Some Thoughts on Geometry, Colourings, and Honza's Very Old Research

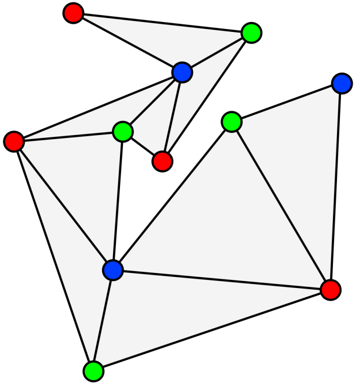
Petr Hliněný

Faculty of Informatics, Masaryk University
Brno, Czech Republic

joint work with

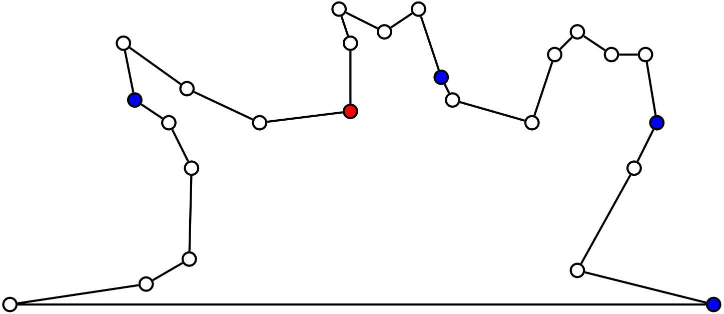
O. Çağırıcı, S.K. Ghosh, B. Roy

1 Guarding a Polygon (The Art Gallery Problem)



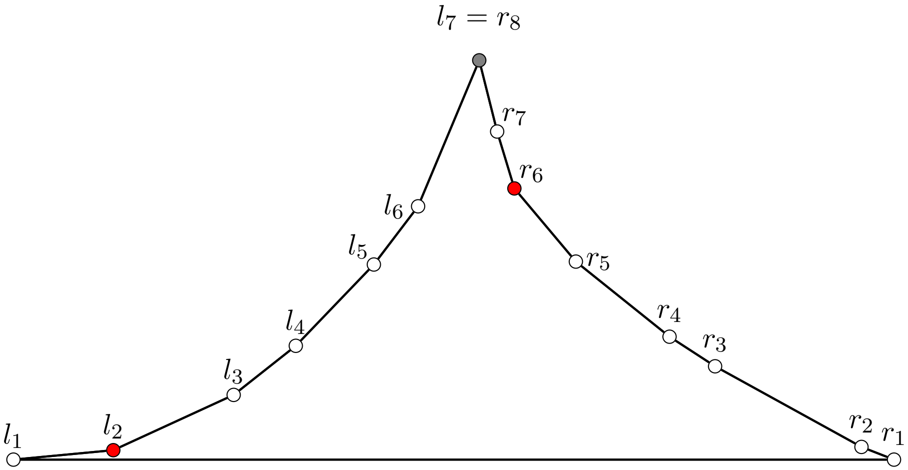
- Having got a polygon – a crazily shaped *art gallery* (non-convex),
- the task is to place static guards protecting every point of the gallery.
- Guard often restricted to the vertices – *vertex-to-point guarding*.

Another View: Conflict-free coloured guards



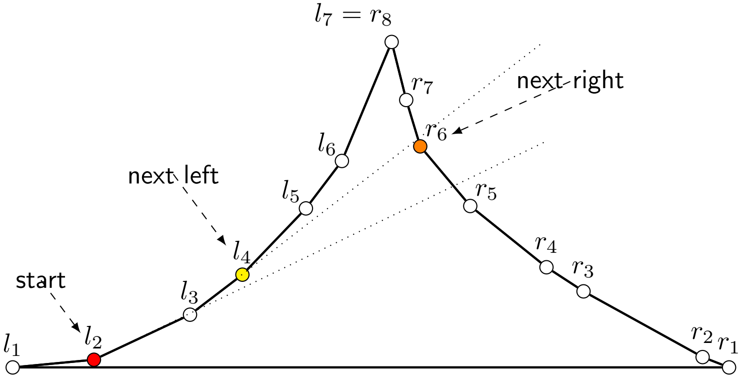
- This time, imagine static sensors (the guards), and
- a mobile robot that needs to communicate with the sensors.
- Each sensor has a fixed radio frequency – its *colour*, and the communication requires a unique avail. frequency due to interference.
- Mathematically:
  Assign *coloured guards* to the vertices of a polygon, s. t. every point in the polygon can see, among all visible guards, one which is of unique colour.

