A GRASP for Graph Planarization

Maurício G.C. Resende

Mathematical Sciences Research

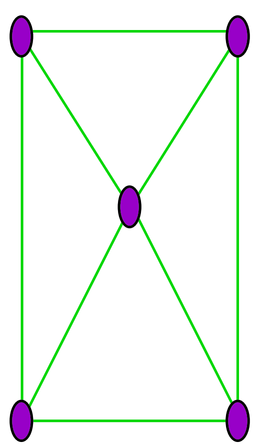
AT&T Bell Laboratories

Murray Hill, New Jersey - USA

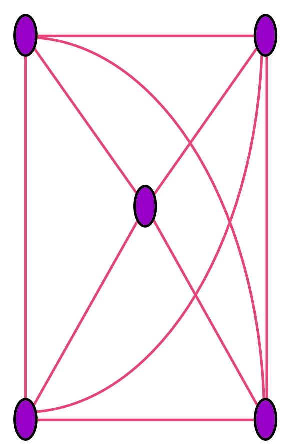
(joint with Celso C. Ribeiro - Catholic U. of Rio de Janeiro)

Graph Planarization

Planar graph: A graph that can be drawn on a plane, such that its edges do not cross.



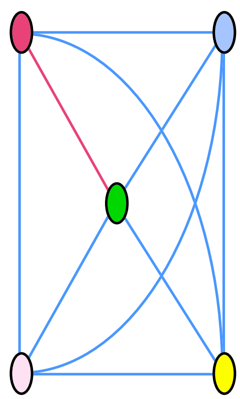
Planar graph



Nonplanar graph

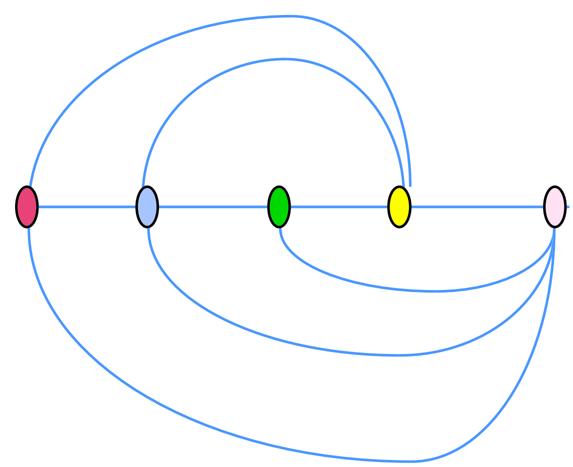
Graph planarization problem

- Given an undirected graph G = (V,E), find the largest subset of edges $A \subseteq E$ such that H = (V,A) is planar.
- The graph planarization problem is also called the **maximum planar subgraph problem**.



nonplanar graph G

maximum planar subgraph of G



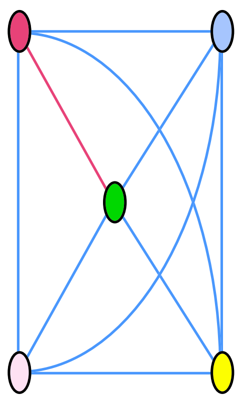
Previous work

- Cimikowski (1995) does extensive empirical evaluation of heuristics, concluding:
  - –Jünger & Mutzel (1993) heuristic is overall best in finding the largest planar subgraphs.
  - –Goldschmidt & Takvorian (1994) is second best, but is slow as input graph size increases.
  - –If time is critical, heuristics based on planarity testing are recommended.

