Transformada Imagem-Floresta (IFT)

Prof. Dr. Paulo A. V. de Miranda

pmiranda@vision.ime.usp.br

Instituto de Matemática e Estatística (IME),

Universidade de São Paulo (USP)

Caminhos no grafo

- Um caminho π no grafo $G = (\mathcal{D}_I, \mathcal{A})$ é uma sequência de pixels distintos $\langle p_1, p_2, \ldots, p_n \rangle$, onde $(p_i, p_{i+1}) \in \mathcal{A}$, $i = 1, 2, \ldots, n-1$.
- O pixel p_1 é a origem $org(\pi)$ do caminho, e p_n é o destino $dst(\pi)$.
- O caminho π é dito **trivial** se $\pi = \langle p_1 \rangle$.
- Seja π um caminho que termina em um pixel p e $(p, q) \in \mathcal{A}$, então $\pi \cdot \langle p, q \rangle$ é dito o caminho resultante da concatenação de π e $\langle p, q \rangle$ com as duas instâncias de p se fundindo em uma.
- Um pixel q é dito **conexo** a um pixel p se existir um caminho de p a q em $G = (\mathcal{D}_I, \mathcal{A})$.

Funções de conexidade

- Uma função de conexidade $f(\pi)$ associa um valor escalar a qualquer caminho no grafo $G = (\mathcal{D}_I, \mathcal{A})$, com base em propriedades da imagem ao longo deste caminho.
- Em segmentação, por exemplo, propriedades locais da imagem (vetor de atributos) e globais do objeto desejado (textura, cor e forma) podem ser exploradas no cálculo de $f(\pi)$ para indicar a força de conexidade entre seus nós terminais através do caminho π.

Funções de conexidade

As funções de conexidade são especificadas por uma regra de inicialização e uma regra de extensão de caminho.

$$
\begin{aligned}
f_{\max}(\langle t \rangle) &= H(t) \\
f_{\max}(\pi_s \cdot \langle s,t \rangle) &= \max\{f_{\max}(\pi_s), w(s,t)\} \qquad (1) \\
f_{sum}(\langle t \rangle) &= H(t) \\
f_{sum}(\pi_s \cdot \langle s,t \rangle) &= f_{sum}(\pi_s) + w(s,t) \qquad (2) \\
f_{euc}(\langle t \rangle) &= \begin{cases} 0 & \text{if } t \in \mathcal{S} \\ +\infty & \text{caso contrário} \end{cases} \\
f_{euc}(\pi_s \cdot \langle s,t \rangle) &= \|t - R(s)\|^2 \qquad (3)
\end{aligned}
$$

onde $H(t)$ é um valor inicial, $R(s) = org(\pi_s)$, $w(s,t)$ é um peso de arco ($w(s,t) \geq 0$ em f_{sum}), e $\mathcal{S} \subset \mathcal{D}_I$ é um conjunto de sementes.

Funções de conexidade

As funções $f_{\max}$ e f_{sum} são casos particulares de funções f_{mi} monotonicamente incrementais.

$$\begin{aligned} f_{mi}(\langle t \rangle) &= H(t), \\ f_{mi}(\pi_s \cdot \langle s,t \rangle) &= f_{mi}(\pi_s) \odot (s,t), \end{aligned} \tag{4}$$

onde $\odot : \mathcal{V} \times \mathcal{A} \to \mathcal{V}$ é uma operação binária entre o valor de um caminho e um arco que satisfaz as condições:

- **(M1)** $a \geq b \Rightarrow a \odot (s,t) \geq b \odot (s,t)$,
- **(M2)** $a \odot (s,t) \geq a$,

para a, $b \in \mathcal{V}$ e quaisquer arcos $(s,t) \in \mathcal{A}$. Uma característica essencial deste modelo de função é que $\odot$ depende apenas do valor de π_s, e não de qualquer outra propriedade deste caminho.

Caminho Ótimo

- Um caminho π_t é ótimo se $f(\pi_t) \leq f(\tau_t)$ para qualquer outro caminho τ_t, independentemente de sua raiz.
- Para cada nó $t \in \mathcal{D}_I$, temos um valor único $V(t)$ que armazena o valor de um caminho ótimo com término em t:

$$V(t) = \min_{\forall \pi_t \text{in}(\mathcal{D}_I, \mathcal{A})} \{f(\pi_t)\}. \quad (5)$$

Transformada Imagem-Floresta

A transformada imagem-floresta (IFT - *Image Foresting Transform*) reduz problemas de processamento de imagem baseados em conexidade ao cálculo:

- de uma **floresta de caminhos ótimos** no grafo derivado da imagem,
- seguido de um pós-processamento simples de atributos da floresta resultante.

Motivação

- Unificação: Vários operadores de imagem são derivados de um algoritmo geral. Isto favorece
  - implementações baseadas em hardware,
  - compreender a relação entre alguns operadores de imagem,
  - possíveis extensões

Motivação

- Eficiência: A maioria dos operadores de imagem podem ser implementados em tempo linear e otimizações adicionais são possíveis com cálculo diferencial,e paralelo,e para algumas aplicações específicas.
- Simplicidade: Os operadores de imagem são reduzidos a escolha de poucos parâmetros no algoritmo da IFT e um processamento local de sua saída.

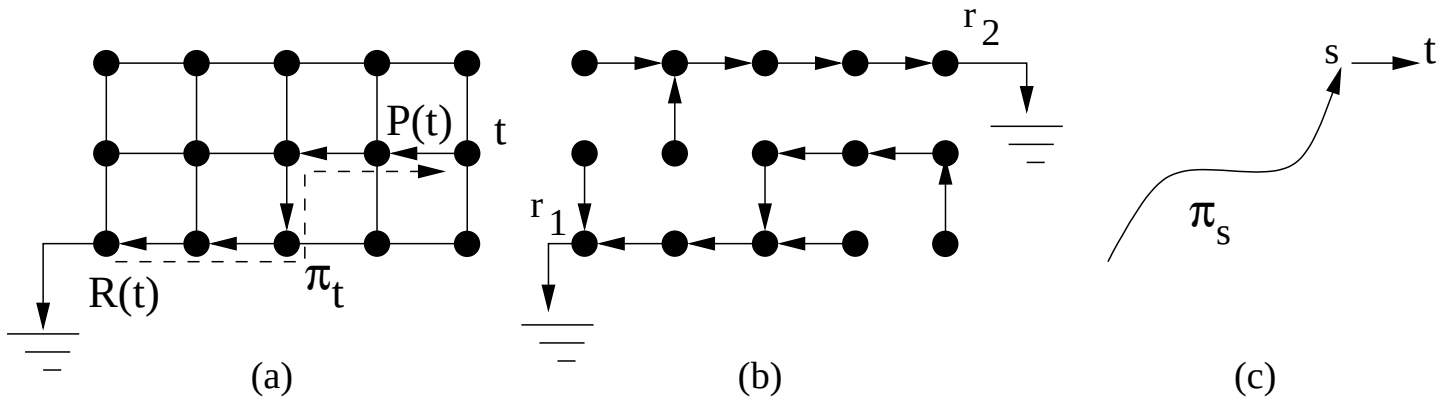
Quais problemas podem ser resolvidos?

- Transformadas de distância e operadores relacionados:
  Euclidean distance transform,multiscale skeletonization,fractal dimensions,shape filtering,shape saliences,shape description,geodesic paths, etc.
- Filtragem e segmentação de imagens:
  Morphological reconstructions, and image segmentation based on watershed transforms,live wire, riverbed, growcut by cellular automaton,and fuzzy-connected components.
- Reconhecimento de padrões:
  Data clustering,and supervised pattern classification.

Floresta de espalhamento

- Um *mapa de predecessores* é um função P que atribui para cada nó t em $\mathcal{D}_I$ algum outro nó adjacente em $\mathcal{D}_I$, ou uma marca distintiva $nil \notin \mathcal{D}_I$ — caso em que t é dito ser uma *raiz* do mapa.
- Uma *floresta de espalhamento* "é" um mapa de predecessores que não contém ciclos — isto é, um que leva cada pixel para nil em um número finito de iterações.
- Para qualquer pixel $t \in \mathcal{D}_I$, uma floresta de espalhamento P define um caminho π_t recursivamente como $\langle t \rangle$ se $P(t) = nil$, e $\pi_s \cdot \langle s, t \rangle$ se $P(t) = s \neq nil$.

Floresta de espalhamento



- O predecessor $P(t)$ de cada nó t leva a um nó raiz $R(t)$ e $P(R(t)) = nil$.
- Um caminho π_t é *trivial* quando $\pi_t = \langle t \rangle$ (i.e., $P(t) = nil$).

Transformada Imagem-Floresta

A IFT essencialmente generaliza o algoritmo de Dijkstra para funções de conexidade, onde para qualquer nó $q \in \mathcal{D}_I$, existe um caminho ótimo π_q que é ou trivial, ou tem a forma $\pi_p \cdot \langle p, q \rangle$ onde:

- **(C1)** $f(\pi_p) \leq f(\pi_q)$,
- **(C2)** π_p é ótimo, e
- **(C3)** para qualquer caminho ótimo τ_p, $f(\tau_p \cdot \langle p, q \rangle) = f(\pi_q)$.

Note que estas condições são aplicadas apenas a caminhos ótimos.

Algoritmo da IFT

Algoritmo 1 — ALGORITMO GERAL DA IFT

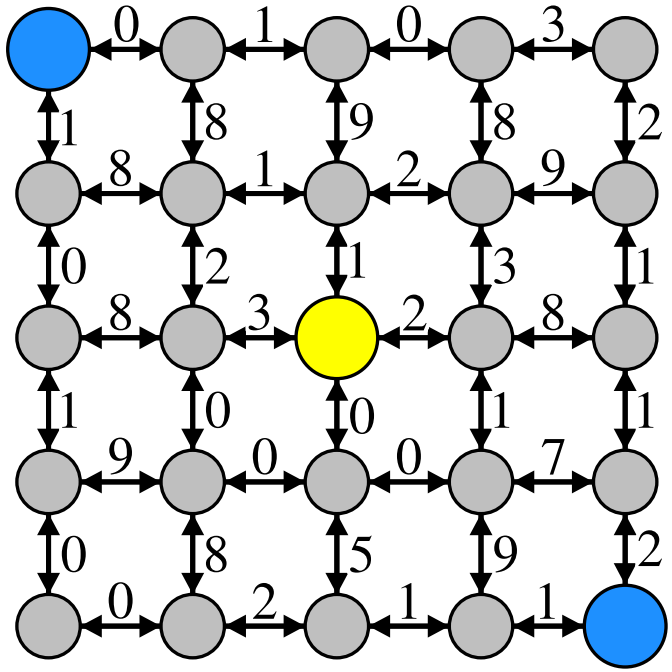
ENTRADA: Imagem $\hat{I} = (\mathcal{D}_I, \vec{I})$, adjacência $\mathcal{A}$, e função de conexidade f.

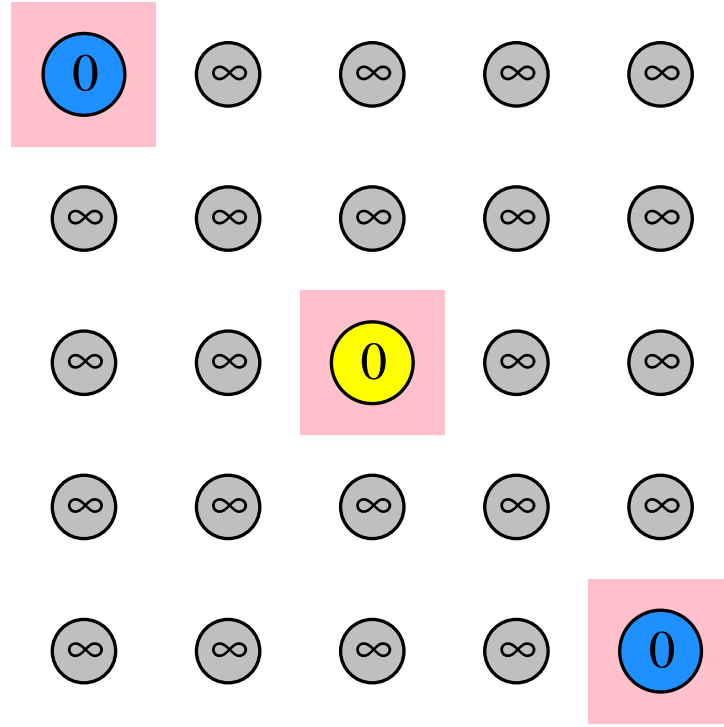
SAÍDA: Imagens $\hat{P} = (\mathcal{D}_I, P)$ de predecessores, e $\hat{V} = (\mathcal{D}_I, V)$ de conexidade.

AUXILIARES: Fila de prioridade Q, variável tmp, e vetor de $estado$ inicialmente zerado.

