

Projet « Réouverture des salles de spectacles en période de pandémie, aidez-nous à les remplir ! »

ENSIMAG 1aa

L'entreprise « Faites-moi rêver ! » gère une salle de spectacles avec une disposition assez classique comme celle de la Figure 1.

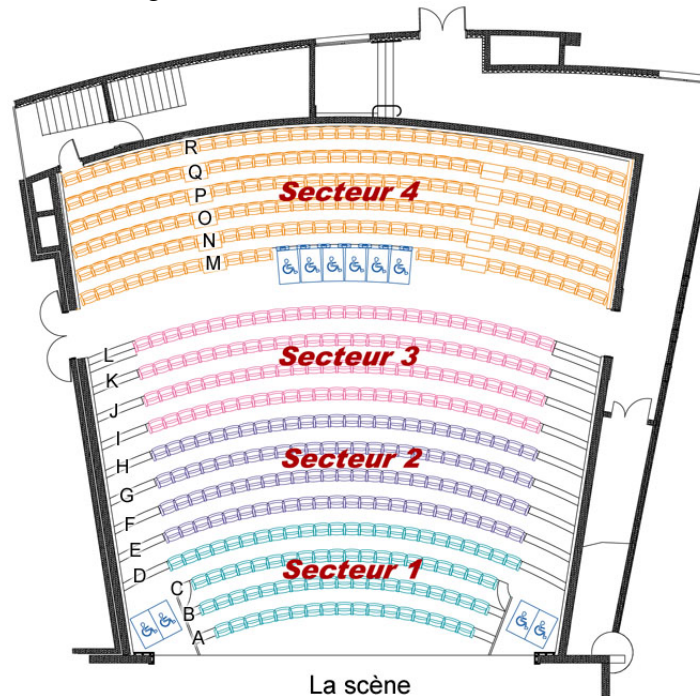


Figure 1* : Configuration d'une salle de spectacle

*<https://commons.wikimedia.org/wiki/File:Plan-salle.jpg>

En temps normal, seules les contraintes physiques de la salle limitent le remplissage des rangées. Or, en période de pandémie, l'entreprise doit proposer un protocole de remplissage qui garantit une certaine distanciation sociale limitant les risques de transmission.

Le protocole **scénario de base** est ainsi énoncé :

- deux rangées consécutives de spectateurs doivent être séparées par au moins P rangées vides, en pratique, on peut considérer sans perdre de généralité $P=1$;
- un groupe de S spectateurs (famille/amis) est indissociable et doit être placé sur une seule rangée, on peut considérer sans perdre de généralité $S \leq K$ avec $K=6$ en pratique ;
- deux groupes de spectateurs qui sont placés sur une même rangée doivent être séparés de Q places, on peut considérer sans perdre de généralité $Q=2$ en pratique.

Le patron de l'entreprise fait donc appel à vous, jeunes ingénieurs maîtrisant les nouvelles technologies, pour l'aider à mettre en place des stratégies de remplissage qui respectent les contraintes.

Pour simplifier les choses, on peut considérer qu'une salle est constituée de G groupes de rangées d'un seul type de places. Dans la Figure 1 par exemple, on a quatre groupes (à ne pas confondre avec les Secteurs de la figure) de rangées. Le groupe 1 est composé des Secteurs 1, 2 et 3 ; tandis que le Secteur 4 est décomposé en 3 groupes (la dernière rangée R est condamnée et toute la partie centrale de la rangée M avec les places pour handicapés est traitée à part). Ainsi une salle de spectacle peut être définie par la donnée de G groupes de

rangées, chaque groupe g contient R_g rangées, chaque rangée r d'un groupe g dispose un nombre C_{gr} de places (capacité de la rangée) et d'une distance D_{gr} de la scène.

Par exemple dans la Figure 1, on a $G=4$; $R_1=12$, $R_2=R_4=5$, $R_3=4$; la rangée A du groupe 1, $C_{1A}=17$, $D_{1A}=1$; la rangée B du groupe 1, $C_{1B}=19$, $D_{1B}=2$; la rangée M des groupes 2 et 4, $C_{2M}=C_{4M}=8$, $D_{2M}=D_{4M}=13$; etc. (on compte le rang à partir de la scène).

