The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 7, Additional Problem 20

Note that the problem is in P if all a_{ii} are at most 2.

Membership in NP is non-trivial, because the entries can be arbitrarily large and yet encoded succinctly, so that we cannot simply guess the sets in the collection. The key to membership in NP is also the key to part of the transformation: since at most $n(n-1)/2$ different intersections exist, it is sufficient to consider at most that many different elements; any larger intersections must arise from simple copying. Thus we need only establish "elements" x_{ij} and guess which of these are contained in each C_i and in how many copies. Verifying that the intersections have the proper size is then trivial.

We transform *G3C* to our problem as follows. The basic tool we have in our problem is setting certain entries to 0 to force disjointness and setting certain others to non-zero values to force overlap. Both are necessary (although they can be scaled up by adding the same quantity to all entries). We shall use one C_i, say C_0, to denote the set of available colors; since there are 3 colors, a_{00} will be set to 3. The other C_i will be used to denote the coloring of each vertex; thus there will be n of them, one per vertex. We force each to intersect with C_0 in exactly one element (the color); thus $a_{i0} = a_{0i} = 1$ for all $1 \leq i \leq n$. Whenever two vertices are connected by an edge, they cannot share a color; thus we set $a_{ij} = a_{ji} = 0$ whenever there is an edge in the original graph between vertices i and j. When two vertices are not connected, they could share a color; thus we set $a_{ij} = a_{ji} = 1$ whenever there is no edge between vertices i and j. However, setting a_{ij} to 1 *forces* overlap between the sets C_i and C_j, whereas the correct coloring may or may not assign the same color to vertices i and j: thus other means of intersection must be provided. This shows us how to assign values to the $a_i i$ for $i > 0$: such values must be large enough to allow the corresponding C_i to have a large number of elements for possible intersections. More specifically, we could have up to half the vertices in the graph colored in the same color, yet not connected to each other nor connected to some fixed vertex colored in another color. The corresponding sets must all intersect; those with the same color intersect by their color, but each must otherwise be disjoint from the others while at the same time intersecting with a common set. This indicates that a minimum value is $\lfloor n/2 \rfloor$; picking n will put all doubt to rest. (To reduce the value, one needs to start with a graph of restricted degree; we know that *GC* remains NP-complete for graphs of degree 3 and thus could use this special case for the transformation, leading to diagonal values of just 3.) The transformation is easily accomplished in polynomial time.

The construction ensures that a solution to our problem implies the existence of a solution to the coloring problem: each vertex gets assigned a single color chosen from a set of three and two vertices connected by an edge have different colors since their sets do not intersect. Conversely, given a legal coloring we can construct the sets C_i piece piece as follows. C_0 has three elements, call them r, g, and b (the colors). Each C_i includes exactly one of these three elements, as indicated by the coloring. Now remove from the matrix the $i0$ and $0i$ elements, subtract one from each ij entry whose corresponding intersection will include one of the color elements and consider the resulting matrix. It is an $n \times n$ matrix; its diagonal entries are all $n - 1$ and its off-diagonal entries are 0 or 1. We need $C_1, \ldots, C_n$, each of size $n - 1$, to provide the intersection pattern. Think of each as a vector; we use the jth entry of C_i to provide an intersection with C_j if necessary (by giving both the same jth element); otherwise, we assign all distinct elements. This provides a solution. Hence we have a solution to the original instance if and only if we have one for the transformed instance.

An alternative solution starts with *Positive 1in3SAT* and creates a matrix of size $m + n + 1$ from an instance with m clauses on n variables. The entries for the first $m + n$ rows and columns are the obvious ones: an entry a_{ij} where i and j denote two variables is set to 0 (the variables cannot share their own truth assignment); one that denotes two clauses is set to the number of literals the two clauses share; and one that denotes a variable and a clause is set to corresponding to presence or absence of a variable from a clause or to the intersection of two clauses. The diagonal elements for the variables are set to 2 (not 1), in order to allow the two truth values; the diagonal elements for the clauses are set to 3. The $m + n + 1$st row and column serve as enforcer; all of their elements are set to 1, except for the bottommost, rightmost element, which is set to n; this row and column correspond to the truth assignment.

The intent is to have as sets collections built from a total of $2n$ elements, namely the elements true(i) and false(i), for each $1 \le i \le n$. The set for variable i is {true(i), false(i)}; the set for a clause contains 3 elements, the true version of each variable in the clause; and the set for the enforcer is the truth assignment, i.e., one of each pair for each variable. The proof follows easily.