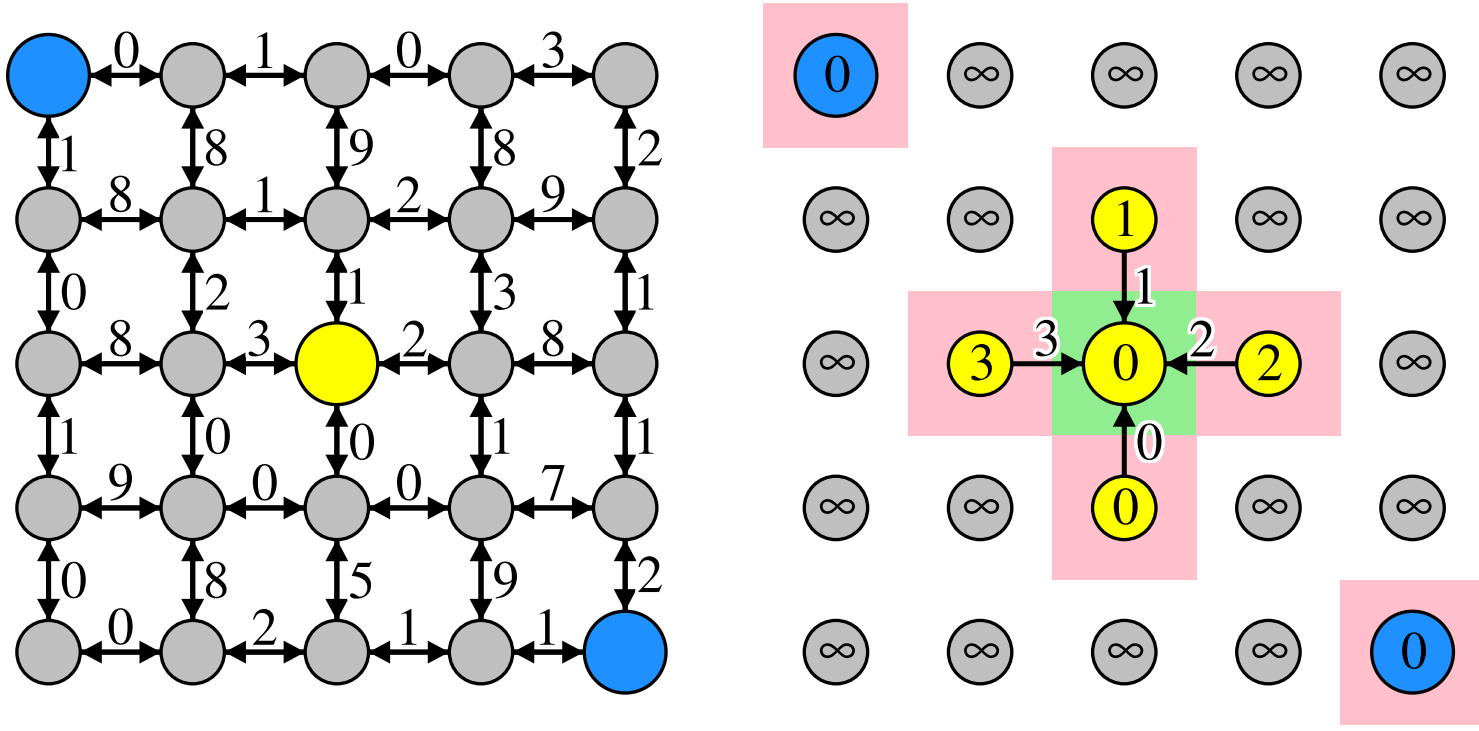
1. ***Para Cada*** $t \in \mathcal{D}_I$, ***Faça*** $P(t) \leftarrow nil$ e $V(t) \leftarrow f(\langle t \rangle)$. ***Se*** $V(t) \neq +\infty$, ***Então*** *insira* t *em* Q.
2. ***Enquanto*** $Q \neq \emptyset$, ***Faça***
3. *Remova um pixel* s *de* Q *cujo valor* $V(s)$ *seja minimo.*
4. $estado(s) \leftarrow 1$.
5. ***Para Cada*** $t \in \mathcal{A}(s)$, *tal que* $estado(t) = 0$, ***Faça***
6. $tmp \leftarrow f(\pi_s \cdot \langle s, t \rangle)$.
7. ***Se*** $tmp < V(t)$, ***Então***
8. ***Se*** $V(t) \neq +\infty$, ***Então*** *remova* t *de* Q.
9. $P(t) \leftarrow s$, $V(t) \leftarrow tmp$, *e insira* t *em* Q.

Propagação dos caminhos



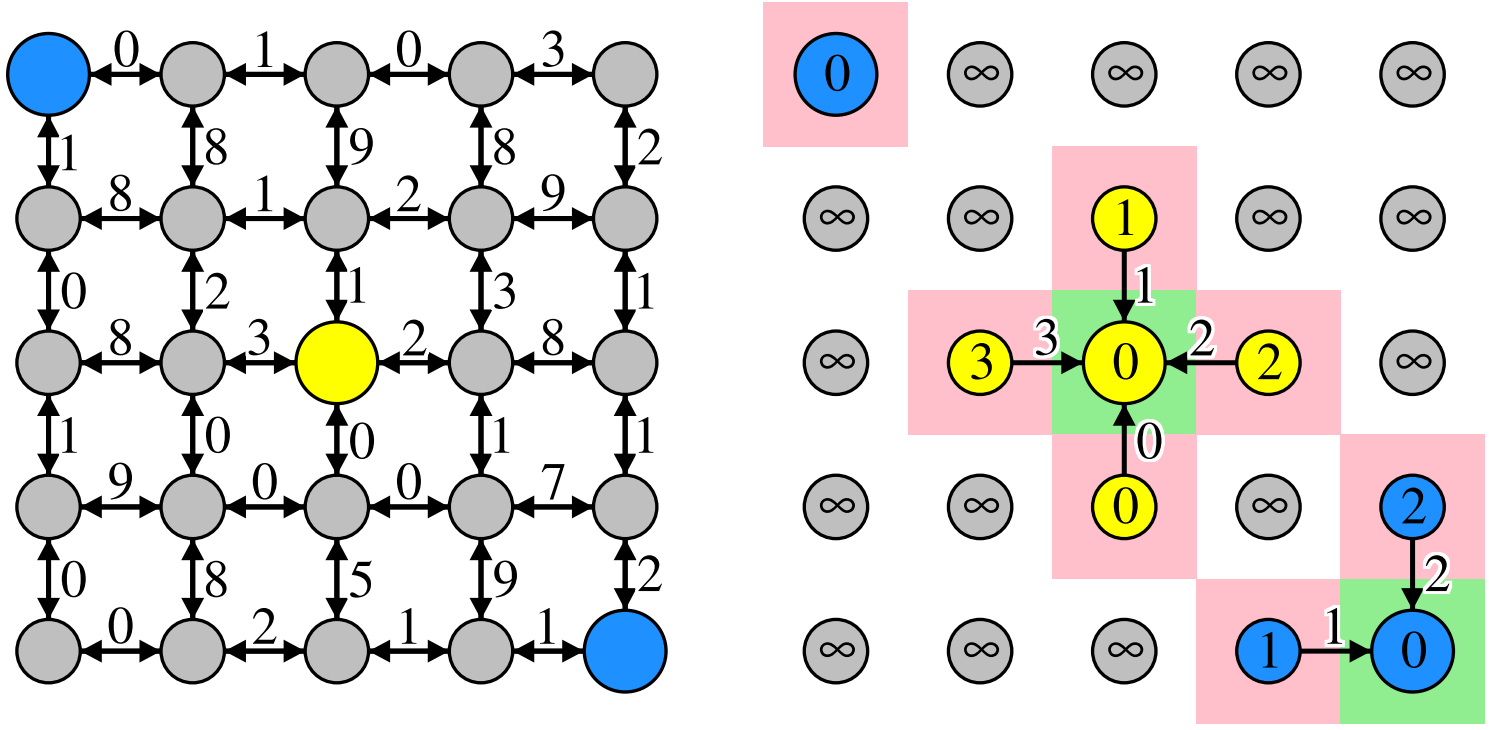


Propagação dos caminhos



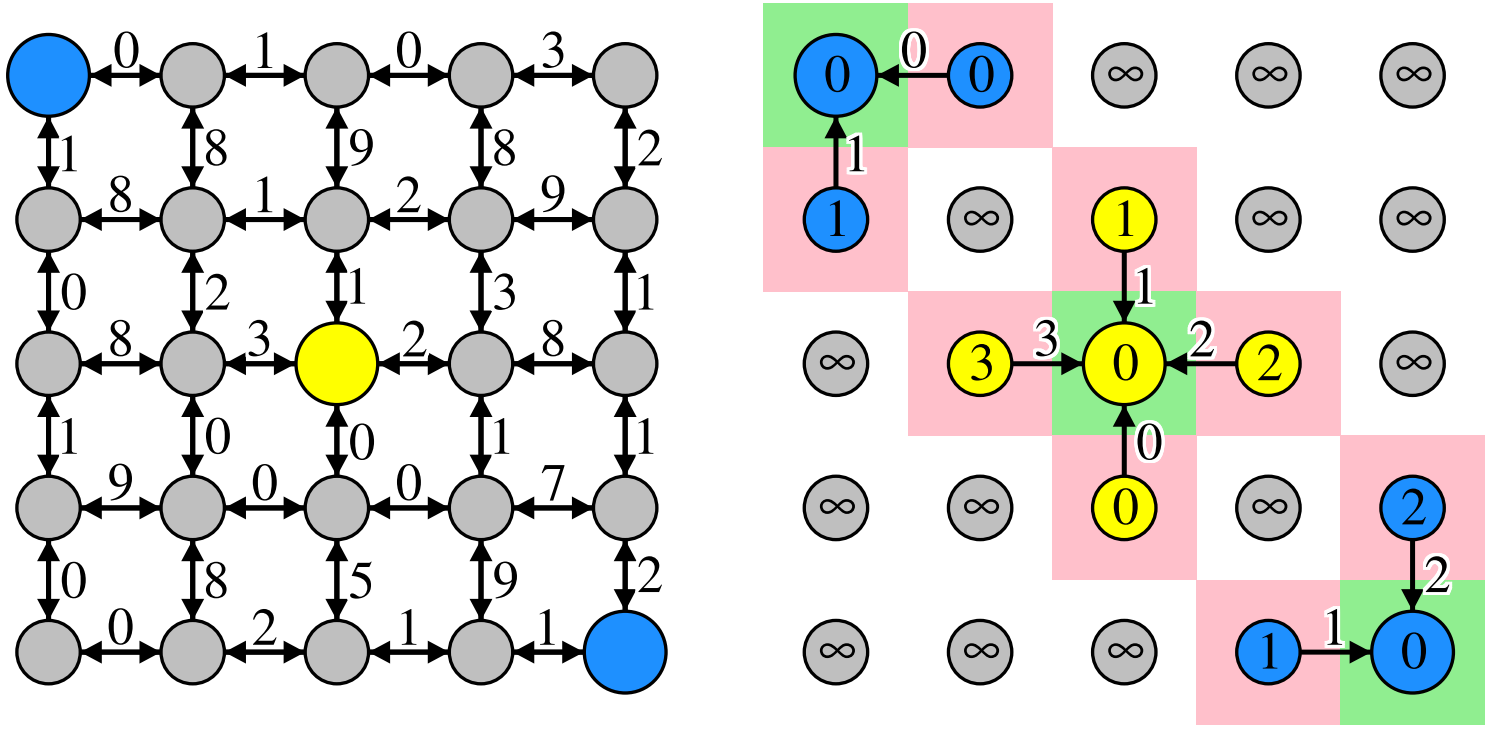
após 1 iteração.

Propagação dos caminhos



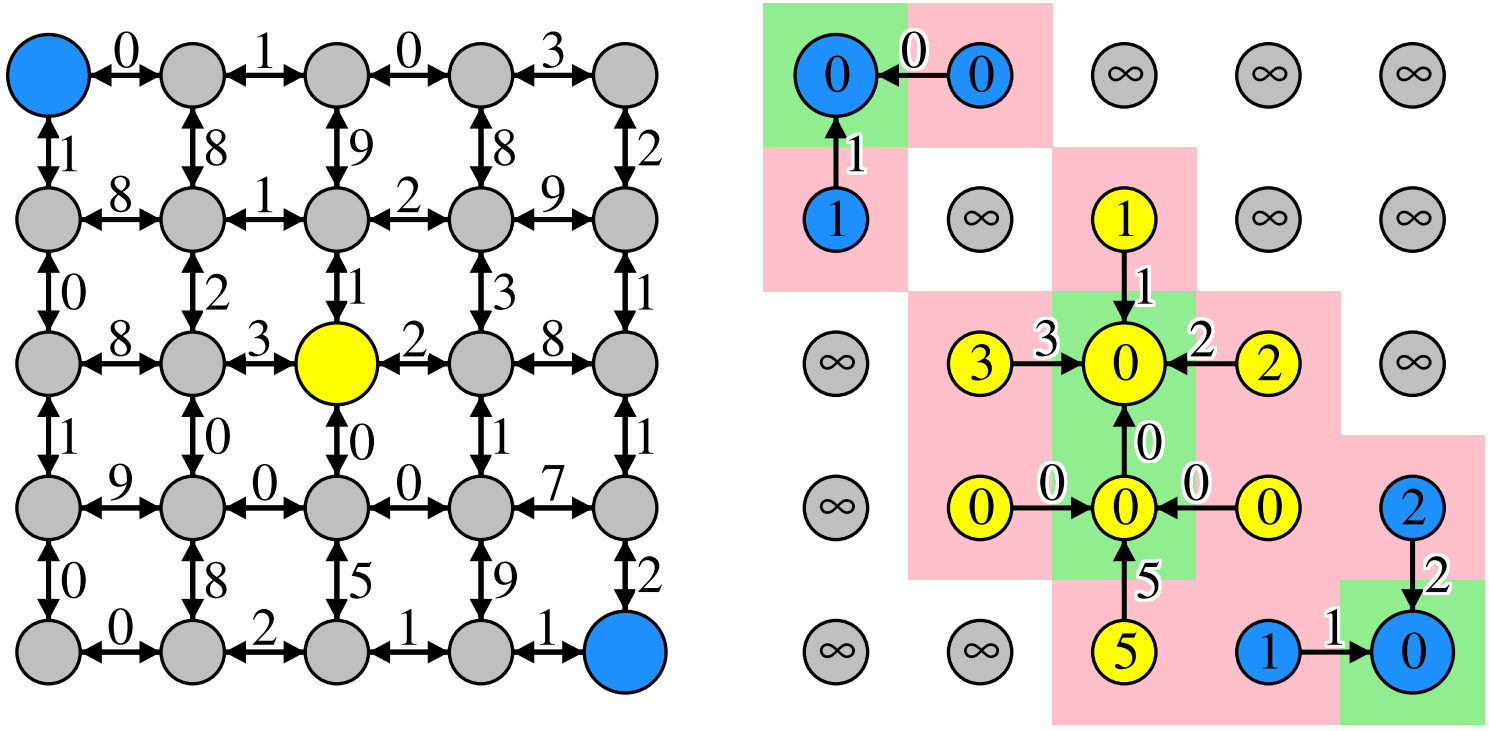
após 2 iterações.

