Advanced Algorithmics

Projects

Deadline: May 24th

Delay: 1 day = 20% of max points

Prerequisite for exam

2010 Spring
Expectations:

• Study the problem
• Implement, Evaluate, Compare, Measure, ...
• Your task is to make the project interesting to others: right questions; cool applications; novel ideas; desire to read; materials to complement next year courses.
• Find a clear objective and focus, state it, study it!
• 20-40h
• Report – 2-3 p text, references; + illustrations, tables; ...
Tasks

• Here is a list of some proposals

• You can propose your own.

• Or select some on your own
  – from international competitions
    • e.g. IOI (ACM) olympics finals series
    • implementation challenges from DIMACS, etc.
    • etc.
Main areas:

• Graph layouts – very large graphs;
• Graph layouts – constraints, self-organising
• Social Network Analysis
• Clustering, bi-clustering, seriation
• Combinatorial optimisation (TSP, SAT, etc...)
• Pattern counting
• Data Structures - speed
Tree layout

- Layout of (really) large trees
- E.g. families – 1000+ members
- Circular layered layouts, “wide Poster”
- Simple: only parental (or maternal)
- Complex: combine
Graph layout

• Graph layout
  – “Physical Spring model” with some extra added constraints or specialised nodes for stars, cliques, connection strength, etc.

• Create a nr of criteria and try to minimize nr of crossings, area of graph, etc.
Constrained Spring Embedding Layout

• Define certain areas (or lines, etc) that “attract” nodes. Allow graph to “layout” itself dynamically.
Graph layout II

• Layout a (weighted) graph by trying to minimize nr of visual connections. I.e. develop some “flows”. E.g. make wider connections if many connections from a “module” or region to another

Figure 9: A communication heat map.
Social Network Analysis

• Q1: Who are the most likely new friends I should connect to (recommendation engine)

• Q2: In a huge graph estimate distance from node A to B. Hint – use some small nr of “landmarks”. Test the quality of distance estimation by comparing actual distance vs the predictions
Order nodes in a graph

- In current social network sites, you often want to search for persons to get in contact with. Some sites provide a kind of search based on names (string matching). But if your friend has several homonymous, which one is the one that you are looking for? It often happens that the person that you are looking for is already in the buddy list of one of your friends (e.g. only one hop away). If you represent social networks as graphs, the statement recalls the “one-source shortest path” problem. The classic solutions to that problem are not suitable given the sizes of current real social networks (e.g. Skype, Facebook, MSN Messenger, etc.) and the inherent time constraints (e.g. you expect an answer to your query as soon as possible). However, you will be happy if the site provides you with a list of candidates ordered according to their distance to you in the graph. For this project, you are asked to implement a method for ranking nodes according to (approximate) distances to a given node using an off-line pre-computation as described in [1].

Combinatorial optimisation and algorithmic competitions

• Any of the public (past) competitions on optimisation, path-finding, sorting etc challenges

• Compare your results with best methods
Greedy Tour

Nearest Neighbor Tour

Savings Tour

Optimal Tour
Visit all cities... - physically!

http://cswww.essex.ac.uk/staff/sml/gecco/PTSPComp.html
http://algoval.essex.ac.uk/ptsp/ptsp.html
• 652, 652
• 648, 636
Biclustering

• Ordering rows and columns to reveal modules/areas of high “coherence”

• Example: A. Tanay, R. Sharan, R. Shamir: Discovering statistically significant biclusters in gene expression data. Bioinformatics 18, Suppl.1, 136-44, 2002
Templates – automatically define and organise data templates.

Seriation

- Serialise matrices
- (2-way)
Compound word generator

• Given a set of words/strings
• Generate maximally overlapping artificial compound words.
  – kahvanäguripäevapiltnik
  – riigieelarveteõiendamine
• Minimal length, maximum nr of words concatenated with overlaps.
• Palindromic! (2-way search)
15-puzzle

• Solver for 15-puzzle; solve larger ones(?)
  – goal is to find shortest path solutions
  – try heuristics
  – different sizes of the problem

• Comment: In case of a well-studied problem, expectations on project achievements, comparisons, reporting, a good tutorial style, etc are much higher; it’s harder to show your own creativity.
I SPEXS: general algorithm

1. \( S = \) input sequences (\( ||S||=n \))
2. \( e = \) empty pattern, \( e.\text{pos} = \{1,\ldots,n\} \)
3. \( \text{enqueue( order , e )} \)

4. \( \text{while } p = \text{dequeue( order )} \)
5. \( \text{generate all allowed extensions } p' \text{ of } p \text{ (}& p'.\text{pos}) \)
6. \( \text{enqueue( order, } p', \text{ priority}(p') \text{ )} \)
7. \( \text{enqueue( output, } p', \text{ fitness}(p') \text{ )} \)

8. \( \text{while } p = \text{dequeue( output )} \)
9. \( \text{Output } p \)

Jaak Vilo: Discovering Frequent Patterns from Strings.

Jaak Vilo: Pattern Discovery from Biosequences
PhD Thesis, Department of Computer Science, University of Helsinki, Finland.
Report A-2002-3 Helsinki, November 2002, 149 pages

Applications in bioinformatics:
- Functional elements in proteins (2002: 32 cit)
Sequence patterns: the basis of the SPEXS

- GCAT (4 positions)
- GCATA (3 positions)
- GCATA.
- GCATA.C
• priority: *e.g.* depth-first; breadth-first; best-first; A*, beam search, ...

• fitness: most frequent; max(freq * length); ...
Similarity join

• From two sets of objects
  – find the best matching pairs

• E.g. two photos from slightly changed angle
  – identify “same” points on two images
  – can have applications in 3-D stereo imaging; photo stitching (panoramic photos), etc.
Graph edit distance

• Calculate edit distance between graphs:
  – add/delete nodes, edges
  – match nodes (e.g. by partially matching labels)

  – Q: How similar are two graphs
Graph: SESE

• Identify region(s) of Single Entry, Single Exit.

• I.e. find subgraphs that can be “collapsed” and studied further recursively, for example

  • Marlon: Here is a set of test cases you could give to the students who choose the project on "identifying SESE regions". The rar file attached contains a README file that explains the contents of the folder and the format used to encode the process graphs.
Graph: counting features

• Count a nr of times certain features exist in graphs; find “most frequent features”
• E.g. small feedback-triangles; feed-forward triangles, etc.
• Hint: count nr of smallest features; add certain feature and recalculate their frequencies.
• Keep expanding the most frequent features (similar to SPEXS idea)
Graph

• Implement Page Rank algorithm
  – Evaluate problem sizes, provide examples
  
  – Abstract graphs
  – Method could be used on scientific literature, social networks, etc...
  – Can you add more features, prevent “spammers”?
The graph of function $G^2$ for $n = 2$. Infeasible solutions were as
Your own projects

- Ask a question
- Study literature
- Propose solution
- Implement
- Experiment and report results of experiments