The Theory of Computation

Bernard M.E. Moret

Addison-Wesley-Longman, 1998

Solution Manual
Chapter 8, Problem 8.17

We prove the problem to be NP-complete with just 2 currencies, then show how to add any fixed number of currencies without affecting the result.

Note that proving it complete with 2 currencies is a stronger result than proving it complete for a floating number of currencies; we can do this last trivially by mapping *VC* as follows: for each vertex there will be a box; for each edge there will be a currency; each box contains one unit of currency for each currency that corresponds to an edge incident to the corresponding box; and the requirement is one unit of each currency—i.e., cover each edge.

We transform *1in3SAT* to our problem restricted to two currencies. Our transformation is essentially the same as for *Partition*; indeed, one could view the transformation as one from *Partition*, but a form of it where the *number* of items on each side of the partition is known in advance. Recall our transformation (in the text) from *1in3SAT* to *Partition*. Each literal gives rise to one object, hence $2n$ objects in all (plus the enforcer object); the size of an object is determined by the identity of the literal and the clauses in which it appears; let the size of object i be denoted by $s(i)$. We saw that the sum of the sizes of the $2n$ elements is an odd number, call it S; we made this number even by adding an enforcer with appropriate size—let us call the even value thus obtained S'.

We create an instance of our problem with $2n$ safe deposit boxes. Each box corresponds to a literal and contains two currencies; specifically, box i contains $s(i)$ units of currency 1 and $2S - s(i)$ units of currency 2. We set the upper bound, k, on the number of boxes to be opened to n and set the target to be $\frac{1}{2}S'$ units of currency 1 and $2nS - \frac{1}{2}S'$ units of currency 2.

Note that, in order to reach the bound on currency 2, we must open at least n of the $2n$ boxes (each box contributes at most $2S$ and we must reach a total larger than $2(n-1)S$); since $k = n$, this means that we must open exactly n boxes. But then we must open n boxes such that the sum of the $s(i)$ for these boxes is at least $\frac{1}{2}S'$ (currency 1) and at most $\frac{1}{2}S'$ (to achieve the total for currency 2), i.e., such that the sum of the $s(i)$ for these boxes is exactly $\frac{1}{2}S'$, which, as we have seen in the transformation to *Partition*, corresponds exactly to the false literals in a solution to *1in3SAT*.

To add a new currency, just place one unit of that currency in each box and set the requirement for that currency to n—we know that exactly n boxes will be opened in any solution, so this new currency does not alter the solution in any way.