République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Jijel
Faculté des Sciences Exactes et Informatique
Département d'Informatique

Optimisation Stochastique

Cours 1

Programmation Dynamique

Master I SIAD 2020/2021

Problèmes d'optimisation auxquels on s'intéresse

- Sont associés à des situations où des décisions doivent être prises de manière séquentielle:
  - ✓ On cherche une ***suite de décisions optimales***.
- Impliquent un système ***dynamique*** à ***temps discret***.
- Sur ***horizon fini***:
  - ✓ Le système évolue sur un nombre fini d' étapes.
- Impliquent une ***fonction coût*** *(à la base)* ***additive***:
  - ✓ c'est-à-dire que le coût s'**accumule** avec le temps.

Modèle de base

Un problème de **Programmation Dynamique Déterministe** implique un système dynamique en temps discret de la forme:

$$x_{k+1} = f_k(x_k, u_k) \quad k = 1,2, \dots, N$$

- k : numérote les périodes ou les étapes et N est l'***horizon ou*** le nombre de fois que doit prendre une décision
- x_k : état du système au début de la période k , $x_k \in S_k$. S_k est l'ensemble de tous les états possibles.
- u_k : décision (ou contrôle) devant être prise à la période k, $u_k \in C_k$. C_k est l'ensemble de toutes les décisions possibles.
- $\boldsymbol{f_k}$ **: fonction de transfert ou de transition qui décrit le mécanisme par lequel l'état est mis à jour de l' étape k à l' étape k + 1.**

Modèle de base (suite)

- Une fonction coût est associée à chaque période du système dynamique ***(coût intermédiaire)*** et qu'on dénote :

$$g_k(x_k, u_k)$$

- Il est possible de définir un ***coût terminal*** dépendant uniquement de l'état atteint à la fin du processus et qu'on dénote:

$$g_{N+1}(x_{N+1}).$$

- Pour un état initial x_1 , le ***coût total*** associé à une suite de décision u_1 , u_2 ,.... u_N est:

$$J(x_1; u_1 \,....\, u_N) = g_{N+1}(x_{N+1}) + \sum_{k=1}^{N} g_k(x_k, \mu_k(x_k))$$

Modélisation des problèmes d'optimisation en tant que des problèmes de PD

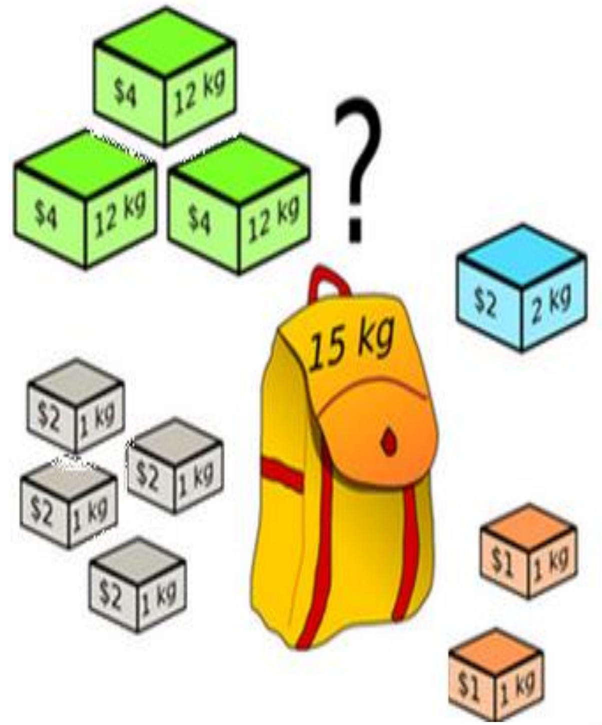
1. Identifier les décisions (u_k) à prendre et quand (k) ?
2. Choisir l'état (x_k) qui doit inclure toute information connu à l'étape k par le décideur et qu'il peut exploiter dans la prise de décision.
3. (…le reste se découle)

Exemple de Modélisation: problème de sac à dos

On veut charger un sac de volume ***b*** par des objets de différents types numérotés de 1 à N.

Chaque objet de type k est caractérisé par une **valeur** $c_k > 0$ entière et un **volume** $a_k > 0$ également entier.

on considère que l'on a plusieurs exemplaires de chaque objet.

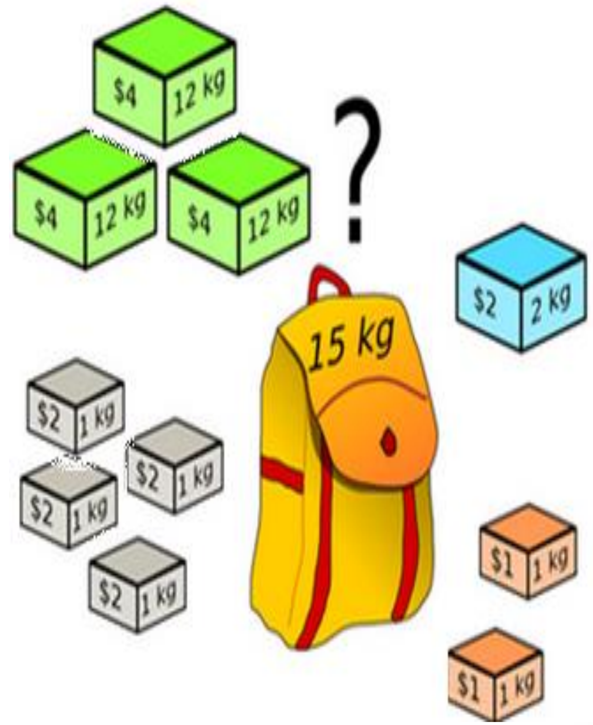


Exemple de Modélisation: problème de sac à dos

On veut charger un sac de volume ***b*** par des objets de différents types numérotés de 1 à N.

