

Planification de Tournoi Sportif par Programmation par Contraintes

Projet de Programmation par Contraintes

10 avril 2025

Table des matières

1	Introduction	2
2	Modélisation avec OR-Tools (Python)	2
2.1	Variables et contraintes	2
2.2	Contraintes principales	2
3	Modélisation avec Choco-Solver (Java)	2
3.1	Description	2
3.2	Comparaison	2
4	Modélisation avec Z3 (Python)	2
4.1	Description	2
4.2	Limites	2
5	Exécution via ChatGPT	3
5.1	Automatisation du solving avec API OpenAI	3
5.2	Méthodologie	3
5.3	Résultat	3
6	Benchmarks comparatifs	3
6.1	OR-Tools	3
6.2	Z3	3
7	Analyse comparative	3
7.1	OR-Tools vs Choco	3
7.2	OR-Tools vs Z3	3
8	Conclusion	4

1 Introduction

La planification de tournois sportifs avec des contraintes complexes constitue un problème classique en Programmation par Contraintes (CP). Ce projet a pour objectif de modéliser un championnat round-robin en respectant des contraintes telles que l’alternance domicile/extérieur, la disponibilité des stades, ou encore la minimisation des enchaînements non alternés appelés “breaks”.

2 Modélisation avec OR-Tools (Python)

2.1 Variables et contraintes

Nous avons utilisé la bibliothèque `OR-Tools` pour modéliser ce problème en Python. Les principales variables sont :

- `X[i, j, d]` : 1 si l’équipe i reçoit l’équipe j au jour d
- `is_home[i,d]`, `is_away[i,d]` : statut de l’équipe i pour le jour d
- `B[i,d]` : 1 si l’équipe i a un break entre d et $d + 1$

2.2 Contraintes principales

- Une rencontre unique entre chaque paire d’équipes
- Une équipe joue un seul match par jour
- Pas plus de X déplacements consécutifs
- Disponibilités des stades respectées
- Objectif : minimiser le nombre total de breaks

3 Modélisation avec Choco-Solver (Java)

3.1 Description

Une solution parallèle a été implémentée en Java avec Choco-Solver. Le modèle utilise des variables `IntVar` pour les équipes et les jours, ainsi que des contraintes `allDifferent`, `count`, `ifThenElse`, etc.

3.2 Comparaison

Bien que Choco offre une modélisation fine et une intégration Java native, les temps de résolution sont moins bons que ceux d’OR-Tools et le modèle est plus verbeux.

4 Modélisation avec Z3 (Python)

4.1 Description

Z3 a été utilisé comme solveur SMT pour modéliser le même problème. Les variables sont manipulées via des expressions logiques (`If`, `Implies`, `Sum`). L’optimisation est formulée avec `opt.minimize()`.

4.2 Limites

Z3 n’étant pas conçu spécifiquement pour la CP, il est moins performant : il manque de contraintes globales optimisées et de stratégie de recherche.

5 Exécution via ChatGPT

5.1 Automatisation du solving avec API OpenAI

Une expérimentation complémentaire a été réalisée en connectant le solveur OR-Tools à l'API OpenAI, à travers une fonction déclarée dans le modèle `gpt-4-0613`. Cette approche permet d'invoquer dynamiquement la génération de planning à travers une interface de dialogue.

5.2 Méthodologie

- Définition d'une fonction `solve_schedule` comme appel distant depuis ChatGPT.
- Appel avec les paramètres `num_teams`, `start_date`, `max_consecutive_away` via `function_call`.
- Résolution avec OR-Tools et retour du planning formaté.

5.3 Résultat

Le modèle a généré une réponse structurée contenant le calendrier complet et le nombre total de breaks. Cette démonstration valide la capacité d'intégrer OR-Tools dans un flux IA pour la génération interactive de plannings sportifs.

6 Benchmarks comparatifs

6.1 OR-Tools

Nombre d'équipes	Temps (s)	Statut	Breaks
6	0.115	Faisable	4
8	1.899	Faisable	6
10	27.423	Faisable	8
12	74.524	Faisable	10
14	11.013	Faisable	12
16	21.640	Faisable	14
18	184.361	Faisable	16
20	341.186	Faisable	18

6.2 Z3

Nombre d'équipes	Temps (s)	Statut	Breaks
6	19.580	Faisable	4
8	575.681	Faisable	6

7 Analyse comparative

7.1 OR-Tools vs Choco

OR-Tools est plus concis et rapide grâce à ses contraintes globales optimisées. Choco est pertinent pour une intégration Java mais moins efficace en résolution.

7.2 OR-Tools vs Z3

OR-Tools surpasse largement Z3 en performance et en capacité à scaler. Z3 reste utile pour des modèles logiques ou symboliques, mais inadapté à grande échelle pour la CP pure.

8 Conclusion

OR-Tools s'est révélé être le meilleur outil pour ce type de problème. Il combine performance, simplicité de modélisation, et scalabilité. Choco et Z3 permettent des alternatives, mais au prix de performances ou de complexité accrue.