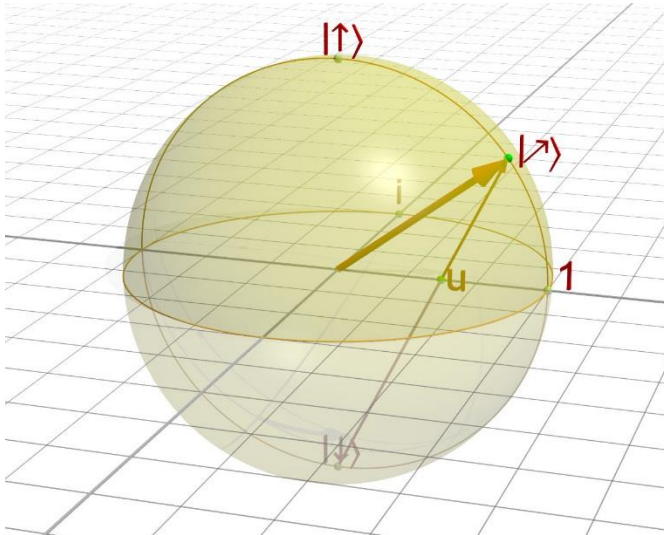


Le contexte



Etant donné **N spins de valeur $s = 1$ ou -1** , et en intégrant dans le modèle les interactions jusqu'à un **niveau K ($K \leq N$)**, le modèle stipule que les valeurs des N spins sont telles que la somme

$$\sum_{i=1}^{K-1} \left(\sum_{j=0}^{N-i-1} s_j \times s_{j+i} \right)^2$$

est **minimisée**

Le sujet

Données d'entrée : deux paramètres

- N : nombre de spins
- K : niveau maximal d'interaction considéré

Sortie : Une séquence (s_0, \dots, s_{N-1}) avec $s_k \in \{-1, 1\}$

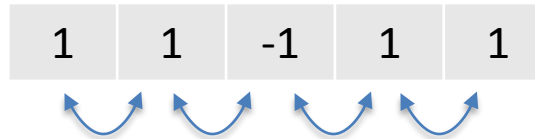
Score d'une séquence :

$$\sum_{i=1}^{K-1} \left(\sum_{j=0}^{N-i-1} s_j \times s_{j+i} \right)^2$$

→ Etant donné N et K, l'objectif est de trouver une séquence de **score minimum**

Illustration du calcul du score :

N = 5, K=2

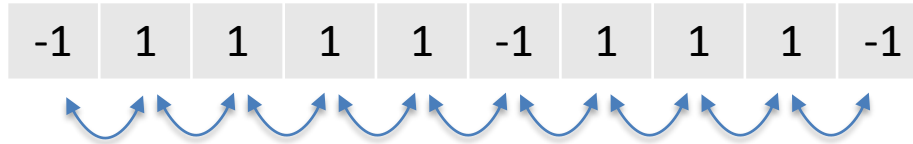


Interactions de niveau 2 : $1 + (-1) + (-1) + 1 = 0$

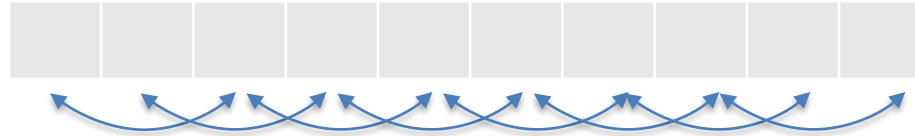
Score : 0

Le sujet

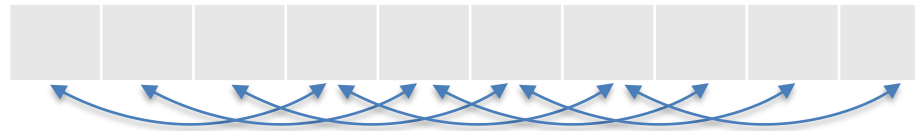
N = 10, K=5



Interactions de niveau 2 : $(-1) + 1 + 1 + 1 + (-1) + (-1) + 1 + 1 + (-1) = 1$



Interactions de niveau 3 : $(-1) + 1 + 1 + (-1) + 1 + (-1) + 1 + (-1) = 0$



Interactions de niveau 4 : $(-1) + 1 + (-1) + 1 + 1 + (-1) + (-1) = -1$

Interactions de niveau 5 : $(-1) + (-1) + 1 + 1 + 1 + 1 = 2$

Score : $1^2 + 0^2 + (-1)^2 + 2^2 = 1 + 0 + 1 + 4 = 6$

Les entrées / sorties

Entrées : un fichier unique **instances.txt** indiquant la liste des instances à résoudre

Format du fichier **instances.txt** :

N	K
5	2
10	5
20	10
40	20
100	50
...	...

