Washington University in St. Louis

Engineering

Review Notes 2

Jon Turner
Computer Science & Engineering
Washington University

www.arl.wustl.edu/~jst

Big Picture

- Kruskal's algorithm and partition data structure
  - » partition operations and how they are used in Kruskal's algorithm
- Analysis of partition data structure
  - » basic lemmas and $O(m \log \log n)$ analysis, multilevel analysis
- Round-robin algorithm and leftist heaps
  - » leftist heap operations; lazy deletion
  - » use of leftist heaps and partition in round-robin; analysis
- Edmonds algorithm for max size matchings
  - » blossom shrinking, use of partition to represent blossoms, finding paths
- Applications of matching
  - » edge coloring, TSP approximations (MST+matching, cycle-patching)
- Linear programming and network optimization
  - » primal/dual formulations, complementary slackness condition
- Edmonds algorithm for max-weight matching
  - » max weight augmenting paths, use of vertex labels, equality edges, label adjustment, odd blossoms

Kruskal's algorithm and Partition

- Kruskal
  - » how the algorithm works; relation to greedy method
  - » role of Partition in algorithm
  - » running time and role of sorting vs. tree-building
- Partition
  - » find and link operations
  - » link-by-rank and path compression

Analysis of Partition

- Basic lemmas
  » properties of *rank*, tree size, number of nodes of rank *k*
- *O*(*m* log log *m*) analysis
  » dominant nodes and their role in dividing analysis into two parts and counting find steps
  » how analysis can be adapted for path compression only
- Multi-level analysis
  » how blocks and levels are defined
  » meaning and implications of singular/non-singular nodes
  » how analysis divides find steps into different categories
  » how bound on # of nodes with rank *k* is used in analysis

Round Robin and Leftist Heaps

- Leftist heaps
  - » how ranks are defined and updated during melds
  - » lazy deletion and melding, including purge and heapify
  - » analysis of heapify
- Round-robin
  - » relation to general greedy method
  - » how Partition and leftist heaps are used in algorithm
    - role of Partition in lazy deletion
  - » how passes are defined and used in the analysis
  - » distinction between "small" and "large" findmins and how this is used to divide analysis into two cases

Max Matching in General Graphs

- Blossoms and blossom-shrinking strategy
- Why Edmonds works
  - » blossom-shrinking preserves existence of augmenting paths
  - » Edmonds keeps shrinking blossoms until path found
- Efficient implementation
  - » how trees are built and the role of odd/even status
  - » role of Partition data structure and origin mapping to represent the "current shrunken graph"
  - » how algorithm determines if two nodes are in same tree and if so, finds their nearest common ancestor
  - » use of bridge mapping and reversible list data structure to recover augmenting path

Applications of Matchings

- Edge coloring in bipartite graphs
  - » find sequence of max degree matchings
- Traveling salesman problem
  - » Christofides' algorithm for symmetric TSP with triangle inequality
    - find MST, then min weight perfect matching in "odd-degree subgraph", combine for Eulerian tour, then take short-cuts
    - tour length at most 1.5 times longer than optimum length
  - » Karp's algorithm for asymmetric TSP with random distances
    - find min weight perfect matching in bipartite graph
    - use to define min weight cycles in original graph
    - patch short cycles into long cycle
    - tour length close to optimum length with high probability

Linear Programming

- Standard matrix form for primal
- LP formulations of common problems
  - max flow, min-cost flow, shortest paths, matchings
- Deriving the dual from the primal
- Relationship between objective function values of primal and dual
- Complementary slackness
  - writing the slack vector for a given solution
  - checking the complementary slackness condition
  - understanding implications for optimality

Edmonds Weighted Matching

- How max weight augmenting paths yield max weight matchings
- Primal/dual strategy
  - » use of equality edges (in both bipartite/general cases)
    - how this yields max weight augmenting paths
  - » label adjustment procedure (both cases)
    - role of δ_1, δ_2,...
    - how adjustment preserves validity of conditions (1), (2)
  - » role of data structures in bipartite case
  - » for general graphs
    - carry-over of blossoms across multiple path searches
    - expanding of odd blossoms during path search