

TP3: Randonnée et optimisation

Evan Voyles

May 25

```
library(tidyverse)
library(purrr)
library(tibble)
library(ggplot2)
library(pracma)
```

Un randonneur se trouve face à un dilemme classique : il aimerait emmener avec lui un maximum d'affaires qui pourraient lui servir mais ne souhaite pas alourdir inutilement son sac, qui ne doit pas dépasser un certain poids P . Il cherche donc à déterminer la combinaison d'affaire qui vérifierait le meilleur rapport utilité/légèreté. Pour ce faire, il associe à chacune de ses M affaires, son poids p_i et sa valeur v_i pour $i \in \{1, \dots, M\}$. L'objectif est de trouver la configuration qui maximise la valeur du sac à dos sans dépasser P .

1. On considère un vecteur σ (qui contient 0 ou 1 pour l'entrée i selon que l'objet i ait été choisi pour remplir le sac à dos). On choisit initialement un vecteur σ tel que

```
# A partir d'une configuration \sigma, proposer \sigma' en modifiant aleatoirement l'un des
# \sigma_i par (1 - \sigma_i)
propose_sigma_prime <- function(sigma) {

  n <- length(sigma)
  i <- sample(1:n, 1)

  sigma[i] <- 1 - sigma[i]
  sigma
}

# Return the total weight of the items carried
total_weight <- function(sigma, poids) {
  sum(sigma * poids)
}

total_valeur <- function(sigma, valeur) {
  sum(sigma * valeur)
}

# We'll use the transition matrix  $Q(\cdot | V) = N(V, 1)$ 
# En sachant que  $\pi(\sigma') / \pi(\sigma) = \exp(V_{\sigma'} - V_{\sigma})$ ,
# On implemente l'algorithme de metropolis hastings comme suivant:
taux_accept <- function(sigma, sigma_prime, valeur) {
  min(exp(total_valeur(sigma, valeur) - total_valeur(sigma_prime, valeur)), 1)
}
```

```
item <- c("map", "water", "sandwich", "glucose", "tin", "banana", "apple", "cheese",
         "beer", "suntan_cream", "camera", "T-shirts", "trousers", "umbrella",
```

```

      "waterproof_trousers", "waterproof_overclothes", "note-case", "sunglasses", "towel",
      "socks", "book")
p <- c(0.05, 1, 0.7, 0.1, 0.5, 0.2, 0.3, 0.4, 0.5, 0.3, 1, 0.8, 0.4, 0.7, 0.4, 0.3, 0.4, 0.1, 0.3, 0.4,
v <- c(150, 300, 160, 60, 45, 60, 40, 30, 180, 70, 30, 100, 10, 40, 70, 75, 50, 80, 12, 50, 30)
P <- 7

```