

# Crazy Charly Day 2026 — Défi Informatique — ToyBoxing

Toys Academy

## Présentation

### Contexte

Le commanditaire de cette édition est **Toys Academy** ([Toys Academy](#)), une association qui collecte, trie, nettoie et revalorise les dons de jeux, jouets et livres pour enfants. La revente est faite à prix solidaires en boutique et en e-shop. L'association propose également des ateliers parentalité et de sensibilisation à l'environnement avec de l'upcycling de jouets.

Dans ce cadre, l'association souhaite proposer un nouveau service : des **toy box**. Le principe est de permettre à des abonnés de recevoir régulièrement une box contenant des jouets reconditionnés, sélectionnés en fonction de leurs préférences et de **l'âge de leur enfant**. Les articles étant issus de dons, ils sont par nature uniques et leur affectation dans les box doit être optimisée pour satisfaire au mieux les abonnés.

Le développement couvre 3 aspects : une application Web de gestion des articles et des abonnements, une optimisation de la composition des box, et un déploiement de la solution globale sur un serveur public.

### Objectifs et décomposition de l'application

Il s'agit de proposer un outil répondant au besoin complet : développement de l'application Web, optimisation de la composition des box et déploiement. Naturellement, vos parcours respectifs favorisent vos contributions à l'un de ces aspects. Cela étant, vous êtes encouragés à ne pas hésiter à dépasser les thématiques qui sont au cœur de vos parcours :

- les différentes tâches ne sont pas forcément de même importance,
- des fonctionnalités et des comportements attendus sont décrits ; vous avez cependant des choix à réaliser, des technologies à sélectionner et il est préférable de les envisager en concertation,
- les tailles d'équipes sont assez conséquentes : la communication, la collaboration et le transfert de compétences seront certainement décisifs pour garantir une bonne progression de votre projet !

### Dernières recommandations

Même si chaque partie fera l'objet d'une évaluation distincte, l'appréciation se fera également sur la chaîne complète de votre projet. Dans cette perspective, veillez à ne pas trop déséquilibrer ces parties, entraidez-vous. Le temps est limité, faites des choix

pragmatiques et efficaces en utilisant des principes et technologies que vous connaissez et que vous êtes **tous** capables d'appréhender au sein de l'équipe. Organisez-vous pour produire un prototype d'une chaîne complète !

## Partie « Application Web »

### Présentation générale des fonctionnalités

Les fonctionnalités concernent :

- les **abonnés** qui souscrivent au service de toy box, précisent la **tranche d'âge** de leur enfant et expriment leurs **préférences** sur les catégories de jouets souhaitées,
- les **gestionnaires** (responsables de l'association) qui administrent le catalogue d'articles, gèrent les abonnés et supervisent la composition des box *via* le back-office.

On y retrouve les caractéristiques attendues classiques d'une application Web : interfaces et expérience utilisateur, gestion de données, privilèges.

**Important** : la responsable de l'association n'a souvent qu'un téléphone à disposition. L'application doit donc être **responsive** et pleinement utilisable sur mobile.

L'esprit de cet événement est de produire un démonstrateur. Pour éviter de devoir gérer l'authentification, on pourra, dans un premier temps, disposer d'une URL pour le back-office, destinée aux gestionnaires (cette adresse ne nécessitera pas d'authentification). Il n'est pas utile de s'intéresser à la création de compte pour les abonnés : ils peuvent simplement saisir les quelques données les concernant à chaque nouvelle requête.

### Description détaillée des fonctionnalités

*On conseille de ne s'intéresser aux fonctionnalités étiquetées comme « avancées » qu'à partir du moment où l'application développée comporte déjà un ensemble de fonctionnalités de base cohérentes (formant une chaîne utilisable). Les fonctionnalités étiquetées « admin » ne doivent pouvoir être entreprises que par les utilisateurs disposant de ce privilège.*

#### Gestion des articles (**admin**)

Les articles (jouets, jeux, livres) sont issus de dons et sont donc uniques. Chaque article est décrit par : une désignation (texte libre), une **catégorie** parmi les six catégories proposées, une **tranche d'âge** cible, un **état** (Neuf, Très bon état, Bon état), un **prix** (en euros, entier) et un **poids** (en grammes, entier).

