

# TỐI ƯU HÓA

## **Shakashaka Puzzle**

Khoa Toán - Cơ - Tin học  
Trường Đại học Khoa học Tự nhiên  
Đại học Quốc gia Hà Nội

Ngày 23 tháng 5 năm 2023

# Sinh viên thực hiện

**VŨ NHẬT TÂN - 20000583**

**NGUYỄN XUÂN TRƯỜNG - 20000591**

**LƯU VĂN VIỆT - 20000598**

# Nội dung

- 1 Giới thiệu đề tài
- 2 Xây dựng mô hình tối ưu nguyên cho bài toán
  - Thiết lập biến cho mô hình
  - Thiết lập ràng buộc cho mô hình
  - Thiết lập hàm mục tiêu
- 3 Sử dụng thư viện Gurobipy để cài đặt mô hình tối ưu tuyến tính cho trò chơi xếp hình Shakashaka
- 4 Tổng kết
- 5 Tài liệu tham khảo

# Giới thiệu đề tài

## Giới thiệu về trò chơi xếp hình Shakashaka

# Giới thiệu đề tài

## Giới thiệu về trò chơi xếp hình Shakashaka

Trò chơi xếp hình Shakashaka là một trong những trò chơi xếp hình được phát triển bởi nhà xuất bản Nikoli.

# Giới thiệu đề tài

## Giới thiệu về trò chơi xếp hình Shakashaka

Trò chơi xếp hình Shakashaka là một trong những trò chơi xếp hình được phát triển bởi nhà xuất bản Nikoli.

Shakashaka được chính thức đưa ra vào năm 2008 bởi Guten và sau đó trở thành một trong những trò chơi xếp hình phổ biến của NXB Nikoli.

# Giới thiệu đề tài

## Luật chơi Shakashaka

# Giới thiệu đề tài

## Luật chơi Shakashaka

Bảng trạng thái bắt đầu của trò chơi Shakashaka là một bảng vuông cỡ  $n$  gồm  $n \times n$  hình vuông đơn vị.



# Giới thiệu đề tài

## Luật chơi Shakashaka

Bảng trạng thái bắt đầu của trò chơi Shakashaka là một bảng vuông cỡ  $n$  gồm  $n \times n$  hình vuông đơn vị.

Mỗi hình vuông đơn vị có màu trắng hoặc đen. Một số hình vuông màu đen có thể chứa chứa một số nguyên  $k \geq 0$ .



# Giới thiệu đề tài

## Luật chơi Shakashaka

# Giới thiệu đề tài

## Luật chơi Shakashaka

Mục tiêu của trò chơi là điền vào các ô màu trắng bằng 1 trong 5 ô sau:



# Giới thiệu đề tài

## Luật chơi Shakashaka

Mục tiêu của trò chơi là điền vào các ô màu trắng bằng 1 trong 5 ô sau:



Sao cho:

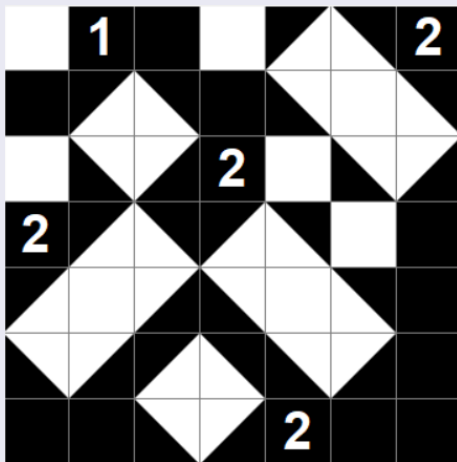
- Xung quanh các ô màu đen có chứa số  $k$  theo hướng dọc và ngang có đúng  $k$  ô vuông có chứa hình tam giác.
- Các khoảng trắng sau khi điền hết toàn bộ bảng phải là các hình chữ nhật không giao nhau.

