1 Model

m machines (indexed by i), n jobs indexed by j. Objective function is Total Weighted Completion time so we can suppose that the jobs are already sorted in the order $\left(\frac{w_j}{p_j}\right)_j$.

Let a_k be the pattern k. It is a binary vector of length n; $a_{k,j} = 1 \Leftrightarrow$ job j is included in the pattern k.

$y_{i,k}$ is the binary decision variable: $y_{i,k} = 1 \Leftrightarrow$ we use pattern k for the machine i. The objective function is:

$$min \sum_i \sum_k C(a_k, y_{i,k})$$

with $C(a_k, y_{i,k}) = y_{i,k} \sum_j a_{k,j} w_j \sum_s^j a_{k,j} p_j^i$

This formulation of C can be justified by analogy to the following programs (that are equivalent):

```
def c1( A, Y, i, k):
    if Y[i,k] == 0:
        return 0

    res = 0
    for j in range(1, n):
        cumul_time = 0
        for s in range(1, j):
            cumul_time += A[k,j] * p[i,j]
        res += A[k, j]

def c_2( A, Y, i, k):
    if Y[i,k] == 0:
        return 0

    res = 0
    cumul_time = 0

    for j in range(1, n):
        cumul_time += A[k,j] * p[i,j]
        res += A[k,j * cumul_time * W[j]
```

Constraints are:

$$\forall i, \sum_k y_{i,k} = 1$$

$$\forall j, \sum_{i,k} a_{k,j} y_{i,k} = 1$$

2 Reduced cost

Let u_i, v_j be the dual variables. The reduced cost is: