A Efficient Approach
for Solving Nonogram Puzzles

Le Quang Dong

University of Engineering and Technology,

Vietnam National University

Abstract. Nonograms, hay còn được biết điến với các tên Japanese Puzzle, Picture-by-Number, là một câu đố khá nổi tiếng trên các tờ báo cũng như cộng đồng giải đố. Mục tiêu của trò chơi là điền vào một lưới ô vuông với các pixels đen và trắng với những gợi ý được cho trước với từng hàng và cột chỉ độ dài các đoạn ô được tô màu đen liên tục trên hàng hoặc cột đó. Dù các Nonogram trong các sách báo giải đố thường có thể giải bằng tay, bài toán giải một Nonogram với gợi ý cho trước là một bài toán NP-hard. Trong đây, tôi sẽ trình bày một số cách suy luận để chương trình có thể giải được một phần của bài toán cũng như một số cách suy nghĩ logic để có thể dựa trên lưới ô vuông hiện có để giải bài toán. Những cách tiếp cận trên dựa trên quy hoạch động và giải thuật heuristic. Kết quả thử nghiệm của tôi cho thấy cách tiếp cận này có thể giải được một số Nonograms không thể giải chỉ bằng lý giải logic đơn lẻ từng hàng hoặc cột. Hầu hết các tính toán trong đây được xử lý trong thời gian đa thức.

1 Introduction

Nonograms, hay còn được biết đến với cái tên Paint-by-Number, là một câu đố nối tiếng xuất phát từ Nhật Bản, với mục tiêu vẽ một bức ảnh hình chữ nhật bằng các pixel màu đen hoặc trắng dựa trên các hằng số gợi ý của các hàng và cột. Thường thấy các hình ảnh được vẽ qua trò chơi này là đen trắng, nhưng các Nonogram với nhiều hơn 2 màu có tồn tại. Figure 1 cho ví dụ về một Nonogram, một Nonogram được giải 1 phần và một Nonogram được giải hoàn toàn.

Câu đố này có dạng một lưới hình chữ nhật, được chia thành các đơn vị ô nhỏ hơn. Với mỗi hàng và cột, người giải sẽ có 1 chuỗi các số gợi ý, gợi ý này cho biết độ dài các đoạn ô đen liên tiếp có trên hàng hoặc cột đó. Ví dụ gợi ý trên một hàng là “2 3" thì trên hàng đó, từ trái qua phải sẽ có 0 hoặc nhiều hơn các ô trắng, tiếp nối bởi 2 ô đen, tiếp nối bởi 1 hoặc nhiều hơn các ô trắng, tiếp nối bởi 3 ô đen, cuối cùng là 0 hoặc nhiều hơn các ô trắng. Nếu như hàng hoặc cột có gợi ý rỗng , toàn bộ hàn g hoặc cột đó được tô toàn bộ là bằng ô trắng. Mục tiêu của câu đố là điền toàn bộ các ô của câu đố vời 1 trong 2 màu đen hoặc trắng thỏa mãn với các gợi ý trên các hàng và cột. Với những gợi ý và các cách giải trên từng hàng đơn lẻ, câu đố sẽ thường chỉ giải được một phần, nghiên cứu này sẽ đưa ra một cách giải không quá tốt để có thể hoàn thiện việc giải câu đố.

// add figure here

// Nonogram can be related to Discrete Tomography

Nonogram có nhiều các cấp bậc độ khó khác nhau, những câu đố thường thấy trên sách báo giải đố thường là những câu đố đơn giản có thể giải bằng những logic thông thường áp dụng trên từng hàng, cột riêng lẻ. Với những câu đố lớn hơn và ngẫu nhiên thường rất khó để giải và có thể có nhiều hơn 1 lời giải cho 1 câu đố. Các thuật toán giải Nonogram thường có tác dụng giải câu đố và thường không tìm được tất cả các lời giải cho 1 câu đố.

