

# IFT-4001 Projet d'exploration

# IFT-7020 Projet de recherche

Enseignant: Claude-Guy Quimper

**Échéances (IFT-4001):** 27 octobre 2024, 15 décembre 2024.

**Échéances (IFT-7020):** 27 octobre 2024, 10 novembre 2024, 15 décembre 2024.

**Équipes:** de 2 à 5 personnes.

**Mixité:** Les étudiants inscrits au cours IFT-4001 peuvent être en équipe avec ceux du cours IFT-7020. Dans ce cas, le projet sera évalué selon les critères du cours IFT-7020.

Pour ce projet d'exploration et ce projet de recherche, vous devez explorer les capacités et les limites d'un solveur. Pour le projet d'exploration (IFT-4001) vous devez **apprendre** quelque chose sur les solveurs ou l'optimisation combinatoire qui n'a pas été mentionné dans le cours et en faire mention dans votre rapport. Pour le projet de recherche (IFT-7020), vous devez **découvrir** quelque chose de nouveau pour la communauté scientifique et en faire mention. Le rapport prendra donc la forme d'un article scientifique pouvant être publié.

Vous pouvez aborder plusieurs sujets avec ce projet. Vous pouvez par exemple choisir un problème, le modéliser avec MiniZinc et le résoudre avec un solveur tel Chuffed. Il existe une liste de problèmes à résoudre sur le site web [www.csplib.org](http://www.csplib.org), mais les meilleurs problèmes viendront probablement de vous. Vous pouvez apporter des améliorations à un solveur en y ajoutant des heuristiques, des contraintes, de nouvelles structures de données ou de nouveaux algorithmes de filtrage. Vous pouvez expérimenter avec des solveurs autres que Chuffed et Gecode.

Pour le projet de recherche (IFT-7020), vous devez convaincre le lecteur de votre rapport que ce que vous avez fait est nouveau. Il vous faudra donc faire une revue de littérature d'au moins **trois articles par membre** de l'équipe inscrit au cours IFT-7020. Ces articles doivent avoir été publiés dans des revues scientifiques ou des actes de conférences avec comité d'arbitrage.

## Les livrables.

**Une proposition de projet** (évaluation d'équipe): Dans ce document d'un **maximum d'une page**, vous devez présenter ce que vous avez l'intention de faire comme projet. Vous décrivez brièvement le problème sur lequel vous voulez travailler et ce que vous avez l'intention d'apprendre ou de découvrir. Cette proposition est à remettre le **27 octobre** en format PDF. Pour les projets de recherches (IFT-7020), incluez les références des articles que vous avez l'intention de lire. Remettez aussi les fichiers PDF de ces articles. Les noms, prénoms et numéros de dossier de chaque membre de l'équipe doivent figurer sur votre proposition de recherche.

L'envergure de votre projet doit être proportionnelle à la taille de votre équipe. Par exemple, si vous avez l'intention de modéliser et résoudre un problème combinatoire, je m'attends à voir autant de modèles qu'il y a de membres dans l'équipe (ex: un modèle de contraintes, un modèle linéaire, un modèle SAT, etc.) Votre projet doit tenter de repousser les limites des solveurs. Prévoyez donc des jeux de données de tailles différentes.

**Revue de littérature** (IFT-7020 seulement, évaluation individuelle): Vous devrez remettre le **10 novembre** un document PDF résumant chaque article que vous avez lu. Chaque membre inscrit au cours IFT-7020 doit remettre, individuellement, une revue de littérature de 3 articles d'une longueur d'**au plus une page et demie**. Les membres d'une même équipe doivent choisir des articles différents. Vous devez non seulement résumer les articles, mais décrire les liens qu'il y a entre chaque article. Vous devez dire comment se positionne chaque article par rapport aux autres. Les traductions des résumés (*abstract*) des articles n'ont pas leur place dans une revue de littérature puisqu'elles ne font aucun lien entre les articles et ne font pas ressortir les principales différences et innovations de chaque article.

**Un rapport de projet** (évaluation d'équipe): Vous devez rédiger un rapport par équipe d'**au plus 10 pages** en vous inspirant du plan suivant et le remettre le **15 décembre** en format PDF.

1. Introduction

2. Description du problème

Dans cette section, vous devez présenter la problématique avec suffisamment de détails pour qu'un lecteur aie tout ce qu'il faut pour qu'il puisse la résoudre, à sa manière. S'il s'agit d'un problème à modéliser, vous devez présenter tous les paramètres d'une instance à votre problème et vous devez décrire ce qu'est une solution valide. Profitez-en pour définir la notation mathématique. Ex: Le problème du commis voyageur prend en entrée une matrice  $D$  où  $D[i, j]$  est la distance, en kilomètres, entre la ville  $i$  et la ville  $j$ .

3. Approche(s) proposée(s)

Si vous décrivez des modèles pour résoudre votre problème, vous devez lister les variables de vos modèles. Pour chaque variable, donnez son domaine. Présentez chaque contrainte de vos modèles sous forme mathématique (ne copiez pas votre code MiniZinc dans votre document). Donnez la fonction objectif de votre problème. Le lecteur doit être en mesure de prendre vos modèles et de les implanter dans le solveur de la même façon que vous l'avez fait. Mentionnez aussi les heuristiques utilisées. Si vous présentez un modèle, inspirez-vous de la notation utilisée dans les exercices du chapitre 1.

