SIMPLE ALGORITHMS BASED ON GRAPH SEARCHES

DEREK CORNEIL

UNIVERSITY OF TORONTO

+ COAUTHORS TO BE NAMED

BACKGROUND ON SEARCHING

- FUNDAMENTAL GRAPH OPERATION
- VISIT EVERY VERTEX AND EDGE OF G(V,E)
- INTERESTED IN THE ORDER VERTICES ARE VISITED.

GENERIC SEARCH (TARJAN)

- AT EACH STAGE VISIT AN UNVISITED VERTEX THAT IS ADJACENT TO SOME PREVIOUSLY VISITED VERTEX.

  WHERE DO WE START ?

  HOW DO WE BREAK TIES ?

- CLASSICAL SEARCHES ARE DFS + BFS
- BOTH DESCRIBED OVER A CENTURY AGO FOR MAZE TRAVERSAL.
- 1970s MANY APPLICATIONS OF DFS AND BFS.

ROSE, TARJAN + LUEKER (1976):

INTRODUCED LEXICOGRAPHIC BREADTH FIRST SEARCH (LBFS) AND SHOWED:

G IS CHORDAL IFF THE REVERSE OF AN ARBITRARY LBFS IS A PERFECT ELIMINATION ORDERING

LBFS: $v_1\ v_2\ \dots\ v_i\ \dots\ v_n$

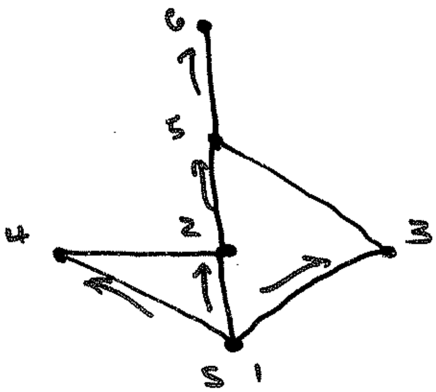
NEIGHBOURHOOD OF v_i IS A CLIQUE.

ALSO SHOWED THIS HOLDS FOR OTHER GRAPH SEARCHES.

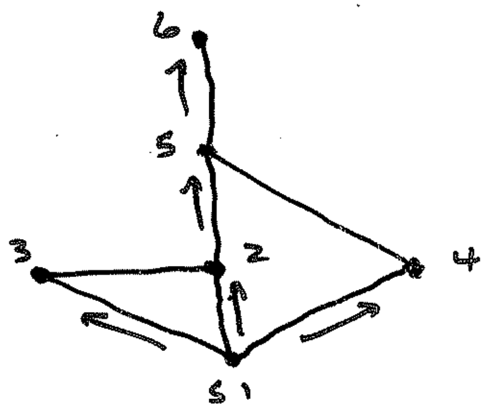
Lexicographic B.F.S.

Overview

- BFS PROCEDURE WHERE "TIES" ARE BROKEN TO FAVOUR VERTICES WITH EARLIER VISITED NEIGHBOURS.

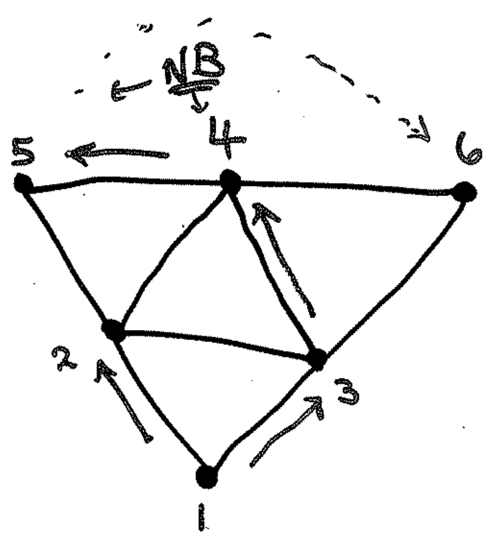


NOT LBFS



LBFS

AT 1, {2,3,4} ARE TIED; AT 2, 4 MUST BE CHOSEN NEXT.



- EASY LINEAR TIME IMPLEMENTATION VIA PARTITION REFINEMENT (~~HABIB~~ HABIB et al.)
- STRAIGHT FORWARD MODIFICATION TO GET AN LBFS OF $\bar{G}$ (WITHOUT CALCULATIN $\bar{G}$)
- <u>MANY</u> APPLICATIONS OF LBFS.

<u>RECOGNITION</u>: CHORDAL, COGRAPHS, INTERVAL, UNIT INTERVAL ..

