

# Découpage optimal d'une barre : construire la solution

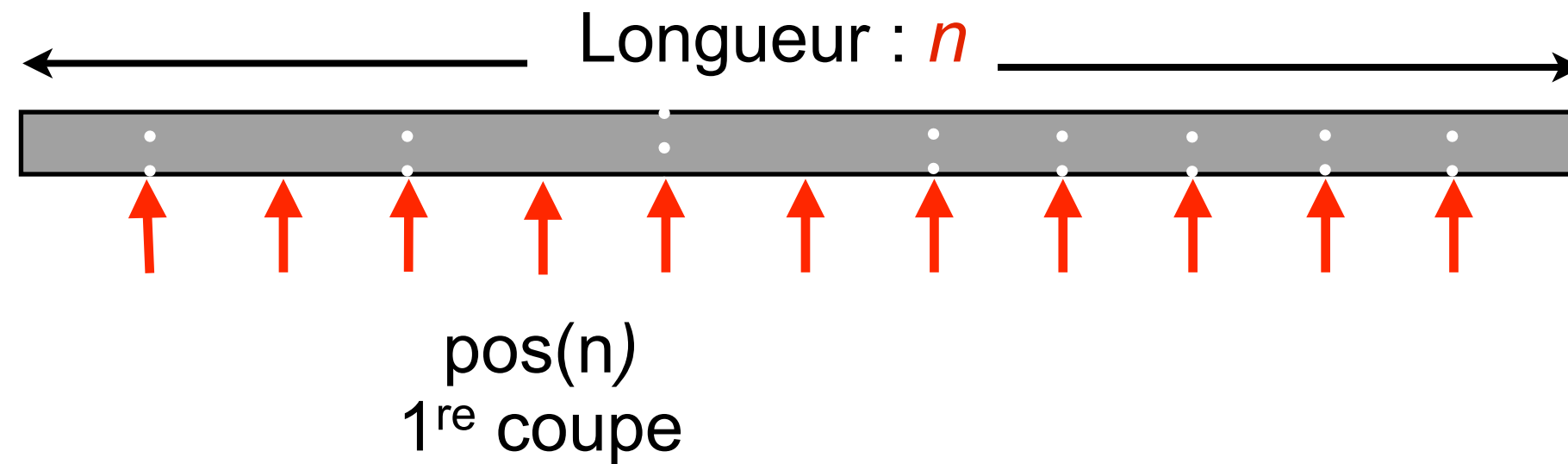
Conception et mise en œuvre d'algorithmes

Programmation Dynamique - leçon 5-3

Benjamin Werner

Ecole Polytechnique

# Retrouver la solution optimale



$\text{pos}(i)$  position de la 1re coupe pour atteindre l'optimum avec une barre de taille  $i$

$\text{pos}(i)$  est calculé simultanément à  $M(i)$

on garde le même code, mais on l'instrumente pour calculer aussi  $\text{pos}(i)$

# Trouver la bonne solution : code instrumenté

```
public static int maxDecoupeDyn(Integer n, int[] P) {
    int[] max = new int[n+1];
    int[] pos = new int[n+1];
    max[0] = 0;
    for (int i = 1; i <= n; i++) {
        max[i] = 0;
        pos[i] = -1;
        if (i < P.length) { max[i] = P[i]; }
        for (int j = 1; j < i; j++) {
            if (j < P.length && P[j] + max[i-j] > max[i])
                max[i] = P[j] + max[i-j];
            pos[i] = j;
        }
    }
    return(max[n]);
}
```

Complexités en temps et en espace restent inchangées dans ce cas

Résumons la situation

But : trouver la meilleure solution d'un problème (ici trouver le découpage donnant le prix maximum)

- On commence par trouver la *valeur* de la meilleure solution (le prix)
- Ensuite, on adapte le code pour trouver la solution correspondant à cette valeur (ici le découpage)

La programmation dynamique peut s'appliquer lorsque :

- L'on cherche un *optimum* (ici le meilleur prix)
- L'ensemble des solutions est indicé par des nombres entiers
- Le problème peut être ramené à des problèmes plus petits