MATH 60617

Mise en œuvre de modèles d'optimisation en gestion

Travail 2

Présenté à

Pr. Sylvain Perron

Par

Christelle GEORGE – 11288106

Samy SENOUNE – 11290255

Joe YOUNES – 11285603

Le 3 mai 2021

À Montréal

Table des matières

Introduction 1
Mise en contexte 2
Définition du problème 3
Données 3
Variables de décision 3
Fonctions objectif 4
Contraintes 5
Analyse des résultats 8
Conclusion 11
Référence 11
Annexes 12
Données du problème du programme OPL 12
Variable de décision du programme OPL 12
Fonction objectif du programme OPL 12
Contraintes du programme OPL 13

Liste des figures

Figure 1 : Déplacement de l'équipe d'Ottawa 9
Figure 2 : Déplacement de l'équipe d'Edmonton 9

Liste des tableaux

Tableau 1 : Calendrier des matchs de la saison régulière 8
Tableau 2 : Tableau des distances parcourues par chaque équipe 8
Tableau 3 : Occupation des arénas 9
Tableau 4 : Nombre de long home stands et road trips 10
Tableau 5 : Analyse de sensibilité de la contrainte de la limite de matchs consécutifs 10

Introduction

La Ligne Nationale de Hockey (LNH) est une association sportive professionnelle nord-américaine qui prend place de septembre à juin, chaque année. Elle se déroule en 2 temps : d'une part, la saison régulière (octobre-avril) pendant laquelle toutes les équipes s'affrontent et d'autre part, la série éliminatoire (*playoff*, avril-juin) qui se déroule entre les meilleures équipes issues de la première phase.

Ce rapport se concentre uniquement sur la première phase du tournoi, soit la saison régulière. Il s'agit de concevoir un horaire de jeu pour les équipes sportives qui permet de respecter les règles propres à la LNH. La particularité de cette étude relève du contexte sanitaire de la Covid-19, qui a donc modifié certaines règles, notamment les équipes impliquées dans le tournoi : dans ce cas, les équipes américaines sont écartées. Seules les 7 équipes canadiennes sont considérées. Il s'agit des *Canadiens* de Montréal, *Maple Leafs* de Toronto, *Sénateurs* d'Ottawa, *Jets* de Winnipeg, *Flames* de Calgary, *Oilers* d'Edmonton et *Canucks* de Vancouver.

La saison régulière se déroule selon un tournoi « double round robin » : chacune de ses 7 équipes va affronter les autres 2 fois ; 1 fois à domicile et 1 fois à l'extérieur. Ainsi, chaque équipe disputera un total de 12 matchs. Globalement, 42 matchs se dérouleront pendant la saison régulière. Finalement, l'attribution des points se fait de la façon suivante : une victoire correspond à 2 points, une défaite en prolongation ou lors des tirs de fusillade vaut 1 point et une défaite lors du temps réglementaire 0 point.

Avec cette approche de « double round robin », le hasard est minimisé puisque toutes les équipes se rencontrent et ont l'opportunité de gagner malgré une mauvaise performance antérieure. Ceci dit, une certaine asymétrie est susceptible de surgir en fin de tournoi, lorsque par exemple une équipe ayant déjà accumulé un bon nombre de points – donc une qualification assurée – pourrait choisir de perdre délibérément. Une question d'équité se pose alors.

Somme toute, concevoir l'horaire de jeu pour les 7 équipes canadiennes de la LNH relève d'un problème de « Traveling Tournament Problem » (TTP) qui s'articule autour de 2 contraintes traditionnelles fondamentales.

Dans un premier temps, il est important de tenir compte de la question de faisabilité : les déplacements des équipes selon l'horaire obtenu doivent éviter une longue période de jeu à domicile (*home stands*) ainsi qu'une longue période successive de déplacement à l'extérieur (*road trips*). En fait, une courte période à domicile est cruciale pour permettre l'entretien de la glace de l'aréna qui se dégrade après 3 matchs successifs. De même, une courte période à l'extérieur permet d'optimiser la performance des athlètes qui ne seraient donc pas épuisés par une longue route.

Dans un deuxième temps, il est primordial d'éviter des voyages excessifs. En d'autres termes, il s'agit de minimiser la distance totale parcourue par les équipes.

Mise en contexte

La revue de littérature a mis en évidence une approche combinatoire pour la résolution de ce type de problème TTP : il s'agit d'utiliser la programmation en nombres entiers ainsi que la programmation par contrainte. Lorsqu'utilisées indépendamment, la programmation en nombres entiers permet de résoudre le problème de « Traveling Salesman » (TSP) , soit de réduire la distance totale parcourue. Ce problème classique a d'ailleurs fait l'objet d'une séance du cours d'optimisation et les similitudes entre le problème du TSP et ce rapport seront identifiées au fur et à mesure. La programmation par contrainte, quant à elle, est très pratique lorsqu'il s'agit d'obtenir les combinaisons adéquates de jeux à domicile et à l'extérieur. C'est en combinant les deux approches que des résultats plus rapides sont obtenus, dans le cadre d'un problème de TTP. Ainsi, dans le cadre de cette étude, c'est également une approche combinatoire qui est employée.

D'autre part, il est important de mentionner que le tournoi « double round robin » suppose un nombre paire d'équipe pour l'appariement, sans quoi une équipe serait mise de côté à chaque étape. Ainsi, l'étude étant une simplification d'un problème plus complexe, l'équipe de Vancouver est écartée de l'analyse afin d'assurer un nombre paire de 6 équipes.

Les hypothèses de notre étude sont les suivantes :

- Chaque équipe dispute exactement 1 match par jour ;
- Chaque équipe joue une fois à domicile et une fois à l'extérieur contre tous ses adversaires ;
- Chaque équipe commence sa tournée et termine sa tournée à domicile. Attention, ceci n'implique pas forcément que l'équipe dispute son premier match à domicile mais assure que le calcul des distances s'effectue à partir du domicile ;
- Chaque équipe dispute un maximum de 3 matchs consécutifs à domicile – soit des *home stands* de 3 jours maximum ;
- Chaque équipe dispute un maximum de 3 matchs consécutifs à l'extérieur – soit des *road trips* de 3 jours maximum ;
- La compétition se déroule sur une période de 10 jours et les équipes n'ont aucun jour de repos.