Propagação dos caminhos



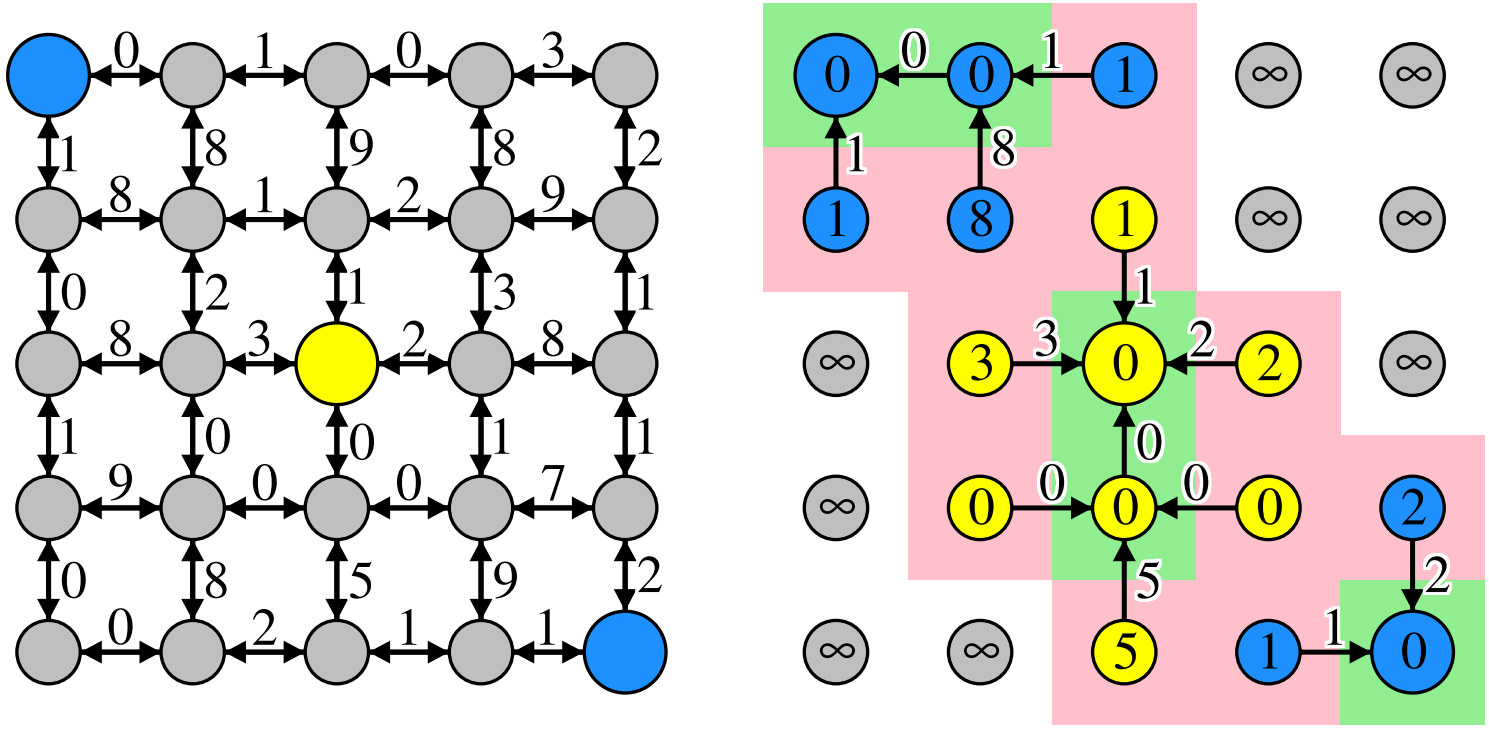
após 3 iterações.

Propagação dos caminhos



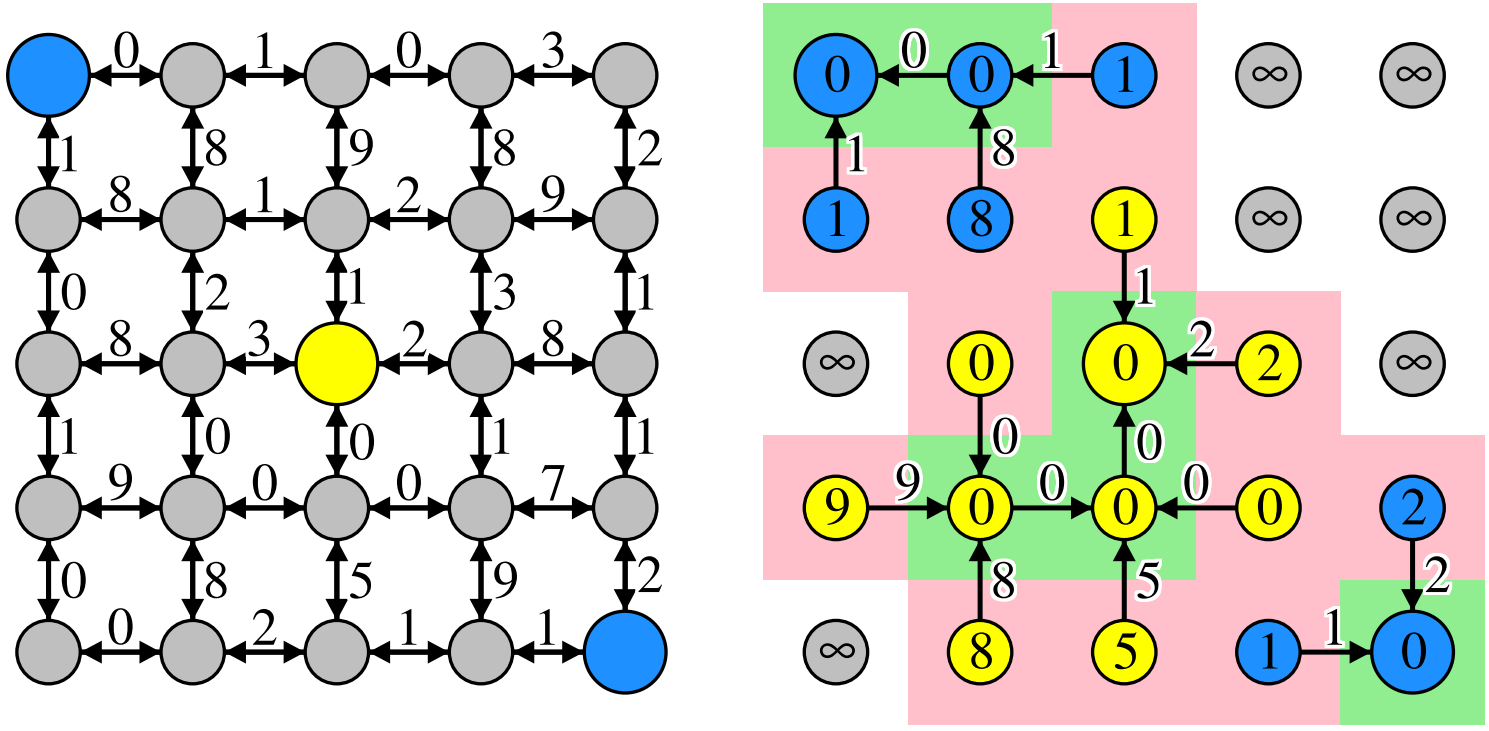
após 4 iterações.

Propagação dos caminhos



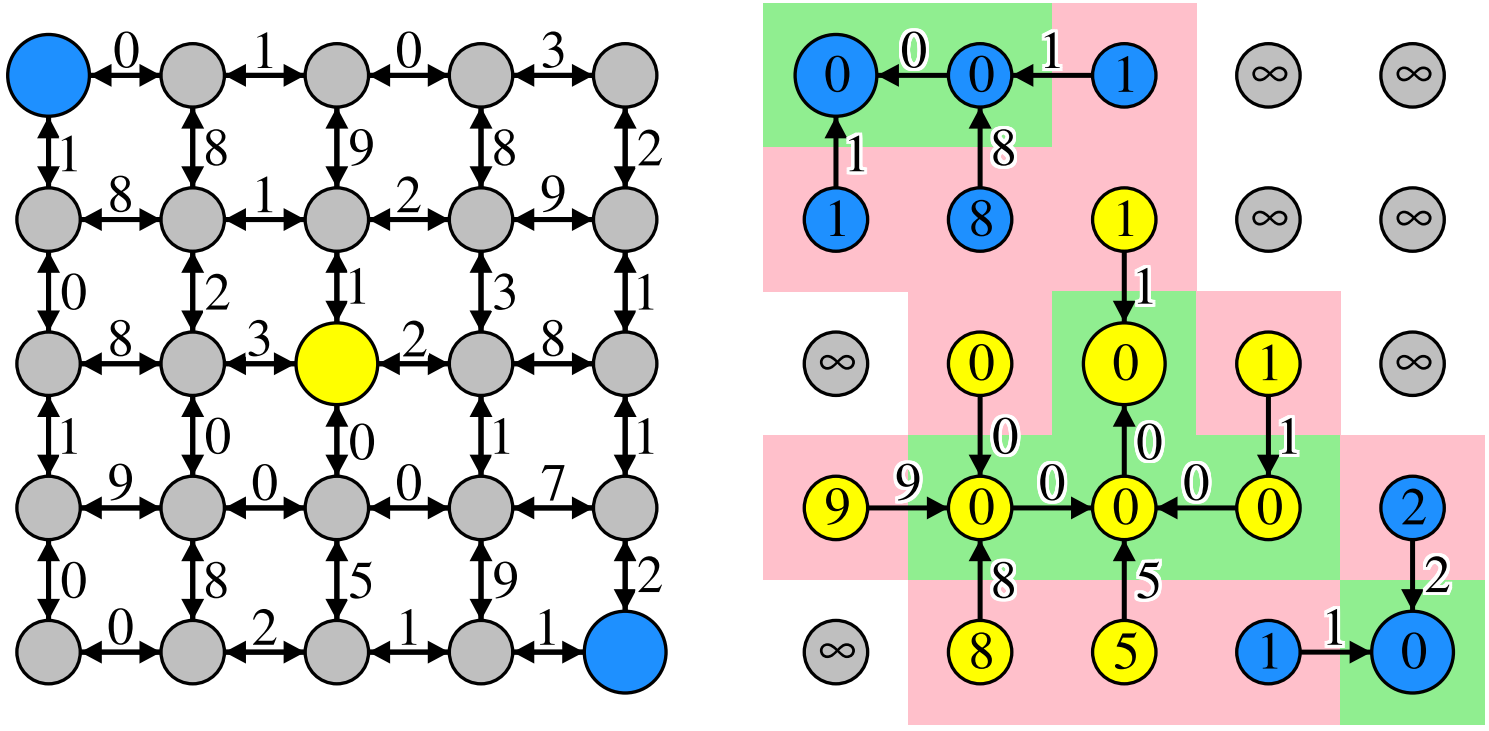
após 5 iterações.

Propagação dos caminhos



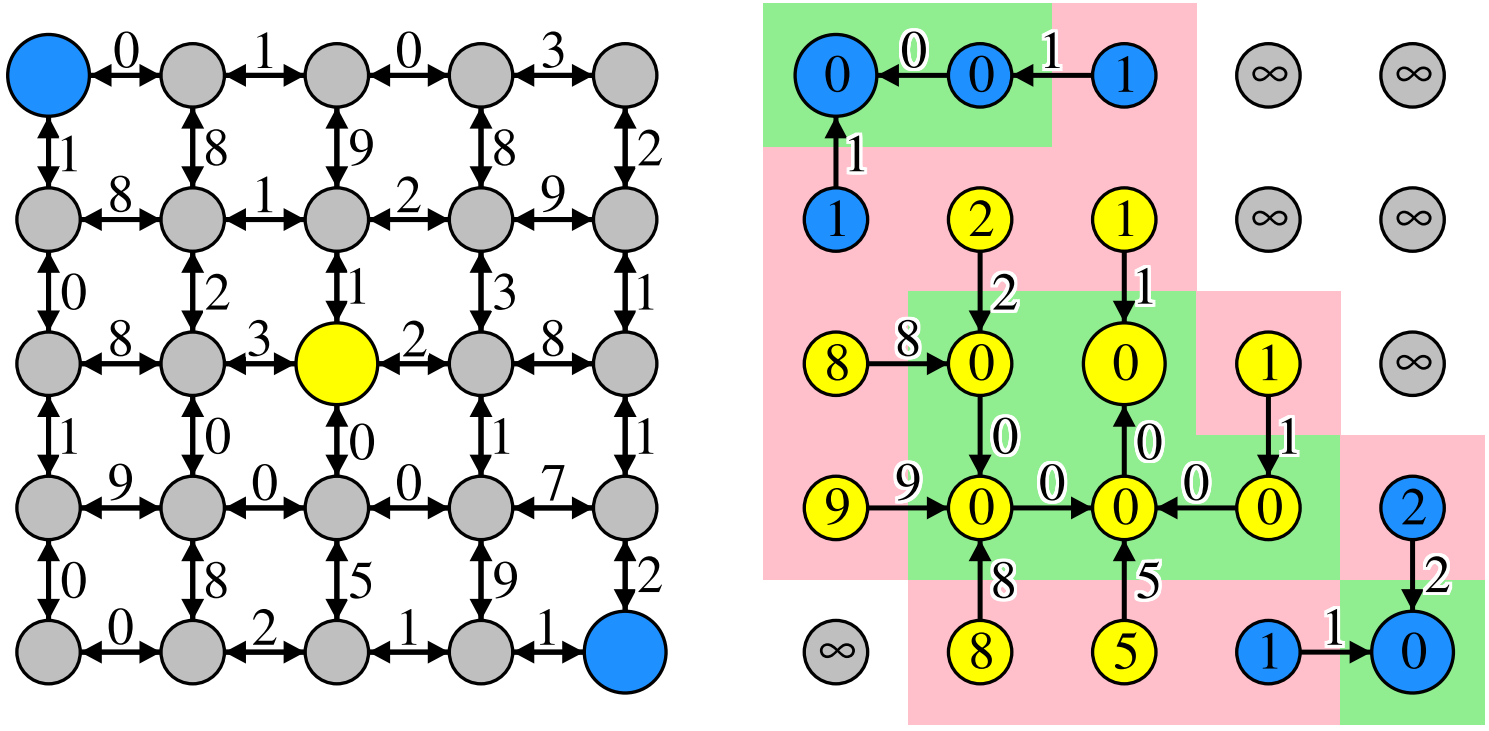
após 6 iterações.