<u>DIAMETER ESTIMATION</u>: LAST VERTEX OF HIGH ECCENTRICIT

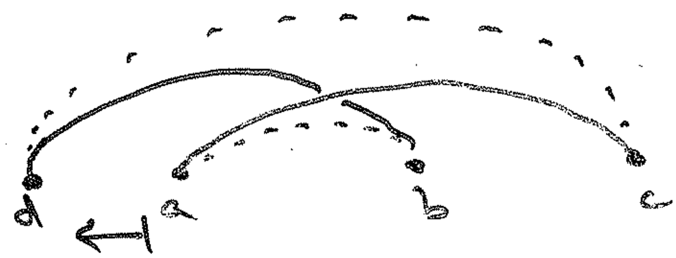
<u>DOMINATING PAIR IN AT-FREE GRAPH</u>:

<u>PREPROCESSING STEP IN DYNAMIC ALGS</u>:

SPLIT DECOMPOSITION, CIRCLE GRAPH RECOGNITION, INTERVAL GRAPH RECOGNITION.

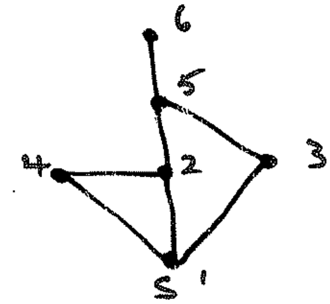
CHARACTERIZATION OF LBFS ORDERINGS

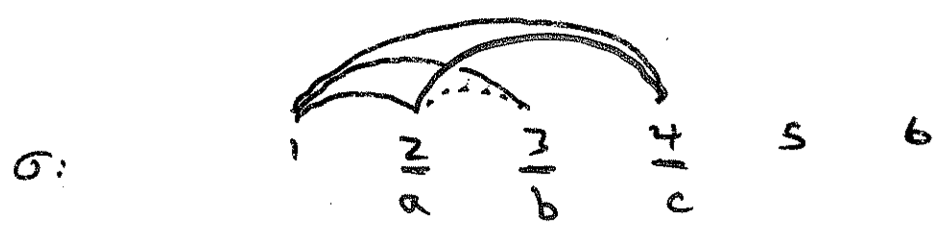
THM: (GOLUMBIC; DRAGAN, NICOLAI + BRANDSTADT): AN ORDERING σ IS AN LBFS ORDERING IFF FOR ALL $a <_\sigma b <_\sigma c$ WHERE $ac \in E$, $ab \notin E$ THERE EXISTS $d <_\sigma a$ SUCH THAT $db \in E$, $dc \notin E$.



$\Rightarrow$

PREVIOUS EXAMBLE





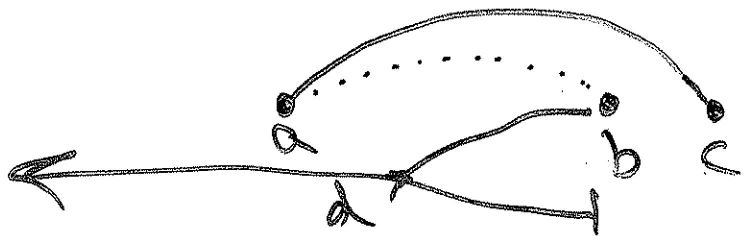
THERE IS NO $d < a$ WITH $db \in E$ $dc \notin E$

NB. THIS PROPERTY IS CRITICAL IN THE PROOFS OF CORRECTNESS OF MULTI-SWEEP LBFS ALGORITHMS

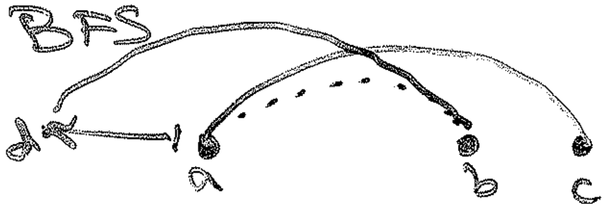
CHARACTERIZATIONS OF OTHER SEARCHES:

(WITH RICHARD KRUEGER)