Goldschmidt-Takvorian Heuristic

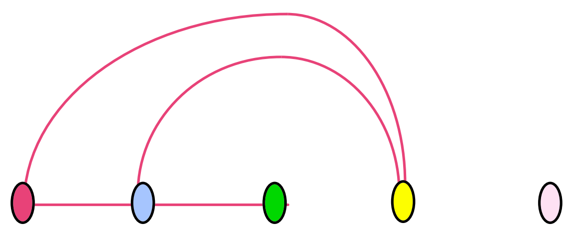
- As in the Takefuji & Lee (1989) heuristic, a 2-phase process is used in the Goldschmidt & Takvorian (G&T) heuristic:
  - phase 1: linear ordering of vertices is produced
  - phase 2: Partition E into edge sets A, B, C, so:
    - » no 2 edges both in A or both in B intersect with respect to linear ordering of phase 1
    - » |A| + |B| is large (hopefully maximum)

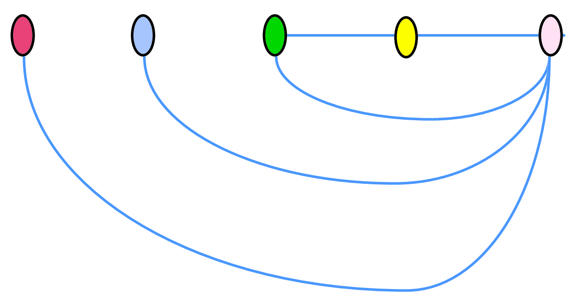
2 Phase Heuristic



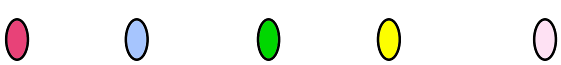
Nonplanar graph

phase 2: subset A of edges





phase 1: linear ordering



phase 2: subset B of edges

Phase 1 of Goldschmidt-Takvorian Heuristic

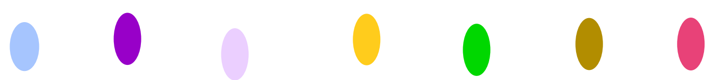
Finding a Hamiltonian cycle in a graph is NP-complete. Greedy heuristic:

- First vertex in ordering O is a vertex with smallest degree.
- After first k vertices of O have been determined, vertex v_{k+1} is vertex adjacent to v_k having smallest degree in the graph induced on $V - \{v_1, \ldots, v_k\}$
- If v_k has no neighbor, v_{k+1} is smallest degree vertex in induced graph.

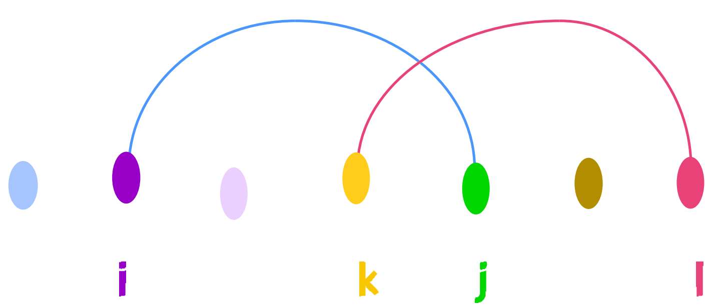
Phase 2 of Goldschmidt-Takvorian Heuristic

Edge partition is made.

Lay vertices on a line according to phase 1:



Definition: let (i,j) and (k,l) be edges of the input graph such that $i < j$ and $k < l$ (assume w.l.g. that $i < k$). Edge (i,j) **intersects** edge (k,l) if $k < j < l$.

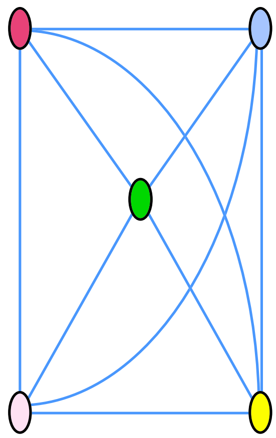


Phase 2 of Goldschmidt-Takvorian Heuristic

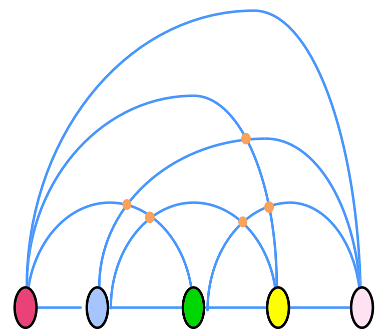
- Define a graph H = (E,A), where:
  - to each edge of input graph G corresponds a vertex in H.
  - Two vertices in H have an edge between them if their corres-ponding edges in G intersect (over-lap) in the linear ordering of phase 1.

