Algorithms: Design and Analysis, Part II

Dynamic Programming

WIS in Path Graphs: Optimal Substructure

Optimal Substructure

Critical step: reason about structure of an optimal solution [in terms of optimal solutions of smaller subproblems]

Motivation: this thought experiment narrows down the set of candidates for the optimal solution; can search through the small set using brute-force search.

Notation: Let $S \subseteq V$ be a max-weight independent set (IS). Let v_n = last vertex of path.

A Case Analysis

Case 1: Suppose $v_n \notin S$. Let G' = G with v_n deleted.

Note: S also an IS of G'.

Note: S must be a max-weight IS of G' – if S^* was better, it would also be better than S in G. [contradiction]

Case 2: Suppose $v_n \in S$.

Note: previous vertex $v_{n-1} \notin S$. [by definition of an IS] Let G'' = G with v_{n-1}, v_n deleted.

Note: $S - \{v_n\}$ is an IS of G''.

Note: must in fact be a max-weight IS of G'' — if S^* is better than S in G'', then $S^* \cup \{v_n\}$ is better than S in G [contradiction]

Tim Roughgarden

Toward an Algorithm

Upshot: a max-weight IS must be either
(i) a max-weight IS of G' or
(ii) v_n + a max-weight IS of G''

Corollary: if we knew whether or not v_n was in the max-weight IS, could recursively compute the max-weight IS of G' or G'' and be done.

(Crazy?) idea: try both possibilities + return the better solution.