Les six catégories d'articles sont les suivantes :

- **SOC** — Jeux de société
- **FIG** — Figurines et poupées
- **CON** — Jeux de construction

- **EXT** — Jeux d'extérieur
- **EVL** — Jeux d'éveil et éducatifs
- **LIV** — Livres jeunesse

Les quatre tranches d'âge sont les suivantes :

- **BB** — 0-3 ans (bébé)
- **PE** — 3-6 ans (petit enfant)
- **EN** — 6-10 ans (enfant)
- **AD** — 10+ ans (adolescent)

Chaque article est destiné à **une seule tranche d'âge**. Cette information est essentielle car elle conditionne les articles pouvant être placés dans la box d'un abonné : **un article ne peut être affecté qu'à un abonné dont la tranche d'âge de l'enfant correspond à celle de l'article.**

Les trois états possibles sont :

- **N** — Neuf (article jamais utilisé ou comme neuf)
- **TB** — Très bon état
- **B** — Bon état

**1.** Ajout d'un article (*admin*). Saisie de la désignation, de la catégorie (parmi les six), de la tranche d'âge (parmi les quatre), de l'état, du prix et du poids. Un identifiant unique est attribué automatiquement à l'article.

**2.** Affichage du catalogue des articles. Présentation de la liste des articles disponibles (non encore affectés à une box) avec leurs caractéristiques. On souhaite un affichage paginé à raison de 10 articles par page.

**3.** Filtrage du catalogue (*avancé*). Possibilité de filtrer les articles par catégorie, par tranche d'âge et/ou par état.

**4.** Modification d'un article (*avancé, admin*). Interface de modification avec pré-remplissage des champs aux valeurs actuelles. Un article déjà affecté à une box validée ne peut pas être modifié.

**5.** Identification par code-barre (*avancé, admin*). Possibilité d'associer un code-barre (ou QR code) à un article lors de sa saisie, et de rechercher un article par scan ou saisie de ce code.

## Gestion des abonnés

**6.** Inscription d'un abonné. Saisie du nom, du prénom et de l'adresse email. L'abonné renseigne la **tranche d'âge de son enfant** (parmi BB, PE, EN, AD) ainsi que ses préférences en **ordonnant les six catégories** de jouets de la plus souhaitée à la moins souhaitée. Il n'est pas utile de s'intéresser à la création de compte : l'abonné saisit ses informations à chaque visite. Un cookie peut être utilisé pour identifier l'abonné pour réutiliser ses informations lors d'une prochaine visite.

**7.** Affichage de la liste des abonnés (*admin*). Présentation des abonnés existants avec la tranche d'âge de leur enfant et leurs préférences de catégories.

**8.** Modification des préférences (*avancé*). Un abonné peut modifier la tranche d'âge de son enfant et l'ordre de ses préférences de catégories (en renseignant son adresse email pour retrouver son profil).

### **Composition et gestion des box**

**9.** Paramétrage d'une *campagne* de box (*admin*). Avant de lancer la composition, le gestionnaire définit le paramétrage de la campagne : poids maximum par box (en grammes).

**10.** Lancement de la composition (*admin*). Envoi des données (articles disponibles, abonnés avec tranche d'âge et préférences, paramètres) à la brique logicielle d'optimisation (*cf.* partie « Optimisation » décrite plus loin). Il sera nécessaire d'être attentif à la communication mise en place (modalité, protocole), autant pour envoyer les données que pour récupérer les résultats.

**11.** Affichage des box composées (*admin*). Visualisation de chaque box proposée par l'optimisation : liste des articles avec leurs caractéristiques (catégorie, tranche d'âge, état), score de la box, poids total et prix total.

**12.** Validation des box (*avancé, admin*). Validation individuelle des box. Une fois validée, les articles de la box sont retirés du stock disponible et l'envoi est enregistré dans l'historique.

**13.** Consultation de sa box. Un abonné peut consulter la composition de la box qui lui est destinée (en renseignant son adresse email). Seules les box validées sont visibles.