Phase 2 of G&T applies a greedy heuristic to find a maximal bipartite subgraph of H.

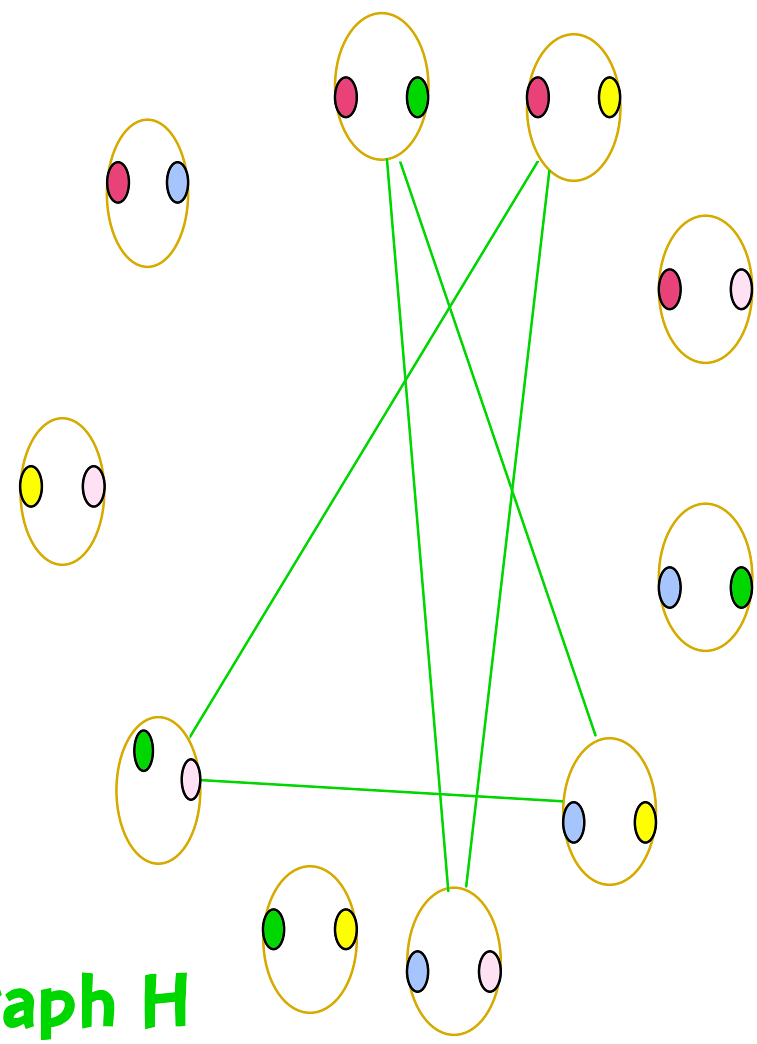
Phase 2 of Goldschmidt-Takvorian Heuristic