GEN

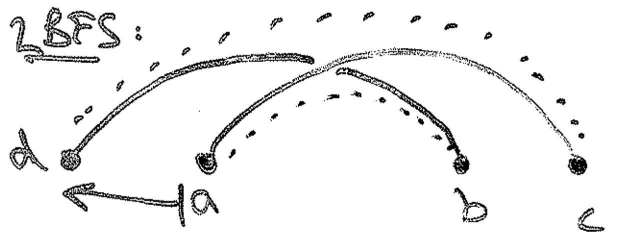


BFS

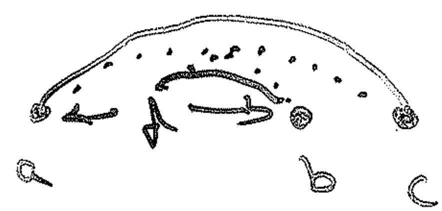


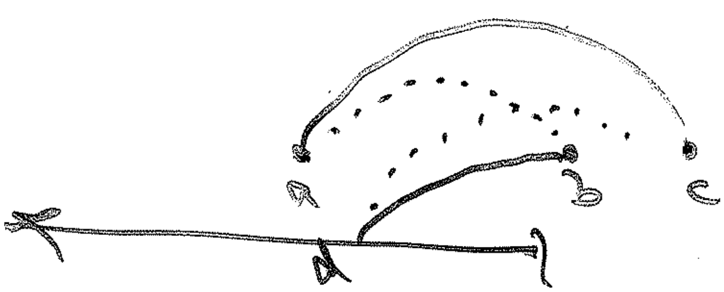
DFS

LBFS:



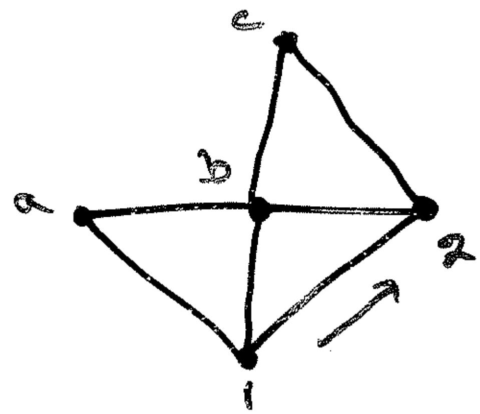
LDFS





MNS.

TEST: COMPLETE THE LDFS.



$1 \to 2 \to b \to c \to a$

IMPLEMENTATION: $O(\min\{n^2, n + m \log\log n\})$

SPINRAD → ???

ARE THERE ANY APPLICATIONS OF LDFS?

HAMILTONIAN PROBLEMS ON COCOMPARABILITY GRAPHS

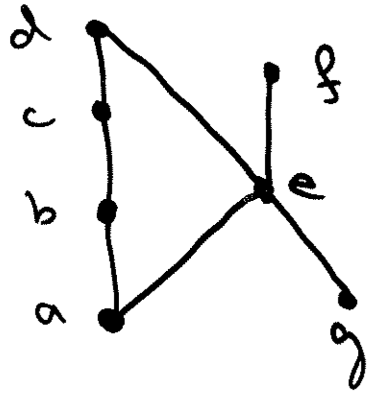
JOINT WORK WITH BARNABY DALTON & MICHEL HAB

COCOMP GRAPHS – THE COMPLEMENTS OF COMPARABILITY GRAPHS (i.e. ∃ T.O.

PREVIOUS APPROACH USED "BUMP NUMBER" FROM POSETS, IN THE MID '90s.

FOR EXAMPLE: FOR THE MINIMUM PATH COVER PROBLEM ON COCOMP G.

1. COMPUTE $\overline{G}$.
2. CONSTRUCT A POSET REPRESENTING THE TRANSITIVE REDUCTION OF $\overline{G}$.
3. FIND A LINEAR EXTENSION OF THIS POSET THAT MINIMIZES THE BUMP #. THIS GIVES THE MIN. PATH COVER ON G.
4. FROM THE CERTIFICATE OF THE BUMP #, GET A SUPERGRAPH H OF G (ON THE SAME VERTEX SET) THAT HAS THE SAME PATH COVER.



A BUMP IN A LINEAR EXTENSION $p_1\ p_2\ \cdots\ p_n$ IS A PAIR $p_i\,p_{i+1}$ S.T. $p_i < p_{i+1}$.

EXAMPLE

LE 1: a g b c e d f   2 BUMPS

$b<c$ $e<d$

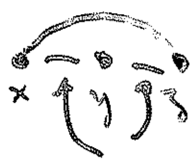
LE 2: a g b e c f d   0 BUMPS

BUMP NUMBER OF POSET P IS THE

$$\min_{\{LE\}} \{b(\text{l.e.})\}$$

NEW ALGORITHM:

NOTE: G IS A COCOMP IFF THERE EXISTS AN ORDERING OF V_G S.T. $\forall\ x < y < z$, $xz \in E \Rightarrow$ AT LEAST ONE OF $xy, yz \in E$



(IMMEDIATE OBSERVATION FROM $\bar{G}$ HAVING A TRANSITIVE ORIENTATION)

STEP 1: LET π BE AN ARBITRARY COCOMP ORDER OF G.