Finalement, le temps d'exécution sur OPL est limité à 10 minutes.

Définition du problème

Données

Les données du problèmes sont les suivantes :

- L'ensemble des équipes :
$$Teams = \{1 = « Montréal », 2 = « Toronto », 3 = « Ottawa », 4 = « Winnipeg », 5 = « Calgary », 6 = « Edmonton »\}$$
- Le nombre de match joué par équipe : $N_i = 10$
- L'ensemble I des matchs joués par équipe : $I = 1 \dots N_i$
- L'ensemble S des déplacements incluant le départ à domicile : $S = 1 \dots N_i + 1$
- Un indicateur indiquant si le match a lieu à l'extérieur ou à domicile :
$$Home = \{Extérieur = «0», Domicile = «1»\}$$

- La matrice des distances en km entre les arénas des équipes :
$$D_{Teams,Teams} = \begin{bmatrix} 0 & 547.6 & 198.2 & 2269.1 & 3526.2 & 3572.1 \\ 547.6 & 0 & 402.0 & 2222.7 & 3406.0 & 3465.3 \\ 198.2 & 402.0 & 0 & 2139.7 & 3330.3 & 3442.7 \\ 2269.1 & 2222.7 & 2139.7 & 0 & 1329.1 & 1304.9 \\ 3526.2 & 3406.0 & 3330.3 & 1329.1 & 0 & 299.5 \\ 3572.1 & 3465.3 & 3442.7 & 1304.9 & 299.5 & 0 \end{bmatrix}$$

Par rapport au problème du TSP vu dans le cours, on voit que la matrice des distances ainsi que l'ensemble des équipes (ou clients) sont des éléments communs aux 2 problèmes. Cependant, notre problème considère davantage de données, telles que le nombre de match joué par équipe et le lieu du match (à domicile ou à l'extérieur).

Variables de décision

12 variables de décisions sont définies dans l'étude, soit 2 par equipe. Pour chaque équipe, une variable permet de retracer les déplacement de son parcours et une variable binaire indique ses matchs :

- $Ville\_precedente\_Mtl_S \in N^+$, ville dans laquelle se situe Montréal avant le déplacement S
- $Ville\_precedente\_Tor_S \in N^+$, ville dans laquelle se situe Toronto avant le déplacement S
- $Ville\_precedente\_Ott_S \in N^+$, ville dans laquelle se situe Ottawa avant le déplacement S
- $Ville\_precedente\_Win_S \in N^+$, ville dans laquelle se situe Winnipeg avant le déplacement S
- $Ville\_precedente\_Cal_S \in N^+$, ville dans laquelle se situe Calgary avant le déplacement S
- $Ville\_precedente\_Edm_S \in N^+$, ville dans laquelle se situe Edmonton avant le déplacement S
- $Mtl_{I,Teams,Home} \in \{0,1\}$, vaut 1 si Montréal affronte l'équipe $Teams$ à l'aréna $Home$ lors du match I
- $Tor_{I,Teams,Home} \in \{0,1\}$, vaut 1 si Toronto affronte l'équipe $Teams$ à l'aréna $Home$ lors du match I
- $Ott_{I,Teams,Home} \in \{0,1\}$, vaut 1 si Ottawa affronte l'équipe $Teams$ à l'aréna $Home$ lors du match I
- $Win_{I,Teams,Home} \in \{0,1\}$, vaut 1 si Winnipeg affronte l'équipe $Teams$ à l'aréna $Home$ lors du match I
- $Cal_{I,Teams,Home} \in \{0,1\}$, vaut 1 si Calgary affronte l'équipe $Teams$ à l'aréna $Home$ lors du match I
- $Edm_{I,Teams,Home} \in \{0,1\}$, vaut 1 si Edmonton affronte l'équipe $Teams$ à l'aréna $Home$ lors du match I

Pour établir un parallèle avec le problème de TSP étudié dans le cours, le TSP possède une logique d'exécution qui sera reprise dans ce rapport. Dans le TSP, une variable binaire est employée pour retracer le parcours du voyageur, soit x_{ij} qui vaut 1 si le nœud *j* suit immédiatement le nœud *i* dans le circuit, 0 sinon. Dans ce rapport, c'est la variable $Ville\_precedente$ qui est privilégiée. Elle prend les valeurs entre 1 et 6 représentant les 6 villes des 6 équipes à l'étude et permet de situer à chaque déplacement S la position géographique de l'équipe en question.

Ceci dit, le problème de TSP est trop simple pour contenir toute l'information à représenter dans le problème du tournoi des matchs TTP. À titre d'exemple, le TSP permet d'indiquer les adversaires d'un match lorsque celui-ci se déroule à l'extérieur puisque l'adversaire correspond à la ville en question. Malheureusement, une de ses limitations serait lors d'un match à domicile puisque dans ce cas, l'adversaire serait inconnu. Pour remédier à ce problème, la variable binaire $Ville_{I,Teams,Home}$ est définie dans le TTP. Pour chaque équipe, il est alors possible de connaitre son adversaire ainsi que le lieu de chaque match, soit à domicile ou à l'extérieur.