nonplanar input graph G

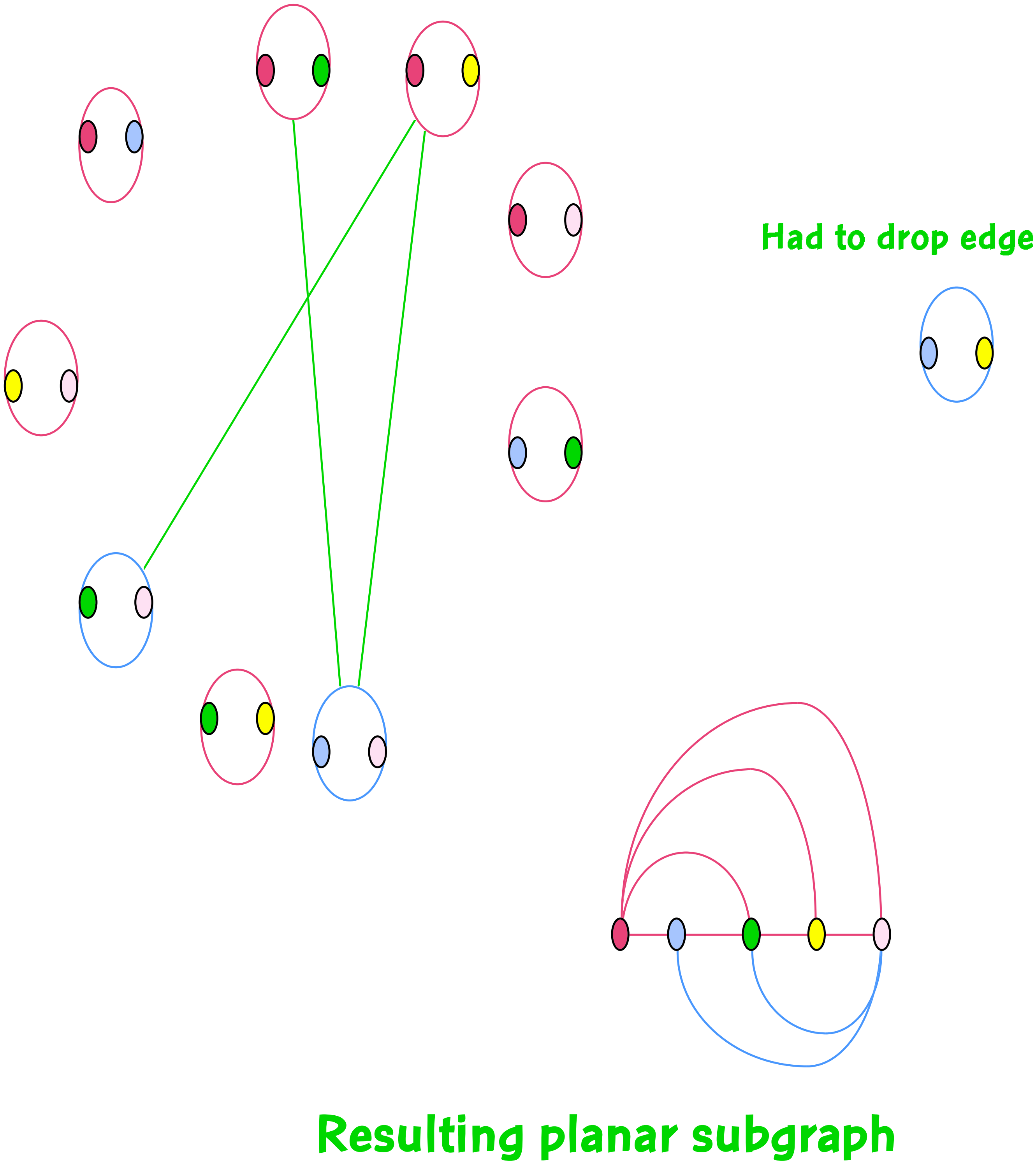


intersections



The overlap graph H

Maximal bipartite subgraph of H?