2 Guarding a Funnel (non-Coloured)



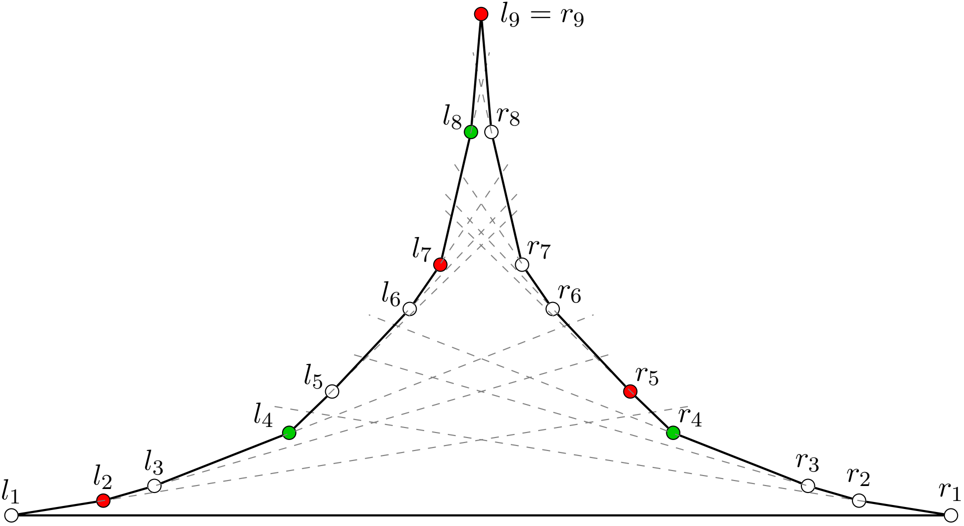
- A special case not studied so far in the literature.
- We can very efficiently find the minimum-size guard set (next)...

Algorithm 1: Very Simple Bottom-Up



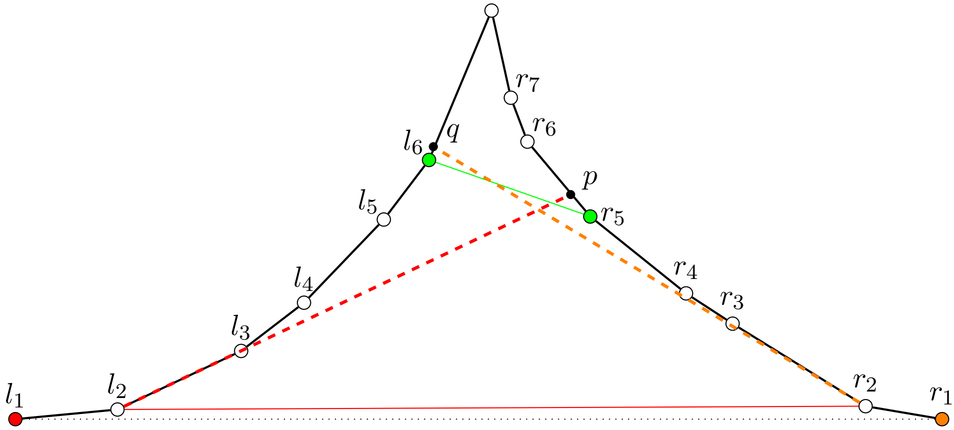
- Start at the bottom-left or (and) bottom-right (nondeterministic).
- Each step, find the topmost next vertex on the left / right, such that you guard a downward-closed region (again nondeterministic).
  Act., enough to guard the upper segment of visible area of the previous g.
- Encode the nondeterministic choices by an auxiliary directed (bottom-up) graph, and find a shortest path.