# Giới thiệu đề tài

## Luật chơi Shakashaka

# Giới thiệu đề tài

## Luật chơi Shakashaka



# Thiết lập biến cho mô hình

## Thiết lập biến của mô hình



# Thiết lập biến cho mô hình

## Thiết lập biến của mô hình

Với mỗi ô vuông  $(i, j)$  màu trắng, ta cần điền vào 1 trong 5 ô vuông dạng



# Thiết lập biến cho mô hình

## Thiết lập biến của mô hình

Với mỗi ô vuông  $(i, j)$  màu trắng, ta cần điền vào 1 trong 5 ô vuông dạng  $\square, \blacksquare, \blacktriangleleft, \blacktriangleright, \blacksquare$ .

Do đó, ta biểu thị trạng thái mỗi ô vuông  $(i, j)$  bởi 5 biến nhị phân:  $x[i, j, \square], x[i, j, \blacksquare], x[i, j, \blacktriangleleft], x[i, j, \blacktriangleright], x[i, j, \blacksquare]$ .

$$\begin{cases} x[i, j, \square] = 1 \text{ nếu ô } (i, j) \text{ điền } \square \\ x[i, j, \blacksquare] = 1 \text{ nếu ô } (i, j) \text{ điền } \blacksquare \\ x[i, j, \blacktriangleleft] = 1 \text{ nếu ô } (i, j) \text{ điền } \blacktriangleleft \\ x[i, j, \blacktriangleright] = 1 \text{ nếu ô } (i, j) \text{ điền } \blacktriangleright \\ x[i, j, \blacksquare] = 1 \text{ nếu ô } (i, j) \text{ điền } \blacksquare \end{cases}$$

# Thiết lập biến cho mô hình

## Thiết lập biến của mô hình

Với mỗi ô vuông  $(i, j)$  màu trắng, ta cần điền vào 1 trong 5 ô vuông dạng  $\square, \blacksquare, \blacktriangleleft, \blacktriangleright, \blacksquare$ .

Do đó, ta biểu thị trạng thái mỗi ô vuông  $(i, j)$  bởi 5 biến nhị phân:  $x[i, j, \square], x[i, j, \blacksquare], x[i, j, \blacktriangleleft], x[i, j, \blacktriangleright], x[i, j, \blacksquare]$ .

$$\begin{cases} x[i, j, \square] = 1 \text{ nếu ô } (i, j) \text{ điền } \square \\ x[i, j, \blacksquare] = 1 \text{ nếu ô } (i, j) \text{ điền } \blacksquare \\ x[i, j, \blacktriangleleft] = 1 \text{ nếu ô } (i, j) \text{ điền } \blacktriangleleft \\ x[i, j, \blacktriangleright] = 1 \text{ nếu ô } (i, j) \text{ điền } \blacktriangleright \\ x[i, j, \blacksquare] = 1 \text{ nếu ô } (i, j) \text{ điền } \blacksquare \end{cases}$$






Với mỗi ô vuông  $(i, j)$  màu đen, ta không được điền ô vuông nào khác vào nên  $x[i, j, *] = 0$ .

# Thiết lập ràng buộc cho mô hình

Ràng buộc đối với các ô vuông màu trắng

# Thiết lập ràng buộc cho mô hình

## Ràng buộc đối với các ô vuông màu trắng

**Constraint 1:** Đối với mỗi ô vuông  $(i, j)$  màu trắng, ta chỉ được điền vào duy nhất một trong 5 ô , , , , .

# Thiết lập ràng buộc cho mô hình

## Ràng buộc đối với các ô vuông màu trắng

**Constraint 1:** Đối với mỗi ô vuông  $(i, j)$  màu trắng, ta chỉ được điền vào duy nhất một trong 5 ô  $\square, \blacksquare, \blacktriangleleft, \blacktriangleright, \blacktriangledown$ .

$$x[i, j, \square] + x[i, j, \blacksquare] + x[i, j, \blacktriangleleft] + x[i, j, \blacktriangleright] + x[i, j, \blacktriangledown] = 1$$

Với mọi  $(i, j)$  mà trong bảng trạng thái xuất phát, ô vuông  $(i, j)$  là ô vuông màu trắng.