Propagação dos caminhos



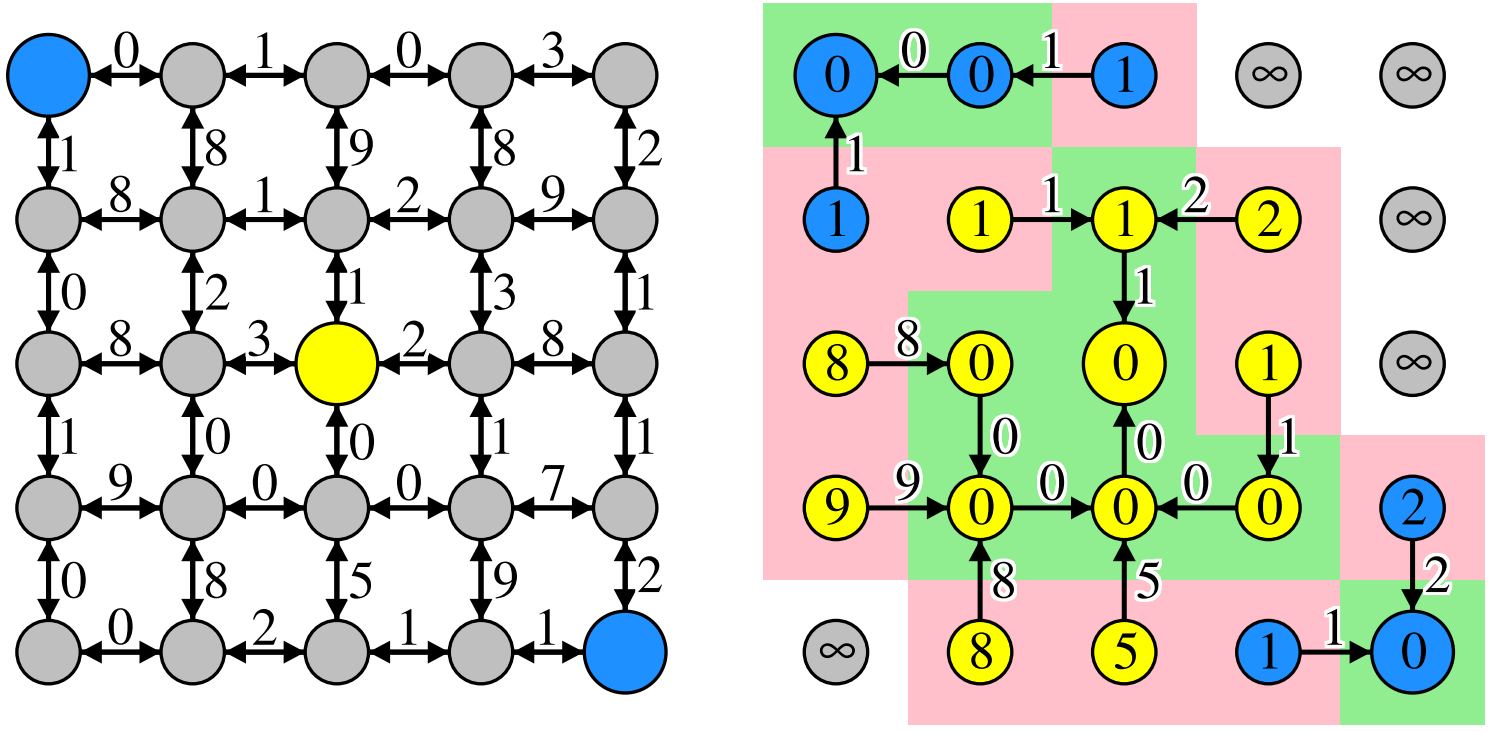
após 7 iterações.

Propagação dos caminhos



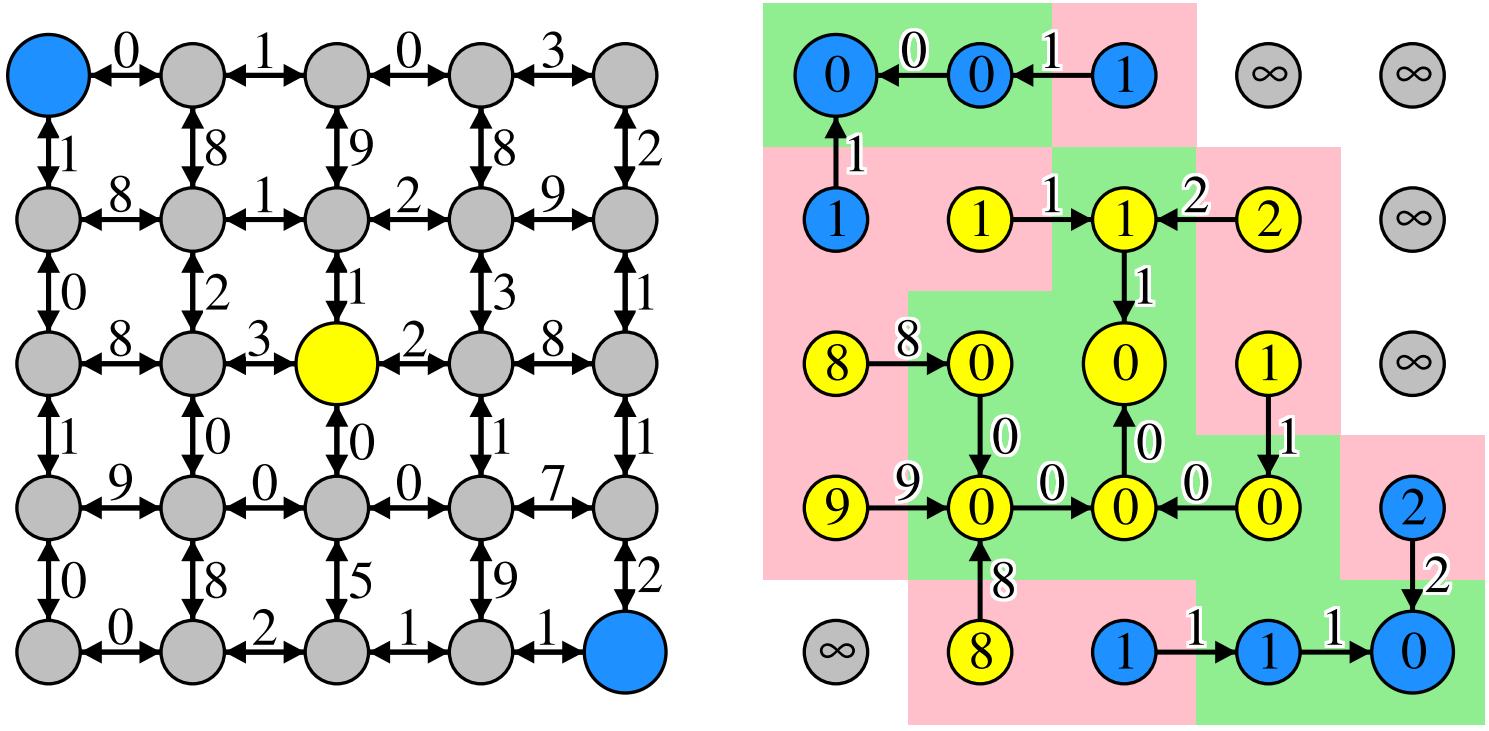
após 8 iterações.

Propagação dos caminhos



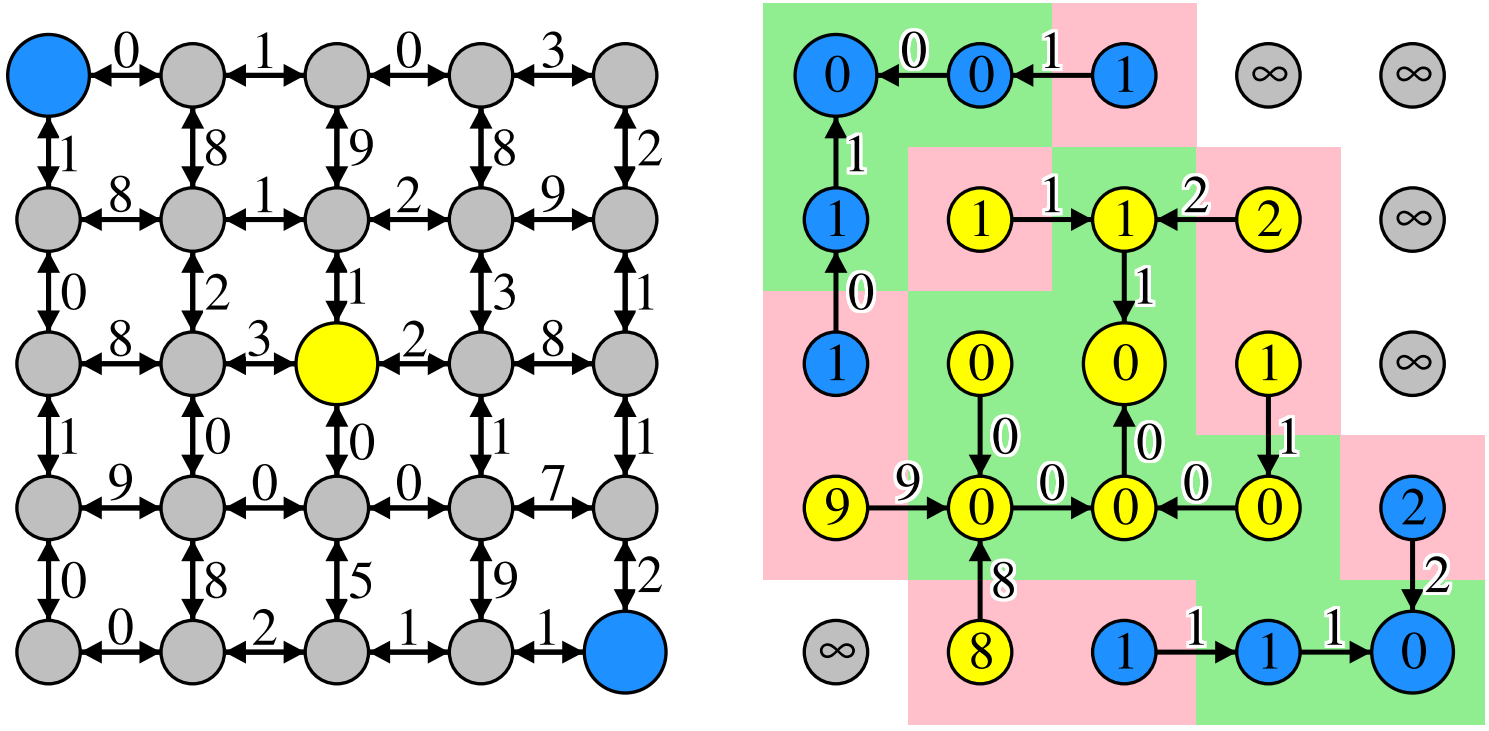
após 9 iterações.