Les entrées / sorties

Sorties : un fichier par instance résolue

Nom proposé pour les fichiers solutions : **res_N_K.csv**

Format du fichier **res_5_2.csv** :

Attention : le séparateur à utiliser dans les fichiers .csv est la virgule « , »

5	2			
1	1	-1	1	1

(contenu du fichier vu sous Excel)

```
5,2  
1,1,-1,1,1
```

(contenu du même fichier vu avec un éditeur de texte)

Algorithme de descente

```
définir une solution  $S = (s_0, \dots, s_{N-1})$  quelconque
best ← score de cette solution
i ← 0
fail ← 0
tant que fail < N
    modifier la valeur de  $s_i$ 
    si la nouvelle solution obtenue a un score meilleur que best alors
        mettre à jour best
        fail ← 0
    sinon
        revenir à la valeur précédente de  $s_i$ 
        fail ← fail + 1
    fin si
    i ← i+1
    si i = N alors
        i ← 0
    fin si
fin tant que
```

Evaluation

Note : moyenne des notes sur l'ensemble des instances

Note pour une instance :

- Aucune solution meilleure que RANDOM * n'a été transmise : 8
- Une solution de score compris entre RANDOM et DESC * a été transmise

Note proportionnelle à la qualité de la solution, sur l'intervalle [10,15]
(note 10 pour un score RANDOM, note 15 pour un score DESC)

- Une solution de score supérieur à DESC a été transmise

Note proportionnelle à la qualité de la solution, sur l'intervalle [15,20]
(note 15 pour un score DESC, note 20 pour la meilleure solution reçue)

* : RANDOM = score d'une séquence remplie aléatoirement
: DESC = score obtenu dans notre implémentation de l'algorithme de descente

Organisation

1. A effectuer seul
2. Se rendre sur : <https://emse.shinyapps.io/challenge-optim-2020/>
ou <https://emse.shinyapps.io/challenge-optim-2020-2/>
3. Créer son compte et se connecter
4. Choisir son langage (et son environnement)
5. Récupérer le fichier d'instances et ces transparents (onglet download)
6. Transmettre les fichiers solutions obtenus au fur et à mesure (*onglet upload*) jusqu'à **19h00**
7. A 19h00, déposer votre code (onglet upload code)

Les meilleurs résultats et les notes sont mises à jour tout au long de la journée

Organisation

Interaction avec les enseignants : à l'aide d'un canal Mattermost

Nom : ISMIN Cha-Opt

https://mattermost.emse.fr/signup_user_complete/?id=sm5m8ad9pjhbbm7hfcdfaigyc

Le canal permettra :

- des échanges publics
- des échanges privés avec les enseignants (MESSAGES PERSONNELS)

Pour contacter un enseignant, un flag (voir liste des membres par exemple) vous indiquera si les enseignants sont disponibles

Liste des enseignants et langages

Nabil ABSI (C, C++)

Maxime AGIUS (C, C++)

Oussama BEN-AMMAR (C, C++)

Valeria BORODIN (C, C++)

Dominique FEILLET (C, C++)

Elodie SUZANNE (C, C++)

Sébastien BERAUDY (Java)

Mario FLORES-GOMES (Python)

Etienne LE-QUERE (Python)

Thibault PRUNET (Python)

Sean SHORTEN (Python)

Le serveur

Onglet upload

Les fichiers solutions doivent être déposés soit un à un (format .csv) , soit par lot (format .csv), soit sous forme d'archive (format .zip)

Il n'est pas nécessaire d'envoyer d'un coup les fichiers de toutes les solutions

Pour chaque groupe et chaque instance, le meilleur score reçu est conservé

Quelques conseils

N'hésitez pas à solliciter de l'aide

Le challenge est noté mais les enseignants sont à votre disposition pour vous aider tout au long de la journée et répondre à vos questions, comme pour un TP

Quelques conseils

Ne mettez pas votre GP en péril

Si une note de 8 suffit pour votre GP, faites-vous plaisir.

Sinon, si vous êtes peu à l'aise en programmation, nous vous conseillons fortement de programmer en C et de suivre la méthodologie suivante :

1. Commencer par définir sur papier les structures de données qui seront utilisées
2. Faire un planning de développement, étape par étape
3. Ne pas hésiter à faire appel aux enseignants pour valider ces structures et ce planning
4. Nous consulter si vous constatez un retard inquiétant par rapport à ce planning



Bon challenge à tous !