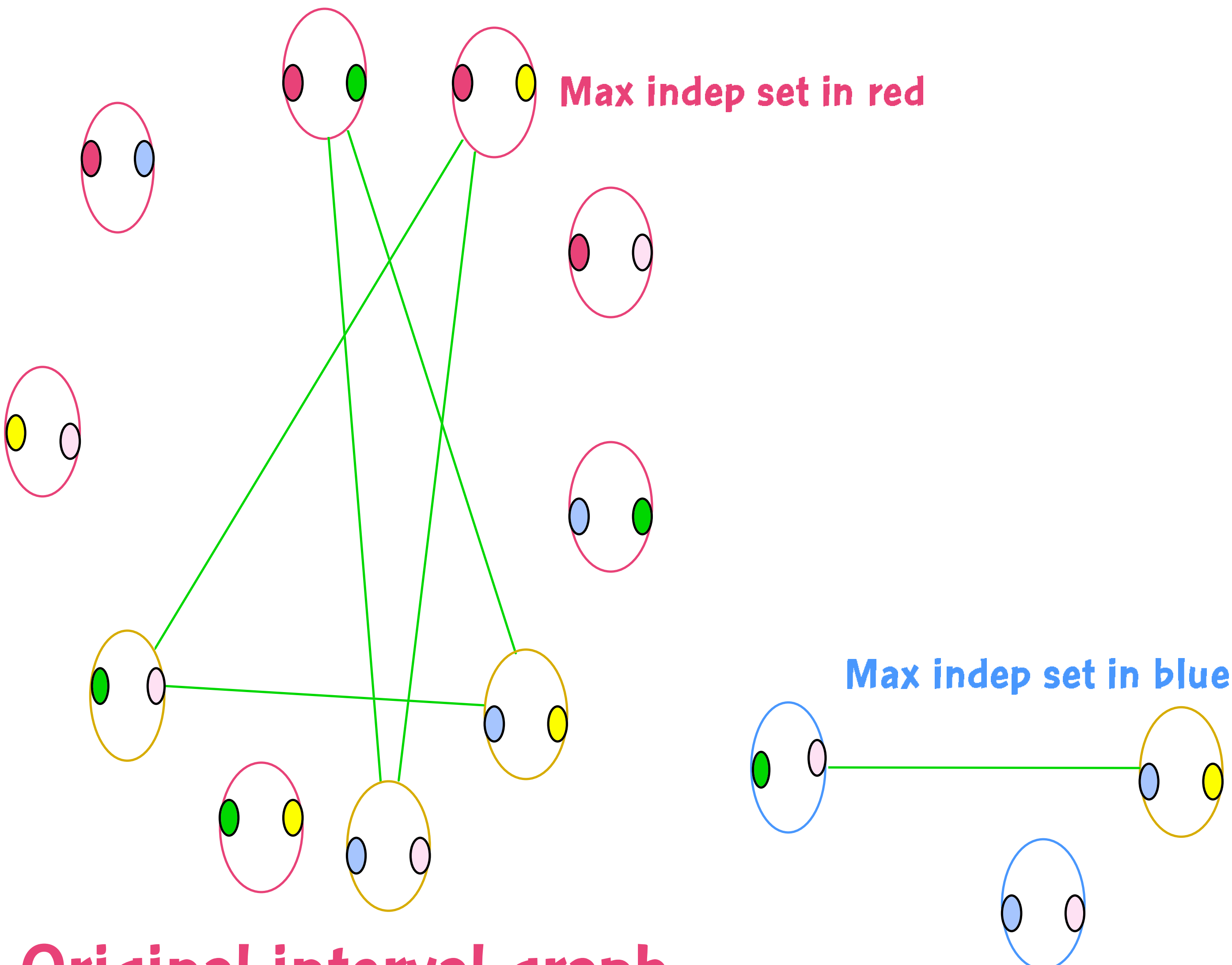


Bipartite subgraph of interval graph

G & T propose the following greedy heuristic to produce a maximal bipartite subgraph of an interval graph:

- Find a maximum independent set of H using Gavril's polynomial-time algorithm [1973]
- Color vertices in max indep set red (these are red edges in original graph)
- Remove vertices in max indep set (and incident edges) from H, and find a max indep set on reduced graph (in polynomial time)
- Color vertices in max indep set blue.

Bipartite subgraph of interval graph



Original interval graph

Reduced graph

The GRASP Metaheuristic

- **G**reedy **R**andomized **A**daptive **S**earch **P**rocedure is a meta-heuristic for combinatorial optimization
- Tutorial: Feo & Resende (1995)
  http://netlib.att.com/netlib/att/math/resende/doc/gtut.ps.Z
- Iterative method (many solutions produced, best one kept as GRASP solution)
- Each GRASP iteration has 2 phases:
  - construction phase
  - local search phase

GRASP construction phase

- Solution is constructed, one element at a time.
- All yet-unselected elements are ranked according to a greedy function.
- Restricted candidate list (RCL) contains highly ranked elements.
- An element is selected, at random, from RCL and is placed in the solution.
- Greedy function is adapted to take into account new element in solution.