### **Historique**

**14.** Historique des box d'un abonné (*avancé*). Liste des box précédemment reçues par un abonné avec le détail des articles.

**15.** Historique global (*avancé, admin*). Vue d'ensemble de toutes les campagnes de box réalisées avec des informations de synthèse (nombre de box, nombre d'articles distribués, score moyen).

### **Autres fonctionnalités**

**16.** Authentification (*avancé*). Implanter une authentification permettant de différencier un profil abonné et un profil gestionnaire.

**17.** Tableau de bord (*avancé, admin*). Statistiques : nombre d'articles en stock par catégorie et par tranche d'âge, répartition par état, nombre d'abonnés actifs par tranche d'âge, score moyen des dernières campagnes.

**18.** Bon de préparation (*super avancé*). Génération d'un document imprimable détaillant le contenu d'une box (articles, poids, prix) pour faciliter sa préparation physique en entrepôt.

**19.** Notification par email (*super avancé*). Envoi automatique d'un email aux abonnés lorsque leur box est prête. Un outil de type MailCatcher peut être intéressant dans ce cadre (pour information, l'envoi d'emails en PHP fonctionne sur webetu uniquement avec les emails univ-lorraine).

## Partie « Optimisation »

### Présentation générale du sujet

L'application d'optimisation récupère (*via* un répertoire partagé, un service web ou autre...) le catalogue des articles disponibles et la liste des abonnés avec la tranche d'âge de leur enfant et leurs préférences de catégories. Elle a pour objectif de **composer une box pour chaque abonné** en sélectionnant des articles du catalogue, de manière à maximiser la satisfaction globale tout en respectant les contraintes de compatibilité d'âge et de poids.

Chaque article est décrit par sa catégorie (parmi les six catégories : SOC, FIG, CON, EXT, EVL, LIV), sa **tranche d'âge** (BB, PE, EN ou AD), son état (N, TB, B), son prix et son poids. Chaque abonné a renseigné la **tranche d'âge de son enfant** et a exprimé un **ordre de préférence** sur les six catégories (de la catégorie la plus souhaitée à la moins souhaitée).

L'application d'optimisation va chercher à composer les box en raisonnant sur plusieurs aspects :

- elle va **respecter la compatibilité d'âge** : un article ne peut être placé que dans la box d'un abonné dont l'enfant correspond à la tranche d'âge de l'article ;
- elle va essayer de placer dans chaque box des articles correspondant aux catégories préférées de l'abonné ;
- elle va encourager la **variété** dans chaque box (éviter de ne mettre que des articles de la même catégorie) ;
- elle va respecter une contrainte de **poids maximum** par box ;
- elle va chercher une répartition **équitable** du nombre d'articles entre les abonnés.

L'application retournera des données (fichier, flux, ...) contenant la composition des box que votre algorithme d'optimisation aura calculée.

### Mesure de la qualité d'une composition

Afin d'optimiser la composition des box, on propose une mesure de la qualité d'une composition (appelée *score*) en fonction des préférences des abonnés et des caractéristiques des articles. L'objectif de l'optimisation sera donc de construire la composition qui **maximise** ce score.

Le score d'une composition repose sur plusieurs règles qui permettent de contraindre les compositions possibles et de donner un score élevé aux compositions qui respectent certains principes.

**Règle 1 (unicité des articles)** : chaque article ne peut apparaître que dans une seule box au plus. Tous les articles ne sont pas forcément utilisés.

**Règle 2 (compatibilité d'âge)** : un article ne peut être placé dans la box d'un abonné que si la tranche d'âge de l'article correspond **exactement** à la tranche d'âge de l'enfant de l'abonné. Par exemple, un article étiqueté PE (3-6 ans) ne peut être affecté qu'à un abonné ayant déclaré un enfant dans la tranche PE.

**Règle 3 (poids limité)** : le poids total des articles contenus dans une box ne doit pas dépasser le poids maximum  $W_{\max}$  défini pour la campagne.

