

NEW MODELIZATION OF THE UNIVERSAL FARING ENGINE WITH CONSTRAINT PROGRAMMING

Introduction

Un solveur de programmation par contraintes prend comme entrée un ensemble de variables auxquelles on associe un domaine de valeurs, ainsi que des contraintes entre ces variables. L'algorithme va créer un arbre de recherche qu'il va élaguer en faisant des invalidations qui vont être propagées dans la recherche.

5	3		7					
6		1	9	5				
9	8				6			
8			6				3	
4		8	3				1	
7			2					6
	6			2	8			
		4	1	9			5	
			8			7	9	

Par exemple, on fait implicitement de la programmation par contraintes en résolvant un sudoku. Une variable est une case du sudoku et chaque variable a un domaine de valeurs, les entiers de 1 à 9. On ajoute une contrainte AllDifferent entre les variables d'une même ligne, d'une même colonne et appartenant au même carré de 9 cases, ce qui signifie qu'aucune variable partageant cette contrainte ne peut avoir la même valeur.

Le moteur de recherche Amadeus est assez statique : une boucle qui va créer des millions de combinaisons et vérifier leur validité. Lorsque nous faisons une invalidation, nous pouvons rétro-propager cette invalidation afin de couper des branches de l'arbre d'exploration. Et si on y ajoutait de la programmation par contraintes ?

Complexité du moteur de recherche Amadeus

Complexity requires massive computing power...

San Francisco to Boston: 10,000 paths



3,604,439,023 possible solutions with 2,389,402,117 that are available for purchase

- Number of outbound flights considered (BOS-SFO): 403
- Number of return flights considered (BOS-SFO): 402
- Total number of journey itineraries: 403 x 402 = 162,006
- Average number of ways to overlay fares onto each journey itinerary: 14,749
- 10,000 paths are normalized by removing unworkable itineraries, e.g., LAX-LHR-BOS, or too short connections

Cette complexité dépend de la granularité.

La plus « simple » des combinatoires étant le Pricing, qui consiste globalement à associer un prix à un ensemble de vols souhaité qui couvre votre itinéraire.

Si vous ne connaissez pas cet ensemble de vols, mais que vous souhaitez voyager avec Air France, c'est un produit Flex Pricer un peu plus complexe : on multiplie la complexité du Pricing par le nombre de combinaisons de vols Air France.

Maintenant, si vous souhaitez trouver le moins cher peu importe les compagnies, c'est le produit Master Pricer (MP) que vous retrouvez par exemple sur Expedia, et d'une complexité encore plus grande, puisqu'on doit désormais comparer tous les ensembles de vols possibles permettant de couvrir l'itinéraire demandé.

Enfin, si vous ne savez pas quand partir, vous pouvez faire un Calendar, qui va multiplier la complexité par le nombre de combinaisons de dates possibles.

Figure courtesy of ITA software (first half of 2000s)

Pour avoir une idée de la combinatoire pour un PAR – NYC round trip sur un Calendar +-3jours :

O(Calendar)	<ul style="list-style-type: none"> • Nombre de combinaisons de dates : 7 (outbound = aller) * 7 (inbound = retour) = 49 MP • Pour chaque combinaison de dates, si on ne considère par exemple que 1000 possibilités pour aller de l'origine du voyage à la destination (on pourrait avoir des vols directs ou des vols avec 1 ou 2 escales), alors on a 1000*1000=10^6 itinéraires possibles pour notre voyage aller-retour.
O(MP)	

amadeus

O(Pricing)

- Pour chaque leg/ensemble de legs, on y associe un prix ; il y en a plusieurs dizaines par segment de vols, avec pour chacun des contraintes à respecter (jour de la semaine, point de vente, combinaison avec d'autres prix, etc.). Si on prend un itinéraire de 6 segments de vol, on a au moins 10^6 prix à évaluer.
- Ces combinaisons de prix doivent suivre des règles dictées par IATA et ATPCO. IATA définit par exemple des contextes de tarification dans lesquels les combinaisons de prix devront être valides. Il y a en moyenne 5 contextes de tarification à valider pour un itinéraire donné.

Donc au moins $49 * 10^6 * 10^6 * 5 = 250$ milliards de combinaisons à tester pour garantir de trouver la recommandation la moins chère sur un Paris - New York aller-retour en Calendar +3 jours. Au fait, PAR représente l'ensemble des aéroports de Paris (CDG + ORY), tout comme NYC... On peut donc encore augmenter la combinatoire ! Et il faut que le moteur de recommandations réponde en moins de 15 secondes.

Contraintes dans le moteur de recherche Amadeus

Les fares contiennent des règles de combinabilité, ATPCO Fare Category ; et il y en a une trentaine !

La catégorie 4, par exemple, indiquera la validité d'un fare sur un ensemble de numéros de vols passant par certains points. Les catégories 6 et 7 spécifient le temps minimum/maximum qu'il est possible de rester à un certain point. La catégorie 10 définit les règles de combinabilité autorisées entre plusieurs fares...

Parmi les millions de combinaisons de fares et les milliards de combinaisons globales à considérer pour trouver la recommandation la moins chère sur notre itinéraire PAR – NYC, seul un faible pourcentage sera valide. Toutes les autres se verront invalidées par des checks de catégorie, qui finalement peuvent être représentés comme des contraintes en programmation par contraintes !

Ensuite, sur l'ensemble de combinaisons valides, il reste globalement à sélectionner la recommandation la moins chère par ensemble de vols et à renvoyer au client N recommandations diversifiées selon des critères propres aux différents clusters : ceux qui veulent voyager au meilleur prix, ceux qui veulent voyager vite, en classe affaires, etc.

Objectifs du TER

Le but de ce travail d'étude et de recherche est de construire un modèle simple de programmation par contraintes du moteur de recommandations d'Amadeus, en ayant éventuellement recours à la programmation dynamique. Ce modèle utilisera les informations sur les vols, les tarifs (fares) et quelques contraintes simples de combinabilité entre eux.

Une phase d'optimisation du modèle de programmation par contraintes visera à essayer de trouver une recommandation valide, voire trouver la moins chère sur un itinéraire donné en quelques secondes voire quelques centaines de millisecondes. Les Etudiants apporterons une nouvelle façon dynamique de modéliser les objets et les relations entre les objets afin de voir si un tel modèle peut être intégré dans le moteur de recommandations actuel.

Livrables du TER

Il est attendu des Etudiants un rapport écrit et un logiciel livrable.

Le rapport détaillé devra à minima décrire l'approche utilisée, en faisant mention des points bloquants et des solutions de contournement, une description technique de la solution logicielle mise en place avec des exemples, ainsi que des tests de performance prouvant l'efficacité, ou pas, de la solution.

Le logiciel livrable pourra être réalisé dans un ensemble de langages différents selon les besoins des Etudiants. Il devra accepter en entrée un ensemble de données prétraitées voire au format brut et fournir en sortie des données pouvant être utilisées par le moteur de recommandations actuel. Si nécessaire à la mesure de la performance, l'intégration par les Etudiants de la solution logicielle dans le moteur de recommandations est attendue, en C++ (simplement via une communication par fichiers textes ou de manière native).