Fonctions objectif

Comme mentionné dans l'introduction, l'objectif le plus important est de minimiser la somme des distances parcourues par l'ensemble des équipes lors de leurs déplacements :

$$
\begin{aligned}
Min \sum_{t=2}^{N_i+1} & D_{Ville\_precedente\_Mtl_{t-1},\, Ville\_precedente\_Mtl_t} + D_{Ville\_precedente\_Mtl_{N_i+1},1} \\
& + \sum_{t=2}^{N_i+1} D_{Ville\_precedente\_Tor_{t-1},\, Ville\_precedente\_Tor_t} + D_{Ville\_precedente\_Tor_{N_i+1},2} \\
& + \sum_{t=2}^{N_i+1} D_{Ville\_precedente\_Ott_{t-1},\, Ville\_precedente\_Ott_t} + D_{Ville\_precedente\_Ott_{N_i+1},3} \\
& + \sum_{t=2}^{N_i+1} D_{Ville\_precedente\_Win_{t-1},\, Ville\_precedente\_Win_t} + D_{Ville\_precedente\_Win_{N_i+1},4} \\
& + \sum_{t=2}^{N_i+1} D_{Ville\_precedente\_Cal_{t-1},\, Ville\_precedente\_Cal_t} + D_{Ville\_precedente\_Cal_{N_i+1},5} \\
& + \sum_{t=2}^{N_i+1} D_{Ville\_precedente\_Edm_{t-1},\, Ville\_precedente\_Edm_t} + D_{Ville\_precedente\_Edm_{N_i+1},6}
\end{aligned}
$$

La fonction objectif est donc identique à celle défini dans le cours portant sur le TSP. Elle est bien entendu adaptée aux variables de décisions du TTP. Grâce à la programmation par contrainte, les indices de la matrice de distance correspondent directement aux trajets en question avec les variables de décision aux indices *t-1* et *t*. Pour chaque déplacement, il s'agit de sommer sur toutes les équipes la distance liant la ville précédente à la ville actuelle de l'équipe pour ensuite additionner tous les pas de déplacements. Il est important de noter qu'une dernière distance est additionnée : celle de la dernière ville dans laquelle se trouve l'équipe avec son domicile. Ceci permet, dans le cas ou une équipe dispute son dernier match à l'extérieur, de prendre en compte son déplacement de retour à domicile dans l'analyse.

Contraintes

Il y a donc une multitude de contraintes à implémenter afin de respecter les hypothèses énoncées plus haut dans la mise en contexte, et d'assurer une certaine logique dans la résolution.

Les contraintes du problème du TTP sont différentes de celles du TSP, et ceci est en parti lié à la définition des variables de décision ainsi qu'au contexte propre au problème étudie. À titre d'exemple, les contraintes pour éliminer les arcs impossibles du TSP de s'appliquent pas dans ce rapport. Il est en effet possible pour les équipes de hockey de rentrer dans leur « nœud » de départ ou de sortir de leur « nœud » d'arrivée, c'est à dire de devoir disputer un match à domicile pendant le tournoi. Également, il est probable que des équipes entrent et sortent plusieurs fois dans chaque nœud, soit dans chaque ville, pour disputer des matchs.

Les contraintes à considérées sont donc présentées ci bas.

Chaque équipe commence son parcours à domicile :

- $Ville\_precedente\_Mtl_1 = 1$
- $Ville\_precedente\_Tor_1 = 2$
- $Ville\_precedente\_Ott_1 = 3$
- $Ville\_precedente\_Win_1 = 4$
- $Ville\_precedente\_Cal_1 = 5$
- $Ville\_precedente\_Edm_1 = 6$

Si une équipe a joué son match précédent à l'extérieur, alors son déplacement se poursuit à partir de cette ville extérieure :

- $\forall\ i \in \{2..N_i + 1\}, j\ \in Teams \begin{cases} Mtl_{i-1,j,"0"} = 1 \Rightarrow Ville\_precedente\_Mtl_i = j \\ Tor_{i-1,j,"0"} = 1 \Rightarrow Ville\_precedente\_Tor_i = j \\ Ott_{i-1,j,"0"} = 1 \Rightarrow Ville\_precedente\_Ott_i = j \\ Win_{i-1,j,"0"} = 1 \Rightarrow Ville\_precedente\_Win_i = j \\ Cal_{i-1,j,"0"} = 1 \Rightarrow Ville\_precedente\_Cal_i = j \\ Edm_{i-1,j,"0"} = 1 \Rightarrow Ville\_precedente\_Edm_i = j \end{cases}$

Si une équipe a joué son match précédent à la maison, alors son déplacement se poursuit à partir de son domicile :

- $\forall\ i \in \{2..N_i + 1\}, j\ \in Teams \begin{cases} Mtl_{i-1,j,"1"} = 1 \Rightarrow Ville\_precedente\_Mtl_i = 1 \\ Tor_{i-1,j,"1"} = 1 \Rightarrow Ville\_precedente\_Tor_i = 2 \\ Ott_{i-1,j,"1"} = 1 \Rightarrow Ville\_precedente\_Ott_i = 3 \\ Win_{i-1,j,"1"} = 1 \Rightarrow Ville\_precedente\_Win_i = 4 \\ Cal_{i-1,j,"1"} = 1 \Rightarrow Ville\_precedente\_Cal_i = 5 \\ Edm_{i-1,j,"1"} = 1 \Rightarrow Ville\_precedente\_Edm_i = 6 \end{cases}$

Chaque équipe ne peut pas jouer contre elle-même :

- $\forall\, i \in \{1..N_i\}, k \in Home \begin{cases} Mtl_{i,1,k} = 0 \\ Tor_{i,2,k} = 0 \\ Ott_{i,3,k} = 0 \\ Win_{i,4,k} = 0 \\ Cal_{i,5,k} = 0 \\ Edm_{i,6,k} = 0 \end{cases}$

Chaque équipe joue selon les règles du tournoi « double round robin », donc joue 1 fois à domicile et 1 fois à l'extérieure contre chacune des autres équipes :