GRASP phase 1 for linear ordering

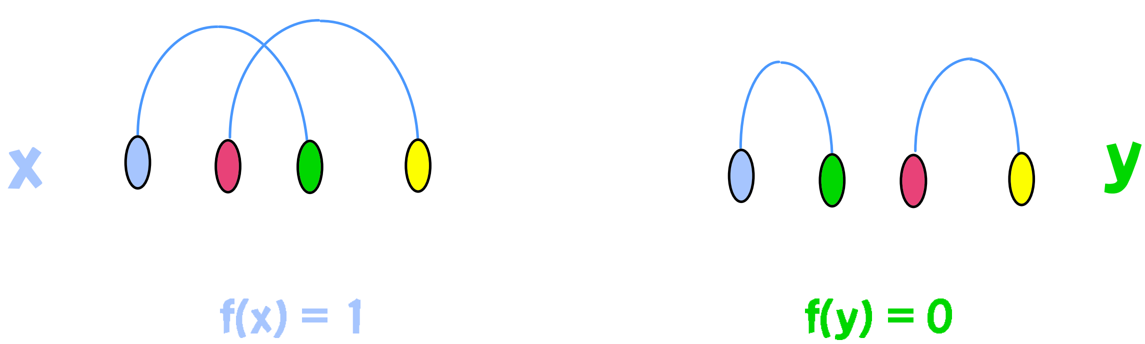
- Let deg(min) & deg(max) be the min & max degrees of vertices in G = (V,E).
- RCL(U) are vertices in vertex subset U with degree < (deg(max)-deg(min))/2.
- Pick node v(1) of ordering:
  - RCL is defined with U=V
  - v(1) is selected at random from RCL
  - V=V\{v(1)}
  - G is graph induced by V

GRASP phase 1 for linear ordering

- Select 2nd, 3rd, ..., n-th nodes in ordering
- To pick k-th node:
  - –If ADJ_NODES[v(k-1)] not empty
    - » then U = ADJ_NODES[v(k-1)]
    - » else U = V
  - –Select v(k) at random from RCL(U)
  - –V=V\v(k)
  - –G is graph induced by V

GRASP phase 2 for linear ordering

- Local search:
  - while (f(x) > f(y) : y in N(x)) do:
    - » pick y in N(x) such that f(y) < f(x)
    - » set x = y
- Objective function f(x): number of crossing edges with respect to ordering
- Neighborhood N(x) definition: 2-exchange



GRASP for graph planarization

Repeat MAXITR times:

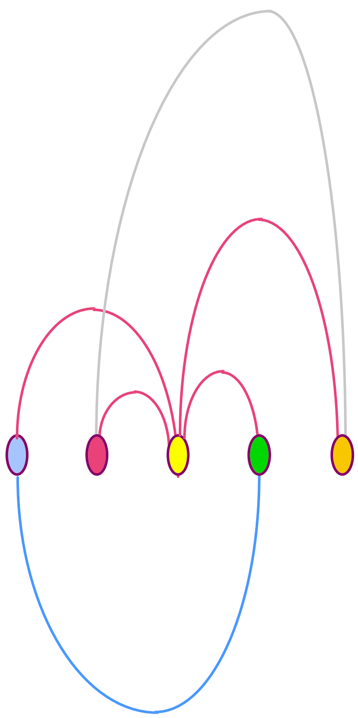
- GRASP phase 1 for linear ordering
- GRASP phase 2 for linear ordering
- Phase 2 of Goldschmidt & Takvorian heuristic
  - » exact method: use Gavril's algorithm (1973)
  - » approx method: use GRASP for max indep set (Feo, Resende & Smith, 1994)
- Local search to improve solution