4. Protocole d'expérimentation

Décrivez les instances de votre banc d'essai. Décrivez les métriques que vous allez mesurer. Un lecteur qui lit cette section de votre rapport (de même que les sections précédentes) devrait pouvoir reproduire les résultats présentés à la section suivante.

5. Résultats

Votre protocole d'expérimentation vous demande de prendre des mesures. Vous présentez dans cette section les mesures qui ont été prises. Ce peut être le temps d'exécution et le nombre de retours arrière pour chaque instance de votre banc d'essai. Ces résultats peuvent être présentés sous forme de graphiques (ex.: le temps de résolution du problème en fonction de la taille de l'instance).

6. Discussion

Comment interprétez-vous les résultats de la section précédente?

7. Conclusion

Résumez votre travail.

8. Annexe: Code de programmation (en pièces jointes)

## Critères d'évaluation

Des points seront accordés pour chaque section du rapport décrit ci-dessus. Portez une attention particulière aux points suivants:

- Qualité du français (jusqu'à 10% de pénalité);
- Présentation claire et détaillée du problème;
- Difficulté de la problématique (vous devez pousser les limites du solveur);
- Qualité de la partie expérimentale;
- Qualité du code de programmation (si applicable);
- **Leçons apprises / découvertes scientifiques.**

L'évaluation prendra en compte le nombre de membres composant l'équipe. Par exemple, une équipe de quatre personnes devrait normalement proposer deux fois plus d'approches pour résoudre un problème qu'une équipe de deux personnes.

## Publication

Ce que vous apprendrez ou allez découvrir dans ce projet pourrait intéresser la communauté scientifique. Si votre rapport peut être peaufiné afin d'être publié dans des actes de conférence avec comité d'arbitrage, un membre de l'équipe pourra voyager aux frais de l'université afin de présenter les travaux à l'une des conférences suivantes.

- CP 2025 (Glasgow, Royaume-Uni)
- CPAIOR 2025 (Melbourne, Australie)
- AAAI 2025 (Philadelphie, États-Unis)
- IJCAI 2025 (Montréal, Canada)

Cette liste n'est pas exhaustive et nous pouvons envisager d'autres conférences ayant une bonne réputation scientifique.

## Prix Pierre-Ardouin

Le meilleur travail recevra un prix du département d'informatique et sera admissible au [prix Pierre-Ardouin](#). Je vous encourage à jeter un coup d'oeil aux lauréats des années précédentes.

## Annexe: Comment bien présenter un modèle?

Lorsque l'on présente un modèle d'optimisation dans un rapport, on le présente comme un modèle mathématique. Un lecteur ne connaissant pas MiniZinc devrait être en mesure de comprendre ce modèle et de l'implémenter dans le langage de son choix (MiniZinc, Prolog, Essence, ...)

1. Privilégiez la notation arithmétique plutôt que la notation `NomDeLaContrainte(Variables)`
2. Utilisez la notation  $\forall$  plutôt que la syntaxe `forall`.
3. Bien que dans un code de programmation, on privilégie les noms descriptifs, en mathématique, les variables sont généralement représentées par un seul symbole.
4. Utilisez la notation  $\text{dom}(X)$  pour définir le domaine d'une variable.

Voici un extrait de code MiniZinc et la façon dont il doit être présenté dans un rapport.

```
set of int: lignes = 1..n;
set of int: colonnes = 1..n;
int: SOMME_MAGIQUE = n * (n * n + 1) div 2;
array[lignes, colonnes] of var 1..n * n: x;
forall(i in lignes)(sum([x[i,j] | j in colonnes]) = SOMME_MAGIQUE;
```

Code MiniZinc

$L = \{1, \dots, n\}$	Ensemble des lignes
$C = \{1, \dots, n\}$	Ensemble des colonnes
$S = \frac{n(n^2 + 1)}{2}$	Somme magique
$\text{dom}(x_{i,j}) = \{1, \dots, n^2\}$	$\forall i \in L, j \in C$
$\sum_{j \in C} x_{i,j} = S$	$\forall i \in L$

Présentation demandée dans le rapport

Écrire en notation mathématique ne veut pas dire qu'il faut décomposer les contraintes. En effet, la contrainte `AllDifferent` n'est pas équivalente aux  $O(n^2)$  contraintes de différence ( $\neq$ ) puisque les algorithmes de filtrage de ces deux contraintes ne sont pas les mêmes et le nombre de contrainte n'est pas le même: 1 vs  $O(n^2)$ .

Les heuristiques de recherche ne font pas partie du modèle mathématique. Vous n'avez donc pas à les décrire mathématiquement, mais vous devez les décrire dans vos propres mots. Encore une fois, l'heuristique doit être compréhensible par quelqu'un qui ne connaît pas MiniZinc. Plutôt que de dire « Nous utilisons les heuristiques `smallest` et `indomain_min` sur le vecteur  $x$  », vous devez plutôt écrire « Le solveur branche sur la variable dans le vecteur  $x$  ayant la plus petite valeur dans son domaine et l'affecte à cette valeur ».