Trong nghiên cứu này, tôi sẽ trình bày một số logic đơn giản cũng như một thuật toán “đoán" ô có hiệu quả với những câu đố ở mức độ vừa phải. Nghiên cứu này được chia thành các phần như sau: Phần 2 giới thiệu về Nonogram một cách chi tiết hơn, phần 3 là các phương pháp giải cho Nonogram,

2 Notation and Concepts

Ta định nghĩa cách ký hiệu cho một hàng hoặc cột qua tập giả trị: $T = \{0, 1, -\}$. Ký hiệu "0" và "1" thể hiện ô được tô màu đen và trắng trong câu đố, trong khi ký hiệu "−" thể hiện một ô chưa được quyết định màu.

Kích thước của câu đố được biểu diễn bằng 1 bảng có kích thước $H * W$, với H là số hàng của bảng và W là số cột của bảng.

Nonogram được mô tả qua những gợi ý hàng và gợi ý cột. Một gợi ý d có độ dài $k > 0$ là một dãy số nguyên có thứ tự $(\mathrm{d}_1, \mathrm{d}_2, \dots, \mathrm{d}_\mathrm{k})$ với $d_i = C_i\{a_i, b_i\}$, với $C_i \in \{0, 1\}$ và $a_i, b_i \in \{0, 1, 2, \dots\}, a_i \leq b_i$, $(i = 1, 2, \dots, \mathrm{k})$; d_i được gọi là 1 đoạn gợi ý và a_i, b_i là điểm đầu và điểm cuối của đoạn thẳng màu c_i trên d. d_i cũng có thể được biểu diễn theo cách $d_i = C_i\{a, \infty\}$, chỉ gợi ý d có độ dài ít nhất là a

Một xâu s gồm các ký tự trong T được dùng để biểu diễn gợi ý d có dạng:

$$(0\{0, \infty\}, 1\{a_1, b_1\}, 0\{1, \infty\}, 1\{a_2, b_2\}, 0\{1, \infty\}, \dots, 1\{a_3, b_3\}, 0\{0, \infty\})$$

Gợi ý của Nonogram có thể được mô tả một cách dễ hiểu hơn là cho một chuỗi $d = \{d_1, d_2, \dots, d_k\}$, chỉ các đoạn ô đen được tô trên một hàng hoặc một cột của bảng Nonogram. Ta quy ước $r(u, v)$ chỉ gợi ý thứ v của hàng u, $c(m, n)$ chỉ gợi ý thứ m của cột n.

Trong bài này, cách quy ước thứ 2 sẽ được sử dụng nhiều hơn.

Ta cũng cần lưu lại những vị trí mà với một gợi ý $r(u, v)$ hoặc $c(m, n)$ có thể bắt đầu và kết thúc, được biểu diễn bởi $firstRow(u, v), lastRow(u, v), firstCol(m, n), lastCol(m, n)$ với khởi tạo sau:

$$firstRow(u, v) = \begin{cases} 1 & v = 1 \\ v + \sum_{i=1}^{v-1} r(u, i) & otherwise \end{cases}$$

$$lastRow(u, v) = W - \left[r(u, v) - 1 + \sum_{i=v+1}^{v\_\max(u)} (r(u, i) + 1) \right]$$

$$firstCol(m, n) = \begin{cases} 1 & m = 1 \\ m + \sum_{j=1}^{m-1} c(j, n) & otherwise \end{cases}$$

$$lastCol(m, n) = H - \left[c(m, n) - 1 + \sum_{j=m+1}^{m\_\max(n)} (c(j, n) + 1) \right]$$

Ta có 2 tập vector phụ α và β để lưu trữ các giá trị đã xác định cho một gợi ý trên bảng, $\alpha(u, v) = \{a_1, a_2, \dots, a_W\}$ cho thông tin của gợi ý thứ v của hàng u, $\beta(m, n) = \{b_1, b_2, \dots, b_H\}$ cho thông tin của gợi ý thứ m của cột n và được khởi tạo như sau:

$$\alpha(u, v) = (x_1, x_2, \dots, x_W)$$

$$x_i = \begin{cases} 1 & firstRow(u, v) \leq i \leq lastRow(u, v) + r(u, v) - 1 \\ 0 & otherwise \end{cases}$$

$$\beta(m,n) = \{b_1, b_2, \dots, b_H\}$$
$$y_i = \begin{cases} 1 & firstCol(m,n) \le i \le lastCol(m,n) + c(m,n) - 1 \\ 0 & otherwise \end{cases}$$