Local search to enlarge subgraph

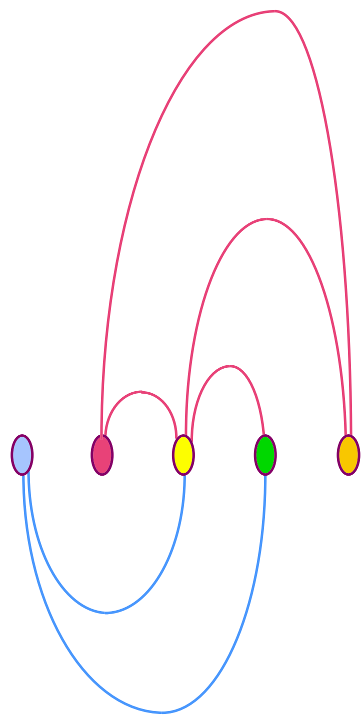
- The heuristic (phase 2 of G&T) used to produce an approximate maximal bipartite subgraph produces 3 sets of edges:
  - set A: red edges
  - set B: blue edges
  - set C: the remaining, or pale, edges
- Attempt to color a red edge blue and a pale edge red, thus increasing size of planar subgraph.

Local search to enlarge subgraph

- Blue edges cannot be colored red, but red edges can be colored blue (if they do not intersect with any blue edge)



|E| = 5



|E| = 6

AT&T

Computational results

- Several variants:
  - –A: Grasp ph 1+2 + G&T ph 2 + local ph 3
  - –B: Grasp ph 1 + G&T ph 2 + local ph 3
  - –C: Grasp ph 1+2 + Grasp MIS + local ph 3
  - –D: Pure greedy GRASP ph 1 + G&T ph 2 (i.e. the G&T heuristic)
- In this talk, we limit ourselves to a subset of the experimental results: we consider variant A.

Hamiltonian Graphs

nodes	arcs	T&L	G&T	Grasp
10	22	20	20	20
10	24	21	24	24
10	25	22	24	24
10	26	22	24	24
10	34	23	24	24

Hamiltonian Graphs

nodes	arcs	T&L	G&T	Grasp
25	70	59	69	69
25	71	58	68	69
25	72	60	68	69
25	90	61	67	69

Hamiltonian Graphs

nodes	arcs	T&L	G&T	Grasp
50	367	70	129	134
50	491	100	138	143
50	582	101	142	144
100	451	92	183	191
100	742	116	215	231
100	922	115	234	243
150	1064	127	291	305

Cimikowski graphs

nodes	arcs	G&T	J&M	Grasp
10	21	19	19	19
60	166	149	165	165
28	75	73	73	73
10	22	19	20	20
45	85	80	82	82
43	63	54	59	58

Random nonplanar graphs

nodes	arcs	G&T	J&M	Grasp
150	387	210	231	219
150	402	213	227	222
150	453	222	229	232
150	473	223	234	236
150	481	227	241	238

Random nonplanar graphs

nodes	arcs	G&T	J&M	Grasp
200	514	268	284	276
200	519	278	283	279
200	644	286	295	300
200	684	296	297	303
200	701	296	300	309

Random nonplanar graphs

nodes	arcs	G&T	J&M	Grasp
300	814	395	398	405
300	1159	431	420	450
300	1176	439	428	449
300	1474	458	469	473
300	1507	467	472	472

Graphs with known optimal solution

nodes	arcs	G&T	J&M	Grasp
100	314	223	294	244
100	334	228	294	243
100	354	242	287	235
100	374	221	281	236
100	394	224	277	229

Graphs with known optimal solution

nodes	arcs	G&T	J&M	Grasp
200	614	439	594	449
200	634	464	591	449
200	654	411	572	447
200	674	423	550	441
200	694	461	536	427

Conclusion

- Extended the heuristic of Goldschmidt and Takvorian
  - substituted G&T's phase 1 by a two phase GRASP
  - added a local search after G&T's phase 2 to try to increase size of bipartite subgraph
- GRASP dominates G&T with very little overhead
- Jünger and Muztel, which dominated G&T, does not dominate GRASP

Conclusion

- We tried using approximate solutions of max indep set with little success (GRASP for MIS)
- We use the $O\left(|E|^3\right)$ time exact algorithm of Gavril (1973) to compute the max indep sets
- Goldschmidt & Takvorian (1994) have described an $O(|V||E|)$ time exact algorithm for max indep set
- GRASP can be implemented in parallel in a straightforward way