Propagação dos caminhos



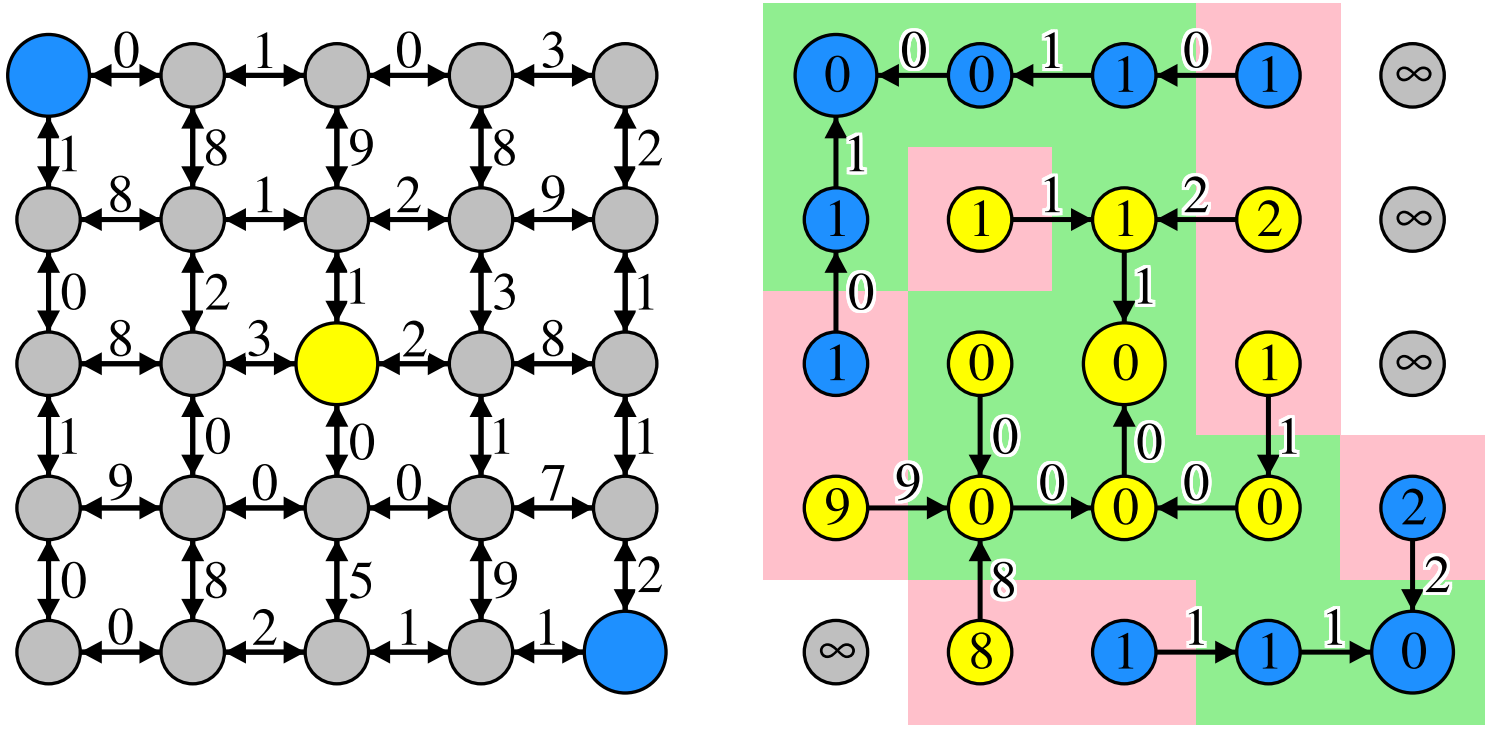
após 10 iterações.

Propagação dos caminhos



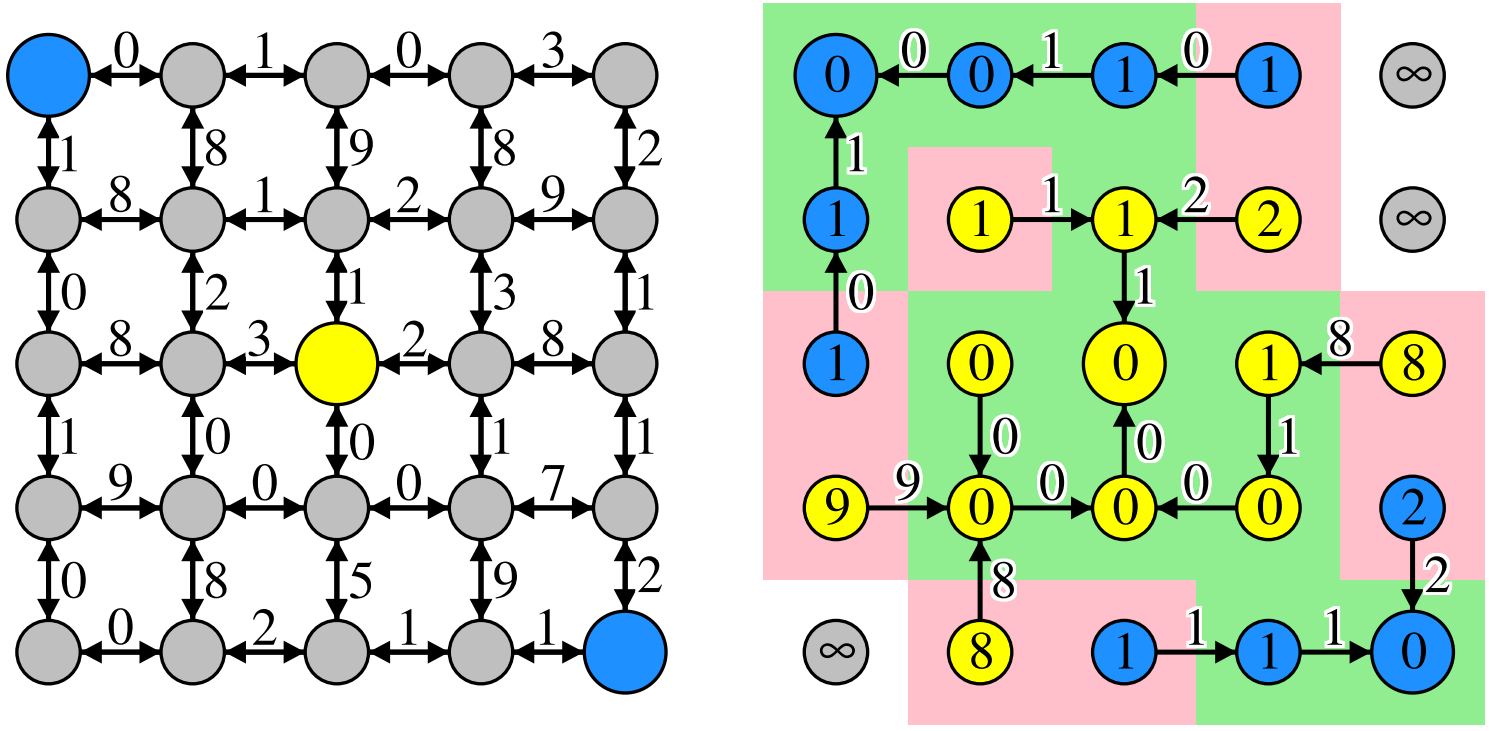
após 11 iterações.

Propagação dos caminhos



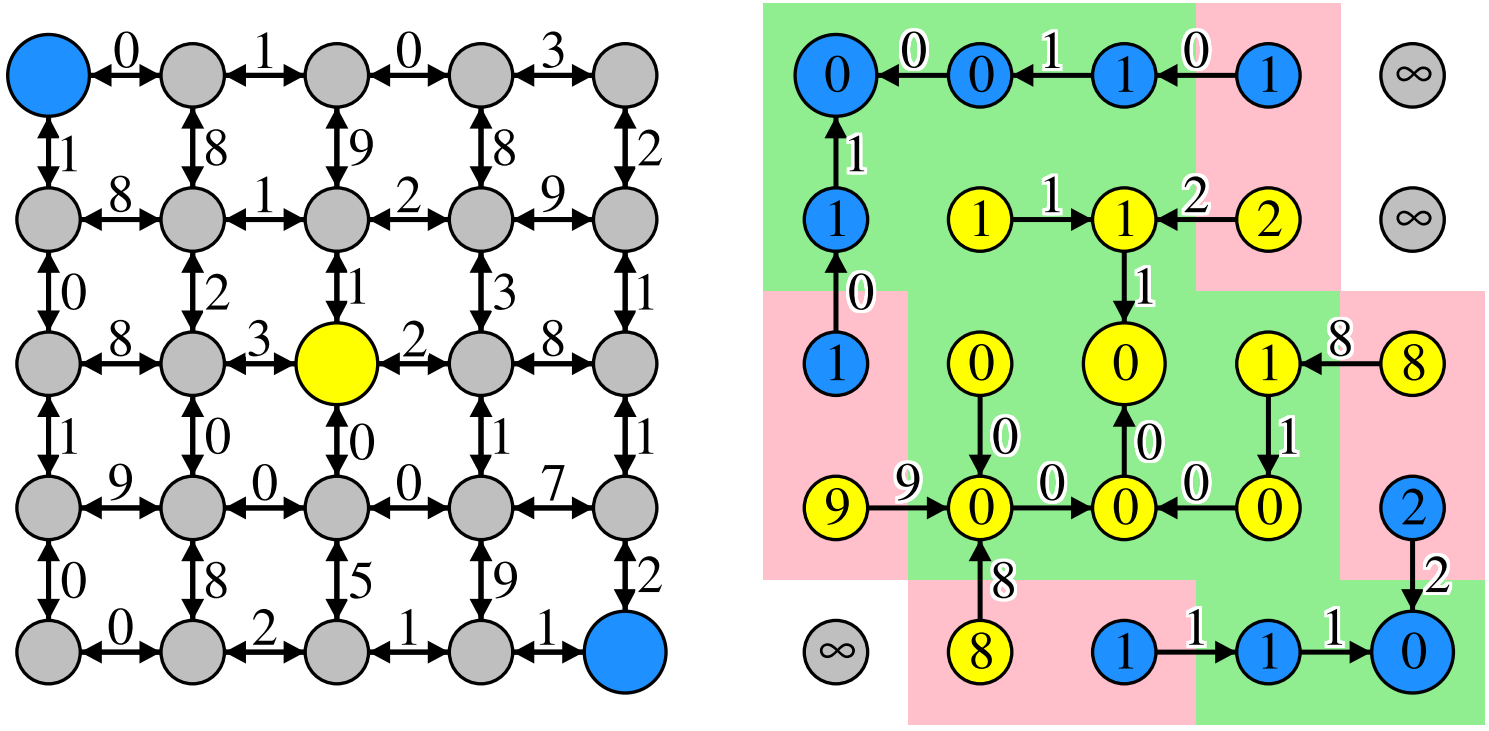
após 12 iterações.

Propagação dos caminhos



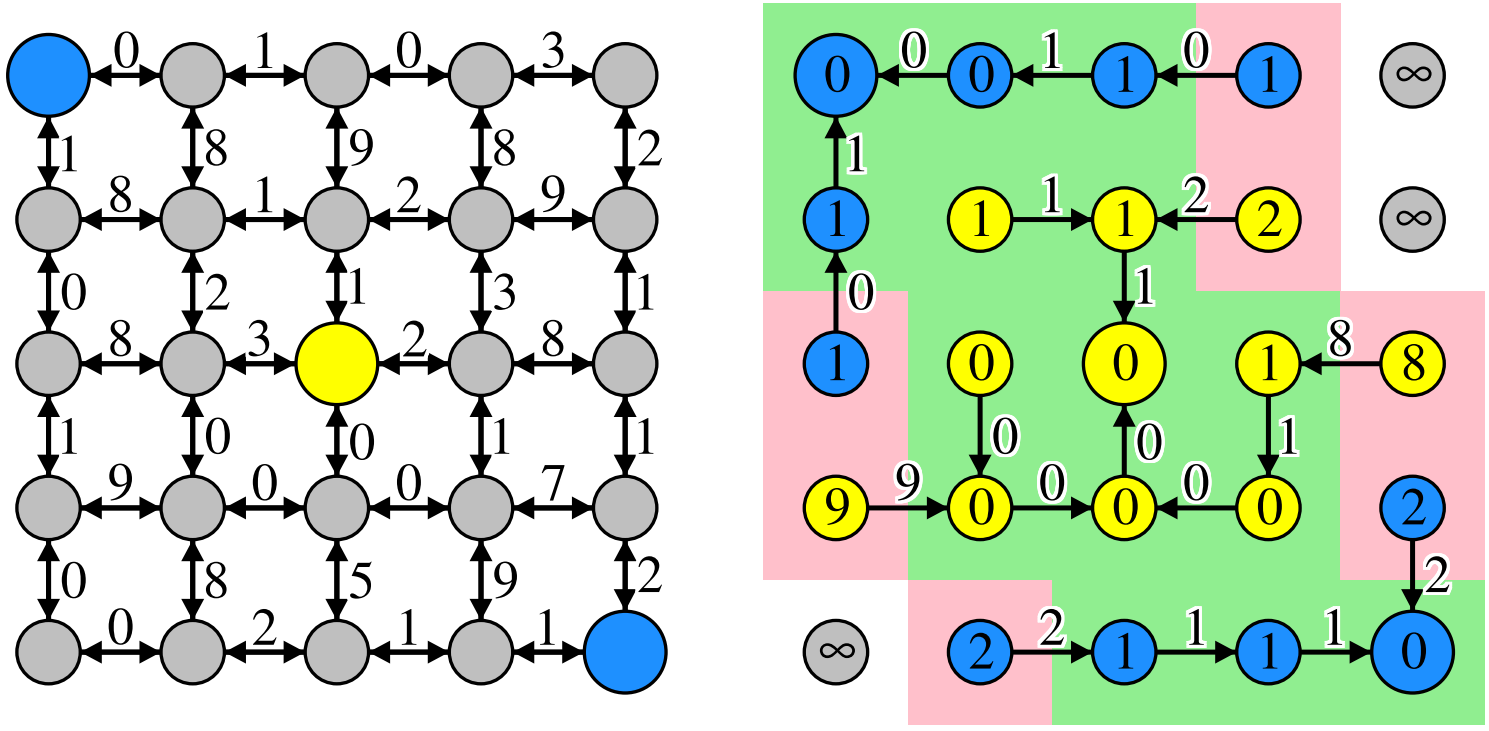
após 13 iterações.

