Complexity of the Hidden Polygon Puzzle

Marzio De Biasi
marziodebiasi [at] gmail [dot] com

August 2013
Version 0.01: proof sketch

Abstract

Given a set P of m integer points on a $n \times n$ square grid and an integer $k \leq m$, the problem of finding a rectilinear simple polygon with k or more vertices $(v_1, v_2, ..., v_t),\ v_i \in P, t \geq k$ (*Hidden Polygon Puzzle* problem) is NP–complete.

1 The Hidden Polygon Puzzle

1.1 Definition

This short paper is inspired by the question *"Complexity of hidden polygon puzzle on square grids?"* by Mohammad Al-Turkistany appeared on the question and answer site cstheory.stackexchange.com.

Definition 1.1. The *Hidden Polygon Puzzle* (for brevity HPP) decision problem is:

Input: a set P of m integer points on a $n \times n$ square grid and an integer $k \leq m$;

Question: does exist a simple rectilinear polygon with k or more vertices $(v_1, v_2, ..., v_t),\ v_i \in P, t \geq k$?

Figure 1 shows an example of a HPP puzzle.

The problem is a slight variant of the puzzle game *Hiroimono* in which the player must collect all the stones on a grid board, he can move in one of the four directions, but he can change it only when he picks up a stone, and he cannot make 180° turns. Hiroimono has been shown to be NP–complete [1].

2 Complexity

The problem is in NP because a solution $(v_1, v_2, ..., v_j)$ can easily be verified in polynomial time; it is enough to check that: 1) $j = k$, 2) for all $i, v_i \in P$, 3) v_i are all distinct, 4) vertices $v_i, v_{(i+1) \bmod k}, v_{(i+2) \bmod k}$ form a 90° corner, 5) check that for every pair of edges e_i, e_j the two edges don't intersect.

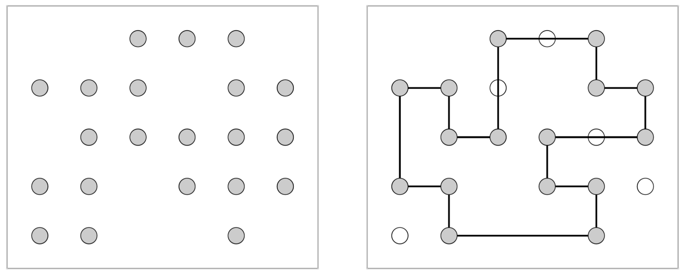

Figure 1: Given the 21 points on the right, can we find a simple rectilinear polygon with at least 16 vertices? A possible solution is shown on the right.

We prove that the problem is NP–hard using a reduction from the Hamiltonian cycle problem on grid graphs with degree ≤ 3 which is NP–complete [2].

Given a $m \times m$ grid graph with degree ≤ 3, $G = (V, E)$ with n nodes, suppose that each node u_i is at coordinate (x_i, y_i), $1 \leq x_i, y_i < n - 1$ in the grid (for simplicity we assume that there is an empty border of width 1). If the graph has a node of degree 1 then build a false dummy HPP instance because it cannot have an Hamiltonian cycle. So we assume that every node has degree $geq2$.

We construct a HPP grid replacing every node n with a *node gadget*.

2.1 Node gadget

The node gadget corresponding to node u_i at cordinates (x_i, y_i) is a $d \times d$ grid placed at coordinates $(d * x_i, d * y_i)$ in the HPP grid with $d = 2 * n * g + 1$ (we will see later which value to use for g); for brevity we set $z = n * g + 1$.

There are three types of node gadgets:

- *T gadget* used to replace nodes of degree three;
- *Turn gadget* used to replace nodes of degree 2 that have two orthogonal incident edges;
- *Line gadget* gadget used to replace nodes of degree 2 that have two align incident edges.

Every node gadget is ideally divided in 4 $ng \times ng$ quadrants (Q_1, Q_2, Q_3, Q_4) with a central cross of width 1. T gadgets have a central node at coordinates (ng, ng); all have two sets ZP_1, ZP_2 of g *Zigzag points* placed in two different quadrants and two *Jump points* J_1, J_2; Turn gadgets have an *Extra point* E on another quadrant.