- $\sum_{i \in I} Mtl_{i,Teams,k} = 1, \quad \forall\, Teams \in \{2,3,4,5,6\}, k \in Home$
- $\sum_{i \in I} Tor_{i,Teams,k} = 1, \quad \forall\, Teams \in \{1,3,4,5,6\}, k \in Home$
- $\sum_{i \in I} Ott_{i,Teams,k} = 1, \quad \forall\, Teams \in \{1,2,4,5,6\}, k \in Home$
- $\sum_{i \in I} Win_{i,Teams,k} = 1, \quad \forall\, Teams \in \{1,2,3,5,6\}, k \in Home$
- $\sum_{i \in I} Cal_{i,Teams,k} = 1, \quad \forall\, Teams \in \{1,2,3,4,6\}, k \in Home$
- $\sum_{i \in I} Edm_{i,Teams,k} = 1, \quad \forall\, Teams \in \{1,2,3,4,5\}, k \in Home$

Chaque équipe joue 1 match par jour :

- $\forall\, i \in I \begin{cases} \sum_{j \in Teams,\ k \in Home} Mtl_{i,j,k} = 1 \\ \sum_{j \in Teams,\ k \in Home} Tor_{i,j,k} = 1 \\ \sum_{j \in Teams,\ k \in Home} Ott_{i,j,k} = 1 \\ \sum_{j \in Teams,\ k \in Home} Win_{i,j,k} = 1 \\ \sum_{j \in Teams,\ k \in Home} Cal_{i,j,k} = 1 \\ \sum_{j \in Teams,\ k \in Home} Edm_{i,j,k} = 1 \end{cases}$

Chaque équipe joue un maximum de 3 matchs consécutifs à l'extérieur :

- $\forall\, i \in \{1..N_i - 3\} \begin{cases} \sum_{z \in \{0..3\}, j \in Teams} Mtl_{i+z,j,"0"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Tor_{i+z,j,"0"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Ott_{i+z,j,"0"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Win_{i+z,j,"0"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Cal_{i+z,j,"0"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Edm_{i+z,j,"0"} \le 3 \end{cases}$

Chaque équipe joue un maximum de 3 matchs consécutifs à domicile :

- $\forall\, i \in \{1..N_i - 3\} \begin{cases} \sum_{z \in \{0..3\}, j \in Teams} Mtl_{i+z,j,"1"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Tor_{i+z,j,"1"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Ott_{i+z,j,"1"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Win_{i+z,j,"1"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Cal_{i+z,j,"1"} \le 3 \\ \sum_{z \in \{0..3\}, j \in Teams} Edm_{i+z,j,"1"} \le 3 \end{cases}$

En denier lieu, il est important de définir les contraintes liantes entre les équipes. À titre d'exemple, le match de l'équipe de Montréal l'opposant à Toronto à l'extérieur est équivalent au match de l'équipe de Toronto l'opposant à Montréal à domicile.

- $$\forall\, i \in I \quad \begin{cases} Mtl_{i,2,0} = Tor_{i,1,1} \\ Mtl_{i,2,1} = Tor_{i,1,0} \\ Mtl_{i,3,0} = Ott_{i,1,1} \\ Mtl_{i,3,1} = Ott_{i,1,0} \\ Mtl_{i,4,0} = Win_{i,1,1} \\ Mtl_{i,4,1} = Win_{i,1,0} \\ Mtl_{i,5,0} = Cal_{i,1,1} \\ Mtl_{i,5,1} = Cal_{i,1,0} \\ Mtl_{i,6,0} = Edm_{i,1,1} \\ Mtl_{i,6,1} = Edm_{i,1,0} \\ Tor_{i,3,0} = Ott_{i,2,1} \\ Tor_{i,3,1} = Ott_{i,2,0} \\ Tor_{i,4,0} = Win_{i,2,1} \\ Tor_{i,4,1} = Win_{i,2,0} \\ Tor_{i,5,0} = Cal_{i,2,1} \\ Tor_{i,5,1} = Cal_{i,2,0} \\ Tor_{i,6,0} = Edm_{i,2,1} \\ Tor_{i,6,1} = Edm_{i,2,0} \\ Ott_{i,4,0} = Win_{i,3,1} \\ Ott_{i,4,1} = Win_{i,3,0} \\ Ott_{i,5,0} = Cal_{i,3,1} \\ Ott_{i,5,1} = Cal_{i,3,0} \\ Ott_{i,6,0} = Edm_{i,3,1} \\ Ott_{i,6,1} = Edm_{i,3,0} \\ Win_{i,5,0} = Cal_{i,4,1} \\ Win_{i,5,1} = Cal_{i,4,0} \\ Win_{i,6,0} = Edm_{i,4,1} \\ Win_{i,6,1} = Edm_{i,4,0} \\ Cal_{i,6,0} = Edm_{i,5,1} \\ Cal_{i,6,1} = Edm_{i,5,0} \end{cases}$$

Analyse des résultats

Pour ce qui est de l'analyse des résultats, il est d'abord important de noter que les valeurs que nous avons obtenues sont les meilleurs résultats qu'OPL studio a trouvé après 10 minutes d'exécution du programme.

Premièrement, le calendrier du tournoi est généré. Celui-ci respecte toutes les contraintes mentionnées plus haut et indique les matchs selon le jour du tournoi. Les équipes sur la même ligne s'affrontent.

Tableau 1 : Calendrier des matchs de la saison régulière

Jour 1		Jour 2		Jour 3		Jour 4		Jour 5	
Extérieur	**Domicile**	**Extérieur**	**Domicile**	**Extérieur**	**Domicile**	**Extérieur**	**Domicile**	**Extérieur**	**Domicile**
Montréal	Toronto	Ottawa	Toronto	Toronto	Montréal	Toronto	Ottawa	Winnipeg	Toronto
Winnipeg	Ottawa	Winnipeg	Montréal	Edmonton	Ottawa	Montréal	Calgary	Ottawa	Calgary
Calgary	Edmonton	Edmonton	Calgary	Calgary	Winnipeg	Edmonton	Winnipeg	Montréal	Edmonton
Jour 6		**Jour 7**		**Jour 8**		**Jour 9**		**Jour 10**	
Extérieur	**Domicile**	**Extérieur**	**Domicile**	**Extérieur**	**Domicile**	**Extérieur**	**Domicile**	**Extérieur**	**Domicile**
Calgary	Toronto	Edmonton	Toronto	Toronto	Winnipeg	Toronto	Edmonton	Toronto	Calgary
Ottawa	Edmonton	Ottawa	Winnipeg	Calgary	Ottawa	Montréal	Ottawa	Ottawa	Montréal
Montréal	Winnipeg	Calgary	Montréal	Edmonton	Montréal	Winnipeg	Calgary	Winnipeg	Edmonton