Chaque objet de type k est caractérisé par une **valeur** $c_k > 0$ entière et un **volume** $a_k > 0$ également entier.

on considère que l'on a plusieurs exemplaires de chaque objet.



Objectif:

Trouver le nombre (***entier***) d'exemplaires à prendre pour chaque type d'objet. La sélection doit être de **valeur maximale** ne dépassant pas le volume du sac

Exemple de Modélisation: problème de sac à dos

- $\boldsymbol{u_k}$: nombre d'objets de type k à inclure dans la sélection.
- Le nombre des étapes étant égal au nombre de décisions à prendre. Alors, les étapes seront numérotées de 1 à N (N : nombre des types des objets).
- Au début de l'étape k , afin de décider combien d'objets mettre, on dois savoir ***l'espace disponible après insertion des objets*** $\boldsymbol{1 \ldots k-1}$ **=** L'état x_k du système
- Chaque décision doit vérifier: $0 \leq u_k \leq \left\lfloor \frac{x_k}{a_k} \right\rfloor$.

Exemple de Modélisation: problème de sac à dos

- Si u_k objets de type k sont sélectionnés, l'espace disponible à l'étape k+1 est :

$$x_{k+1} = f_k(x_k, u_k) = x_k - a_k u_k, \quad \text{k=1,2,…,N}$$

- La valeur des objets sélectionnés à l'étape k est définie comme suit:

$$g_k(x_k, u_k) = c_k u_k$$

- Le **profit total** sera alors :

$$\sum_{k=1}^{N} g_k(x_k, u_k) = \sum_{k=1}^{N} c_k u_k$$

- Aucun **bénéfice terminal** n'est défini :

$$g_{N+1}(x_{N+1}) = 0$$

Modèle stochastique sur horizon fini

- A chaque étape k, on observe l'état x_k et on prend une décision u_k
- Puis, une variable aléatoire $\boldsymbol{\omega_k}$ est générée selon une loi de probabilité :

$$\boldsymbol{P_k(\omega_k | x_k, u_k)}$$

- On observe ω_k et on paye un coût:

$$\boldsymbol{g_k(x_k, u_k, \omega_k)}$$

- L'état à la prochaine étape est :

$$\boldsymbol{x_{k+1} = f_k(x_k, u_k, \omega_k)}$$

- On parlera alors **d'espérance** du coût total:

$$J(x_1; u_1 \ldots. u_N) = \boldsymbol{E_{\omega_k}}\left[g_{N+1}(x_{N+1}) + \sum_{k=1}^{N} g_k(x_k, u_k, \boldsymbol{\omega_k})\right]$$

Espérance Mathématique: Rappel

Considérons une variable aléatoire Y discrète prenant les valeurs y_i avec les probabilités p_i. L'espérance est définie comme-suit :

$$E(Y) = \sum_i p_i y_i$$

Exemple

On lance un dé équilibré à six faces, numérotées de 1 à 6. On gagne 6 u.m, si on obtient 1 ou 6 et on perd 2 u.m sinon.

$$\mathbf{P(Y = 6) = \frac{2}{6}}$$

$$\mathbf{P(Y = -2) = \frac{4}{6}}$$

$$\mathbf{E(Y) = \frac{2}{3} * (-2) + \frac{1}{3} * (6) = \frac{2}{3}}$$

Exemple: Gestion de Stock

- x_k: niveau du stock au début de l'étape k (i.e avant de commander).
- u_k: nombre de biens commandés (et reçus).
- ω_k : nombre de bien demandés par les clients durant l'étape k.

(on ne peut pas savoir quelle sera la demande clientèle avant de prendre notre décision)

- Fonction de transition:

$$\boldsymbol{x_{k+1} = x_k + u_k - \omega_k}$$

$$\boldsymbol{x_{k+1} = Max\,\{x_k + u_k - \omega_k, 0\}}$$

Algorithme de la PD déterministe

1. Construction de la politique optimale

a. Initialisation

$$J_{N+1} := g_{N+1}(x_{N+1}) \quad \forall\, x_{N+1} \in S_{N+1}$$

b. Pour $k = N, N-1, \dots, 1$ et pour tout $x_k \in S_k$ résoudre

$$J_k(x_k) := \min_{u_k \in U_k(x_k)} \{ g_k(x_k, u_k) + J_{k+1}(f_k(x_k, u_k)) \} \quad \dots\dots(*)$$

et stocker la valeur $J_k(x_k)$ ainsi que la décision $\mu_k^*(x_k) := u_k^*$ pour laquelle le minimum est atteint.

Fonction Coût Additive

2. Lecture de la solution optimale (pour l'état initial x_1)

a. Le coût optimal est $J^* = J_1(x_1)$.

b. La suite de décisions optimales est $u_1^* = \mu_1^*(x_1)$

Et $u_k^* = \mu_k^*(f_{k-1}(x_{k-1}, u_{k-1}^*)),\ k = 2,3,\dots,N$

Résolution du problème de sac à dos par l'algorithme de la PD

On cherche le chargement optimal d'un sac de ***volume 6***. Les caractéristiques des objets susceptibles d'être inclus dans ce sac sont données par le tableau suivant :

Objet K	1	2	3
Valeur c_k	5	3	7
Volume a_k	3	2	4