BUT, 'Very Simple' has a problem. . .



- Algorithm 1 selects the 4 red guards.
- The 3 green guards are enough!

Algorithm 2: Simple Bottom-Up with Double-Move

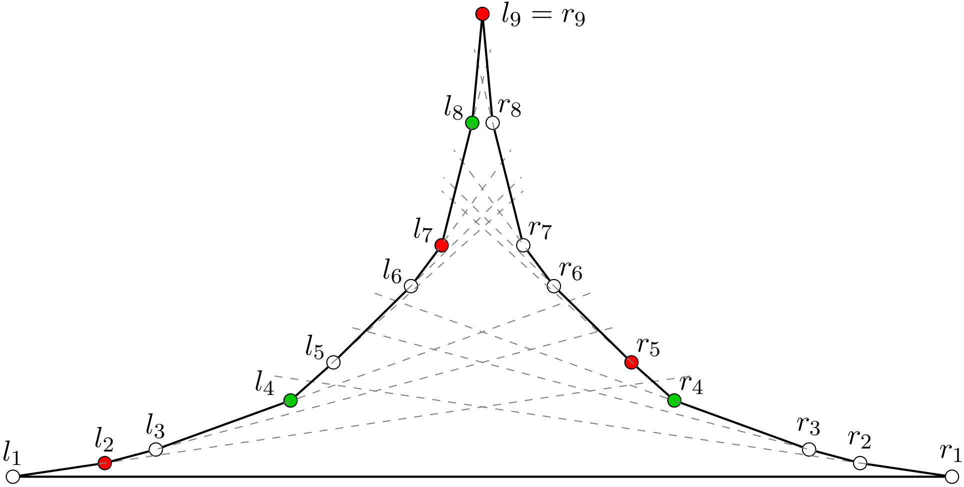


- Start with everything as in Algorithm 1.
- At each step, add a third move;
  find the topmost left-right pair of guards that (again) guard a downward-closed region (nondeterministic third choice).
- Find a shortest path in the (now weighted, 1 or 2) aux. graph.

Theorem. Both Algorithms run in polynomial time.

Algorithm 2 computes a feasible vertex guard set of the optimal cardinality.

Algorithm 1 computes a feasible vertex guard set of cardinality $\leq OPT + 1$.



3 Guarding a Funnel – Conflict-Free Coloured

Know the *ruler sequence*?

1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,5,1,2,1,3,1,2,1,4,1,2,1,3,1,2,1,6,1,2,1,3,1,2,1,...

– the i-th member c_i is the largest exponent of 2 that divides $2i$.

Algorithm C. (C as '*Conflict-free Colouring*')

1. Get the guard set of Algorithm 1 (recall, $\leq OPT+1$), ordered bottom-up:

$$A = (a_1, a_2, a_3, \ldots, a_{s-1}, a_s)$$

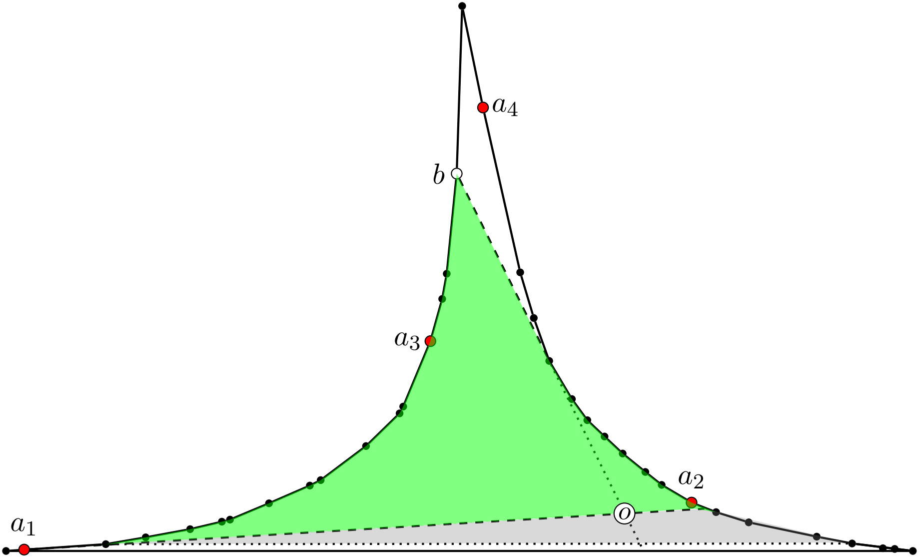
2. Give guard a_i the colour c_i.

