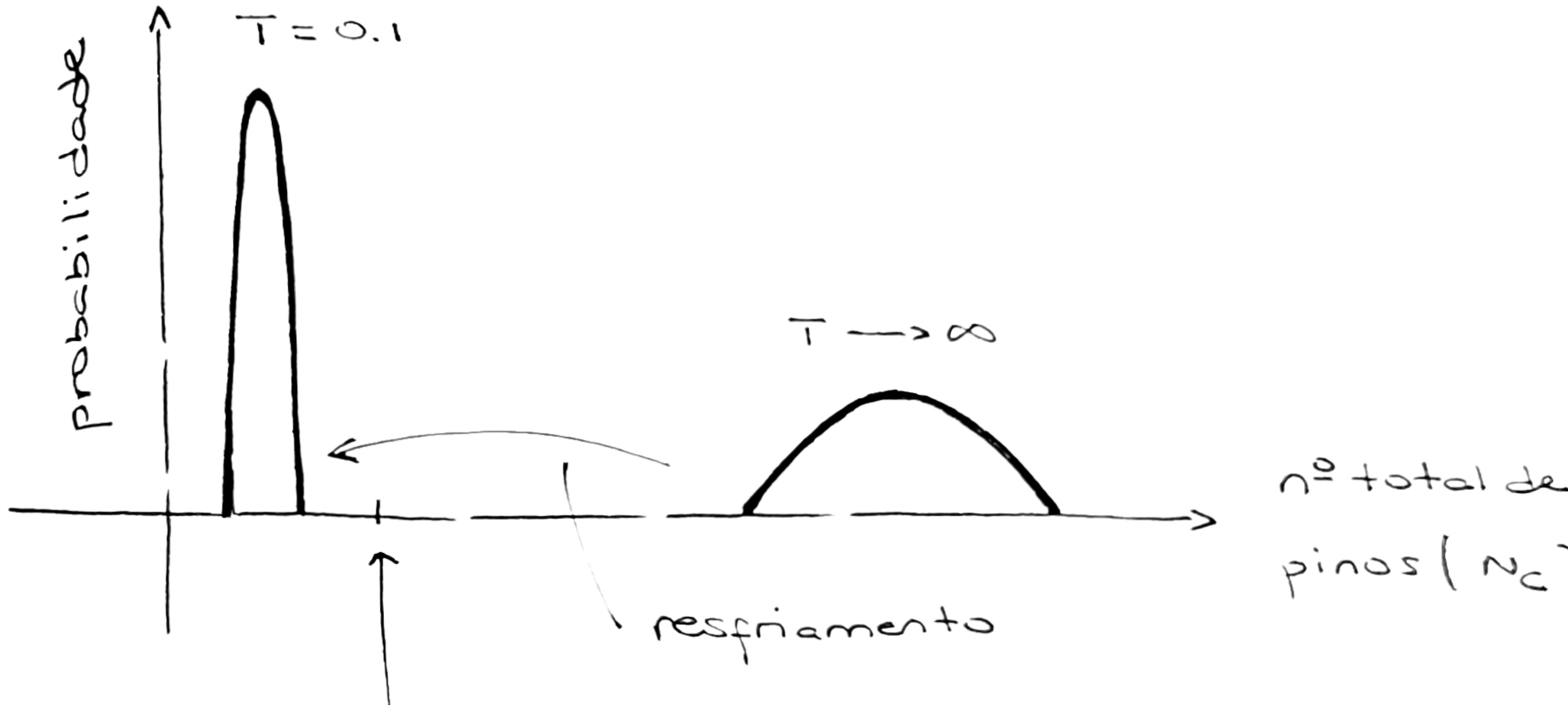
com $J(\mu) = \sum_{i>j} \frac{a_{ij}}{4} (\mu_i - \mu_j)^2$, onde os números a_{ij} são dados.



Método de perturbação: escolher uma posição aleatória do vetor μ e alterá-la de "-1" para "+1" ou vice-versa.

Método de resfriamento: exponencial (ou seja, não logarítmico). Resulta em minimo local).

Resultado:



Solução com o menor número de pinos conhecida antes

Obs.1: (micro)processador IBM 370

Obs.2: próximas duas etapas: posicionamento dos chips na placa e roteamento da placa — metodologia semelhante.

Exemplo 2 — Travelling Salesman Problem (TSP — "Caixeiro" viajante)

São dadas K cidades, situadas nas coordenadas c_i , $i = 1, 2, \dots, K$.

As cidades são visitadas na ordem $X = [i_1, i_2, \dots, i_K]$. O vetor X corresponde a uma permutação dos números naturais de 1 até K .

Problema de otimização "difícil"; ver Figura 9 no paper (Kirkpatrick, 1983).

[en.wikipedia.org/wiki/Travelling-salesman-problem](https://en.wikipedia.org/wiki/Travelling_salesman_problem)

(en.wikipedia.org)

www.math.uwaterloo.ca/tsp/world

Método de perturbação: escolher aleatoriamente duas posições do vetor X trocá-las uma pela outra.

Exemplo 3 — Minimização de $J(x)$ das cinco partículas.

Lembrando: $J(x) = \frac{1}{S} \sum_{i=1}^S \|x_i\|^2 + \lambda \sum_{j \geq i} \frac{1}{\|x_i - x_j\|^2}$

Método de Perturbação: $\hat{x} = x + \epsilon R$. R gaussiano $N(0,1)$ e " $\epsilon = 0.1$ ".

Método de Resfriamento: logarítmico, 8 temperaturas diferentes.

Soluções sobre "pentágono" e sobre "quadradão" — ver Prova de 2012, questão 1.

Na figura mostrada no slide a seguir, observe a razão entre $P(J)$ (em

azul) e fatores $e^{-J/T}$ (em vermelho): $\frac{e^{-J/T}}{z} \cdot "N_J" = P(J)$

Azul: $P(J)$. Vermelho: $\exp(-J/T)/Z$

