

Thomas **CORNIER**
Vincent **COURSAC**
Arthur **HEUILLARD**
Matthieu **PILAUDEAU**
Paul **TORTEL**

Groupe : **F3**

Projet d'Aide à la décision

Moteur PSC

Dans le cadre de la troisième année d'études à l'EFREI, nous avons dû concevoir un programme de résolution de problèmes par les techniques de Propagation et Satisfaction de Contraintes.

Table des matières

I.	L'architecture du fichier d'entrée	2
A.	Variables	2
B.	Contraintes	2
C.	Affichage.....	3
II.	Les structures de données.....	4
A.	La classe Engine	4
B.	La classe Variable.....	4
C.	La classe Constraints	5
D.	La classe Node	6
III.	La recherche d'une solution au problème	6
A.	createNode(), la fonction qui crée associe une variable à une valeur	7
B.	La réduction de domaines	7
C.	La cohérence d'arête	9
IV.	Traces d'exécution.....	9
A.	Problème 1	10
B.	Problème 2	12
C.	Problème 3	14
D.	Problème 4	16
E.	Problème 5	17
F.	Problème 6	18
G.	Problème 7	20
H.	Problème customisé	20
I.	SEND + MORE = MONEY	23

I. L'architecture du fichier d'entrée

Il est conseillé pour bien suivre cette partie d'avoir à côté un de nos fichiers XML ouvert.

Format du fichier

Notre fichier est au **format XML**, permettant une meilleure lisibilité du point de vue de l'utilisateur tout en restant compréhensible pour le programme.

Le fichier se découpe en deux grandes parties principales, et une partie optionnelle gérant l'affichage de la solution.

A. Variables

Les variables se glissent dans le nœud **Vars** grâce aux sous nœuds **Var**

```
<PSC>
  <Vars>
  </Vars>
  <Constraints>
  </Constraints>
  <Display names = "false">
  </Display>
</PSC>
```

		Attribut	Valeurs possibles	Description
Nœud Principal	Var	name	Toute chaîne de caractères	Nom de la variable
Sous-nœuds	Index	Crée des sous variables à partir du nom de base. Par exemple : X[1], X[2], ... , X[n]		
		min	Entiers	Index minimal de la variable
		max	Entiers	Index maximal de la variable
	Domain	value	Intervalles sous la forme : n;p ([n;p]) ou n ({n}) Séparés par des espaces	Domaine de définition de la variable

B. Contraintes

Les contraintes se glissent dans le nœud **Constraints** grâce aux sous nœuds **Constraint**.

		Attribut	Valeurs possibles	Description	
Nœud Principal	Var	type	"bin" ou "sum"	Décrit le type de contrainte : binaire ou somme	
Contraintes binaires		var1	Chaîne de caractères	Nom de la première variable de la contrainte	
		var2	Chaîne de caractères	Nom de la deuxième variable de la contrainte	
		value	Entier	Valeur constante de référence pour la contrainte	
		ref	Chaîne du type X[i - 1] NomVar[i (+ - * / %) n]	Permet de référencer une variable en fonction de l'index de la variable 1.*	
					Uniquement un des trois!

Contraintes de somme		value	Chaine de caractères	Valeur constante de reference pour la contrainte	Uniquement un des deux!
		var	Chaine de caractères	Nom de la variable de reference pour la contrainte	
Sous-nœuds	Var	Permet d'ajouter une variable à la somme			
		name	Chaine de caractères	Nom de la variable ajoutée à la contrainte de somme	
		op	>= ou > ou <= ou < ou != ou =	Operateur binaire utilisé pour la contrainte (binaire ou somme)	

* Exemple d'utilisation de **ref** :

var1 = "X[5][6]" et ref = "[i - 1][i + 2]"

Alors la deuxième variable sera X[4][8]

D'autre part, dans les attributs var1, var2 et name (dans une contrainte de somme), il est possible d'utiliser la notation [n,p] pour ajouter plusieurs variables rapidement.

Par exemple : X[1 ;5] ajoutera une contrainte sur X[1], X[2], ..., X[5].