# Thiết lập ràng buộc cho mô hình

Ràng buộc đối với các ô vuông màu đen

# Thiết lập ràng buộc cho mô hình

## Ràng buộc đối với các ô vuông màu đen

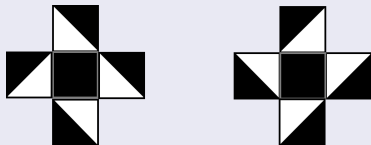
**Constraint 2:** Xung quanh các ô vuông màu đen theo hướng dọc và ngang không được điền các ô vuông có chứa hình tam giác mà tạo thành góc  $45^\circ$ .



# Thiết lập ràng buộc cho mô hình

## Ràng buộc đối với các ô vuông màu đen

**Constraint 2:** Xung quanh các ô vuông màu đen theo hướng dọc và ngang không được điền các ô vuông có chứa hình tam giác mà tạo thành góc  $45^\circ$ .



$$\begin{aligned} x[i-1, j, \blacksquare] &= x[i, j-1, \blacktriangle] = x[i+1, j, \blacksquare] = x[i, j+1, \blacktriangle] \\ &= x[i-1, j, \blacktriangle] = x[i, j-1, \blacksquare] = x[i+1, j, \blacktriangle] = x[i, j+1, \blacksquare] = 0 \end{aligned}$$

Với mọi  $(i, j)$  mà trong bảng trạng thái xuất phát, ô vuông  $(i, j)$  là ô vuông màu đen.

# Thiết lập ràng buộc cho mô hình

Ràng buộc đối với các ô vuông màu đen

# Thiết lập ràng buộc cho mô hình

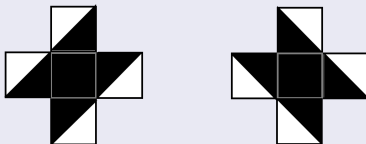
## Ràng buộc đối với các ô vuông màu đen

**Constraint 2:** Xung quanh các ô vuông màu đen có chứa số  $k$  theo hướng dọc và ngang không cần có đúng  $k$  ô vuông có chứa hình tam giác.

# Thiết lập ràng buộc cho mô hình

## Ràng buộc đối với các ô vuông màu đen

**Constraint 2:** Xung quanh các ô vuông màu đen có chứa số  $k$  theo hướng dọc và ngang không cần có đúng  $k$  ô vuông có chứa hình tam giác.



$$\begin{aligned} & x[i-1, j, \blacktriangleleft] + x[i, j-1, \blacktriangleleft] + x[i+1, j, \blacktriangleright] + x[i, j+1, \blacktriangleright] \\ & + x[i-1, j, \blacktriangleright] + x[i, j-1, \blacktriangleleft] + x[i+1, j, \blacktriangleleft] + x[i, j+1, \blacktriangleright] = k \end{aligned}$$

Với mọi  $(i, j)$  mà trong bảng trạng thái xuất phát, ô vuông  $(i, j)$  là ô vuông màu đen và chứa số  $k$ .

# Thiết lập ràng buộc cho mô hình

Ràng buộc đối với các ô vuông có chứa hình tam giác

# Thiết lập ràng buộc cho mô hình

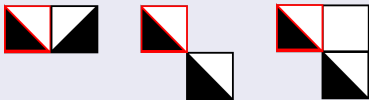
Ràng buộc đối với các ô vuông có chứa hình tam giác

**Constraint 3:** Với mỗi ô vuông chứa các hình tam giác được điền, các ô vuông xung quanh cần phải được điền để tạo thành một góc  $90^\circ$  hoặc  $180^\circ$ .

# Thiết lập ràng buộc cho mô hình

## Ràng buộc đối với các ô vuông có chứa hình tam giác

**Constraint 3:** Với mỗi ô vuông chứa các hình tam giác được điền, các ô vuông xung quanh cần phải được điền để tạo thành một góc  $90^\circ$  hoặc  $180^\circ$ . **Ví dụ:** Giả sử ô vuông  $(i, j)$  được điền  $\blacksquare$ . Khi đó các ô vuông xung quanh cần phải được điền như sau để tạo thành một góc một góc  $90^\circ$  hoặc  $180^\circ$ .



$$\begin{aligned}x[i, j, \blacksquare] &\leq x[i, j + 1, \blacksquare] + x[i + 1, j + 1, \blacksquare] \\x[i, j, \blacksquare] + x[i + 1, j + 1, \blacksquare] &\leq x[i, j + 1, \square] + 1\end{aligned}$$