**Règle 4 (gain par préférence de catégorie)** : chaque article placé dans la box d'un abonné rapporte un certain nombre de points au score global. Ce nombre de points dépend de la position de la catégorie de l'article dans l'ordre de préférence de l'abonné :

Ordre de préférence	1 <sup>er</sup>	2 <sup>nd</sup>	3 <sup>ème</sup>	4 <sup>ème</sup>	5 <sup>ème</sup>	6 <sup>ème</sup>
Points octroyés	10	8	6	4	2	1

Ainsi, si Alice a classé la catégorie « Jeux de société » (SOC) en première position et qu'un article SOC compatible avec la tranche d'âge de son enfant est placé dans sa box, cette affectation rapporte 10 points.

**Règle 5 (bonus d'état)** : en plus des points de préférence, chaque article apporte un bonus lié à son état :

- état **N** (Neuf) : **+2** points
- état **TB** (Très bon état) : **+1** point
- état **B** (Bon état) : **+0** point

**Règle 6 (utilités dégressives)** : pour encourager la variété dans les box, si plusieurs articles de la **même catégorie** sont placés dans une même box, les points diminuent pour chaque article supplémentaire de cette catégorie.

Pour cela, on classe les articles d'une même catégorie présents dans la box par ordre d'ajout :

- le premier article de cette catégorie rapporte les points correspondant au tableau de la règle 4 ;
- le deuxième article de la même catégorie rapporte les points correspondant au rang de préférence **décalé de 1** (par exemple, si la catégorie est au 2<sup>ème</sup> rang, le deuxième article rapporte les points du 3<sup>ème</sup> rang) ;
- et ainsi de suite (le troisième article subit un décalage de 2 rangs, etc.) ;
- les rangs au-delà du 6<sup>ème</sup> rapportent tous 1 point.

Le bonus d'état (règle 5) reste inchangé quel que soit le nombre d'articles de la même catégorie.

**Règle 7 (tout le monde est servi) :** lorsqu'un abonné participant à la campagne ne reçoit aucun article (box vide), la composition subit un malus global de **-10**.

**Règle 8 (équité) :** lorsqu'un abonné reçoit **2 articles de moins** (ou plus) qu'un autre abonné dans la même campagne, la composition subit un malus global de **-10** (appliqué une seule fois par abonné concerné).

### Exemple détaillé

Supposons la présence de huit articles disponibles :

Id	Désignation	Catégorie	Âge	État	Prix	Poids
a1	Monopoly Junior	SOC	PE	N	8 €	400 g
a2	Barbie Aventurière	FIG	PE	TB	5 €	300 g
a3	Puzzle éducatif	EVL	PE	TB	7 €	350 g
a4	Cubes alphabet	CON	PE	N	4 €	300 g
a5	Livre cache-cache	LIV	PE	N	3 €	200 g
a6	Kapla 200 pièces	CON	EN	B	10 €	600 g
a7	Cerf-volant Pirate	EXT	EN	N	6 €	400 g
a8	Le Petit Nicolas	LIV	EN	TB	5 €	200 g

Trois abonnés sont inscrits à la campagne :

- **Alice** — enfant en tranche **PE** (3-6 ans) — préférences : SOC, FIG, EVL, CON, LIV, EXT
- **Bob** — enfant en tranche **EN** (6-10 ans) — préférences : EXT, CON, SOC, EVL, FIG, LIV
- **Clara** — enfant en tranche **PE** (3-6 ans) — préférences : EVL, LIV, FIG, SOC, CON, EXT

Le poids maximum par box est  $W_{\max} = 1200$  g.

Les articles compatibles avec chaque abonné sont donc :

- **Alice (PE)** : a1 (SOC), a2 (FIG), a3 (EVL), a4 (CON), a5 (LIV)
- **Bob (EN)** : a6 (CON), a7 (EXT), a8 (LIV)
- **Clara (PE)** : a1 (SOC), a2 (FIG), a3 (EVL), a4 (CON), a5 (LIV)

On notera qu'Alice et Clara sont en **concurrence pour les mêmes 5 articles PE**. L'optimisation devra déterminer le meilleur partage de ces articles entre elles, en fonction

de leurs préférences respectives. Bob, seul abonné dans la tranche EN, n'a que 3 articles à sa disposition.

