**Example 1: Solve the instance of 0/1 knapsack problem using dynamic Programming : n = 4, M = 25, (P1, P2, P3 P4) = (10, 12, 14, 16), (W1, W2, W3, W4) = (9, 8, 12, 14)**

**Solution:**

Knapsack capacity is very large, i.e. 25, so tabular approach won’t be suitable. We will use the set method to solve this problem

Initially, S0 = { (0, 0) }

**Iteration 1:**

Obtain S10 by adding pair (p1, w1) = (10, 9) to each pair of S0

S10 = S0 + (10, 9) = {(10, 9)}

Obtain S1 by merging and purging S0 and S10

S1 = MERGE\_PURGE (S0, S10)

= { (0, 0), (10, 9) }

**Iteration 2:**

Obtain S11  by adding pair (p2, w2) = (12, 8) to each pair of S1

S11 = S1 + (12, 8) = {(12, 8), (22, 17)}

Obtain S2 by merging and purging S1 and S11

S2 = MERGE\_PURGE(S1, S11)

= { (0, 0), (12, 8), (22, 17) }

Pair (10, 9) is discarded because pair (12, 8) dominates (10, 9)

**Iteration 3**:

Obtain S12 by adding pair (p3, w3) = (14, 12) to each pair of S2

S12 = S2+ (14, 12)

= { (14, 12), (26, 20), (36, 29) }

Obtain S3 by merging and purging S2 and S12 .

S3 = MERGE\_PURGE (S2, S12 )

= { (0, 0), (12, 8), (22, 17), (14, 12), (26, 20) }

Pair (36, 29) is discarded because its w > M

**Iteration 4:**

Obtain S13 by adding pair (p4, w4) = (16, 14) to each pair of S3

S13 = S3+ (16, 14)

= { (16, 14), (28, 22), (38, 31), (30, 26), (42, 34) }

Obtain S4 by merging and purging S3 and S13.

S4 = MERGE\_PURGE (S3, S13)

= { (0, 0), (12, 8), (14, 12), (16, 14), (22, 17), (26, 20), (28, 22) }

Pair (38, 31), (30, 26) ,and (42, 34) are discarded because its w > M

**Find optimal solution**

Here, n = 4.

Start with the last pair in S4, i.e. (28, 22)

(28, 22) ∈ S4 but (28, 22) ∉ S3

So set xn = x4 = 1

Update,

p = p – p4 = 28 – 16 = 12

w = w – w4 = 22 – 14 = 8

n = n – 1 = 4 – 1 = 3

Now n = 3, pair (12, 8) ∈ S3 and (12, 8) ∈ S2

So set xn = x3 = 0

n = n – 1 = 3 – 1 = 2

Now n = 2, pair(12, 8) ∈ S2 but (12, 8) ∉ S1

So set xn = x2 = 1

Update,

p = p – p2 = 12 – 12 = 0

w = w – w2 = 8 – 8 = 0

Problem size is 0, so stop.

Optimal solution vector is (x1, x2, x3, x3) = (0, 1, 0, 1) Thus, this approach selects pair (12, 8) and (16, 14) which gives profit of 28.

**Example 2: Generate the sets Si, 0 ≤ i ≤ 3 for following knapsack instance. N = 3, (w1, w2, w3) = (2, 3, 4) and (p1, p2, p3) = (1, 2, 5) with M = 6. Find optimal solution.**

**Solution:**

Initially, S0 = {(0, 0) }

**Iteration 1:**

Obtain S10 by adding pair (p1, w1) = (1, 2) to each pair of S0

S10 = S0 + (1, 2) = {(1, 2)}

Obtain S1 by merging and purging S0 and S10

S1 = MERGE\_PURGE (S0, S10)

= { (0, 0), (1, 2) }

**Iteration 2**:

Obtain S11 by adding pair (p2, w2) = (2, 3) to each pair of S1

S11 = S1 + (2, 3) = {(2, 3), (3, 5)}

Obtain S2 by merging and purging S1 and S11

S2 = MERGE\_PURGE(S1, S11)

= { (0, 0), (1, 2), (2, 3), (3, 5) }

**Iteration 3**:

Obtain S12 by adding pair (p3, w3) = (5, 4) to each pair of S2

S12 =  S2+ (5, 4) = {(5, 4), (6, 6), (7, 7), (8, 9) }

Obtain S3 by merging and purging S2 and S12 .

S3 = MERGE\_PURGE (S2, S12)

= { (0, 0), (1, 2), (2, 3), (5, 4), (6, 6) }

Pair (7, 7) and (8, 9) are discarded because their w > M

Pair (3, 5) is discarded because pair (5, 4) dominates (3, 5)

**Find optimal solution**:

Here, n = 3.

Start with the last pair in S3, i.e. (6, 6)

(6, 6) ∈ S3 but (6, 6) ∉ S2

So set xn = x3 = 1

Update,

p = p – p3 = 6 – 5 = 1

w = w – w3 = 6 – 4 = 2

Now n = 2, pair(1, 2) ∈ S2 and (1, 2) ∈ S1

So set xn = x2 = 0

Now n = 1, pair(1, 2) ∈ S1 but (1, 2) ∉ S0

So set xn =   x1 = 1

Optimal solution vector is (x1, x2, x3) = (1, 0, 1)

Thus, this approach selects pair (1, 2) and (5, 4)