Propagação dos caminhos



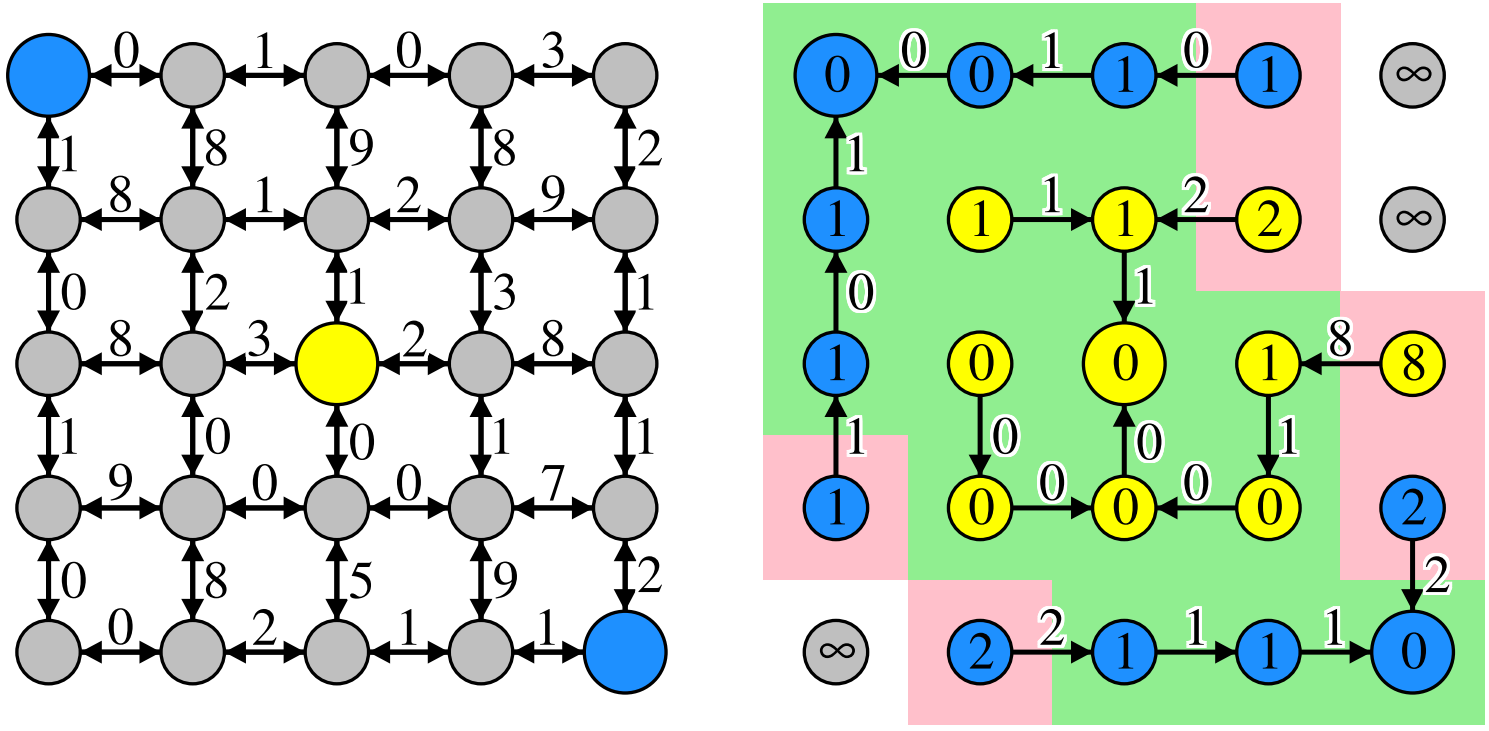
após 14 iterações.

Propagação dos caminhos



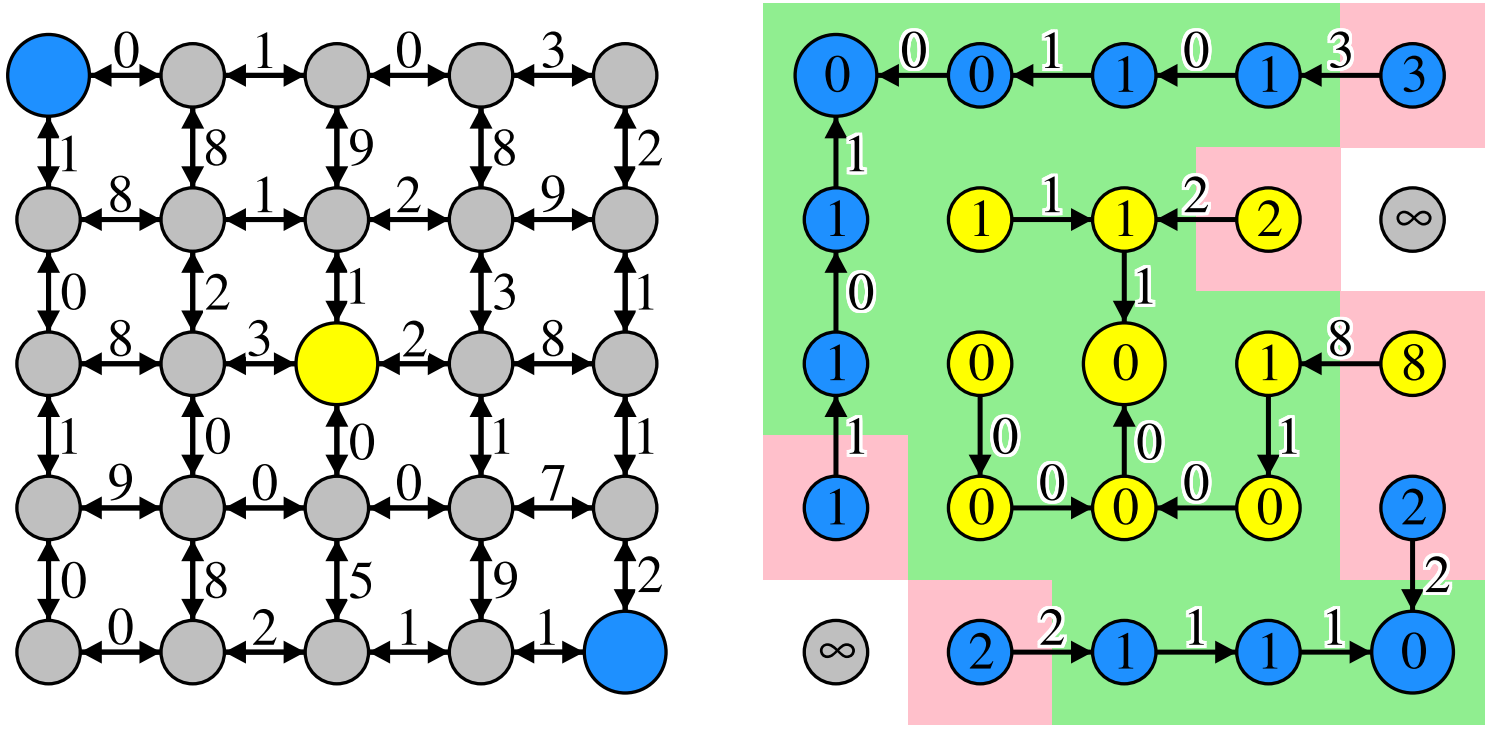
após 15 iterações.

Propagação dos caminhos



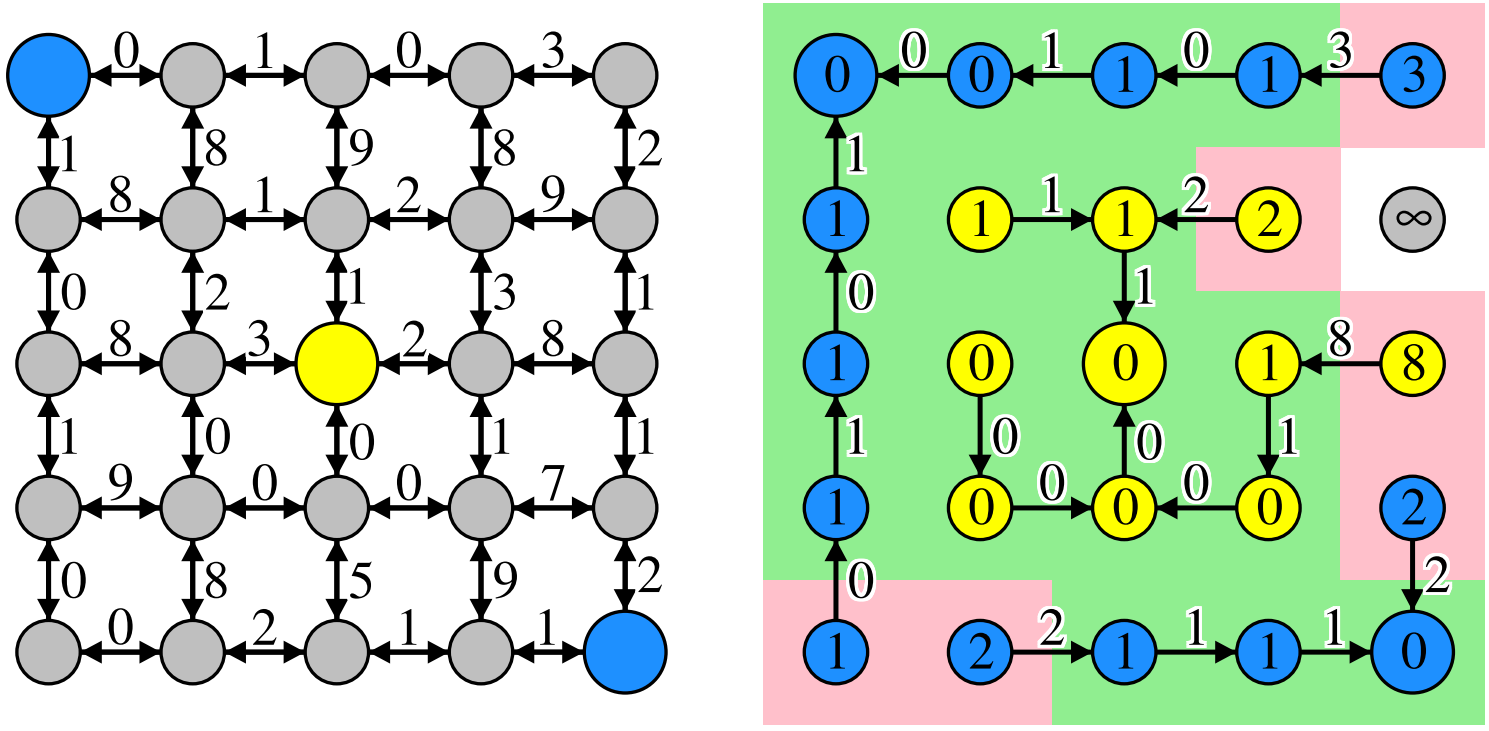
após 16 iterações.

Propagação dos caminhos



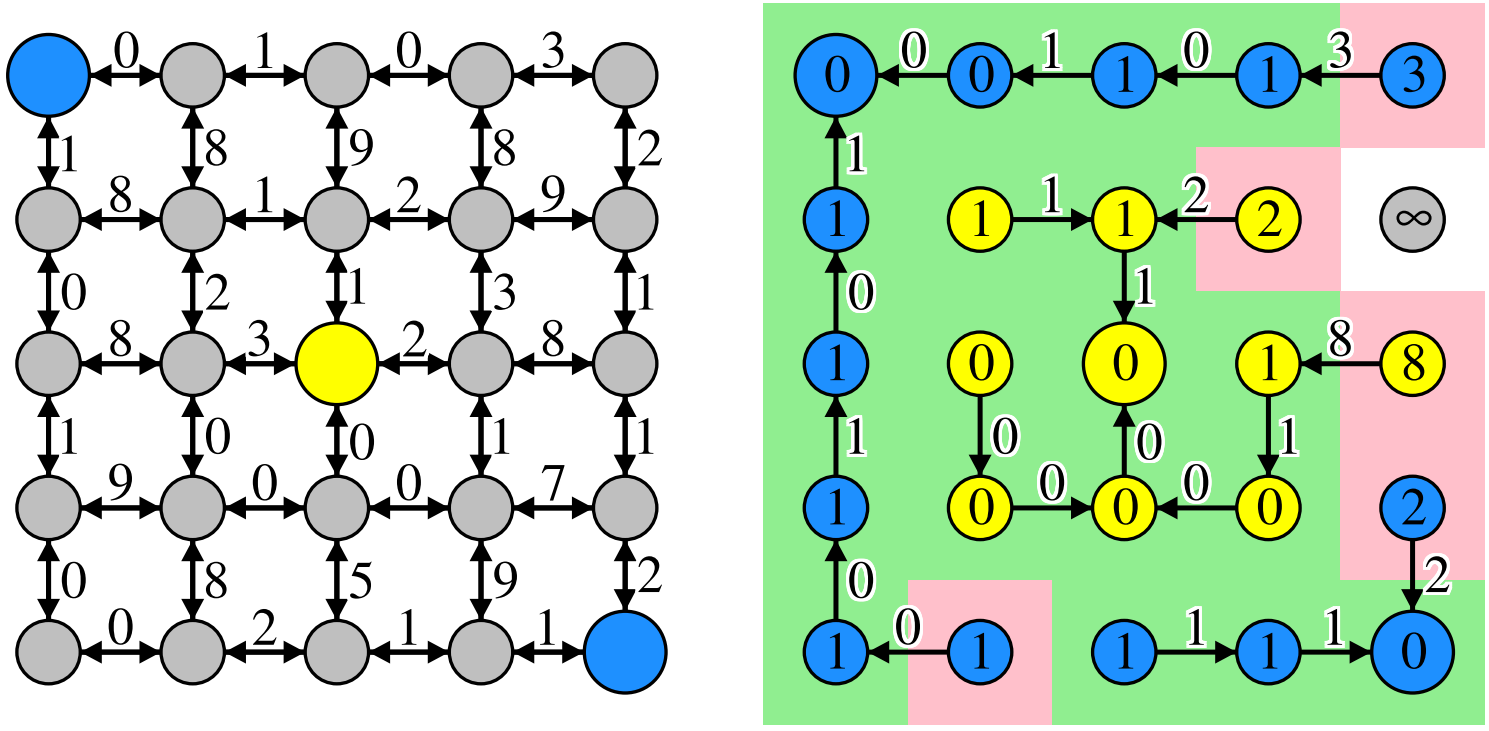
após 17 iterações.

