The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 7, Problem 7.27

1. The problem is clearly in NP. To prove that it is NP-complete, we transform *X3C* to it. Given an instance S, $|S| = 3q$, $C \subset 2^S$, $(c \in C \Rightarrow |c| = 3)$ of *X3C*, we construct an instance $G = (V, E)$, $K \leq |V| - 1$, $w: E \to \Re$ of *BDST* as follows. We set $V = S \cup \{z_0, z_1, z_2\} \cup \{x_1, \dots, x_{|C|}\}$ (where the x_is are intended to represent 3-sets), and $E = \{\{a_i, x_i\}, \{b_i, x_i\}, \{c_i, x_i\} | \{a_i, b_i, c_i\} \in C\} \cup \{\{z_0, x_i\} | 1 \leq i \leq |C|\} \cup \{\{x_i, y_j\} | x_i, y_j \in C\} \cup \{\{z_0, z_1\}, \{z_1, z_2\}\}$. Then we set the maximum diameter to 4 and assign unit cost to all edges except the $\{z_0, x_i\}$ edges which get assigned a cost of 3; the bound on the total cost is then set to $5q + |C| + 2$. This transformation is clearly polynomial-time.
   The bound is easily reached with an exact cover by using a spanning tree with the root at z_0, with the solutions 3-sets as the first-level children, and with the set elements, as well as the discarded 3-sets, as grandchildren. Conversely, the diameter of 4 drastically limits the type of solution that can be obtained in our problem; in fact, since the eccentricity of z_0 is exactly 2 (due to the "tail" of z_is and the fact that it is also 2 edges away from the set elements), we must have z_0 as center ("root") of the solution tree. The required cost allows at most q edges of cost 3, which then designate the solution sets, since all set elements must be reachable from these q vertices.
   Another reduction starts from *1in3SAT*. The construction is exactly as in the text's reduction from *NAE3SAT* to *G3C*, except that an extra "tail" of 3 vertices is attached to the "root" of the construction. All edges are given length 1 or 2: the edges in the tail, those forming the clause triangles, and those connecting the complemented and uncomplemented versions of each variable are assigned unit lengths; the others are assigned length 2. The diameter bound is then set to 6 and the length bound to $4k + 3n + 3$ (for n variables and k clauses). The tail ensures that the eccentricity of the "root" is 3, so that the vertices of the clause triangles must be reachable from said root in

no more than 3 edges—which prevents any possibility of reaching the vertices of one triangle from a vertex of another. Spanning the tail will contribute 3 to the total length, leaving $4k + 3n$. Now reaching all three vertices of a clause triangle can be done in one of three ways: (i) through 3 edges from the variables triangles, for a length contribution of $3 \cdot 2 = 6$; (ii) through 2 such edges plus a triangle edge, for a length contribution of $2 \cdot 2 + 1 = 5$; or (iii) through one such edge plus two triangle edges, for a length contribution of $2 + 2 \cdot 1 = 4$. If all clauses are spanned in this last manner (corresponding to one literal per clause being true—that touched by the edge from the variable triangles), the contribution to the total length reaches a minimal value of $4k$. Finally, the variable triangles must be spanned by one edge from the "root" (of length 2)—indicating the true literal—and one from the true literal to the false (of length 1); any other possibility contributes more than $3n$ to the total length. Notice that each path from the "root" to the clause vertices must go through the true literals, since otherwise there would be paths of 4 edges ("root" to true literal, true literal to false, false literal to first clause vertex, first clause vertex to other clause vertex) and such a path would result in a tree diameter of 7. Thus we have a direct correspondence between the true literal of each clause and a truth assignment.

Yet another construction is from *3SAT*; similar to above, but each clause is simply a single vertex. The diameter is set to 4 and the tail has only two edges. The edge connecting literal and complement for each variable has length 1; the edges connecting these pairs to the "root" have length 2; the edges connecting to the clause vertices have length 3. The bound is set to 3*(#vars+#clauses). The proof follows easily.

2. We restrict this problem to *Hamiltonian Path* by restricting its instances to those where D is set to 2. Since every node in the tree must have degree 2 or less, the tree is in fact just a simple path; since it is a spanning tree, it is a Hamiltonian path.

3. This is the other variant of the spanning tree problem that is not just a Hamiltonian path in disguise. Fortunately, we can reuse the construction used in Exercise 7.26 (Steiner tree).

   One construction is exactly the same as in our solution for Exercise 7.26; the bound is set to $K = 2q + p$, where $n = 3q$ is the number of elements and p the number of 3-sets in the instance of *X3C*. We claim that the "root" of the constructed graph must be the "root" of the spanning tree. If it were not, all 3-set vertices would have to be reached through other 3-set vertices and their elements; but then we cannot have enough leaves in all. (The "best" case—although one in which there is no solution in either instance—has one element belonging to every 3-set; reaching the other $3q - 1$ elements requires $\lceil (3q-1)/2 \rceil$ 3-set elements; the resulting maximal number of leaves is $(3q-1) + (k - \lceil (3q-1)/2 \rceil) + 1$, which simplifies to $k + 3q - \lceil (3q-1)/2 \rceil$, which is smaller than $k + 2q$ for all $q > 1$. For $q = 1$, the values are equal, but that number is only reachable with a no-instance, since all 3-sets overlap.) (This whole reasoning could easily be avoided by endowing the "root" with a "tail," as in many similar constructions.) This being established, only one other observation need be made: any edge from an element to a 3-set is advantageously replaced by an edge from the "root" to the 3-set: such a change does not impair the spanning property and can only increase the number of leaves. The proof follows easily.

An alternate construction starts from *3SAT*. For each variable x, set up 3 vertices, say x, x_{mid}, and $\overline{x}$, connected in that order by two edges. Set up an additional vertex (root R) and connect it to all vertices corresponding to complemented and uncomplemented literals ($2n$ edges). Finally, set up one vertex for each clause and connect it to the 3 corresponding literal vertices. Set K to the number of clauses plus twice the number of variables. Since the x_{mid} vertices have degree 2, one of the x and $\overline{x}$ vertices cannot be a leaf; the proof follows easily.

4. We restrict this problem to *Hamiltonian Path* by restricting its instances to those where the bound is set to 2. A tree that has only two leaves (two nodes of degree 1) is a simple path; since it is a spanning tree, it is a Hamiltonian path.

5. We restrict this problem to *Hamiltonian Path* by restricting its instances to those where the subset of leaves has exactly two leaves. A tree with two leaves, as just remarked, is a simple path; thus any solution tree is a Hamiltonian path between the two specified nodes.

6. We restrict this problem to *Hamiltonian Path* by restricting its instances to those where the tree is in fact a simple path of $|V| - 1$ edges. Such a tree has exactly the number of edges needed for a spanning tree, but it is a path and thus the solution must be a Hamiltonian path.