Tel que mentionné, toutes les équipes jouent 2 fois l'une contre l'autre, dont une fois à domicile et une fois à l'extérieur. Dans le calendrier, il est important de relever que le programme OPL a tendance à faire jouer les équipes avec les distances les plus courtes dans les premiers jours. Il prend ensuite de plus en plus de matchs affrontant deux équipes à de grandes distances. Par exemple, sachant que les distances entre Montréal, Ottawa et Toronto sont relativement courtes, de même pour Calgary et Edmonton, et que Winnipeg se trouve au milieu des 5 autres villes, il apparait que dès le jour 1, le calendrier indique 2 matchs « courte distance » et un match « moyenne distance ». Cette tendance se poursuit le jour 2 et commence à se dégrader à partir du jour 3. Par contraste, le jour 8 – par exemple – comporte un match « moyenne distance » et 2 matchs « longue distance ».

D'autre part, un autre pattern se dessine : les équipes qui sont sur un *road trips* et qui passent par Montréal vont aussi passer par Ottawa (voir Winnipeg au jours 1 et 2). Il en va de même pour les équipes en *road trips* passant par Calgary, qui vont alors passer par Edmonton (voir Montréal aux jours 4 et 5, Ottawa aux jour 5 et 6 et Toronto aux jour 9 et 10).

En somme, ce calendrier minimisait la somme des distances parcourues par toutes les équipes avec une distance totale de **62 931,6 km**. Le tableau suivant indique les distances parcourues par chacune des équipes :

Tableau 2 : Tableau des distances parcourues par chaque équipe

Ville	Montréal	Toronto	Ottawa	Winnipeg	Calgary	Edmonton	**Total**
Distance [km]	8891.3	8380,9	8274,8	11985,9	10739,3	14659,4	**62 931,6**

L'équipe d'Ottawa est donc l'équipe qui se déplace le moins avec 8274,8 km ; à contrario, l'équipe d'Edmonton est celle qui se déplace le plus avec 14659,4 km. Leur trajet respectif est résumé dans les figures suivantes :

Figure 1 : Déplacement de l'équipe d'Ottawa

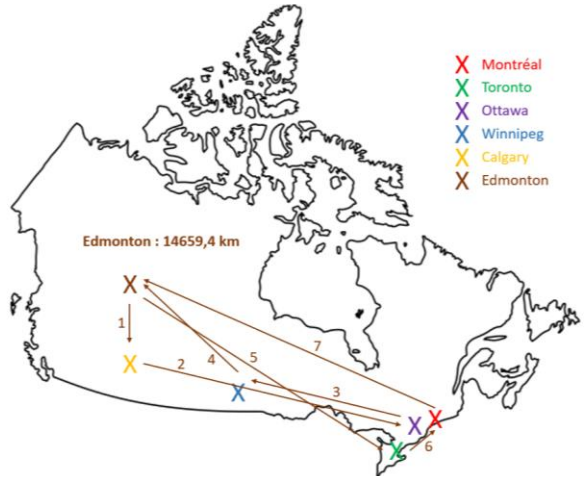

Figure 2 : Déplacement de l'équipe d'Edmonton

Deuxièmement, le problème est considéré d'un point de vue des arénas. En effet, il est important de savoir quelles arénas seront utilisées à quels jours pour pouvoir planifier la maintenance des patinoires et des installations ainsi qu'effectuer la réservation des infrastructures nécessaires pour qu'il n'aille pas d'événements conflictuels avec les matchs du tournoi. Le tableau suivant présente l'occupation des arènes *Centre Bell* (Montréal), *Scotiabank Arena* (Toronto), *Canadian Tire Centre* (Ottawa), *Bell MTS Place* (Winnipeg), *Scotiabank Saddledome* (Calgary) et *Rogers Place* (Edmonton) selon le jour du match et l'équipe qu'ils reçoivent.

Tableau 3 : Occupation des arénas

Montréal		**Toronto**		**Ottawa**	
Centre Bell		***Scotiabank Arena***		***Canadian Tire Centre***	
Jour	Visiteurs	Jour	Visiteurs	Jour	Visiteurs
2	Winnipeg	1	Montréal	1	Winnipeg
3	Toronto	2	Ottawa	3	Edmonton
7	Calgary	5	Winnipeg	4	Toronto
8	Edmonton	6	Calgary	8	Calgary
10	Ottawa	7	Edmonton	9	Montréal
Winnipeg		**Calgary**		**Edmonton**	
Bell MTS Place		***Scotiabank Saddledome***		***Rogers Place***	
Jour	Visiteurs	Jour	Visiteurs	Jour	Visiteurs
3	Calgary	2	Edmonton	1	Calgary
4	Edmonton	4	Montréal	5	Montréal
6	Montréal	5	Ottawa	6	Ottawa
7	Ottawa	9	Winnipeg	9	Toronto
8	Toronto	10	Toronto	10	Winnipeg

La contrainte définie d'un nombre maximal de 3 jours consécutifs à domicile se reflète dans ce tableau : chacune des arénas possède des temps de pause qui seront importants pour une maintenance ou une certaine logistique. Aussi, on note que les arènes de Toronto et Winnipeg exploitent la contrainte à sa limite avec 3 jours consécutifs de matchs disputés.