Với mỗi hàng và cột, ta còn có $\alpha(u,0)$ và $\beta(0,n)$ biểu diễn kết quả của hàng hoặc cột đó, mỗi giá trị trong được thể hiện qua tập giá trị T. Điều đó khiến chúng phải thỏa mãn các điều kiện sau:

$$\forall u, v, \quad \alpha(u,v) \rightarrow \sum_{i=1}^{W} x_i = r(u,v)$$
$$\forall m, n, \quad \beta(m,n) \rightarrow \sum_{j=1}^{H} y_j = c(m,n)$$
$$\forall u, \quad \alpha(u,0) \rightarrow \sum_{i=1}^{v\_max} \alpha(u,i)$$
$$\forall v, \quad \beta(0,n) \rightarrow \sum_{j=1}^{m\_max} \beta(j,n)$$

3 Partially solving solution

3.1. Giải trên 1 hàng

3.1.1 Simple Boxes / Simple Spaces

Simple Boxes được dùng để quyết định một ô có được điền "1" hay không dựa vào số khả năng có thể tô màu ô đó. Hình 2 và 3 thể hiện cách thực hiện của Simple Boxes và Simple Spaces:

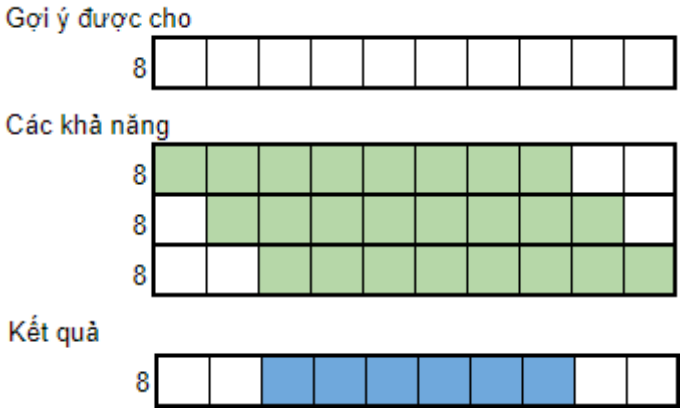


Figure 2. Simple Boxes

Phương thức Simple Boxes được xác định như sau:

- Với hàng:

  Cho $r(u,v)$, $\alpha(u,v) = \{x_1, x_2, \dots, x_W\}$, $firstRow(u,v)$ và $lastRow(u,v)$:

$$\forall u, v, firstRow(u,v) + r(u,v) - 1 \ge lastRow(u,v) - r(u,v) + 1$$
$$\rightarrow \begin{cases} \forall x_i \in \alpha(u,0), lastRow(u,v) - r(u,v) + 1 \le i \le firstRow(u,v) + r(u,v) - 1, & x_i = 1 \\ \forall i, y_u \in \beta(0,i), lastRow(u,v) - r(u,v) + 1 \le i \le firstRow(u,v) + r(u,v) - 1, & y_u = 1 \end{cases}$$

- Với cột:

  Cho $c(m,n)$, $\beta(m,n) = \{y_1, y_2, \dots, y_H\}$, $firstCol(m,n)$ và $lastCol(m,n)$:

$$\forall m,n, firstCol(m,n) + c(m,n) - 1 \geq lastCol(m,n) - c(m,n) + 1$$

$$\rightarrow \begin{cases} \forall y_i \in \beta(0,n), lastCol(m,n) - c(m,n) + 1 \leq i \leq firstCol(m,n) + c(m,n) - 1, & y_i = 1 \\ \forall i, y_n \in \alpha(i,0), lastCol(m,n) - c(m,n) + 1 \leq i \leq firstCol(m,n) + c(m,n) - 1, & x_n = 1 \end{cases}$$

Tương tự như trên, Simple Spaces được dùng để quyết định 1 ô có được điền "0" hay không.