Maintenant, le patron reçoit les réservations jusqu'à 24h avant chaque spectacle. Pour chaque spectacle, il dispose les informations sur le nombre B de réservations. Une réservation b est égale à un groupe de S_b spectateurs indissociables, avec $S_b \leq \min_{g,r}(C_{gr})$, i.e. on peut toujours placer un groupe de spectateurs dans une rangée dans la salle.

Les données du problème sont donc fournies sous la forme trois **fichiers de données** suivant :
Fichier Salle01.dat :

```
4          // G rangées dans la salle
12 5 4 5 // Rg rangées par groupe
17 1      // chaque rangée r d'un groupe g : Cgr places disponibles et distance Dgr de la scène
19 2
20 3
...
8 13
...
...
8 13
...
```

Fichier Contraintes01.dat :
1 6 2 // les valeurs de P, K, et Q

Fichier Reservations01.dat :
6 // B réservations
5 2 6 5 4 3 // S_b spectateurs pour chaque réservation b

Un fichier résultats Remplissage01.res est attendu comme suit :

```
3 9 25/56 // Nb. de rgs utilisées, somme des distances à la scène et taux de remplissage
1 1 1 2 7 // Num. groupe, de la rangée, des groupes de spectateurs, nb. places utilisées
1 3 3 4 11
1 5 5 6 7
Non placés // Numéros des groupes de spectateurs non placés
-1
```

Vous notez ici que les solutions sont données avec trois critères de performance sur le remplissage. Il peut en avoir d'autres, comme par exemple le taux de remplissage, i.e. nombre de places occupées/nombre de places total des rangées engagées. Une rangée est considérée comme engagées s'elle contient au moins un spectateur placé.

On peut également avoir plus de réservations que la capacité de la salle avec ou sans contrainte de distanciation sociale.

Suggestion : Commencer déjà par remplir un groupe de rangées avant d'étendre le remplissage à l'ensemble des groupes de rangées dans leur globalité.

Tâches à effectuer dans ce projet : Scénario de base

Tâche 1 : Proposer un algorithme et son programme JAVA associé qui implémentent une stratégie **heuristique** (fondée sur du bon sens) de remplissage d'une salle.

Il s'agit de trouver ici juste une **solution réalisable** (satisfaisant toutes les contraintes de distanciation sociale).

Pour passer de la définition de la stratégie au programme JAVA en passant par un algorithme, il vous est recommandé de suivre la démarche ci-après.

1. Faire une analyse descendante du problème au niveau 0:

- Pour les données : Identifiez toutes les classes d'objets ainsi que tous les attributs de chaque classe permettant de représenter les données que vous avez à manipuler pour créer un plan de découpe, et dressez un diagramme de relations entre ces classes (à la UML). Si des structures de données sont employées, précisez et justifiez-les.
- Pour les traitements : Donnez pour chacune des classes, les noms des méthodes, leurs entrées et sorties, et les grandes étapes de chacune des méthodes. N'hésitez pas à faire des schémas pour expliquer ce que vous voulez faire. Un schéma simple vaut parfois mille lignes d'explication.

2. Préciser votre analyse descendante au niveau i : pour chacune des méthodes, détaillez algorithmiquement les traitements. Si une étape de votre méthode est trop complexe n'hésitez pas à la décomposer en de nouveaux sous-algorithmes que vous préciserez par la suite et séparément, et ainsi de suite.

Tâche 2 : Proposer un algorithme **énumérative totale** et son programme JAVA associé qui améliore celui développé dans la Tâche 1 en trouvant une **solution optimale** par rapport à l'un des deux critères (nombre de rangées utilisées ou somme des distances des rangées par rapport à la scène), puis modifier le programme pour l'autre critère.

Tâche 3 : Proposer un algorithme **énumérative implicite** à base d'**évaluation et séparation** et son programme JAVA associé qui trouve une **solution optimale** par rapport à un des deux critères (nombre de rangées utilisées ou somme des distances des rangées par rapport à la scène), puis modifier le programme pour l'autre critère.

Tâche 4 : Développer un « checker » qui, prend en entrée l'ensemble des fichiers de données et du fichier de résultats, et vérifie la validité des contraintes de la solution trouvée ainsi que celles des deux critères. Cette vérification doit être effectuée sans avoir à recalculer la solution avec les programmes précédemment développés.