Troisièmement, le nombre de *home stands* et de *road trips* de 3 jours consécutifs pour chaque équipe est analysé :

Tableau 4 : Nombre de long home stands et road trips

	Nombre de *Home stands* de 3 jours	**Nombre de *Road trips* de 3 jours**
Montréal	0	1
Toronto	1	1
Ottawa	0	1
Winnipeg	1	0
Calgary	0	1
Edmonton	0	1

Les observation suivantes sont tirées :

- Winnipeg est la seule équipe qui a uniquement un *home stand* de 3 jours consécutifs lorsqu'elle reçoit Montréal, Ottawa et Toronto ;
- Toronto est la seule équipe avec à la fois avec un *home stand* de 3 jours et un *road trip* de 3 jours.
- Les 4 autres équipes ont uniquement des *road trips* de 3 jours consécutifs. Dans le cas des équipes de l'Est, i.e. Montréal, Toronto et Ottawa, elles visitent de façon consécutive les 3 équipes de l'Ouest (Winnipeg, Calgary et Edmonton), et vice versa pour Calgary et Edmonton qui visitent les 3 équipes de l'Est de façon consécutive. Cette tendance est normale puisque c'est la façon la plus efficace de limiter l'impact de la distance des matchs à l'extérieur.

Finalement, une analyse de sensibilité est menée sur la contrainte « ne pas avoir plus de 3 matchs consécutifs à domicile ou à l'extérieur » afin d'évaluer son impact sur l'objectif. 6 scénarios différents sont définis :

- 3 matchs à domicile et 2 à l'extérieur ;
- 2 matchs à domicile et 3 à l'extérieur ;
- 2 matchs à domicile et 2 à l'extérieur ;
- 4 matchs à domicile et 3 à l'extérieur ;
- 3 matchs à domicile et 4 à l'extérieur ;
- 4 matchs à domicile et 4 à l'extérieur.

Les effets obtenus sur la distance totale parcourue par toutes les équipes sont résumés dans le tableau suivant :

Tableau 5 : Analyse de sensibilité de la contrainte de la limite de matchs consécutifs

Limite de matchs consécutifs		**Distance totale (km)**
À l'extérieur	**À domicile**	
3	3	62 931,60

2	3	79 483,40
3	2	70 665,00
2	2	82 188,70
4	3	59 719,00
3	4	58 784,90
4	4	55 860,70

Ce tableau indique qu'augmenter la limite à 4 matchs consécutifs pour les deux types de matchs minimise la fonction-objectif avec 55 860,70 km parcourus (une amélioration de 7 070,90 km par rapport à la solution originale de 62 931,60 km). À contrario, réduire la limite à 2 matchs consécutifs pour les deux types de matchs entraine le pire résultat avec 82 188,70 km parcours, soit une détérioration de 19 257,10 km par rapport à la solution originale de 62 931,60 km. En d'autres mots, le changement marginal de 3 matchs à 2 matchs est beaucoup plus important que le changement marginal de 3 matchs à 4 matchs.

De plus, il y a une différence entre la modification de la limite sur les matchs consécutifs à domicile et celle sur les matchs consécutifs à l'extérieur. En moyenne, la valeur absolue d'une variation de 1 sur la limite de matchs consécutifs à l'extérieur est de 9 882,2 km, comparativement à 5 940,05 km pour une variation de 1 sur la limite de matchs consécutifs à domicile.

Conclusion

En conclusion, le « Traveling Tournament Problem » est un cas fort intéressant qui a été simplifié dans le cadre de ce travail. Dans un contexte réel, des contraintes additionnelles devraient être rajoutées. À titre d'exemple, les coûts de transport seraient pertinents à considérer dans l'analyse, tout comme l'ajout de jours de repos entre les matchs, ou des jours de pratique minimum dans la saison.

Une analyse à objectifs multiples pourrait également être menée, avec les objectifs de minimisation de la distance et des coûts. Ceci permettrait, somme toute, d'obtenir dans le meilleurs des cas des solutions Pareto optimales avec la courbe de Pareto.

Finalement, une des limites de cette étude réside dans le nombre pair d'équipe mandaté par le tournoi « double round robin ». Pour aller plus loin, l'équipe de Vancouver devrait être considérée quitte à avoir recours à un adversaire « imaginaire » pour obtenir un chiffre pair et procéder aux appariements. En soit, cela implique qu'à chaque jour, une équipe ne disputera pas de match.

Référence

Easton, Kelly & Nemhauser, George & Trick, Michael. (2002). Solving the Travelling Tournament Problem: A Combined Integer Programming and Constraint Programming Approach. 100-112. 10.1007/978-3-540-45157-0_6.

Annexes

Données du problème du programme OPL

```
Home = {"0", "1"};

N_i = 10;

Distance= [
[0 547.6 198.2 2269.1 3526.2 3572.1]
[547.6 0 402.0 2222.7 3406.0 3465.3]
[198.2 402.0 0 2139.7 3330.3 3442.7]
[2269.1 2222.7 2139.7 0 1329.1 1304.9]
[3526.2 3406.0 3330.3 1329.1 0 299.5]
[3572.1 3465.3 3442.7 1304.9 299.5 0]
];
```

```
using CP;

range Teams=1..6;
int N_i = ...;
range I = 1..N_i;
range S = 1..(N_i+1);
setof(string) Home=...;
float Distance[Teams][Teams]= ...;
```

Variable de décision du programme OPL

```
dvar int Ville_precedente_Mtl[S];
dvar int Ville_precedente_Tor[S];
dvar int Ville_precedente_Ott[S];
dvar int Ville_precedente_Win[S];
dvar int Ville_precedente_Cal[S];
dvar int Ville_precedente_Edm[S];


dvar boolean Mtl[I][Teams][Home];
dvar boolean Tor[I][Teams][Home];
dvar boolean Ott[I][Teams][Home];
dvar boolean Win[I][Teams][Home];
dvar boolean Cal[I][Teams][Home];
dvar boolean Edm[I][Teams][Home];
```

