Advanced Algorithmics (6EAP) Project proposals

Jaak Vilo

2013 Spring
Key info

- Project = 1-2-3 person teams
- Poster session: XXXXXX
  - 2pm - room YYYYY
  - Poster file (PDF?) has to be uploaded before
- Prerequisite for exam
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 / per person
• Report – Poster
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.
• Compare some alternative algorithms, and try to improve on them.
• Take a problem and try to apply algorithmic problems to solve it.
• Take an algorithm from the literature, implement and test.
• Find cool ways to study/visualise algorithmic ideas presented in the course.
• Optimisation using different heuristics
  – **Complex fitness functions** (layout, MDL, etc)
  – Game-playing eninges
  – **Differential Evolution**
• Data structures and speed comparisons
• Succinct data structures – bitlist, tree, graph
Succinct data structures

- Test in practice the size and speed for some succinct data structures (binary search trees, heaps ...)

- Could you support some (any) dynamic update operations? Which kind? What would be needed basic operations for updates? (however bad the time is)
Clustering using Differential Evolution

• Define an objective function
• Define a numeric vector representing a clustering
• Optimise the objective function using differential result
• Report the types of clusters discovered, time and convergence, strengths and weaknesses as compared to other standard clustering methods
Combinatorial search (BFS)

- **Optimal** solution from a particular (or any) state of a 2x2x2 cube
- A* algorithm ???
- “Discover” the short assembly step algorithms
- Provide solutions
GPS navi

• Free map
• Calculate and visualise shortest paths...
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.
• Calculate a short feature vector about the image
• Make a database where images are stored that are “similar”
• 2-or 3-level?
• **Bloom filter?**
• Regions in images that do not have matches?
• Blend images on edges?
Constrained Spring Embedding Layout

- Define certain areas (or lines, etc) that “attract” nodes. Allow graph to “layout” itself dynamically.
TSP variant, but with physical laws of velocity...

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
Vector formula optimisation

A* algorithm mixed with focused wavefront expansion. When “not knowing the track” idea failed, we decided to use wavefront path-planning along with A* algorithm. The resulting algorithm covers racetrack with calculated weights from finish to start and then trying to find the trajectory with best-first search, moving to smaller weights. When failing the algorithm turns back and chooses the second best possible heading.

Calculating weights:
Starting from finish line, every not yet marked cell on the track gets a weight. On every next wave the previously weighted cells orthogonal neighbors get the value +3. After orthogonal weighting the calculated same cells diagonal neighbors are assigned with value +4.

First objective was to find the best trajectory by knowing nothing about the track ahead.

A*, depth-first search:
Finding the fastest trajectory. If car drives out of the track, the speed and/or steering

Visualization
Search results are represented visually using C++ OpenGL.
Seriation

- Serialise matrices
- (2-way)
Biclustering (matching column attributes...)

• Select “best” rows, cluster those

• 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
Show legend >>
Annotations >>
Jump to DiffExp | g:Profiler

51 genes total. Used datasets:

1. A gene expression signature identifies two prognostic subgroups of basal breast cancer (E-GEO-21653)
• I saved a wrong file, lost some examples...
Alizadeh et al., Nature 403:503-11, 2000
Query of OCT4 (POU5F1)
(210265_X_AT)

StdDev < 0.29
Query of OCT4 (POU5F1)
(210265_X_AT: 50 top StdDev datasets in query)

http://eid.ee/b4
Some algorithmic competition

• Test your skills on some algorithmic competition
Perfect!
You completed the level.
Moves: 24 (Best: 24)
Time: 29.54s (Best: 15.01s)

Next
Again
• How big tasks?
• Is random task solvable?
• Make a generator? Assess their complexity?
• Solve some complicated problem?
• Any other layout and rules?
Finally, 15,000 pages later:

\begin{verbatim}
-7  260  0
  7 -260  0
 1072 1070  0
-15 -14 -13 -12 -11 -10  0
-15 -14 -13 -12 -11  10  0
-15 -14 -13 -12  11 -10  0
-15 -14 -13 -12  11  10  0
-7  -6  -5  -4  -3  -2  0
-7  -6  -5  -4  -3  2  0
-7  -6  -5  -4  3  -2  0
-7  -6  -5  -4  3  2  0
 185  0
\end{verbatim}

Combinatorial search space of truth assignments: \textit{HOW?}

\[2^{50000} \approx 3.160699437 \times 10^{15051}\]

Current SAT solvers solve this instance in approx. 1 minute!
Your own projects

• Ask a question
• Study literature
• Propose solution
• Implement
• Experiment and report results of experiments
Evolving patterns or snowflakes