STEP 2: $\sigma = LDFS^{+}(\pi)$ PREPROCESSING STEP

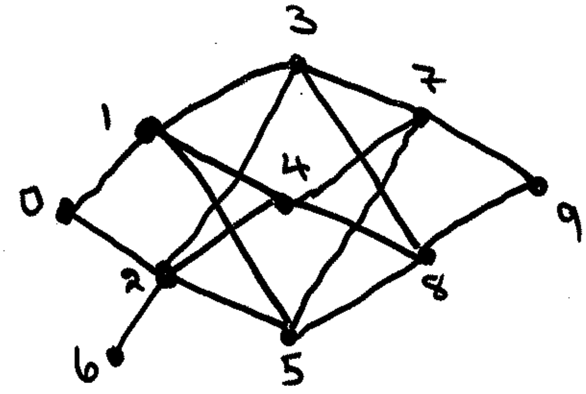
DO AN LDFS ALWAYS BREAKIN TIES BY THE LAST TIED VERTEX IN π.

STEP 3: $\tau = RMN^{+}(\sigma)$ RIGHT MOST NEIGHBOUR FROM THE R.H. END OF σ.

THIS IS THE MIN. PATH COVER

STEP 4: IF τ IS NOT A HAM. PATH, THEN FROM τ CONSTRUCT SEPARATOR S THAT CERTIFIES τ.

EXAMPLE:



π: 0 1 2 3 4 5 6 7 8 9

$LDFS^+(\pi)$

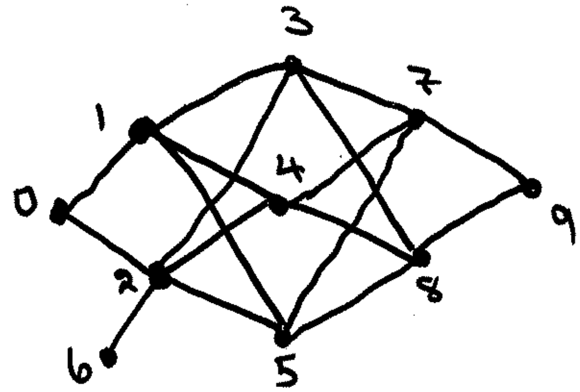
σ: 9 8 5 7 4 2 3 1 0 6

	8	5		4	2
	vs.	vs.		vs.	vs.
	7	4,3		3	1

$Rmn^+(\sigma)$

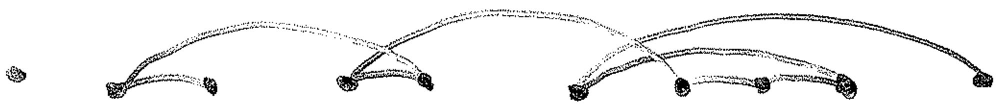
τ:

EXAMPLE:



π: 0 1 2 3 4 5 6 7 8 9

LDFS⁺(π)



σ: 9 8 5 7 4 2 3 1 0 6

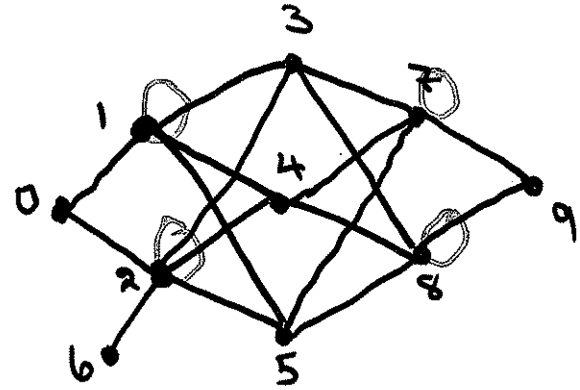
8	5	4	2
vs. 7	vs. 4,3	vs. 3	vs. 1

RMN⁺(σ)

τ: 6 2 0 1 3 7 4 8 5 ‖ 9

FIND S, CERTIFICATE CUT SET

EXAMPLE:



π: 0 1 2 3 4 5 6 7 8 9

LDFS⁺(π)