### Première composition

Si la composition des box est la suivante (Alice s'est vu affecter les articles les plus rentables pour elle, au détriment de Clara) :

- Box d'**Alice** : a1 (SOC, PE, N, 400 g) + a2 (FIG, PE, TB, 300 g) + a3 (EVL, PE, TB, 350 g) → poids total = 1050 g ✓
- Box de **Bob** : a7 (EXT, EN, N, 400 g) + a6 (CON, EN, B, 600 g) → poids total = 1000 g ✓
- Box de **Clara** : a4 (CON, PE, N, 300 g) + a5 (LIV, PE, N, 200 g) → poids total = 500 g ✓

Article non affecté : a8.

Le score se calcule ainsi :

- **Alice** :
    - a1 (SOC) : SOC est au 1<sup>er</sup> rang → 10 pts + 2 (état N) = **12**
    - a2 (FIG) : FIG est au 2<sup>ème</sup> rang → 8 pts + 1 (état TB) = **9**
    - a3 (EVL) : EVL est au 3<sup>ème</sup> rang → 6 pts + 1 (état TB) = **7**
    - Sous-total Alice : **28**
  - **Bob** :
    - a7 (EXT) : EXT est au 1<sup>er</sup> rang → 10 pts + 2 (état N) = **12**
    - a6 (CON) : CON est au 2<sup>ème</sup> rang → 8 pts + 0 (état B) = **8**
    - Sous-total Bob : **20**
  - **Clara** :
    - a4 (CON) : CON est au 5<sup>ème</sup> rang → 2 pts + 2 (état N) = **4**
    - a5 (LIV) : LIV est au 2<sup>ème</sup> rang → 8 pts + 2 (état N) = **10**
    - Sous-total Clara : **14**
- Vérification de l'équité : Alice a 3 articles, Bob en a 2, Clara en a 2. La différence maximale est de 1, ce qui est inférieur à 2 : pas de malus.

**Score total : 28 + 20 + 14 = 62 points.**

On constate que Clara reçoit un article CON (son 5<sup>ème</sup> choix) car Alice a pris a3 (EVL), qui est pourtant le **1<sup>er</sup> choix** de Clara.

### Meilleure composition

En cédant l'article a3 (EVL) à Clara et en attribuant a4 (CON) à Alice, tout en complétant la box de Bob avec l'article a8 restant :

- Box d'**Alice** : a1 (SOC, PE, N, 400 g) + a2 (FIG, PE, TB, 300 g) + a4 (CON, PE, N, 300 g)  
→ poids total = 1000 g ✓

- Box de **Bob** : a7 (EXT, EN, N, 400 g) + a6 (CON, EN, B, 600 g) + a8 (LIV, EN, TB, 200 g)  
→ poids total = 1200 g ✓
- Box de **Clara** : a3 (EVL, PE, TB, 350 g) + a5 (LIV, PE, N, 200 g) → poids total = 550 g ✓

Le score se calcule ainsi :

- **Alice** :
  - a1 (SOC) : 1<sup>er</sup> rang → 10 + 2 (N) = **12**
  - a2 (FIG) : 2<sup>ème</sup> rang → 8 + 1 (TB) = **9**
  - a4 (CON) : 4<sup>ème</sup> rang → 4 + 2 (N) = **6**
  - Sous-total Alice : **27**
- **Bob** :
  - a7 (EXT) : 1<sup>er</sup> rang → 10 + 2 (N) = **12**
  - a6 (CON) : 2<sup>ème</sup> rang → 8 + 0 (B) = **8**
  - a8 (LIV) : 6<sup>ème</sup> rang → 1 + 1 (TB) = **2**
  - Sous-total Bob : **22**
- **Clara** :
  - a3 (EVL) : 1<sup>er</sup> rang → 10 + 1 (TB) = **11**
  - a5 (LIV) : 2<sup>ème</sup> rang → 8 + 2 (N) = **10**
  - Sous-total Clara : **21**
- Vérification de l'équité : Alice et Bob ont 3 articles, Clara en a 2. La différence maximale est de 1 : pas de malus.

