Project Report CS6363

Sxp175331, Sushrut Pathaik.

Running type analysis of the DP for the maximum profit and number of optimal solutions is

$O(n \cdot G \cdot x)$ where $x = \max_{1 \le i \le n} x_i$

Recurrence for Max profit

$$jwlr(i,g) = \begin{cases} 0 & , i=0, g>=0 \\ \max_{0 \le q_i \le x_i} \{ q_i P_i + jwlr(i-1, g - q_i w_i) - \min(C_i, f_i(n_i - q_i)) \} \end{cases}$$

Proofs of correctness:-

feasibility :- There exists a solution with value $r(n)$

Proof — Feasible by induction on i (no. of items)

Base case $i=0 \Rightarrow jwlr(i,g) = 0$

Step $\Rightarrow i-1 < i$

By IH, $jwlr(i-1, g^*)$ is feasible $g^* > 0$.

$\downarrow$

$g - q_i w_i$

Optimality

Let opt(i,g) be an optimal solution of 'i' items and 'g' gold.

Claim :- $jwlr(i,g) \geq opt(i,g)$

Base :- When $i=0$, $jwlr(0,g) = 0$ $\{g_r=0\}$

$opt(i,j) = opt(0,g) = 0$.

No profit when number of items is zero.

Step we can check the cases where $i>0$.

$q_i = q_{opt}$

$$jwlr(i,g) = \max_{0 \leq q_i \leq n_i} \left(q_i p_i + jwlr(i-1, g - q_i w_i) \right)$$

$$= \min(c_i, f^i(n - q_i))$$

$$= q_{opt} p_i + jwlr(i-1, g - q_{opt} w_i) - \min(c_i, f^i(n - q_{opt}))$$

$jwlr(i-1, g - q_{opt} w_i) = opt(i-1, g - q_{opt} w_i)$

By I.H.

$\therefore jwlr(i,g) \geq opt(i,g)$

Recurrence for Count (Max)

$$\text{max\_count}(x_i, g) = \sum \text{count}(x_{i-1}, g - w_i q)$$
$$0 \le q \le \min\left(x_i, \frac{g}{w_q}\right)$$

The recurrence is feasible and optimal on i. (increasing number of items)

The total count for 'i' items will be the sum of the count of the $(i-1)$th items.

Pseudocode for DP

```
Process(int G, Jewel [] items, int n)
{
  profit_max = Most. int value.
  over_count [n+1][0 ... G+1], jwlr[0...n+1]
                                  [0...G+1]
  ctr
  for i: 1 to n                                   n times
    for g = 0 to G                                G times
      over_count[0][g] = 1
      profit_max = Most. int. value.
      for q = 0 to xi                             Q times
        if (g - (q * wi) < 0)
          break;
        if (q < ni)
          profit = q * Pi + jwlr[i-1]
                   [g - q * wi] -
                   Min[a; fi(ni - q)]
        else
          profit = q * Pi + jwlr[i-1]
                   [g - q wi]
        if (Profit < max < profit) {
          profit_max = Profit
          ctr = over_count[i-1]
                [g - q * wi]
        }
```

O(n · G · xi)
where xi = xi max.

```
else if (profit_max == profit)
    ctr = ctr + arr_count[i-1][j-q*w_i]
}
}
arr_count[i][j] = ctr ;
final[i][j] = profit_max
return (profit_max, ctr).
```