That is all! And we have used $\lfloor \log_2 s \rfloor + 1$ colours.

Theorem. Algorithm C computes in polynomial time a feasible conflict-free coloured vertex guard set using $\leq OPT + 4$ colours.

Funnel Interval, its Observer and Shadow

To prove a logarithmic lower bound, we use a bisection approach based on the notion of intervals of the funnel; a *k-interval* includes k guards of our A.



Proving the lower bound

By induction:

- If Q is a k-interval in the funnel F and $k \geq 16 \cdot 2^n - 3$, then any conflict-free colouring of F must use at least $n+1$ colours on the vertices of Q or the shadow of Q.

The base is easy, and the induction step follows from:

- The observer of the interval Q must see a unique colour.
- If p is any vertex of a k-interval Q, and Q_1 and Q_2 are the sections of Q "above" and "below" Q, where Q_i is a k_i-interval, then $k_1 + k_2 \geq k - 3$.

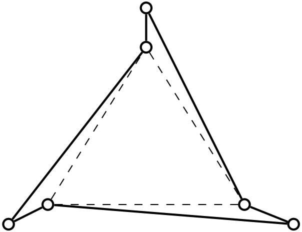
  (In other words, at most 3 of the A-guards in Q are not in $Q_1 \cup Q_2$.)

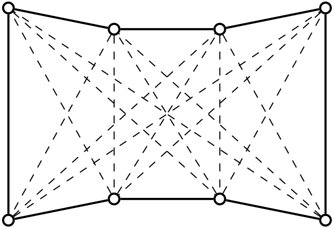
□

4 Vertex-to-Vertex Guarding (Coloured)

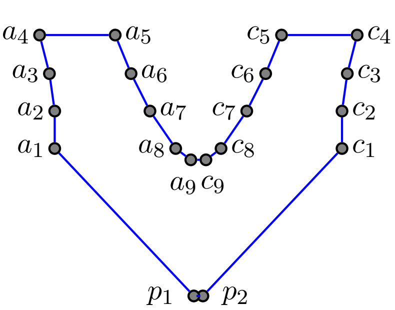
Now, also the observer (not a mobile robot) is restricted to the vertices.

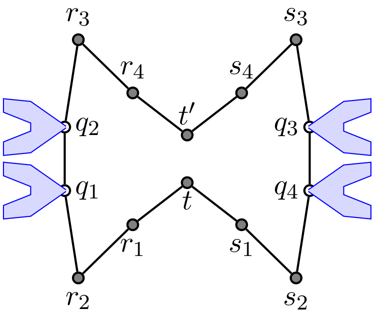
- This is exactly about conflict-free colouring of the visibility graph.
- While conflict-free colouring of general graphs (and its complexity) has recently got an attention, there seem to be no published investigation of it for the polygon visibility graphs.





Claim. There exist polygons requiring at least 2 colours.