Propagação dos caminhos



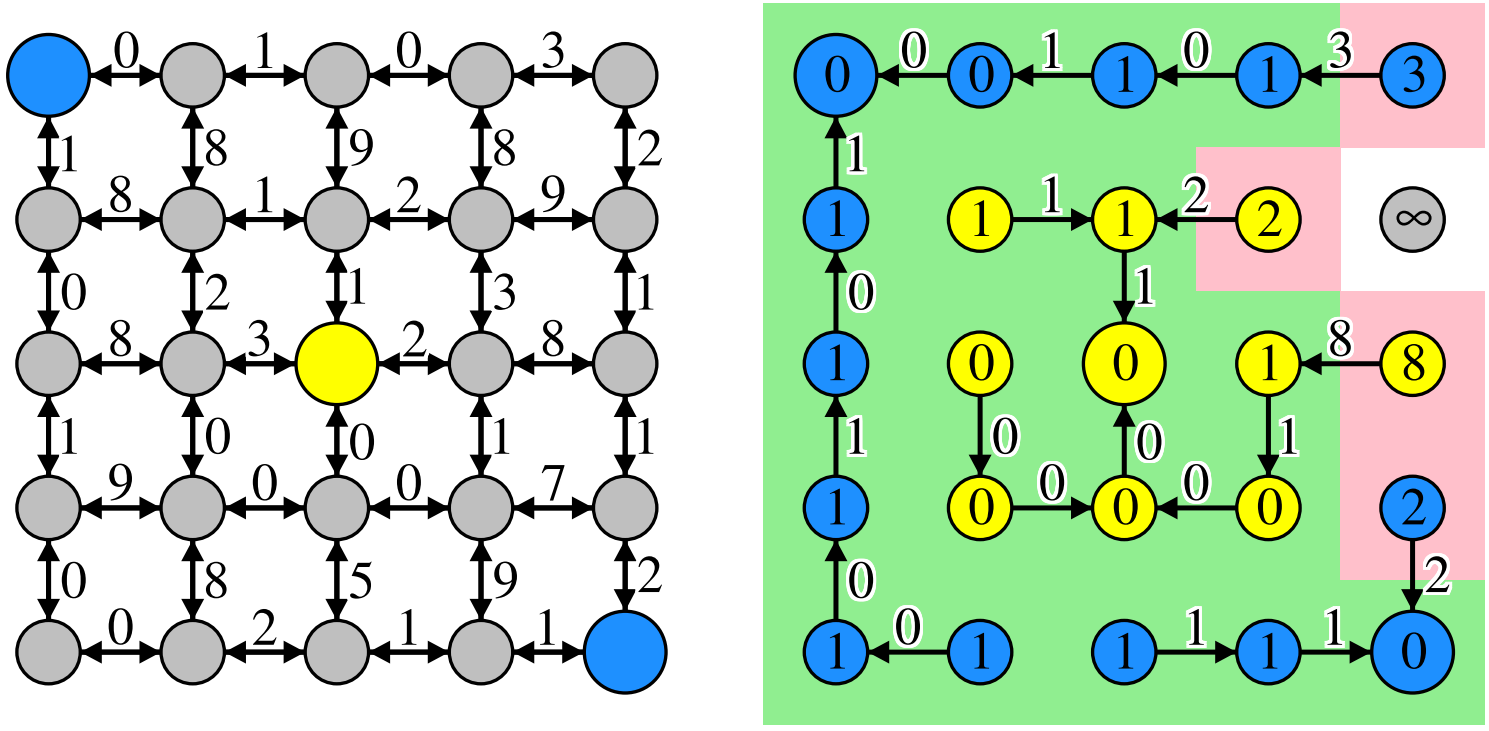
após 18 iterações.

Propagação dos caminhos



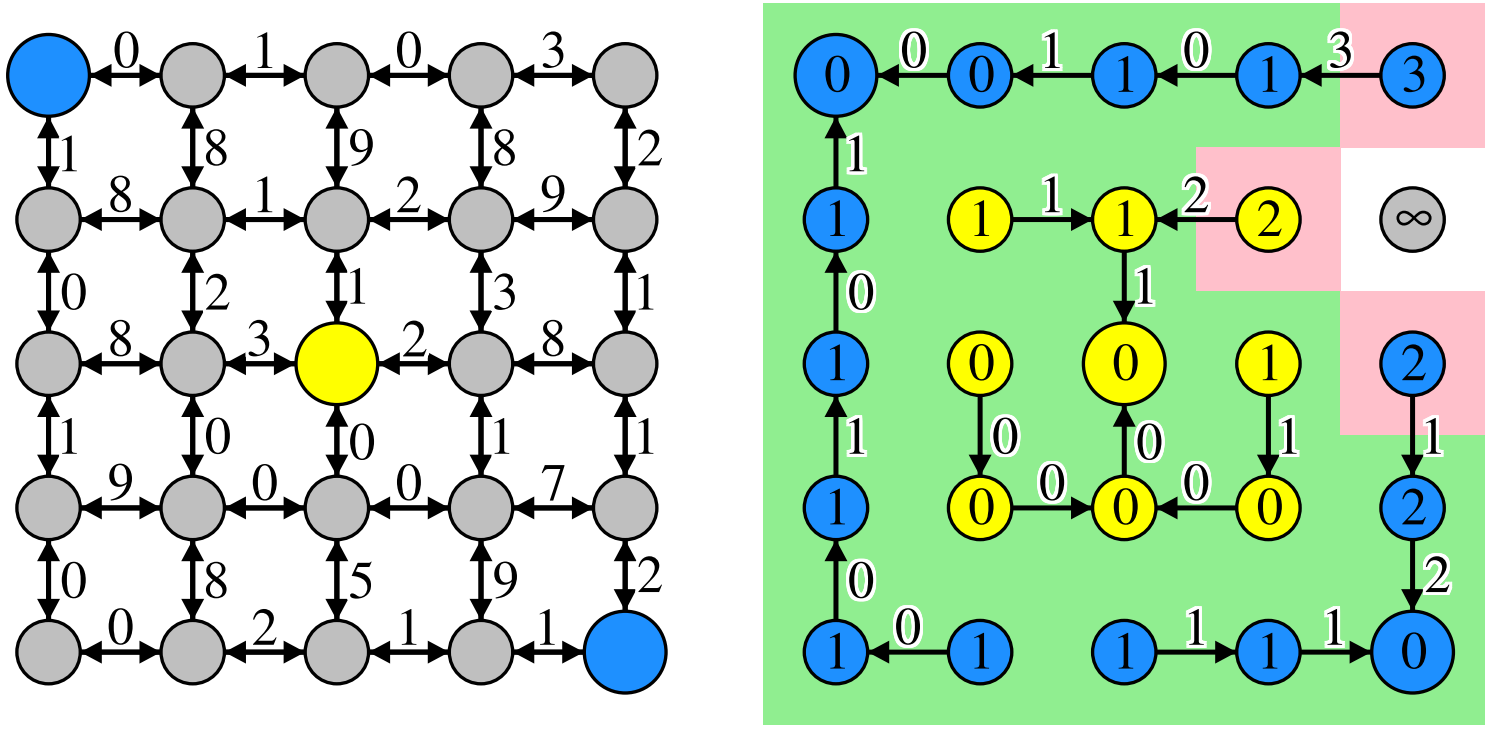
após 19 iterações.

Propagação dos caminhos



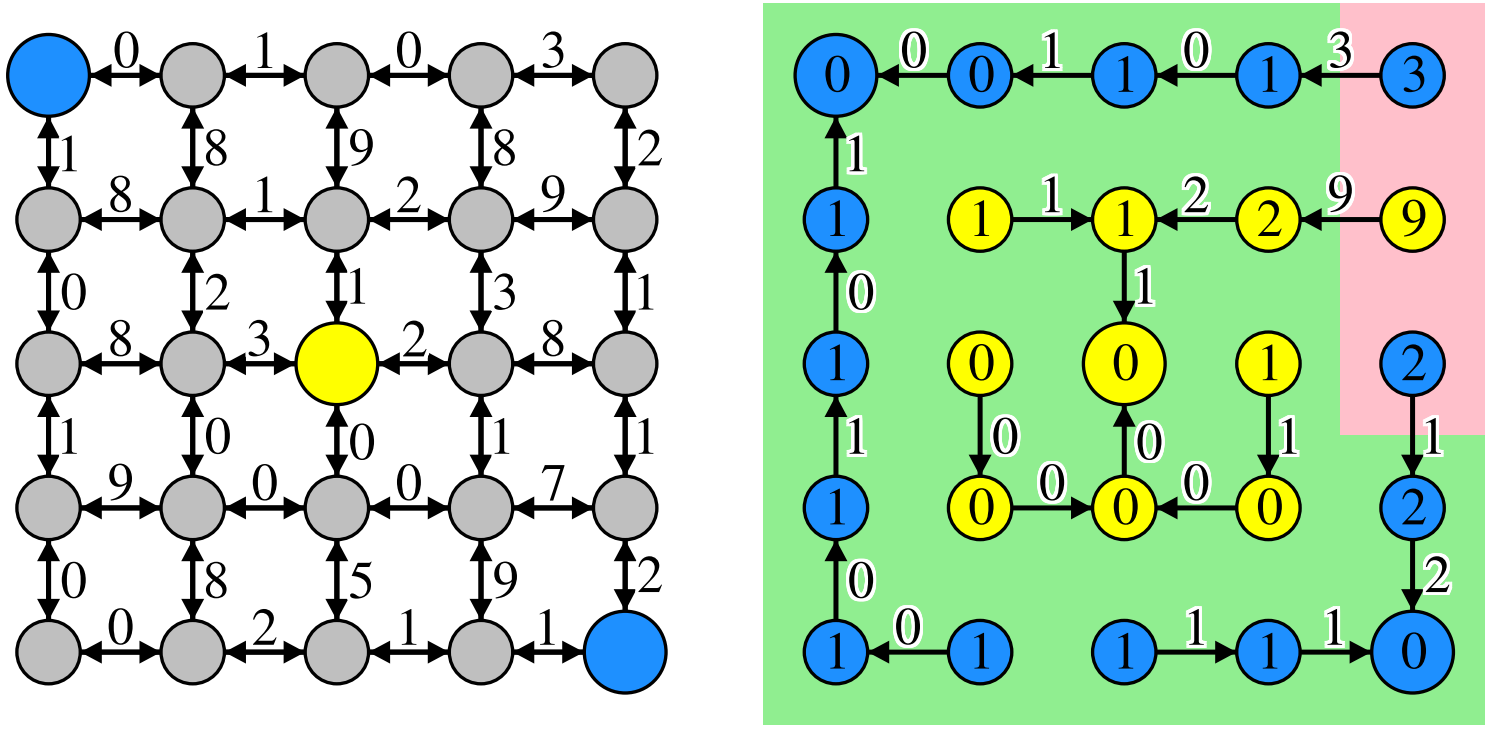
após 20 iterações.

Propagação dos caminhos



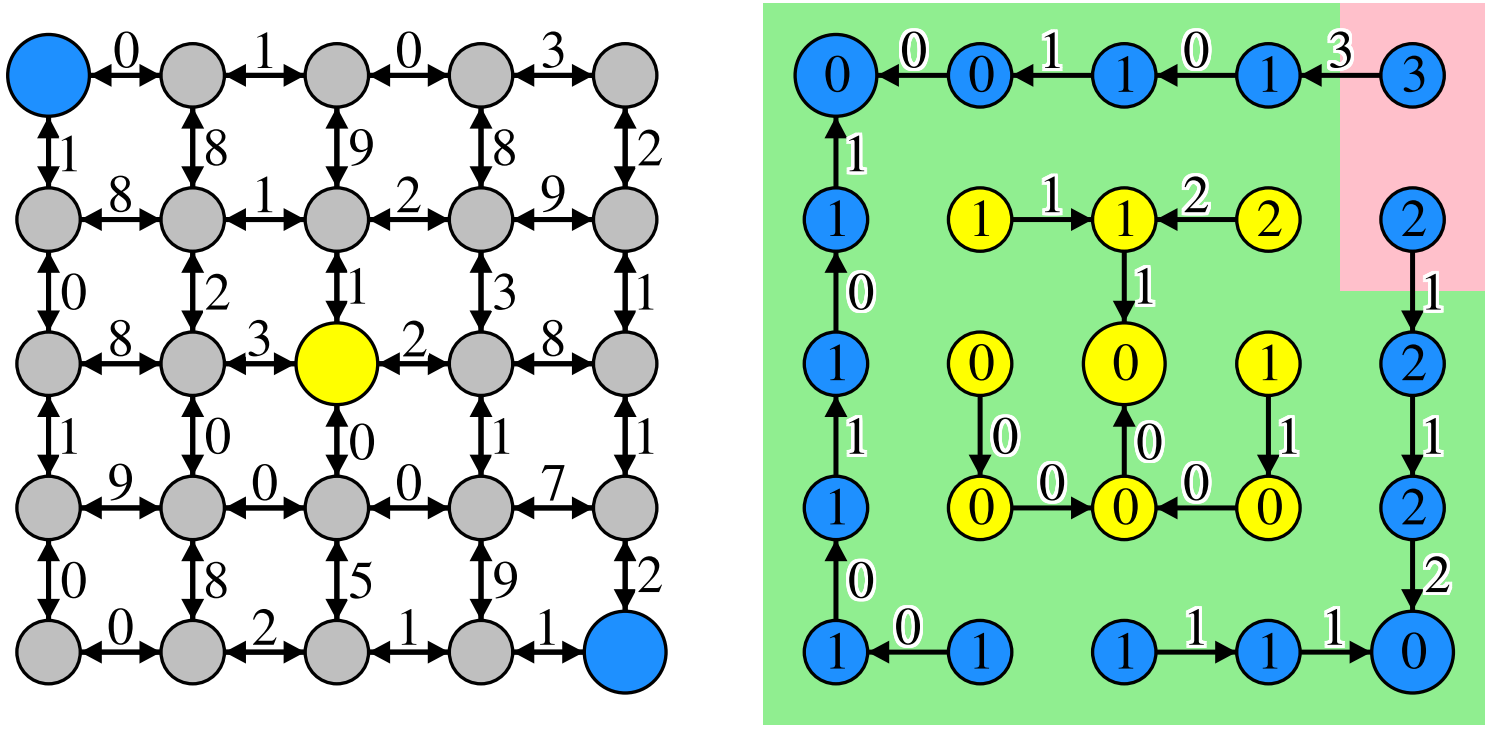
após 21 iterações.