The Zigzag points for a gadget representing node u_i (at coordinates (x_i, y_i) in grid G) are ideally placed in two $g \times g$ *zigzag boxes* at relative coordinates $(i * g, i * g)$ for quadrant Q_1 (resp. $(i * g + ng + 1, i * g), (i * g, i * g + ng + 1), (i * g + ng + 1, i * g + ng + 1)$ for Q_2, Q_3, Q_4). So the absolute position in the HPP grid for the zigzag box in quadrant Q_1 is $(d*x_i+i*g, d*y_i+i*g)$ (add $ng+1$ for the other quadrants). In this way every pair of zigzag points of two distinct gadgets have distinct horizontal and vertical coordinates. In a zigzag box, the g Zigzag points are aligned horizontally or vertically depending on the gadget, except the first point of $ZP1$ and the last point of $ZP2$ which are shifted by one. In every gadget the g Zigzag points of ZP_1 are aligned (horizontally or vertically) with the g Zigzag points of ZP_2.

Jump points are placed in the central cross and are aligned with one zigzag point: jump point J_1 is aligned with the first point of ZP_1, Jump point J_2 is aligned with the last unaligned point of ZP_2, except in Turn gadget in which J_2 is aligned with the Extra point E and E is aligned with the last point of ZP_2.

The ten possible node gadgets and the positions of the Zigzag, Jump and Extra points are shown in Figure 2a–i.

We describe how a polyline can *traverse* the T gadget shown in Figure 2a. By construction, the Zigzag points of a gadget don't share any coordinate with the ZigZag points of the other gadgets, so the only way that a polyline can enter and exit the gadget is through Jump points J_1 or J_2, or the central point C. It cannot self–intersect so the valid combinations are:

t_1: $out \rightarrow J_1 \rightarrow ZP_1 \leftrightarrow ZP_2 \rightarrow J_2 \rightarrow out$
t_2: $out \rightarrow J_1 \rightarrow ZP_1 \leftrightarrow ZP_2 \rightarrow J_2 \rightarrow C \rightarrow out$
t_3: $out \rightarrow J_1 \rightarrow C \rightarrow out$
t_4: $out \rightarrow J_2 \rightarrow C \rightarrow out$
t_5: $out \rightarrow J_1 \rightarrow C \rightarrow J_2 \rightarrow out$
t_6: $out \rightarrow C \rightarrow out$ (vertically)

or the corresponding traversals in the reverse direction (Figure 3. Note that in traversals t_5 and t_6 the polyline doesn't make any turn.

So there are two traversal t_1 and t_2 in which the polyline enters and exits the Zigzag boxes (we will see later what this means); but it is important to notice that if the polyline follows t_3, t_4, t_5, it will never be able to enter and exit the Zigzag boxes later in its path because those traversals "throw out" at least on Jump point and both are needed to enter and exit the Zigzag points boxes; if it follows t_6 then it preclude the possibility to move from ZP_1 to ZP_2 (or vice versa) without creating a self-intersecting polyline. So the only way to gain access to the ZP_1, ZP_2 points is to follow the underlying edges of the grid graph.

In the traversals t_1, t_2 the polyline can use the Zigzag points of ZP_1 and ZP_2 to make a series of turns as shown in Figure 4. In particular, if g is odd, it can make $2g$ turns (i.e. *collect* $2g$ vertices). Figure 4 shows a T gadget and a Turn gadget that correspond to two nodes of the original grid graph. The bold polyline shows a valid traversal that enters the Zigzag points of both gadgets. Note that, by construction, the Zigzag points of the two gadgets don't have any horizontal or vertical coordinate in common.

The traversals of the Turn gadgets and Line gadgets are similar, but in this case there is only one valid traversal that can use the Zigzag points (which follows the two edges incident to the corresponding node).