C. Affichage

Pour l'affichage, il n'y a qu'un attribut pour le nœud principal, et qu'un sous nœud possible.

L'attribut **names** supporte les valeurs **true** et **false**, et permet d'indiquer si l'on doit afficher ou non le nom des variables dans la solution.

Viennent ensuite les sous-nœuds **Line**, qui possèdent les attributs **blank** et **filled**, qui acceptent tous deux des entiers et correspondent respectivement au nombre d'espace en début de ligne à afficher, et au nombre de variables à afficher sur la fin de la ligne.

Exemple simple : **le sudoku**

```
<Display names = "false">
  <Line blank = "0" filled = "9" />
  <Line blank = "0" filled = "9" />
  <Line blank = "0" filled = "9" />
  <Line blank = "0" filled = "9" />
  <Line blank = "0" filled = "9" />
  <Line blank = "0" filled = "9" />
  <Line blank = "0" filled = "9" />
  <Line blank = "0" filled = "9" />
  <Line blank = "0" filled = "9" />
</Display>
```

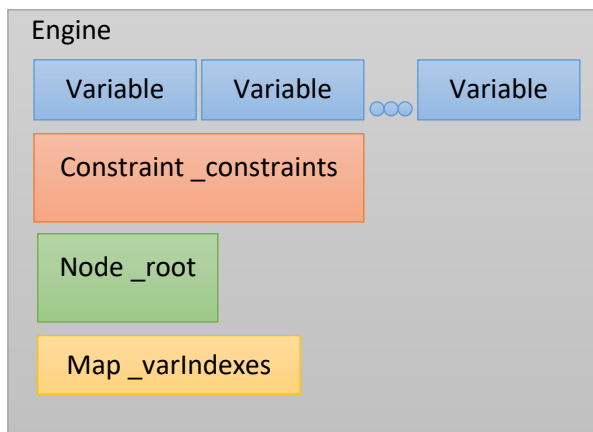
II. Les structures de données

Le fichier étant lu en intégralité, la grande majorité de nos structures de données est remplie. Commençons par voir la classe *Engine*.

La classe *Engine* sert à regrouper les différentes parties du moteur PSC : elle stocke les variables, les contraintes qui s'y appliquent, la racine de l'arbre de recherche de solution. C'est à partir de cette classe que l'on va lancer les différentes techniques de résolution de problème, à savoir la méthode :

- La variable la plus contrainte d'abord
- La variable la moins contrainte d'abord
- Les variables dans leur ordre de déclaration
- La variable ayant le plus petit domaine d'abord

A. La classe *Engine*



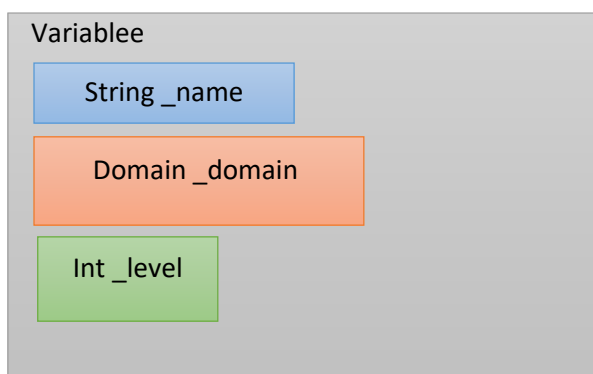
Vector d'instance de la classe *Variable*. Stocke toutes les variables d'un problème.

Instance de la classe *Constraint*. Regroupe toutes les contraintes existantes sur les variables du problème.

Instance de la classe *Node*, est la racine de notre arbre de recherche.