Propagação dos caminhos



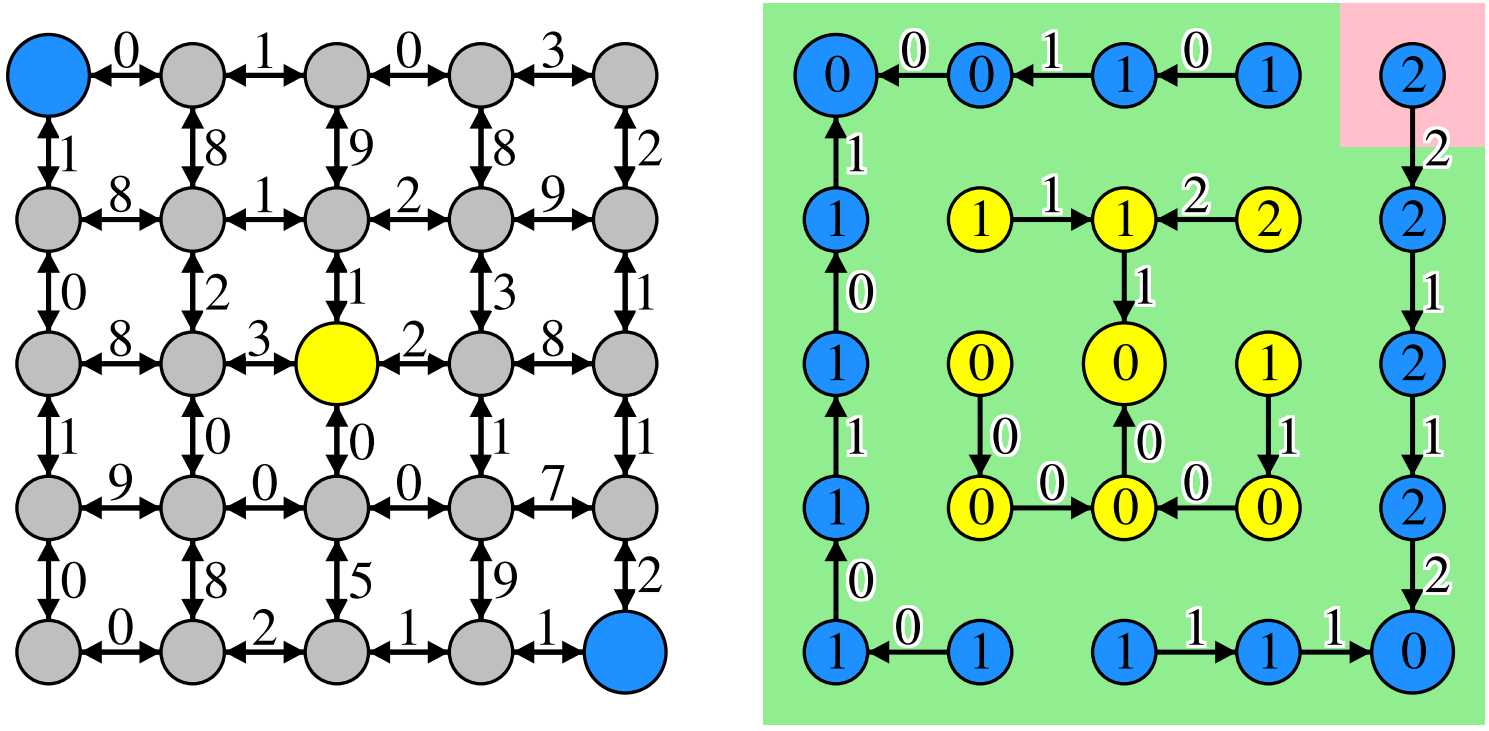
após 22 iterações.

Propagação dos caminhos



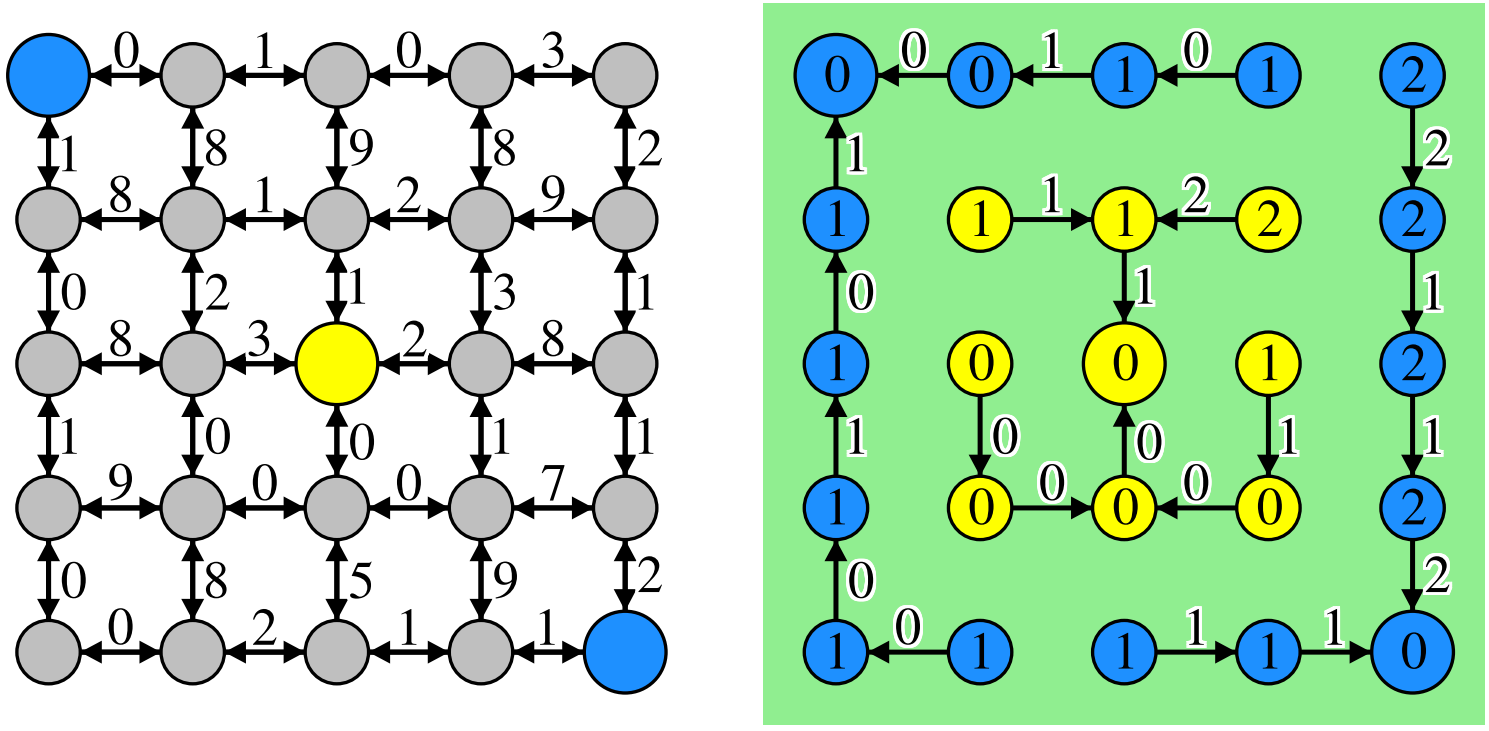
após 23 iterações.

Propagação dos caminhos



após 24 iterações.

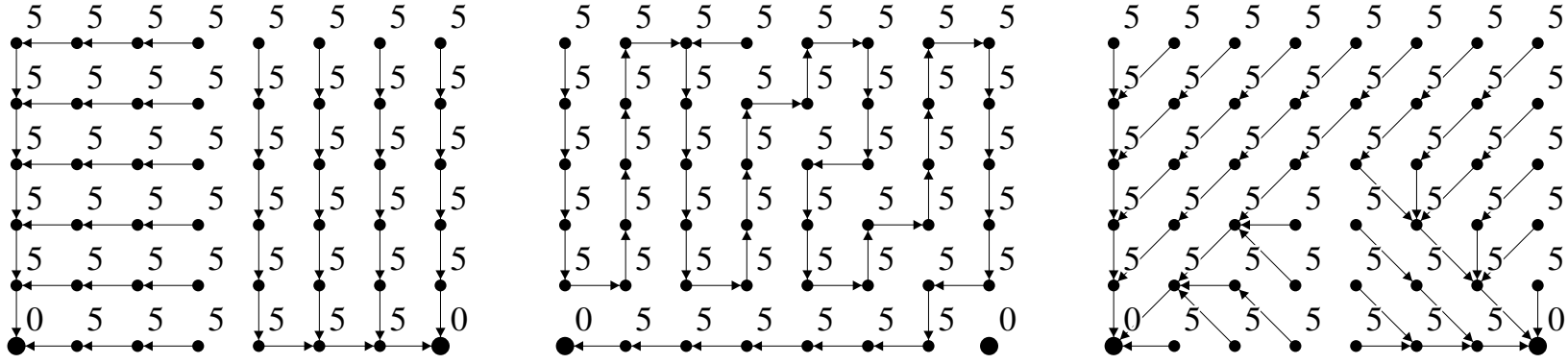
Propagação dos caminhos



após 25 iterações.

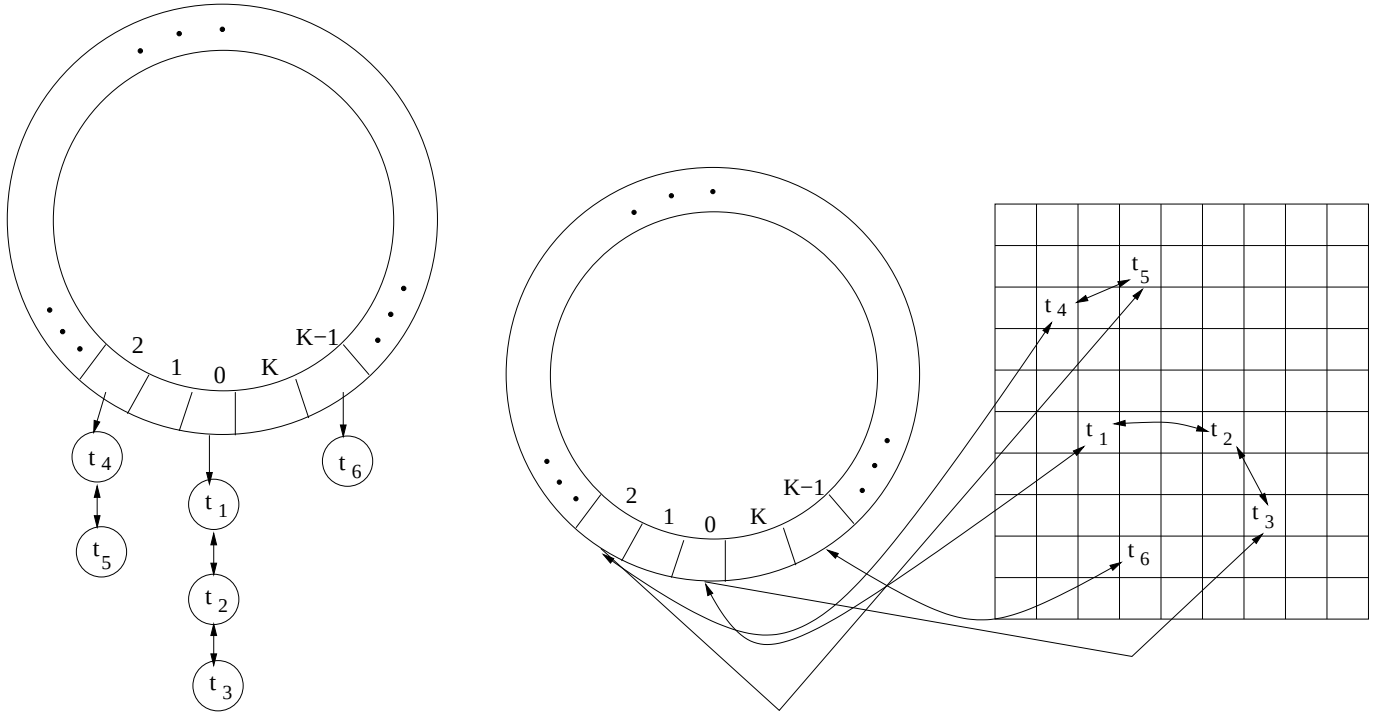
Resolvendo empates

O que fazer quando um pixel é alcançado por dois ou mais caminhos de mesmo custo?



Exemplos de *tie-breaking*. (a) Política FIFO. (b) Política LIFO. (c) Política FIFO com adjacência vizinhos-8.

Estrutura da fila de prioridade



(a) Estrutura de Dial para a fila Q. (b) Estrutura proposta por Falcão.