# Thiết lập ràng buộc cho mô hình

Ràng buộc đối với các ô vuông có chứa hình tam giác



# Thiết lập ràng buộc cho mô hình

Ràng buộc đối với các ô vuông có chứa hình tam giác

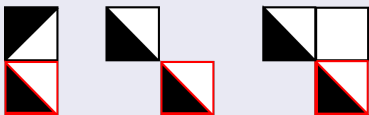
**Constraint 3:** Với mỗi ô vuông chứa các hình tam giác được điền, các ô vuông xung quanh cần phải được điền để tạo thành một góc  $90^\circ$  hoặc  $180^\circ$ .

# Thiết lập ràng buộc cho mô hình

## Ràng buộc đối với các ô vuông có chứa hình tam giác

**Constraint 3:** Với mỗi ô vuông chứa các hình tam giác được điền, các ô vuông xung quanh cần phải được điền để tạo thành một góc  $90^\circ$  hoặc  $180^\circ$ .

**Ví dụ:** Giả sử ô vuông  $(i, j)$  được điền  $\blacksquare$ . Khi đó các ô vuông xung quanh cần phải được điền như sau để tạo thành một góc một góc  $90^\circ$  hoặc  $180^\circ$ .



$$\begin{aligned}x[i, j, \blacksquare] &\leq x[i - 1, j, \blacksquare] + x[i - 1, j - 1, \blacksquare] \\x[i, j, \blacksquare] + x[i - 1, j - 1, \blacksquare] &\leq x[i - 1, j, \square] + 1\end{aligned}$$

# Thiết lập ràng buộc cho mô hình

Ràng buộc để cắt các góc lõm

# Thiết lập ràng buộc cho mô hình

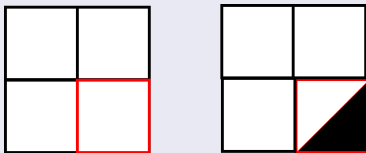
## Ràng buộc để cắt các góc lõm

**Constraint 4:** Các góc lõm được tạo thành từ 3 ô vuông màu trắng đôi một kề nhau (theo hướng dọc và ngang), khi đó vị trí còn lại phải được điền để tránh tạo góc lõm.

# Thiết lập ràng buộc cho mô hình

## Ràng buộc để cắt các góc lõm

**Constraint 4:** Các góc lõm được tạo thành từ 3 ô vuông màu trắng đôi một kề nhau (theo hướng dọc và ngang), khi đó vị trí còn lại phải được điền để tránh tạo góc lõm.



$$x[i, j, \square] + x[i + 1, j, \square] + x[i, j + 1, \square] \leq x[i + 1, j + 1, \square] + x[i + 1, j + 1, \blacksquare] + 2$$

# Thiết lập ràng buộc cho mô hình

Ràng buộc để cắt các góc lồi

# Thiết lập ràng buộc cho mô hình

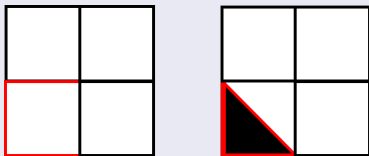
## Ràng buộc để cắt các góc lõm

**Constraint 4:** Các góc lõm được tạo thành từ 3 ô vuông màu trắng đôi một kề nhau (theo hướng dọc và ngang), khi đó vị trí còn lại phải được điền để tránh tạo góc lõm.

# Thiết lập ràng buộc cho mô hình

## Ràng buộc để cắt các góc lõm

**Constraint 4:** Các góc lõm được tạo thành từ 3 ô vuông màu trắng đôi một kề nhau (theo hướng dọc và ngang), khi đó vị trí còn lại phải được điền để tránh tạo góc lõm.



$$x[i, j, \square] + x[i + 1, j, \square] + x[i, j - 1, \square] \leq x[i + 1, j - 1, \square] + x[i + 1, j - 1, \blacksquare] + 2$$



# Thiết lập ràng buộc cho mô hình

Ràng buộc để cắt các góc lõm

# Thiết lập ràng buộc cho mô hình

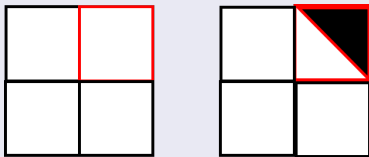
## Ràng buộc để cắt các góc lõm

**Constraint 4:** Các góc lõm được tạo thành từ 3 ô vuông màu trắng đôi một kề nhau (theo hướng dọc và ngang), khi đó vị trí còn lại phải được điền để tránh tạo góc lõm.

