Algorithmique
M2 Data Science

cours 4 : Programmation dynamique
Packages R avec Rcpp
Etude en simulations

Vincent Runge

Laboratoire de Mathématiques et Modélisation d'Évry

LaMME

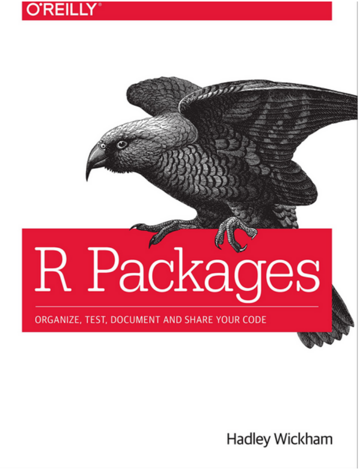
université evry val-d'essonne

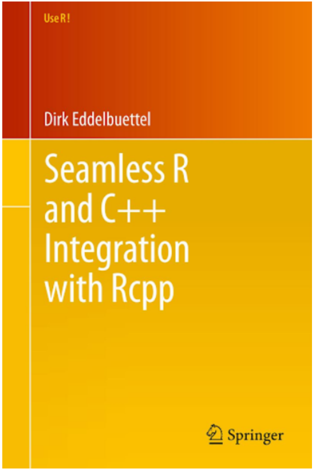
université PARIS-SACLAY

Section 1

R et Rcpp

R et Rcpp


O'REILLY
R Packages
ORGANIZE, TEST, DOCUMENT AND SHARE YOUR CODE
Hadley Wickham


Use R!
Dirk Eddelbuettel
Seamless R and C++ Integration with Rcpp
Springer

Voir aussi

https://github.com/vrunge/M2algorithmique

Section 2

Un 1er exemple

le problème du sac à dos (*knapsack problem*)

Le problème :

1. Nous avons n objets à transporter sur la station spatiale internationale
2. Chacun d'entre eux à une importance $v_i > 0$ et un poids $p_i \in \mathbb{N}^*$
3. Le lanceur est limité par un poids maximal de charge utile $M \in \mathbb{N}^*$

En d'autres termes:

$$\max \sum_{i=1}^{n} v_i x_i$$

sous la contrainte

$$\sum_{i=1}^{n} p_i x_i \leq M\,, \quad x_i \in \{0,1\}$$

Un exemple

Objet	Poids	Importance
1	160	45
2	60	25
3	65	25
4	20	30
5	40	10

Limite de poids de 80.

- Stratégie 1 : Choisir en priorité les objets les plus importants, le 1 puis le 4: résultat 75
- Stratégie 2 : Choisir en priorité les objets les moins lourds, le 4, le 5, puis 2 : résultat 65
- Stratégie 3 : Choisir en priorité des objets de poids moyen et d'utilité moyenne. Par exemple les objets 2, 3 et 4 : résultat 80

Programmation dynamique

On résout des sous-problèmes de solution optimale $f(l, m)$:

$$f(l, m) = \max \sum_{i=1}^{l} v_i x_i$$

sous la contrainte

$$\sum_{i=1}^{l} p_i x_i \leq m \,, \quad x_i \in \{0, 1\}$$

- On ne prend que l objets pour un poids maximal m
- On obtient une famille de $n(M+1)$ problèmes
- La solution recherchée est $f(n, M)$

Relation de récurrence

$$f(l,m) = \max \sum_{i=1}^{l} v_i x_i$$

$$\sum_{i=1}^{l} p_i x_i \le m\,, \quad x_i \in \{0,1\}$$

récurrence par ajout d'objet:

$$f(l+1,m) = \max\big(f(l,m), v_{l+1} + f(l, m - p_{l+1})\big)$$

Conditions aux bords (1 objet):

$$f(1,m) = \begin{cases} v_1 & \text{si } p_1 \le m \\ 0 & \text{sinon} \end{cases}$$

Section 3

student-project allocation

Un problème encore ouvert

Handling preferences in student-project allocation (2019)

https://link.springer.com/article/10.1007/s10479-017-2710-1

Notre problème avec $n = p$

On attribut un score de préférence s_{ij} pour l'étudiant i au projet j

$$\max \sum_{i=1}^{n} \sum_{j=1}^{n} s_{ij} x_{ij}$$

sous les contraintes

$$\sum_{j=1}^{n} x_{ij} = 1\,, \quad x_{ij} \in \{0,1\} \quad \text{un projet par étudiant}$$

$$\sum_{i=1}^{n} x_{ij} = 1\,, \quad x_{ij} \in \{0,1\} \quad \text{un étudiant par projet}$$

Il y a beaucoup de contraintes ! (solution possible (mais lente) par le *multiple knapsack*)

Par programmation dynamique?

On attribut un score s_{ij} pour l'étudiant i au projet j

$$\max \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}x_{ij} + \sum_{i=1}^{n} f(\sum_{j=1}^{n} x_{ij}) + \sum_{j=1}^{n} g(\sum_{i=1}^{n} x_{ij})$$

sous la contrainte

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} \leq n\,, \quad x_{ij} \in \{0,1\}$$

Les fonctions f et g sont choisies de telle sorte à avoir $f(1) = g(1) = 0$ et $f(x \neq 1)$ et $g(x \neq 1)$ très négatif.

MAIS l'algo de programmation dynamique ne s'applique pas ici...

Par programmation dynamique #2

On attribut un score s_{ij} pour l'étudiant i au projet j

$$\max \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}x_{ij} + \sum_{i=1}^{n} a_i(\sum_{j=1}^{n} x_{ij}) + \sum_{j=1}^{n} b_j(\sum_{i=1}^{n} x_{ij})$$

sous la contrainte

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} \leq n\,, \quad x_{ij} \in \{0,1\}$$

On peut appliquer la programmation dynamique **MAIS** peut-on trouver des coefficients (a_i, b_i) qui renvoit une solution satisfaisant les contraintes initiales?

Le TP

On attribut un score s_{ij} pour l'étudiant i au projet j

$$\max \sum_{i=1}^{n}\sum_{j=1}^{n} s_{ij}x_{ij} + \sum_{i=1}^{n} a_i(\sum_{j=1}^{n} x_{ij}) + \sum_{j=1}^{n} b_i(\sum_{i=1}^{n} x_{ij}) = \max \sum_{i=1}^{n}\sum_{j=1}^{n}(s_{ij}+a_i+b_j)x_{ij}$$

sous la contrainte

$$\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} \leq n\,, \quad x_{ij} \in \{0,1\}$$