Fonction objectif du programme OPL

```
minimize sum(test in 2..N_i+1) (Distance[Ville_precedente_Mtl[test-1]][Ville_precedente_Mtl[test]]) + Distance[Ville_precedente_Mtl[N_i+1]][1] +
         sum(test in 2..N_i+1) (Distance[Ville_precedente_Tor[test-1]][Ville_precedente_Tor[test]]) + Distance[Ville_precedente_Tor[N_i+1]][2] +
         sum(test in 2..N_i+1) (Distance[Ville_precedente_Ott[test-1]][Ville_precedente_Ott[test]]) + Distance[Ville_precedente_Ott[N_i+1]][3] +
         sum(test in 2..N_i+1) (Distance[Ville_precedente_Win[test-1]][Ville_precedente_Win[test]]) + Distance[Ville_precedente_Win[N_i+1]][4] +
         sum(test in 2..N_i+1) (Distance[Ville_precedente_Cal[test-1]][Ville_precedente_Cal[test]]) + Distance[Ville_precedente_Cal[N_i+1]][5] +
         sum(test in 2..N_i+1) (Distance[Ville_precedente_Edm[test-1]][Ville_precedente_Edm[test]]) + Distance[Ville_precedente_Edm[N_i+1]][6];
```

Contraintes du programme OPL

```
subject to {
 // MONTREAL
// Contrainte liante de ville précédente

    //Mtl commence part de montreal même si son premier match est ailleurs!
    Ville_precedente_Mtl[1] == 1;

    //Si une équipe à jouer son match précédent à l'extéreur alors elle arrive de labas
    forall(i in 2..N_i+1, j in Teams) (Mtl[i-1][j]["0"]==1) => (Ville_precedente_Mtl[i] == j);

    //Si Mtl joue son match précédent à la maison alors elle arrive de Mtl
    forall(i in 2..N_i+1, j in Teams) (Mtl[i-1][j]["1"]==1) => (Ville_precedente_Mtl[i] == 1);


// Mtl ne joue pas contre lui-même
forall(i in 1..N_i, k in Home) Mtl[i][1][k]==0 ;

// Mtl joue 1 fois a domicile contre chaque équipe et 1 fois a l'exterieur contre chaque équipe
forall(Teams in 2..6, k in Home) sum(i in I) Mtl[i][Teams][k]==1;

// Mtl joue un match à la fois
forall(i in I) sum(j in Teams, k in Home) Mtl[i][j][k]==1 ;

// Mtl joue maximum 3 matchs à l'exterieur de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Mtl[i+z][j]["0"]) <= 3 ;

// Mtl joue maximum 3 matchs à domicile de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Mtl[i+z][j]["1"]) <= 3 ;
```

```
//TORONTO
// Contrainte liante de ville précédente

    //Tor part de Tor même si son premier match est ailleurs!
    Ville_precedente_Tor[1] == 2;

    //Si une équipe à jouer son match précédent à l'extéreur alors elle arrive de labas
    forall(i in 2..N_i+1, j in Teams) (Tor[i-1][j]["0"]==1) => (Ville_precedente_Tor[i] == j);

    //Si Tor joue son match précédent à la maison alors elle arrive de Tor
    forall(i in 2..N_i+1, j in Teams) (Tor[i-1][j]["1"]==1) => (Ville_precedente_Tor[i] == 2);


// Tor ne joue pas contre lui-même
forall(i in 1..N_i, k in Home) Tor[i][2][k]==0 ;

// Tor joue 1 fois a domicile contre chaque équipe et 1 fois a l'exterieur contre chaque équipe
forall(Teams in {1,3,4,5,6}, k in Home) sum(i in I) Tor[i][Teams][k]==1;

// Tor joue un match à la fois
forall(i in I) sum(j in Teams, k in Home) Tor[i][j][k]==1 ;

// Tor joue maximum 3 matchs à l'exterieur de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Tor[i+z][j]["0"]) <= 3 ;

// Tor joue maximum 3 matchs à domicile de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Tor[i+z][j]["1"]) <= 3 ;
```

```
//Ottawa
// Contrainte liante de ville précédente

    //Ott  part de ott même si son premier match est ailleurs!
    Ville_precedente_Ott[1] == 3;

    //Si une équipe à jouer son match précédent à l'extéreur alors elle arrive de labas
    forall(i in 2..N_i+1, j in Teams) (Ott[i-1][j]["0"]==1) => (Ville_precedente_Ott[i] == j);

    //Si Ott joue son match précédent à la maison alors elle arrive de Ott
    forall(i in 2..N_i+1, j in Teams) (Ott[i-1][j]["1"]==1) => (Ville_precedente_Ott[i] == 3);

// Ott ne joue pas contre lui-même
forall(i in 1..N_i, k in Home) Ott[i][3][k]==0 ;

// Ott joue 1 fois a domicile contre chaque équipe et 1 fois a l'exterieur contre chaque équipe
forall(Teams in {1,2,4,5,6}, k in Home) sum(i in I) Ott[i][Teams][k]==1;

// Ott joue un match à la fois
forall(i in I) sum(j in Teams, k in Home) Ott[i][j][k]==1 ;

// Ott joue maximum 3 matchs à l'exterieur de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Ott[i+z][j]["0"]) <= 3 ;

// Ott joue maximum 3 matchs à domicile de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Ott[i+z][j]["1"]) <= 3 ;
```

```
//Winnipeg
// Contrainte liante de ville précédente

    //Win commence part de Winnipeg même si son premier match est ailleurs!
    Ville_precedente_Win[1] == 4;

    //Si une équipe à jouer son match précédent à l'extéreur alors elle arrive de labas
    forall(i in 2..N_i+1, j in Teams) (Win[i-1][j]["0"]==1) => (Ville_precedente_Win[i] == j);