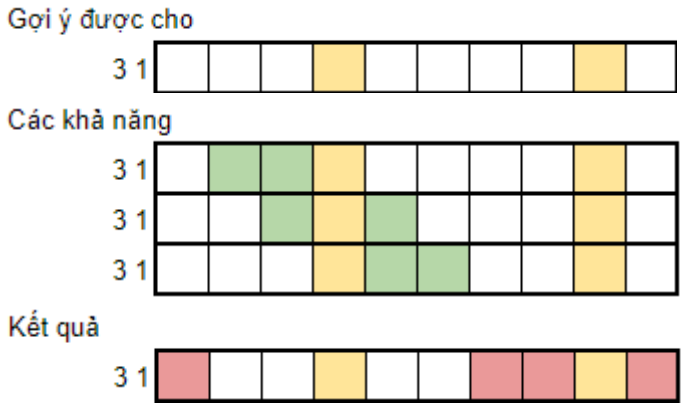


Figure 3. Simple Spaces

Với phương thức Simple Space, ta định nghĩa $z(u,v,i)$ là phần tử thứ i trong $r(u,v)$, và $w(m,n,i)$ là phần từ thứ i trong $c(m,n)$, ta có:

- Với hàng:

$$\forall u,i \sum_{v=1}^{v\_max} z(u,v,i) = 0 \rightarrow z(u,0,i) = 0 \wedge w(j,i,u) = 0, j = 0,1,\dots,m\_max$$

$$\forall u,i,\exists!\, v, z(u,v,i) = 1 \wedge z(u,0,i) = 1$$

$$\rightarrow z(u,v,j) = 0, j = 1,2,\dots,i - r(u,v) \wedge z(u,v,j) = 0, j = i + r(u,v),\dots,W$$

- Với cột:

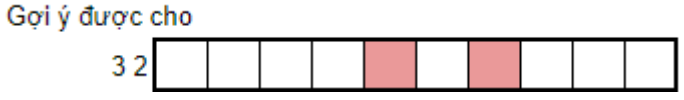
$$\forall n,i \sum_{m=1}^{m\_max} w(m,n,i) = 0 \rightarrow w(0,n,i) = 0 \wedge z(i,j,n) = 0, j = 0,1,\dots,v\_max$$

$$\forall n,i,\exists!\, m, w(m,n,i) = 1 \wedge w(0,n,i) = 1$$

$$\rightarrow w(m,n,j) = 0, j = 1,2,\dots,i - c(m,n) \wedge w(m,n,j) = 0, j = i + c(m,n),\dots,H$$

3.1.2 Forcing

Trong cách xử lý này, những chỗ trống trong một hàng chưa được hoàn thiện sẽ được tính toán xem có thể ép một đoạn màu đen "1" vào một đoạn ô chưa được điền nào đó, cũng như một đoạn ô chưa được điền nào đó có thể quá nhỏ để có thể cho vừa một đoạn màu trắng "0". Hình 4 diễn tả cách hoạt động của phương pháp này.



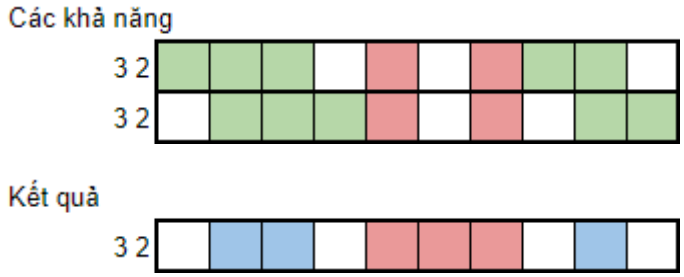


Figure 4. Forcing

Phương thức Forcing được định nghĩa như sau:

- Cho hàng:

$$\forall u,v,i\ \ z(u,v,i)=0 \wedge i \le r(u,v) \rightarrow z(u,v,j)=0, j=1,2,\dots,i-1$$
$$\forall u,v,\exists i,j, z(u,v,i)=0 \wedge z(u,v,j)=0 \wedge j-i \le r(u,v)$$
$$\rightarrow z(u,v,p)=0, \qquad p=i+1,i+2,\dots,j-1$$
$$\forall u,v,i\ \ z(u,v,i)=0 \wedge i+r(u,v)>W \rightarrow z(u,v,j)=0,\ \ j=i+1,i+2,\dots,W$$