**Score total : 27 + 22 + 21 = 70 points.**

Alice perd seulement 1 point (28 → 27) en cédant a3 à Clara, tandis que Clara gagne 7 points (14 → 21) car elle reçoit son 1<sup>er</sup> choix. Bob gagne aussi 2 points en exploitant a8. Le **partage judicieux des articles PE entre les deux abonnées de même tranche d'âge**, combiné à une meilleure utilisation du catalogue EN, améliore significativement le score global.

## Données fournies

Les données sont fournies sous la forme de fichier texte CSV (*comma-separated values*, séparation avec le caractère ';') respectant le format suivant :

- la première ligne contient l'intitulé "articles" ;
- les lignes suivantes contiennent les descriptifs des articles avec
  - (1) un id,
  - (2) une désignation,
  - (3) une catégorie parmi les catégories possibles,
  - (4) une tranche d'âge (BB, PE, EN ou AD),

- (5) un état (N, TB ou B),
- (6) le prix en euros (entier) et
- (7) le poids en grammes (entier) ;
- une ligne vide sépare la suite ;
- une ligne contenant “abonnes” introduit la liste des abonnés ;
- chaque abonné est ensuite décrit sur une ligne avec
  - (1) un id,
  - (2) son prénom (supposé unique),
  - (3) la tranche d’âge de son enfant (BB, PE, EN ou AD) et
  - (4) son ordre de préférence listant les catégories de la plus souhaitée à la moins souhaitée ;
- une ligne vide sépare la suite ;
- une ligne contenant “paramètres” introduit les paramètres ;
- une ligne indique le poids maximum par box en grammes (entier).

**Attention :** du fait des ouvertures et fermetures sur des tableurs, il est possible que des ‘;’ aient été rajoutés après les intitulés. Ne testez donc pas directement une égalité sur la chaîne mais utilisez la méthode `contains`.

Les intitulés utilisés pour les catégories sont les suivants :

- **SOC** — Jeux de société
- **FIG** — Figurines et poupées
- **CON** — Jeux de construction
- **EXT** — Jeux d’extérieur
- **EVL** — Jeux d’éveil et éducatifs
- **LIV** — Livres jeunesse

Les intitulés utilisés pour les tranches d’âge sont : **BB** (0-3 ans), **PE** (3-6 ans), **EN** (6-10 ans), **AD** (10+ ans).

Les intitulés utilisés pour les états sont : **N**, **TB**, **B**.

#### Exemple de fichier d’entrée

```
articles
a1;Monopoly Junior;SOC;PE;N;8;400
a2;Barbie Aventurière;FIG;PE;TB;5;300
a3;Puzzle éducatif;EVL;PE;TB;7;350
a4;Cubes alphabet;CON;PE;N;4;300
a5;Livre cache-cache;LIV;PE;N;3;200
```

```
a6;Kapla 200 pièces;CON;EN;B;10;600  
a7;Cerf-volant Pirate;EXT;EN;N;6;400  
a8;Le Petit Nicolas;LIV;EN;TB;5;200
```

```
abonnes  
s1;Alice;PE;SOC,FIG,EVL,CON,LIV,EXT  
s2;Bob;EN;EXT,CON,SOC,EVL,FIG,LIV  
s3;Clara;PE;EVL,LIV,FIG,SOC,CON,EXT
```

```
parametres  
1200
```

## Résultat attendu

La réponse de votre algorithme sera une composition globale qui prendra la forme d'un fichier CSV. Ce fichier contiendra :

- **la première ligne avec le score de la composition** (un entier),
- puis **une ligne par article affecté** décrivant la box dans laquelle il a été placé. Si un article n'est affecté à aucune box, il n'apparaît pas.

