Institut for Matematik & Datalogi
Syddansk Universitet

February 21, 2013
JBJ

Combinatorial Optimization II (DM209)) — Ugeseddel 4

Handout material in week 8

- Tarjan og Yannakakis, Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs and selectively reduce acyclic hypergraphs, Siam J. Computing 13 (1984) 566-579. We only cover pages 566-569.
- Lucena, A new lower bound for tree-width using maximum cardinality search, siam J. Discrete Math. 16 (2003) 345-353. We just cover the conclusion that every maximum cardinality search provides a lower bound for the tree-width of the graph (namely, the maximum number of edges any vertex has to lower numbered vertices).
- Thomasse, A quadratic Kernel for feedback vertex set, SODA 2009 pages 115-119.

Stuff covered in Week 8

- Chordal graphs based on Golumbic chapter 4 (except section 4).
- Tree width based on Niedermeier sections 10.1-10.4. We covered in full detail a dynamic programming algorithm for the minimum vertex cover problem and for the chromatic number problem (see notes below).

Week 9

- Fixed parameter tractability and kernels based on the paper by Thomasse on the feedback vertex set problem.
- The minimum strong spanning subdigraph problem based on the paper Bang-Jensen and Yeo, The minimum spanning strong subdigraph problem is fixed parameter tractable, Discrete Applied mathematics 156 (2008) 2924-2929.
- 2-sat based on BJG Section 1.10.
- Niedermeier Section 10.6 (briefly).

Finding the chromatic number of a graph G by dynamic programming based on a tree-decomposition of G:

- I gave a proof that for every graph G we have $\chi(G) \leq tw(G) + 1$, where $\chi(G)$ is the chromatic number of G and $tw(G)$ is the tree-width of G, that is, $\beta - 1$ where β is the maximum bag size of some tree decomposition of G. The proof uses that we have $\chi(G) = tw(G) + 1$ for chordal graphs: Given a tree-decomposition $(\{X_i : i \in I\}, I)$ of G we add new edges E' to G so that in the resulting graph G' each of the subgraphs $G'[X_i]$, $i \in I$ are cliques. Clearly $\chi(G) \leq \chi(G')$ and the claim now follows from the fact that G' is a chordal graph whose maximal cliques are exactly those induced by the X_i's.

- I also suggested a dynamic programming algorithm for finding $\chi(G)$ when we are given a tree-decomposition $(\{X_i : i \in I\}, I)$ of G: Let ω denote the size of a largest bag (X_i) and consider all possible colourings of the X_i's by colours $1, 2, \ldots, \omega$ (there are $|X_i|^\omega$ of these. For each such colouring $C_i : X_i \to \{1, 2, \ldots, \omega\}$ we initialize $m(C_i)$ as ∞ if C_i is not a legal colouring (some edge in $G[X_i]$ received the same colour in both ends) and otherwise $m(C_i)$ is $\beta(C_i)$ which is the number of different colours used. Furthermore, we also keep a bit-vector $\gamma(C_i)$ which codes which of the ω colours are used in the colouring C_i (so $\beta(C_i)$ equals the number of 1's in $\gamma(C_i)$).

  After this initialization we are ready to start updating the value $\gamma(C_i)$ and hence $\beta(C_i)$ and $m(C_i)$ using dynamic programming guided by the tree I: When we update the info for X_i from the info for a child X_j we first indentify the set $Z = X_I \cap X_J$ and then, for all of the $|Z|^\omega$ different colourings of $|Z|$ in turn: if C is such a colouring then for every proper colouring C_i of X_i which agrees (that is, uses exactly the same colours on Z as C) we update as follows: For every colouring C_j of X_j which agrees with C and which is a legal colouring of X_j consider the number of 1's in the OR-sum of the bitvectors $\gamma(C_i)$ and $\gamma(C_j)$ and make the new $\gamma(C_i) := \gamma(C_i)\ OR\ \gamma(C_{j'})$, where j' is chosen such that the total number of used colours (bits that are 1) in C_i and $C_{j'}$ is minimum.

  We preform this updating for all children of X_i and continue around the tree in an in-order traversal of I. It can be shown that this will result in the root bag X_r containing a colouring $C_r(X_r)$ whoose value $m(C_r) = \chi(G)$.

  Note that the process above considers the "same" colouring MANY times because a lot of the colourings in the X_i's are identical up to a renumbering of the colours.