- Cho cột:

$$\forall m,n,i\ \ w(m,n,i)=0 \wedge i \le c(m,n) \rightarrow w(m,n,j)=0, j=1,2,\dots,i-1$$
$$\forall m,n,\exists i,j, w(m,n,i)=0 \wedge w(m,n,j)=0 \wedge j-i \le c(m,n)$$
$$\rightarrow w(m,n,p)=0, \qquad p=i+1,i+2,\dots,j-1$$
$$\forall m,n,i\ \ w(m,n,i)=0 \wedge i+c(m,n)>H \rightarrow w(m,n,j)=0,\ \ j=i+1,i+2,\dots,H$$

3.2 Giải trên cả bảng

3.2.1 Guessing

Khi các cách giải logic không thể hoàn thiện câu đố, cách đoán này sẽ được áp dụng. Dựa vào công thức đơn giản tính mức độ ảnh hưởng của các ô chưa được điền màu để xếp hạng các ô trong vòng quay roulette, sau đó tiến hành bánh xe xổ số và chọn ngẫu nhiên một ô chưa được hoàn thiện trong bảng để thực hiện đoán màu của ô này.

Hình 5 mô tả cách thực hiện của phương pháp này:

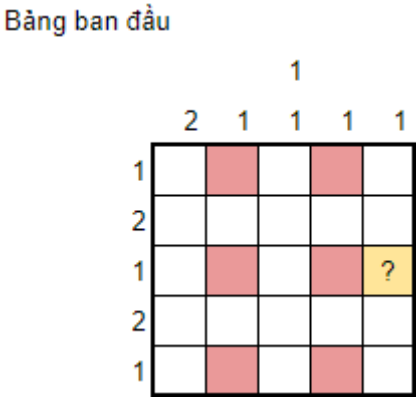


Figure 5a. “?” pixel denote the pixel is being guessed

Thuật toán đệ quy sẽ được sử dụng trong việc đoán màu của ô để tìm xem việc đặt màu cho ô đó có gây ra xung đột trên bảng hay không, từ đó đưa ra kết quả của ô được đoán.

Với bài toán sửa 1 hàng này, phương pháp được đặt ra là xây dựng một hàm đệ quy $Fix(i,j)$ với mọi $i \geq 0$ & $j \geq 0$ như sau. Gọi xâu $S_i = s_1 s_2 \dots s_i$ là xâu biểu diễn i ký tự đầu tiên trong một hàng, $D_j = (d_1, d_2, \dots, d_j)$ là tập giá trị các đoạn gợi ý từ 1 đến j của hàng đó. $Fix(i,j)$ sẽ kiểm tra xem là xâu S_i có thể được sửa với các gợi ý D_j còn lại không. Gọi $Fix0(i,j)$ kiểm tra việc xâu S_i có thể sửa được khi đặt $s_i =$ ”0” với gợi ý D_j, $Fix1(i,j)$ kiểm tra tương tự nhưng với $s_i =$ "1". Cách tính cho các phương thức trên được mô ta như sau

$$Fix(i,j) = \begin{cases} true, & if\ i = 0, j = 0 \\ false, & if\ i = 0, j > 0 \\ Fix0(i,j) \vee Fix1(i,j), & otherwise \end{cases}$$

$$Fix0(i,j) = \begin{cases} Fix(i-1,j), & if\ s_i \in \{0,-\} \\ false, & otherwise \end{cases}$$

$$Fix1(i,j) = \begin{cases} Fix(i-d_j-1, j-1), & if\ j > 0, i-d_j-1 \geq 0, s_{i-d_j}, \dots, s_i\ not\ conflict \\ false, & otherwise \end{cases}$$