Pour chaque article affecté, la ligne précisera les éléments suivants (séparés par ';') :

- (1) le prénom de l'abonné
- (2) l'id de l'article
- (3) la catégorie de l'article
- (4) la tranche d'âge de l'article
- (5) l'état de l'article

## Exemple de fichier de sortie

```
70  
Alice;a1;SOC;PE;N  
Alice;a2;FIG;PE;TB  
Alice;a4;CON;PE;N  
Bob;a7;EXT;EN;N  
Bob;a6;CON;EN;B  
Bob;a8;LIV;EN;TB  
Clara;a3;EVL;PE;TB  
Clara;a5;LIV;PE;N
```

Pour manipuler facilement votre code, vous proposerez en outre :

- une méthode permettant de générer la chaîne correspondant au fichier CSV,
- une méthode permettant de sauver cette composition dans un fichier dédié.

## Travail proposé

Le travail proposé est ouvert. Nous vous conseillons cependant de suivre la démarche ci-dessous pour avancer progressivement. Bien entendu, vous pouvez vous organiser comme vous le souhaitez pour travailler en parallèle.

### (1) Mise en place du problème d'optimisation

Proposez dans un premier temps les structures de données utiles pour représenter le problème et une composition de box. Proposez et implémentez ce qu'il faut pour mesurer la qualité d'une composition.

Pour simplifier cette partie, vous pouvez débuter par implémenter une partie des règles de score (par exemple, 1 point par article affecté à un abonné compatible en âge) puis implémenter de plus en plus de détails du calcul de score : points par préférence de catégorie, bonus d'état, utilités dégressives, malus d'équité, etc.

Cela permet de disposer d'une première fonction de scoring pour travailler en parallèle sur l'algorithme d'optimisation.

### (2) Algorithme d'optimisation

Proposez un algorithme cherchant à maximiser cette mesure de qualité. On vous conseille de commencer par un **algorithme simple** (voire trivial) qui constituera une base de comparaison avec les algorithmes plus évolués que vous pourrez proposer dans un second temps.

### (3) Directions possibles de travail

Parmi les pistes que vous pouvez utiliser pour votre programme d'optimisation, vous pouvez :

- utiliser des approches **gloutonnes** comme point de départ, en remplissant les box article par article en choisissant à chaque étape l'affectation la plus rentable parmi les articles compatibles ;
  - [https://fr.wikipedia.org/wiki/Algorithme\\_glouton](https://fr.wikipedia.org/wiki/Algorithme_glouton)
- utiliser un algorithme de type **recuit simulé**, qui permet d'explorer l'espace des solutions en acceptant parfois des dégradations temporaires pour échapper aux optima locaux ;
  - [https://fr.wikipedia.org/wiki/Recuit\\_simulé](https://fr.wikipedia.org/wiki/Recuit_simulé)
- raisonner sur des approches de type **backtracking** ou **branch and bound** en remettant en cause des affectations déjà effectuées ;
  - [https://fr.wikipedia.org/wiki/Séparation\\_et\\_évaluation](https://fr.wikipedia.org/wiki/Séparation_et_évaluation)

- s'inspirer des algorithmes de **bin packing** (problème du sac à dos), particulièrement adaptés aux problèmes de remplissage sous contrainte de capacité ;
  - [https://fr.wikipedia.org/wiki/Problème\\_du\\_sac\\_à\\_dos](https://fr.wikipedia.org/wiki/Problème_du_sac_à_dos)
- utiliser des **algorithmes évolutionnaires** (génétiques) qui font évoluer une population de solutions candidates.
  - [https://fr.wikipedia.org/wiki/Algorithme\\_évolutionniste](https://fr.wikipedia.org/wiki/Algorithme_évolutionniste)

#### (4) Livrables attendus

Le rendu attendu consistera :

- à décrire l'algorithme que vous avez utilisé (écriture de l'algorithme général, détails éventuels) en justifiant vos choix ;
- à programmer l'application correspondante qui sera sollicitée par l'application web (*cf.* autres sujets) ;
- à donner les compositions obtenues sur les données exemples fournies que vous avez traitées ;
- à donner le descriptif éventuel des différentes extensions que vous avez proposées en détaillant votre raisonnement.

Ces éléments pourront être présentés dans un rapport d'une page (ou deux si besoin) pour donner une vision synthétique de votre travail.