Un affichage « simple » du plan de remplissage de la salle associé au checker est une option appréciée.

Tâche 5 : Votre travail sera synthétisé dans un mini-rapport (10-15 pages max. pour le cœur du rapport, pensez à l'usage d'annexes si vous devez dépasser les 15 pages) qui contiendra votre analyse du problème, les schémas conceptuels de données, les algorithmes ainsi que leur complexité algorithmique (en temps, et en mémoire si besoin est), des tests et leurs résultats numériques (structures et valeurs des solutions, temps CPU, etc.).

Pour les mordus/challengers : Scénario avancé !

On suppose ici des salles « rectangulaires » où les rangées ont toutes le même nombre de places disponibles.

La contrainte b) est relâchée. Un groupe de spectateurs peut-être placé sur plusieurs rangées, mais les écarts latéraux avec un autre groupe de Q places sur une même rangée doivent toujours être respectés. On peut placer deux groupes de spectateurs sur deux rangées consécutives mais l'écart de Q places (en « verticale », comptage en distance « manhattan ») est aussi valable. Aussi, un groupe de spectateurs doit rester « groupé », n'importe quel membre d'un groupe doit être en « contact » (zéro place d'écart) avec au moins un autre membre du même groupe. L'exemple ci-dessous illustre, une solution qui respecte ces contraintes (à gauche), et une autre qui ne respecte pas ces contraintes (à droite).

0001000000030	0001110000300
0111002220033	0010002200033
0001100222000	0011002202000

Tâche 6 : Modifier le programme JAVA issu de la Tâche 3 pour optimiser le critère du taux de remplissage en gardant la contrainte b).

Tâche 7 : A l'instar de la Tâche 1, proposer un algorithme et son programme JAVA associé qui implémentent une stratégie **heuristique** (fondée sur du bon sens) de remplissage d'une salle rectangulaire.

Il s'agit de trouver ici juste une **solution réalisable** (satisfaisant toutes les contraintes de distanciation sociale).

Tâche 8 : Proposer un algorithme **énumérative implicite** à base d'**évaluation et séparation** et son programme JAVA associé qui trouve une **solution optimale** par rapport au critère du taux de remplissage.

Tâche 9 : Comparer les solutions pour les salles rectangulaires, sur le critère du taux de remplissage, obtenues par la Tâche 6 et celles obtenues par la Tâche 8.

REMARQUE : Si vous devez changer des spécifications des fichiers (notamment pour le scénario avancé) ou faire d'autres hypothèses sur les scénarios, c'est possible. Il faut les valider avec l'enseignant avant et les préciser clairement dans votre rapport. Attention aux hypothèses trop simplificatrices.

Pour les inconditionnels : Scénario proactif !!!

A moins de 24h, certains groupes de spectateurs annulent leur réservation et d'autres nouvelles réservations sont prises en compte.

Tâche 10 : Proposer une solution pour compléter les Tâches 3/6/8 afin de prendre en compte ces modifications dans les réservations sur la journée même du spectacle, SANS changer tous les plans de remplissage préétablis à 24h.

Ce travail est à effectuer par **groupe de 4 à 5 étudiants**.

Date de rendus : date à définir ensemble, dépôt sur Teide.

Pour les rendus, chaque groupe met à disposition un fichier archive compressé de type GroupeX.zip (X est le numéro du groupe) que vous téléverserez sur Teide. Je créerai N groupes sur Teide selon le nombre.

Ce fichier archive doit contenir la structure de répertoires suivante :

GroupeX/Programs/ // répertoire contenant vos programmes sources

GroupeX/Report/ // répertoire contenant votre mini-rapport (10-15 pages max.)
reprenant et détaillant les points clés de votre projet.

Les fichiers sources (.tex ou .docx ou autres) sont demandés en plus du .pdf.

GroupeX/Data/ // répertoire contenant vos fichiers de données

GroupeX/Results/ // répertoire contenant vos fichiers de résultats expérimentaux

GroupeX/Articles/ // répertoire contenant les articles de la littérature si vous en avez
les .pdf, .docx, etc. Pensez à les référencer dans votre rapport