# Thiết lập ràng buộc cho mô hình

## Ràng buộc để cắt các góc lõm

**Constraint 4:** Các góc lõm được tạo thành từ 3 ô vuông màu trắng đôi một kề nhau (theo hướng dọc và ngang), khi đó vị trí còn lại phải được điền để tránh tạo góc lõm.



$$x[i, j, \square] + x[i-1, j, \square] + x[i, j+1, \square] \leq x[i-1, j+1, \square] + x[i-1, j+1, \blacksquare] + 2$$

# Thiết lập ràng buộc cho mô hình

Ràng buộc để cắt các góc lõm

# Thiết lập ràng buộc cho mô hình

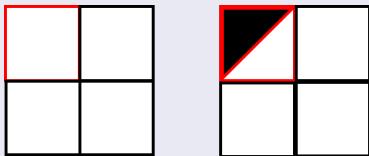
## Ràng buộc để cắt các góc lõm

**Constraint 4:** Các góc lõm được tạo thành từ 3 ô vuông màu trắng đôi một kề nhau (theo hướng dọc và ngang), khi đó vị trí còn lại phải được điền để tránh tạo góc lõm.

# Thiết lập ràng buộc cho mô hình

## Ràng buộc để cắt các góc lõm

**Constraint 4:** Các góc lõm được tạo thành từ 3 ô vuông màu trắng đôi một kề nhau (theo hướng dọc và ngang), khi đó vị trí còn lại phải được điền để tránh tạo góc lõm.



$$x[i, j, \square] + x[i - 1, j, \square] + x[i, j - 1, \square] \leq x[i - 1, j - 1, \square] + x[i - 1, j - 1, \blacksquare] + 2$$

# Thiết lập ràng buộc cho mô hình

Ràng buộc để cắt các góc lõm

# Thiết lập ràng buộc cho mô hình

## Ràng buộc để cắt các góc lõm

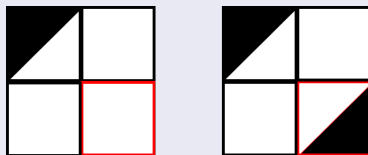
**Constraint 4:** Các góc lõm được tạo thành từ 1 ô vuông chứa tam giác và 2 ô vuông màu trắng.



# Thiết lập ràng buộc cho mô hình

## Ràng buộc để cắt các góc lõm

**Constraint 4:** Các góc lõm được tạo thành từ 1 ô vuông chứa tam giác và 2 ô vuông màu trắng.



$$x[i, j, \blacksquare] + x[i + 1, j, \square] + x[i, j + 1, \square] \leq x[i + 1, j + 1, \square] + x[i + 1, j + 1, \blacksquare] + 2$$

# Thiết lập hàm mục tiêu

Thiết lập hàm mục tiêu

# Thiết lập hàm mục tiêu

## Thiết lập hàm mục tiêu

Mô hình với các điều kiện tuyến tính đã được xây dựng không có hàm mục tiêu.

# Cài đặt mô hình

## Cài đặt mô hình

Cài đặt mô hình bằng thư viện Gurobipy.

## Tổng kết

- Giới thiệu về trò chơi xếp hình Shakashaka.
- Xây dựng mô hình tối ưu tuyến tính để giải trò chơi Shakashaka.
- Sử dụng thư viện Gurobipy để cài đặt và chạy mô hình đã xây dựng. Mô hình có kết quả khá khả quan khi giải đúng một số test được đưa ra.

- [1] Erik D. Demaine, Yoshio Okamoto, Ryuhei Uehara, Yushi Uno, *Computational complexity and an integer programming model of Shakashaka*, 2013.

**THANK YOU FOR YOUR ATTENTION!**