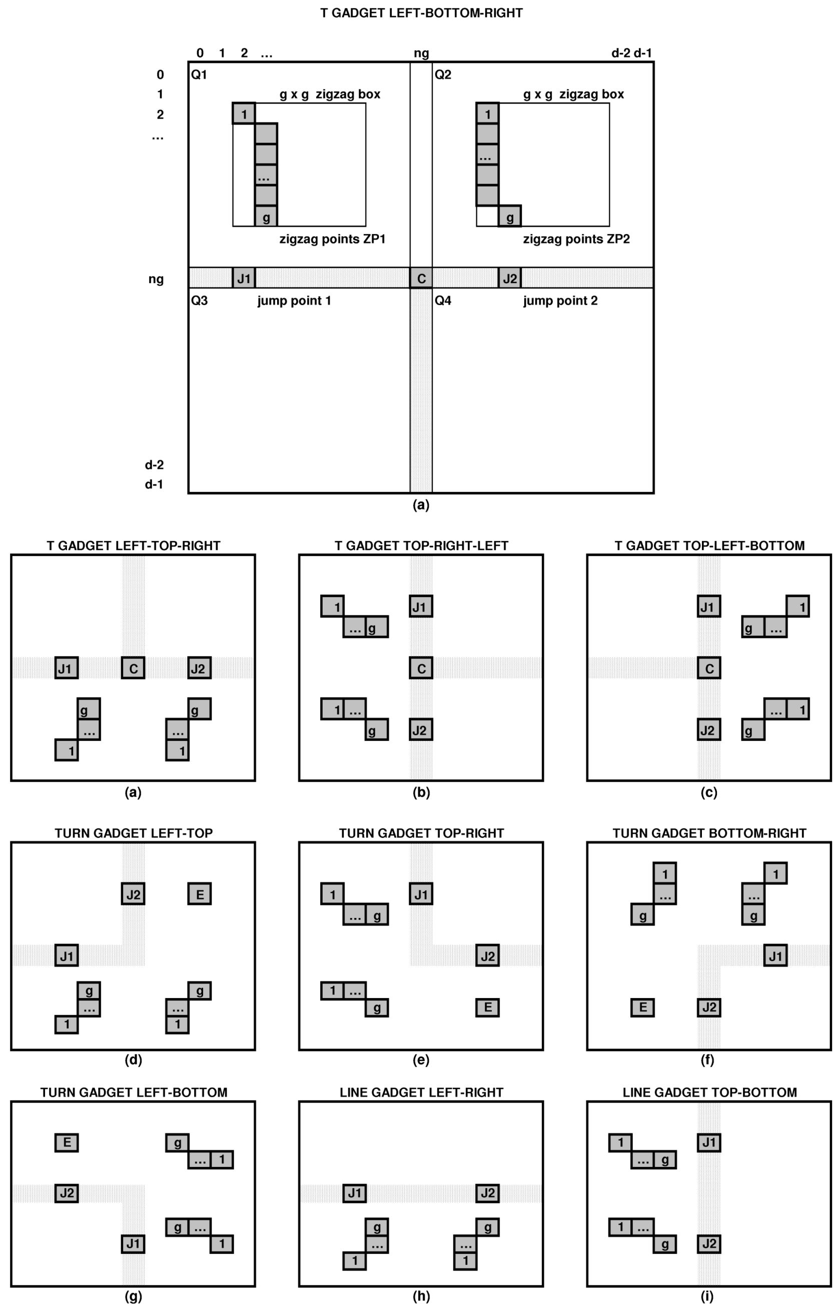

Figure 2: The ten $d \times d$ node gadgets that are used to simulate a valid traversal of the original grid graph with degree ≤ 3.

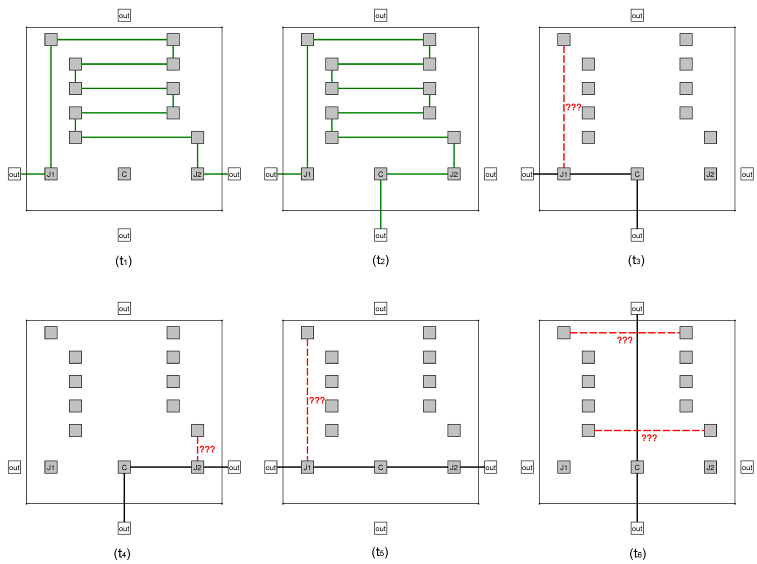

Figure 3: The possible traversals of a T gadget. Only t_1 and t_2 use the Zigzag points; the others are valid, but block the access to the Zigzag points in.

