Approximation Algorithms

CS255
Chris Pollett
May 8, 2006.

Outline

- Performance Ratios
- The Vertex-Cover Problem
- The Traveling Salesman Problem

Performance Ratios

- Since it seems hard to find exact solutions to the optimization problems associated with a given **NP**-complete problem, it is natural to ask if one can get approximate solutions in polynomial time?
- We say an algorithm for a problem has an **approximation ratio** of $r(n)$, if for any input of size n, the cost C of the solution produced by the algorithm is within a factor of $r(n)$ of the cost C^* of the optimal solution. That is, $\max(C/C^*, C^*/C) \leq r(n)$.
- We call an algorithm that achieves an $r(n)$ approximation-ratio an ***r*(*n*)-approximation algorithm.**
- Some **NP**-complete problems have a trade-off between the approximation ratio and the runtime.
- An **approximation scheme** for an optimization problem is an algorithm that take both an instance of the problem as well as a constant ε and then runs a $(1 + \varepsilon)$-approximation on the instance.
- If for any ε, approximation scheme run in p-time, then it is called a **polynomial time approximation scheme**.
- We say that an approximation scheme is a **fully p-time approximation scheme** if it is an approximation scheme and its run time is p-time in both $1/\varepsilon$ and the instance size n.

The Vertex Cover Problem

- The optimization problem associated with VERTEX-COVER is to find the least vertex cover of a instance graph *G*.
- The following algorithm takes a graph *G* and outputs a vertex cover within twice the optimal.

APPROX-VERTEX-COVER(*G*)

1. $C = \emptyset$
2. $E' = E[G]$
3. while $E' \neq \emptyset$
    1. do let (u, v) be an arbitrary edge of E′
    2. $C = C \cup \{u, v\}$
    3. Remove from E' every edge incident with either u or v
4. return C.

Analysis

Theorem. APPROX-VERTEX-COVER is a p-time 2-approximation algorithm

Proof. First, the algorithm runs in time O(|*V*| +|*E*|), as we delete two vertices and at least one edge each time through the loop.

The set *C* returned by the algorithm is a vertex cover, since each edge that is removed is covered by some vertex in *C*. And the loop continues till no edges left.

To see that the cover returned is at most twice the optimal, let *A* denote the set of edges which were picked in line 3.1. In order to cover the edges in *A*, any vertex cover (including the optimal C^*) -must include at least one endpoint of each edge in *A*. No two edges in *A* share an endpoint, so no two edge from *A* are covered by the same vertex from C^*. So $|C^*| \geq |A|$. On the other hand $|C| = 2|A|$.

The Traveling-Salesman Problem

- The optimization problem associated with TSP is to find a tour of least cost.
- Here is a 2-approximation algorithm for this problem when the triangle inequality holds

APPROX-TSP-TOUR(*G*, *c*)

1. select a vertex *r* to be a root vertex
2. compute the minimal spanning tree for *G* from root *r* using Prim's algorithm
3. let *L* be the list of vertices visited in a preorder tree walk of *T*
4. return the hamiltonian cycle *H* that visits the vertices in order *L*.

Analysis

Theorem. APPROX-TSP-TOUR is a p-time 2 approximation algorithm for *TSP* with triangle-inequality holding on the cost function.

Proof. The minimal spanning tree algorithm runs in time $O(|V|^2)$. The remaining step take at most $O(|G|)$ time.

Let H^* denote the optimal tour of the vertices.Since we can obtain a spanning tree from any tour by deleting an edge, we have $c(T) \leq c(H^*)$ where T is our minimal spanning tree. A **full walk** F of T lists the vertices when they are first visited and also whenever they are returned to after a visit to a subtree. So $c(F) = 2c(T) \leq 2c(H^*)$. On the other, the H returned by the algorithm satisfies $c(H) \leq c(F)$, since it is obtains by deleting vertices from the full walk and since the triangle inequality holds. Thus, the theorem holds.