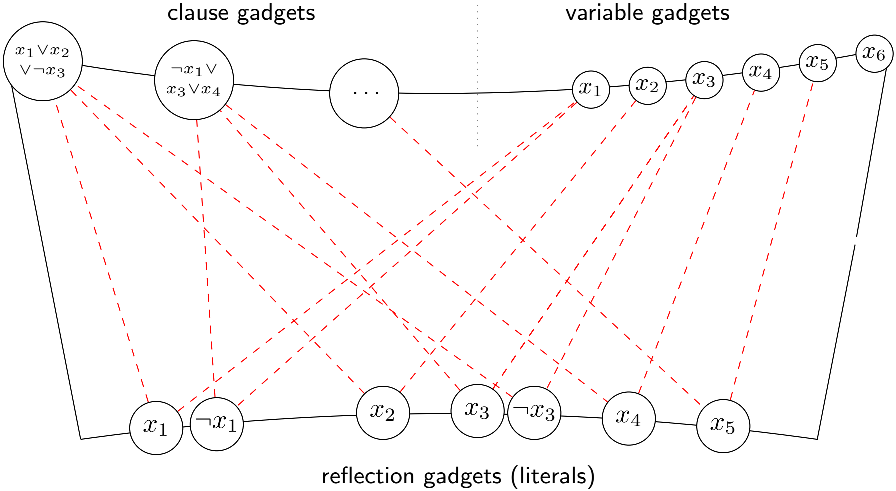
Claim. There exist polygons requiring at least 3 colours. ▮

Conjecture. (to attract research interest)

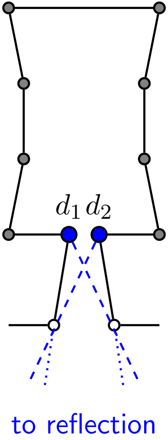
For every simple polygon, its visibility graph has a conflict-free colouring with at most 3 colours.

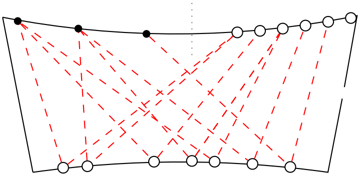
Hardness of vertex-to-vertex guarding

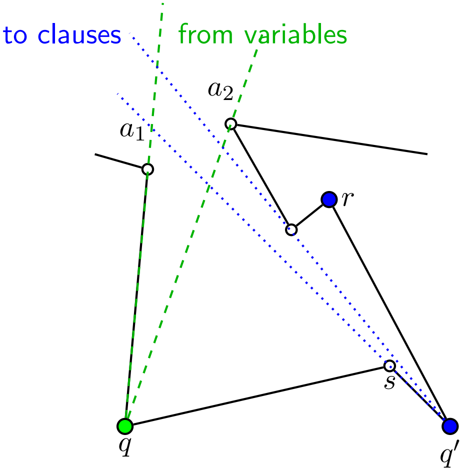
Theorem. For $c \in \{1,2\}$, the question whether for a given simple polygon its visibility graph can be conflict-free coloured using at most c colours, is NP-complete.



The case of 1-colouring







5 Relation to Honza's Reserach?

ON THE COMPUTATIONAL COMPLEXITY OF CODES IN GRAPHS

JAN KRATOCHVÍL and MIRKO KŘIVÁNEK*
Charles University, Prague

Abstract. This paper links to continuing research of the first author on codes in graphs [7-11]. Here codes are studied from the point of view of their computational complexity. It is shown that the problem of perfect code recognition is *NP*-complete even when restricted to k-regular graphs ($k \geq 4$) or to 3-regular planar graphs. On the other hand in the case of trees and graphs of bounded tree-width an optimal $\Theta(n)$ algorithm is developed. Some optimization problems are also investigated.

II. Introduction. The theory of self-correcting codes belongs to thoroughly investigated parts of applied combinatorics. Special attention was paid to the most effective codes, so-called perfect codes. Such codes were shown to be fairly rare, namely in the case of the classical Hamming metrics [2,14]. The classical concept of perfect codes was generalized by Biggs [3] to perfect codes in graphs. However, Biggs and others [6,14] studied only distance regular graphs for which a strong necessary condition for the existence of perfect codes was derived [3].

Perfect codes in general graphs (and their cartesian powers) were studied in [7,9,10,11]. On the other hand within the context of

6 Conclusions

Happy Birthday!