σ: 9 8 5 7 4 2 3 1 0 6

8	5	4	2
vs. 7	vs. 4,3	vs. 3	vs. 1

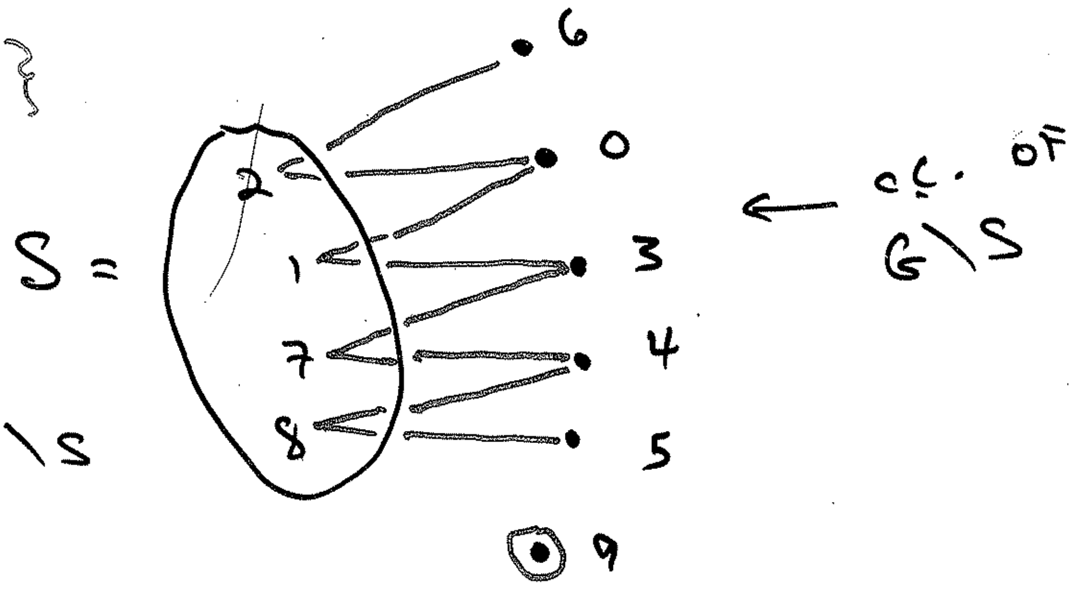
RMN⁺(σ)

τ: 6 (2) 0 (1) 3 (7) 4 (8) 5 || 9

"INNER" FLYERS (2, 1) "OUTER" FLYERS (7, 8)

FIND S, CERTIFICATE CUT SET

S = {1, 2, 7, 8}



MPC # =

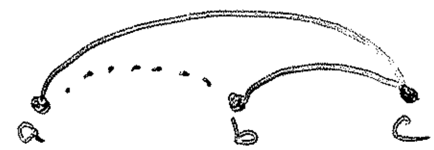
OF CCs OF G \ S − |S|

— MIN PATH COVER.

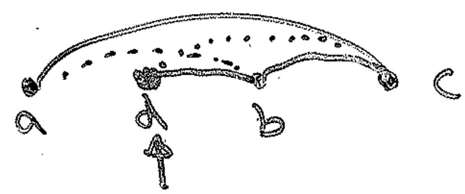
FURTHER COMMENTS:

1. WHY $LDFS^+$?

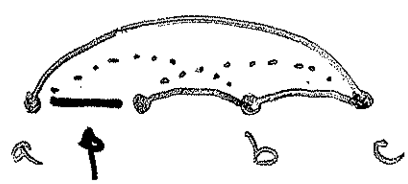
(a) SUPPOSE YOU HAVE IN σ



(b) BY LDFS RULE



(c) BUT $LDFS^+$ OF A COCOMP ORDER IS A COCOMP ORDER



"C_4 PROPERTY"

NOTE RMN^+ OF A COCOMP ORDER IS ALSO A COCOMP ORDER.

2. WHAT DOES THIS ALGORITHM SAY ABOUT BUMP NUMBERS IN POSETS?

(a) $LDFS^+$, WHEN TRANSLATED TO A TRANSITIVE REDUCTION OF $\overline{G}$, GIVES A PREPROCESSING STEP ON THE POSET

(b) USING THIS NEW (EQUIVALENT) POSET, THE TRANSLATION OF RMN^+ GIVES A SIMPLE GREEDY ALGORITHM TO CONSTRUCT A LINEAR EXTENSION OF THE POSET THAT MINIMIZES THE BUMP NUMBER.

(c) S (THE CUTSET) GIVES A NEW, AND EASIER, CERTIFICATE OF THE BUMP NUMBER.

3. TIMING?

(a) COMPUTING π TAKES LINEAR TIME (BUT O(MM) TO VERIFY) INSTEAD OF VERIFYIN USE A "ROBUST" APPROACH SINCE BOTH OUR PATH COVER AND CERTIFICATE CAN BE VERIFIED IN LINEAR TIME.

(b) $LDFS^+$ TAKES $O(n + m \log\log n)$ CURRENTLY

(c) RMN^+ TAKES LINEAR TIME

(d) COMPUTING S TAKES LINEAR TIME.