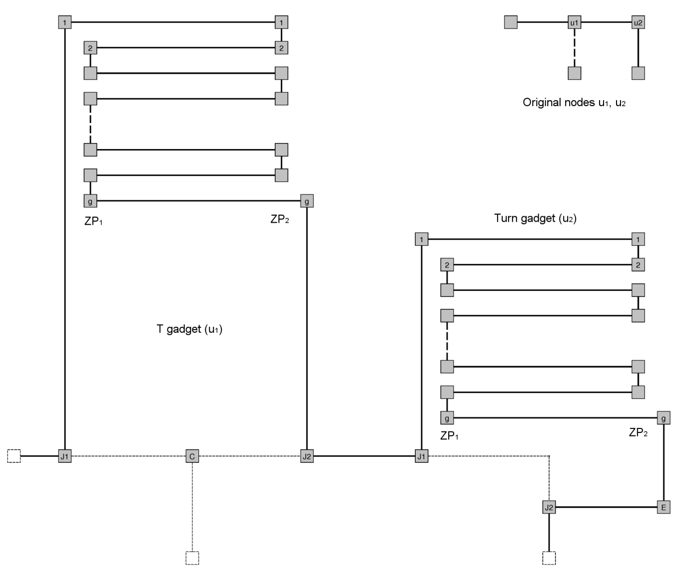

Figure 4: A T gadget and a Turn gadget that correspond to the node u_1, u_2 of the portion of the grid graph shown in the upper right corner. The bold polyline shows a valid traversal that enters the Zigzag points of both gadgets.

So we can summarize the maximum number of vertices that a polyline can collect traversing each gadget and using the Zigzag points:

Gadget	Vertices
T gadget	$2g+2$ (trav. t_1) or $2g+3$ (trav. t_2)
Turn gadget	$2g+3$
Line gadget	$2g+2$

If we choose the value $g = n$, and $k = n * (2g + 2) = 2n^2 + 2n$ then the only way to build a k vertices polygon is to traverse all zigzag point of all node gadgets: if only $n - 1$ gadgets are traversed by the polyline using their Zigzag points and scoring the maximum number of vertices $2g + 3$ and the remaining gadget is traversed scoring 3 vertices (J_1, J_2, C) then the total number of vertices is:

$$(n-1)(2g+3)+3 = 2ng-2g+3n = 2n^2-2n+3n = 2n^2+n < 2n^2+2n = k$$

so the polyline cannot be a valid solution. Furthermore every gadget can be traversed accessing the Zigzag points only once, so if there is a solution to the HPP problem, the sequence of gadgets traversed by the polygonal line corresponds to an Hamiltonian cycle in the original grid graph G. Vice versa, if the original grid graph G has an Hamiltonian cycle, then using the sequence of nodes of the cycle we can build a polygonal line in the corresponding HPP puzzle with a number of vertices greater than or equal to $k = n * (2g + 2)$.

Theorem 2.1. *HPP puzzle is* NP*–complete.*

Proof. The problem is in NP and the reduction from the Hamiltonian cycle problem on grid graph with degree ≤ 3 described above can be done in polynomial time, so it is also NP–hard. □

Figure 5 shows a $n = 10$ nodes grid graph, the corresponding HPP instance with $k = 2n^2 + 2n = 220$ in which Zigzag points are collapsed (every bold box contains n points), and a simple rectilinear polygon with $30 + 20 * 10 = 230 > k$ vertices that is a valid solution and that corresponds to an Hamiltonian cycle on the original grid graph.

3 Conclusion

What about programming a *real* hidden polygon puzzle "app" that can be played online in a browser or downloaded in your favourite smartphone ?!? ...