Vous n'oublierez pas de préciser les éléments utiles pour utiliser correctement votre application. Cela inclut :

- la manière de lancer votre application ;
- la manière de changer facilement le fichier de données en entrée ;
- la manière d'accéder au résultat de votre algorithme ;
- la manière de choisir le fichier dans lequel sauver votre résultat.

#### Extensions possibles

Le sujet proposé initialement est un sujet ouvert à de nombreux algorithmes et variantes. Une fois que vous avez des résultats qui vous semblent pertinents sur les critères proposés, vous pouvez étendre le travail en choisissant une des directions suivantes :

- **Compatibilité d'âge élargie** : la compatibilité d'âge n'est plus stricte. Un article étiqueté pour une tranche d'âge peut convenir à la tranche adjacente, avec une pénalité. Par exemple, un article PE (3-6 ans) placé dans la box d'un abonné EN (6-10 ans) est autorisé mais ses points de préférence sont réduits de moitié (arrondi à l'entier supérieur, minimum 1 point). Les tranches non adjacentes restent incompatibles.

- **Box multi-enfants** : un abonné peut avoir **plusieurs enfants** d'âges (et donc de tranches) différents. La box doit contenir au moins un article compatible avec chaque enfant déclaré. Cela ajoute une contrainte supplémentaire à satisfaire tout en diversifiant les articles éligibles.
- **Historique et diversité** : prise en compte de l'historique des box précédemment envoyées. Un article dont la catégorie est identique à celle d'un article reçu lors de la dernière box du même abonné voit ses points de préférence réduits de 2 (minimum 1 point). Cela favorise le renouvellement.
- **Calibrage en prix** : en plus du poids maximum, la composition doit respecter une fourchette de prix cible  $[P_{\min}, P_{\max}]$  par box. Si le prix total d'une box est en dehors de cette fourchette, un malus de -5 est appliqué.
- **Modification de la qualité d'une composition** : la fonction de score proposée est un premier exemple. Si vous le jugez utile, n'hésitez pas à proposer d'autres mesures de qualité et à comparer les résultats obtenus.

## Partie « Déploiement »

Cette partie est la finalité du projet : elle permet au commanditaire d'accéder à tout ce qui a été développé. Sans ce déploiement, la plus belle application du monde reste inaccessible... Il ne faut donc surtout pas la négliger, d'autant plus que les lauréats auront une petite démonstration à réaliser lors de la remise des prix, depuis l'ordinateur de l'amphi. La disponibilité sur un site public est donc cruciale.

### Présentation

Il s'agit de mettre en production la solution globale (application Web, brique logicielle d'optimisation, dépendances...) sur un serveur public. Le serveur peut être physique, virtuel ou dans le cloud (AWS, Google Cloud, docketu...). Le déploiement doit être optimisé pour une mise en production (« poids » des images, sécurité). Il accueillera :

- Le composant de l'application web de gestion des articles, des abonnés et des box,
- Le composant de l'application d'optimisation qui échange ou partage des informations avec l'application web (exploitation du catalogue d'articles et de la liste des abonnés, génération des compositions de box).

Les 2 conteneurs doivent donc pouvoir s'échanger des données (web-service ou autre...). Par ailleurs, dans l'optique d'un déploiement automatisé, il est demandé de mettre en place une chaîne d'intégration continue (et ainsi d'exploiter les tests unitaires qui ont pu être définis) et de déploiement continu (CI/CD).

La solution globale doit être accessible avec une simple URL publique.

On pourra, après avoir traité les points précédents, envisager :

- l'ajout d'un load balancer sur le serveur de production,

- l'ajout d'un serveur SMTP s'il est intégré à certaines des fonctionnalités de l'application web (qui pourront donc être agrémentées). Un outil de type MailCatcher peut être intéressant à exploiter dans ce cadre.

## Livrable

Le rendu attendu consistera :

- à décrire les choix que vous avez effectués en les justifiant ;
- à présenter l'architecture globale de ce que vous avez mis en œuvre (un schéma peut permettre de synthétiser cette information) ;
- à présenter les résultats obtenus, de façon qualitative et quantitative.