Approximation Algorithms

Sometimes we have problems that cannot be solved optimally efficiently - polynomial time. But in practice, it is often good enough to have a near-optimal solution, which is what approximation algorithms do. We say they have an approximation ratio $\rho(n)$ if

$$max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n)$$

There are also approximation schemes, which are algorithms that take an input ε, so that the scheme is a $(1+\varepsilon)$-approximation algorithm. We say the scheme is fully polynomial if it is polynomial in both $1/\varepsilon$ and n.

Consider the optimization problem of the NP-complete optimization problem Vertex cover. The algorithm takes an arbitrary edge and add both endpoints to C, and does this until all edges are covered. We show it is a polynomial time 2-approximation algorithm:

It is polynomial since it checks at most all edges once removes edges from the corresponding vertices. It runs in $\mathcal{O}(V+E)$ if we use adjacency lists. When we pick an edge to put in A, at least one of the vertices must be in the optimal solution C^*. No two edges in A are covered by the same vertex. We have

$$|A| \leq |C^*|$$

Since the number of vertices in the produced solution C is exactly $2|A|$, we have

$$\begin{aligned} |C| &= 2|A| \\ &\leq 2|C^*| \end{aligned}$$

Another problem is the traveling salesman problem where no efficient solution exist (a hamiltonian cycle with minimum cost). We show there is a 2-approximation algorithm when it is a complete undirected graph and the cost function c in TSP satisfies the triangle inequality, that is

$$c(u,w) \leq c(u,v) + c(v,w)$$

It works by generating a minimum spanning tree and listing the nodes in order when they are visited in a preorder tree walk, and this hamiltonian circle is returned. It is clearly polynomial running time ($\mathcal{O}(|E| \lg |V|)$ with binary heaps and adjacency list - $\mathcal{O}(|E| + |V| \lg |V|)$ with fib heaps).
Proof: The spanning tree T provides a lower bound for the cost of an optimal tour H^*, so

$$c(T) \leq c(H^*)$$

Now let us consider the walk W. Every vertex is visited twice, so

$$c(W) = 2c(T)$$

Which we can simply substitute, so

$$c(W) \leq 2c(H^*)$$

The solution produced is just the walk between each vertex (visit only once), but when the triangle inequality we can deduce $c(H) \leq c(W)$ and therefore $c(H) \leq 2c(H^*)$.

There is no $\rho(n)$ approximation algorithm as it prove $P = NP$ by making a complete undirected graph, the cost function $c(u,v) = 1$ when $(u,v) \in E$ and $c(u,v) = \rho(|V|) + 1$ if not, and then we could find a hamiltonian cycle.