

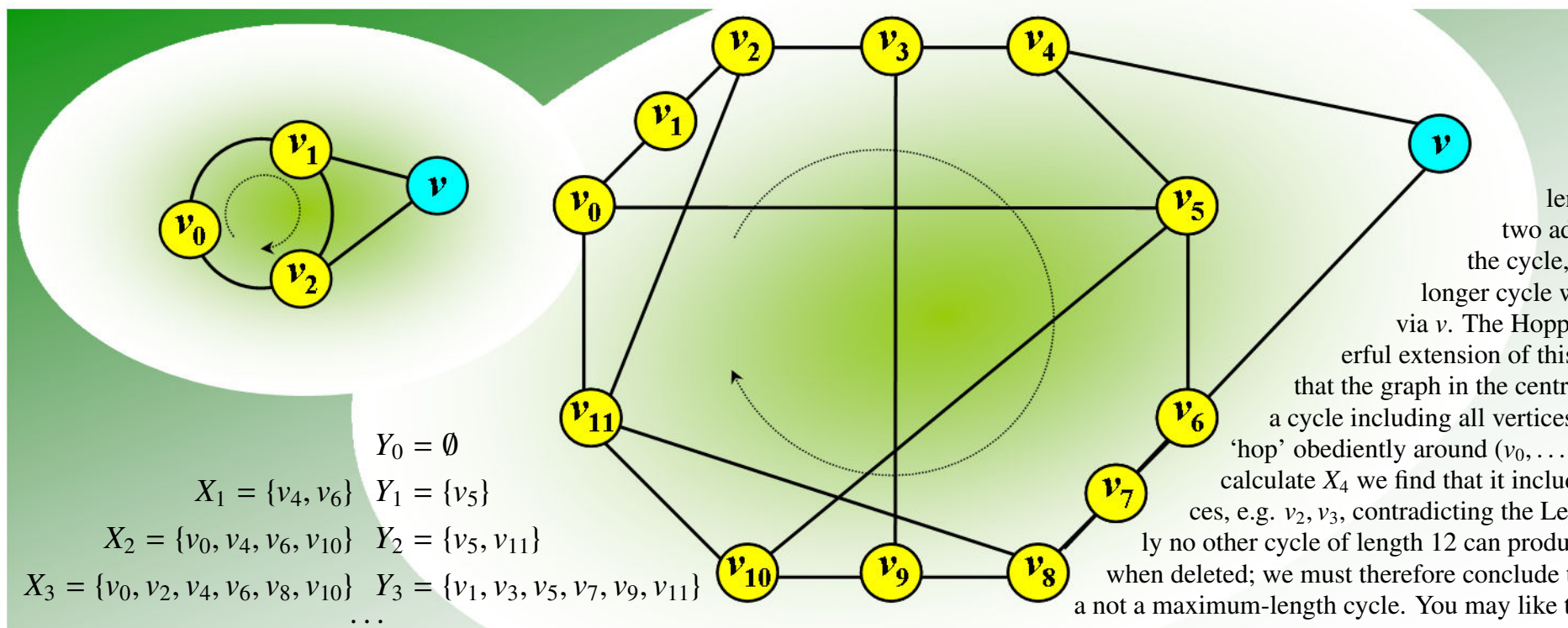


THEOREM OF THE DAY

Woodall's Hopping Lemma Among the maximum-length cycles in a graph G , choose one which, when its vertices are deleted, breaks G into the smallest possible number of connected components, and suppose that one such component is a single vertex v . Let the chosen cycle be $C = (v_0, \dots, v_{t-1}, v_0)$ (the v_i being indexed modulo t). Now define two collections of vertex sets by:

$$Y_0 = \emptyset; \text{ and for } i = 1, 2, \dots, X_i = N(Y_{i-1} \cup \{v\}), \text{ and } Y_i = \{v_j \in C \mid v_{j-1}, v_{j+1} \in X_i\},$$

where for a vertex set S , $N(S)$ is the set of all vertices adjacent to vertices in S . Then for all $i, j \geq 1$, X_i is a subset of C which is disjoint from Y_j and contains no two consecutive vertices of cycle C .



In the graph on the far left, is (v_0, v_1, v_2, v_0) a cycle of maximum length? Because v has two adjacent neighbours on the cycle, we see at once that a longer cycle will pass from v_1 to v_2 via v . The Hopping Lemma is a powerful extension of this idea. Here it reveals that the graph in the centre is Hamiltonian (has a cycle including all vertices): the elements of X_3 'hop' obediently around $(v_0, \dots, v_{11}, v_0)$ but when we calculate X_4 we find that it includes consecutive vertices, e.g. v_2, v_3 , contradicting the Lemma. Why? Certainly no other cycle of length 12 can produce fewer components when deleted; we must therefore conclude that $(v_0, \dots, v_{11}, v_0)$ is a not a maximum-length cycle. You may like to try and find a longer one.

Let graph G have vertex set V and consider $b(G) = \min |N(S)|/|S|$, minimising over nonempty, proper subsets S of V for which $N(S) \neq V$. In 1973, Douglas Woodall proved that if $b(G) \geq 3/2$ then G is Hamiltonian, inventing his Lemma for the purpose. It has since found and continues, in many variants, to find, numerous related applications.

Web link: doc.utwente.nl/65696/

Further reading: *Graph Theory* by J.A. Bondy and U.S.R. Murty, Springer, 2008, Chapter 18.

