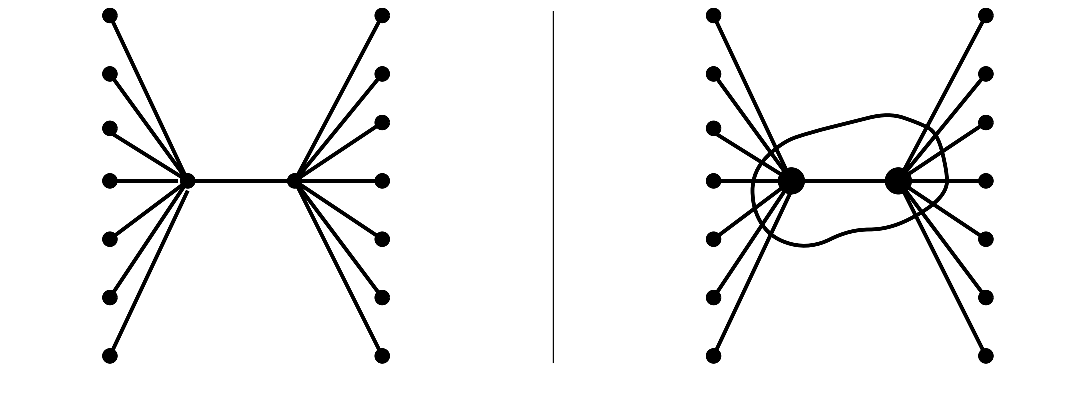

INPUT OUTPUT

16.3 Vertex Cover

Input description: A graph $G = (V, E)$.

Problem description: What is the smallest subset of $S \subset V$ such that each edge $(x, y) \in E$ contains at least one vertex of S?

Discussion: Vertex cover is a special case of the more general *set cover* problem, which takes as input an arbitrary collection of subsets $S = (S_1, \ldots, S_n)$ of the universal set $U = \{1, \ldots, m\}$. We seek the smallest subset of subsets from S whose union is U. Set cover arises in many applications associated with buying things sold in fixed lots or assortments. See Section 18.1 (page 621) for a discussion of set cover.

To turn vertex cover into a set cover problem, let universal set U represent the set E of edges from G, and define S_i to be the set of edges incident on vertex i. A set of vertices defines a vertex cover in graph G iff the corresponding subsets define a set cover in this particular instance. However, since each edge can be in only two different subsets, vertex cover instances are simpler than general set cover. Vertex cover is a relative lightweight among NP-complete problems, and can be more effectively solved than general set cover.

Vertex cover and independent set are very closely related graph problems. Since every edge in E is (by definition) incident on a vertex in any cover S, there can be no edge both endpoints are in $V - S$. Thus, $V - S$ must be an independent set. Since minimizing S is the same as maximizing $V - S$, the problems are equivalent. This means that any independent set solver can be applied to vertex cover as well. Having two ways of looking at your problem is helpful, since one may appear easier in a given context.

The simplest heuristic for vertex cover selects the vertex with highest degree, adds it to the cover, deletes all adjacent edges, and then repeats until the graph is empty. With the right data structures, this can be done in linear time, and should "usually" produce a "pretty good" cover. However, this cover might be $\lg n$ times worse than the optimal cover for certain input graphs.

Fortunately, we can always find a vertex cover whose size is at most twice as large as optimal. Find a *maximal* matching M in the graph—i.e. , a set of edges no two of which share a vertex in common and which cannot be enlarged by adding additional edges. Such a maximal matching can be constructed incrementally, by picking an arbitrary edge e in the graph, deleting any edge sharing a vertex with e, and repeating until the graph is out of edges. Taking *both* of the vertices for each edge in a maximal matching gives us a vertex cover. Why? Because *any* vertex cover must contain *at least* one of the two vertices in each matching edge just to cover the edges of M, this cover is at most twice as large as the minimum cover.

This heuristic can be tweaked to perform somewhat better in practice, if not in theory. We can select the matching edges to "kill off" as many other edges as possible, which should reduce the size of the maximal matching and hence the number of pairs of vertices in the vertex cover. Also, some of the vertices from M may in fact not be necessary, since all of their incident edges might also have been covered using other selected vertices. We can identify and delete these losers by making a second pass through our cover.

The vertex cover problem seeks to cover all edges using few vertices. Two other important problems have similar sounding objectives:

- *Cover all vertices using few vertices* – The *dominating set* problem seeks the smallest set of vertices D such that every vertex in $V - D$ is adjacent to at least one vertex in the dominating set D. Every vertex cover of a nontrivial connected graph is also a dominating set, but dominating sets can be much smaller. Any single vertex represents the minimum dominating set of complete graph K_n, while $n-1$ vertices are needed for a vertex cover. Dominating sets tend to arise in communications problems, since they represent the hubs or broadcast centers sufficient to communicate with all sites/users.

  Dominating set problems can be easily expressed as instances of set cover (see Section 18.1 (page 621)). Each vertex v_i defines a subset of vertices consisting of itself plus all the vertices it is adjacent to. The greedy set cover heuristic running on this instance yields a $\Theta(\lg n)$ approximation to the optimal dominating set.

- *Cover all vertices using few edges* – The *edge cover* problem seeks the smallest set of edges such that each vertex is included in one of the edges. In fact, edge cover can be solved efficiently by finding a maximum cardinality matching (see Section 15.6 (page 498)) and then selecting arbitrary edges to account for the unmatched vertices.

Implementations: Any program for computing the maximum clique in a graph can be applied to vertex cover by complementing the input graph and selecting the vertices which do not appear in the clique. Therefore, we refer the reader to check out the clique-finding programs of Section 16.1 (page 525).

COVER [RHG07] is a very effective vertex cover solver based on a stochastic local search algorithm. It is available at *http://www.nicta.com.au/people/richters/*.

JGraphT (*http://jgrapht.sourceforge.net/*) is a Java graph library that contains greedy and 2-approximate heuristics for vertex cover.

Notes: Karp [Kar72] first proved that vertex-cover is NP-complete. Several different heuristics yield 2-approximation algorithms for vertex cover, including randomized rounding. Good expositions on these 2-approximation algorithms include [CLRS01, Hoc96, Pas97, Vaz04]. The example that the greedy algorithm can be as bad as $\lg n$ times optimal is due to [Joh74] and presented in [PS98]. Experimental studies of vertex cover heuristics include [GMPV06, GW97, RHG07].

Whether there exists a better than 2-factor approximation for vertex cover is one of the major open problems in approximation algorithms. Hastad [Has97] proved there does not exist a better than 1.1666-factor approximation algorithm for vertex cover.

The primary reference on dominating sets is the monograph of Haynes et al. [HHS98]. Heuristics for the connected dominating set problem are presented in [GK98]. Dominating set cannot be approximated to better than the $\Omega(\lg n)$ factor [ACG$^+$03] of set cover.

Related Problems: Independent set (see page 528), set cover (see page 621).