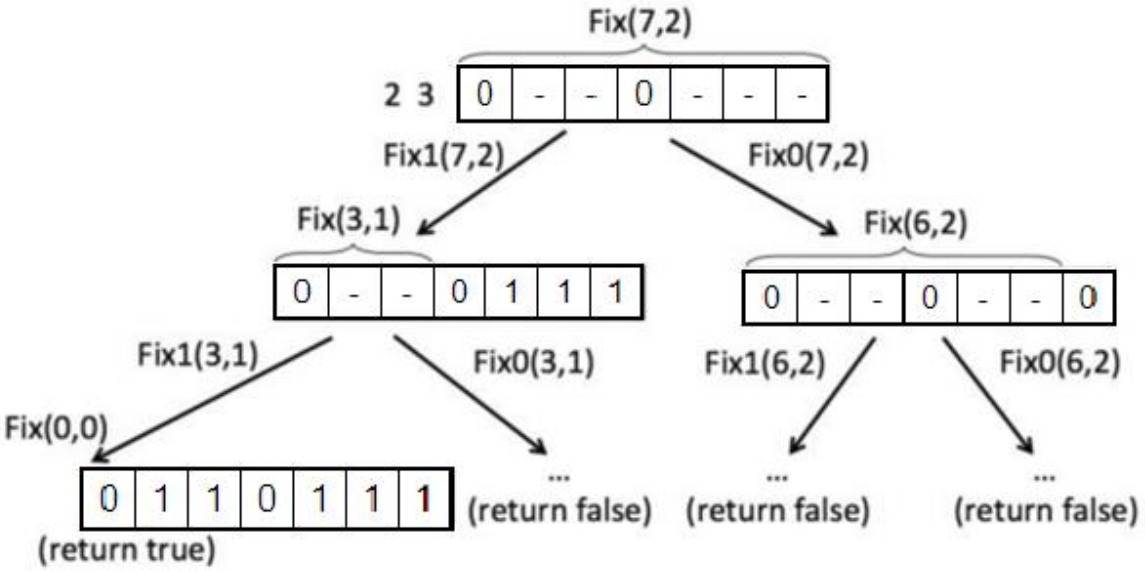


Figure 5b. $Fix(7,2)$ với xâu s $=$ “$0--0---$“

Với việc kiểm tra 1 hàng có thể được giải hay không như trên, ta có phương thức $Paint(i,j)$ để tô màu cho các ô được kiểm tra trên hàng đó:

$$Paint(i,j) = \begin{cases} \emptyset, & i = 0 \\ Paint'(i,j), & otherwise \end{cases}$$

$$Paint'(i,j) = \begin{cases} Paint0(i,j), & if\ Fix0(i,j) AND\ NOT\ Fix1(i,j) \\ Paint1(i,j), & if\ NOT\ Fix0(i,j) AND\ Fix1(i,j) \\ Merge(Paint0(i,j), Paint1(i,j), & otherwise \end{cases}$$

$$Paint0(i,j) = Paint(i-1,j)0$$
$$Paint1(i,j) = Paint\left(i-d_j-1, j-1\right)1^{d_j}0$$
$$Merge(S_i,T_i) = q_1 q_2 \dots q_i$$
$$q_j = \begin{cases} 0, & if\ s_j = t_j = 0 \\ 1, & if\ s_j = t_j = 1 \\ -, & otherwise \end{cases}$$

4 Kết quả thử nghiệm

Với các thuật toán kể trên, tuy không được tối ưu hoàn toàn, chương trình đã có thể giải một số các câu đố không chỉ dùng những logic trên một hàng, cột đơn lẻ mà cần có thêm những suy luận giải bài trên nhiều dòng. Tính hiệu quả của chương trình này được kiểm định qua các câu đố được tạo bởi webpbn – một nguồn tổng hợp các chương trình giải Nonogram và các test Nonogram với nhiều độ khó và độ phức tạp trong cách giải khác nhau. Qua chạy thử, chương trình cho kết quả ở hình 6 như sau:

No	Test	Size	Avg Time to Solve
1	Dancer	5 x 10	0
2	Cat	20 x 20	0.02s
3	Skid	14 x 25	0.03s
4	Bucks	27 x 23	0.86s
5	Edge	10 x 11	x
6	Knot	34 x 34	5.73s
7	Swing	45 x 45	0.78s
8	Mum	34 x 40	x

0: Solve time less than 0.01s

x: Program cannot finish the puzzle

Figure 6. Result