    //Si Win joue son match précédent à la maison alors elle arrive de Ott
    forall(i in 2..N_i+1, j in Teams) (Win[i-1][j]["1"]==1) => (Ville_precedente_Win[i] == 4);


// Win ne joue pas contre lui-même
forall(i in 1..N_i, k in Home) Win[i][4][k]==0 ;

// Win joue 1 fois a domicile contre chaque équipe et 1 fois a l'exterieur contre chaque équipe
forall(Teams in {1,2,3,5,6}, k in Home) sum(i in I) Win[i][Teams][k]==1;

// Win joue un match à la fois
forall(i in I) sum(j in Teams, k in Home) Win[i][j][k]==1 ;

// Win joue maximum 3 matchs à l'exterieur de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Win[i+z][j]["0"]) <= 3 ;

// Win joue maximum 3 matchs à domicile de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Win[i+z][j]["1"]) <= 3 ;
```

```
//Calgary
// Contrainte liante de ville précédente

    //Cal commence part de Calgary même son premier match est ailleurs!
    Ville_precedente_Cal[1] == 5;

    //Si une équipe à jouer son match précédent à l'extéreur alors elle arrive de labas
    forall(i in 2..N_i+1, j in Teams) (Cal[i-1][j]["0"]==1) => (Ville_precedente_Cal[i] == j);

    //Si un équipe joue son match précédent à la maison alors elle arrive de la maison
    forall(i in 2..N_i+1, j in Teams) (Cal[i-1][j]["1"]==1) => (Ville_precedente_Cal[i] == 5);


// Cal ne joue pas contre lui-même
forall(i in 1..N_i, k in Home) Cal[i][5][k]==0 ;

// Cal joue 1 fois a domicile contre chaque équipe et 1 fois a l'exterieur contre chaque équipe
forall(Teams in {1,2,3,4,6}, k in Home) sum(i in I) Cal[i][Teams][k]==1;

// Cal joue un match à la fois
forall(i in I) sum(j in Teams, k in Home) Cal[i][j][k]==1 ;

// Cal joue maximum 3 matchs à l'exterieur de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Cal[i+z][j]["0"]) <= 3 ;

// Cal joue maximum 3 matchs à domicile de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Cal[i+z][j]["1"]) <= 3 ;
```

```
//Edmonton
// Contrainte liante de ville précédente

    //Edm commence part de Edmonton même son premier match est ailleurs!
    Ville_precedente_Edm[1] == 6;

    //Si une équipe à jouer son match précédent à l'extéreur alors elle arrive de labas
    forall(i in 2..N_i+1, j in Teams) (Edm[i-1][j]["0"]==1) => (Ville_precedente_Edm[i] == j);

    //Si un équipe joue son match précédent à la maison alors elle arrive de la maison
    forall(i in 2..N_i+1, j in Teams) (Edm[i-1][j]["1"]==1) => (Ville_precedente_Edm[i] == 6);

// Edm ne joue pas contre lui-même
forall(i in 1..N_i, k in Home) Edm[i][6][k]==0 ;

// Edm joue 1 fois a domicile contre chaque équipe et 1 fois a l'exterieur contre chaque équipe
forall(Teams in {1,2,3,4,5}, k in Home) sum(i in I) Edm[i][Teams][k]==1;


// Edm joue un match à la fois
forall(i in I) sum(j in Teams, k in Home) Edm[i][j][k]==1 ;

// Edm joue maximum 3 matchs à l'exterieur de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Edm[i+z][j]["0"]) <= 3 ;

// Edm joue maximum 3 matchs à domicile de suite
forall (i in 1..(N_i-3)) sum (z in 0..3, j in Teams) (Edm[i+z][j]["1"]) <= 3 ;
```

```
// Contraintes qui lient les équipes

forall(i in I) Mtl[i][2]["0"] == Tor[i][1]["1"];
forall(i in I) Mtl[i][2]["1"] == Tor[i][1]["0"];
forall(i in I) Mtl[i][3]["0"] == Ott[i][1]["1"];
forall(i in I) Mtl[i][3]["1"] == Ott[i][1]["0"];
forall(i in I) Mtl[i][4]["0"] == Win[i][1]["1"];
forall(i in I) Mtl[i][4]["1"] == Win[i][1]["0"];
forall(i in I) Mtl[i][5]["0"] == Cal[i][1]["1"];
forall(i in I) Mtl[i][5]["1"] == Cal[i][1]["0"];
forall(i in I) Mtl[i][6]["0"] == Edm[i][1]["1"];
forall(i in I) Mtl[i][6]["1"] == Edm[i][1]["0"];

forall(i in I) Tor[i][3]["0"] == Ott[i][2]["1"];
forall(i in I) Tor[i][3]["1"] == Ott[i][2]["0"];
forall(i in I) Tor[i][4]["0"] == Win[i][2]["1"];
forall(i in I) Tor[i][4]["1"] == Win[i][2]["0"];
forall(i in I) Tor[i][5]["0"] == Cal[i][2]["1"];
forall(i in I) Tor[i][5]["1"] == Cal[i][2]["0"];
forall(i in I) Tor[i][6]["0"] == Edm[i][2]["1"];
forall(i in I) Tor[i][6]["1"] == Edm[i][2]["0"];

forall(i in I) Ott[i][4]["0"] == Win[i][3]["1"];
forall(i in I) Ott[i][4]["1"] == Win[i][3]["0"];
forall(i in I) Ott[i][5]["0"] == Cal[i][3]["1"];
forall(i in I) Ott[i][5]["1"] == Cal[i][3]["0"];
forall(i in I) Ott[i][6]["0"] == Edm[i][3]["1"];
forall(i in I) Ott[i][6]["1"] == Edm[i][3]["0"];

forall(i in I) Win[i][5]["0"] == Cal[i][4]["1"];
forall(i in I) Win[i][5]["1"] == Cal[i][4]["0"];
forall(i in I) Win[i][6]["0"] == Edm[i][4]["1"];
forall(i in I) Win[i][6]["1"] == Edm[i][4]["0"];

forall(i in I) Cal[i][6]["0"] == Edm[i][5]["1"];
forall(i in I) Cal[i][6]["1"] == Edm[i][5]["0"];
```