References

[1] Daniel Andersson. Hiroimono is np-complete. In Pierluigi Crescenzi, Giuseppe Prencipe, and Geppino Pucci, editors, *Fun with Algorithms*, volume 4475 of *Lecture Notes in Computer Science*, pages 30–39. Springer Berlin Heidelberg, 2007.

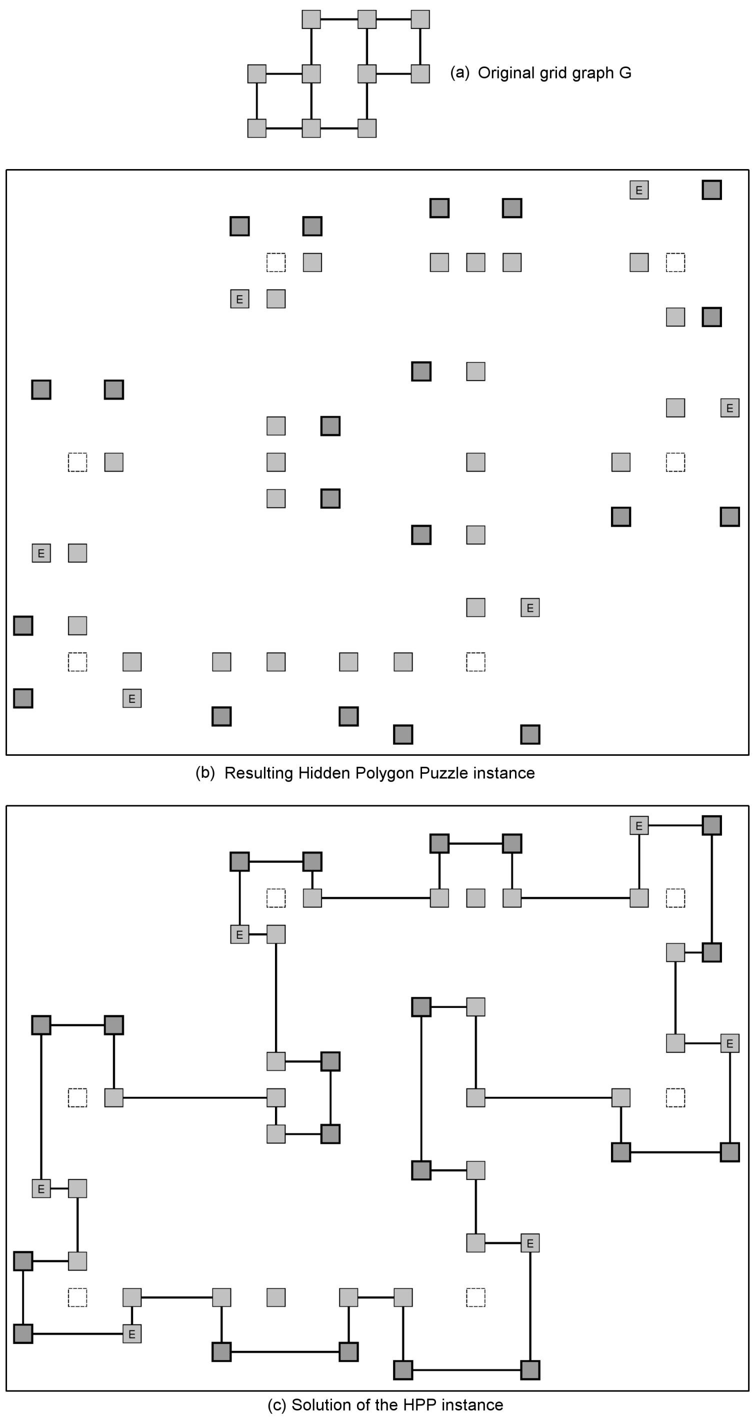

Figure 5: a) A grid graph G with $n = 10$ nodes and b) the corresponding HPP instance with $k = 220$. The Zigzag points are collapsed to make the drawing more readable; each bold box contains n Zigzag points. The bottom drawing c) shows a simple rectilinear polygon with $230 > k$ vertices that is a valid solution and that corresponds to an Hamiltonian cycle on the original grid graph.

[2] Christos H Papadimitriou and Umesh V Vazirani. On two geometric problems related to the travelling salesman problem. *Journal of Algorithms*, 5(2):231 – 246, 1984.