Map stockant pour chaque variable un index (par exemple :

B. La classe *Variable*



_name : le nom de la variable, qui se compose d'un nom et d'un index

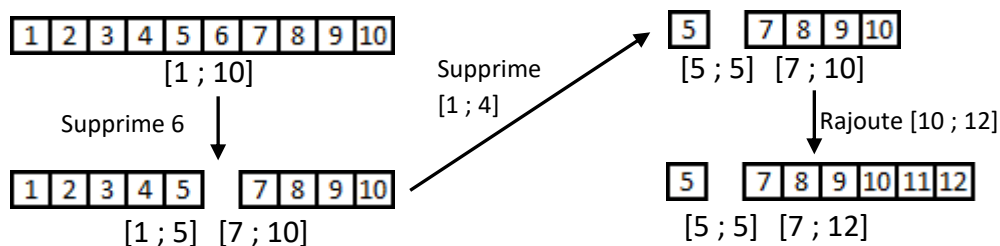
_domain : les variables ont différents domaines au début du problème et il est important de pouvoir conserver en mémoire les caractéristiques de chaque variable.

_level : permet de connaître le niveau de la variable dans l'arbre.

Le moteur de résolution des contraintes ne gérant que les variables simples (non-indexées), les variables indexées (par exemple, $X[9][9]$ pour représenter une grille de sudoku) sont décomposées en plusieurs variables simples ($X[1][1]$, $X[2][1]$, ..., $X[9][9]$).

L'ensemble des variables obtenues à partir du fichier sont stockées dans un tableau, et chaque variable se voit ainsi attribuer un numéro (l'index dans le tableau).

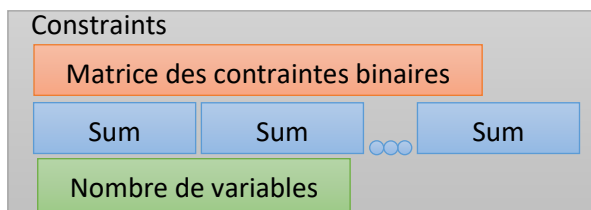
Chaque variable se voit également affecté son domaine de définition, représenté par la classe *Domain*. Ces domaines sont stockés sous forme d'une liste d'intervalles disjoints et croissants et supportent l'ajout et la suppression de valeurs.



Manipulation de domaines

C. La classe *Constraints*

La classe *Constraints* sert à rassembler l'ensemble des contraintes que notre moteur PSC peut traiter. On se sert de cette classe pour vérifier si les valeurs associées aux variables respectent bien toutes les contraintes.



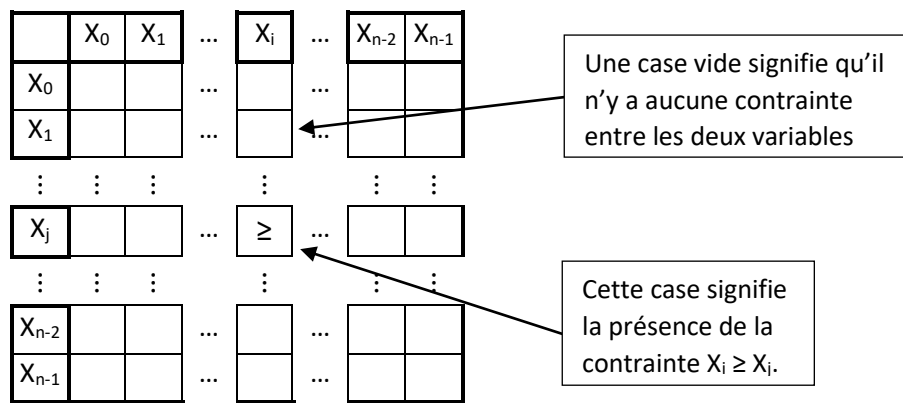
Vector de *Sum* répertoriant toutes les sommes existantes dans le problème.

Le programme supporte deux types de contraintes :

- Les contraintes binaires (par exemple, $A < B$) ;
- Les contraintes sur des sommes pondérées (par exemple, $2A + B - C = 10$ ou encore $2A + B - C = D$)

Les contraintes binaires sont représentées à l'aide d'une matrice $N \times N$, où N est le nombre de variables du problème à résoudre. Les types de contraintes possibles sont :

- $A = B$
- $A \neq B$
- $A \geq B$
- $A \leq B$
- $A > B$
- $A < B$

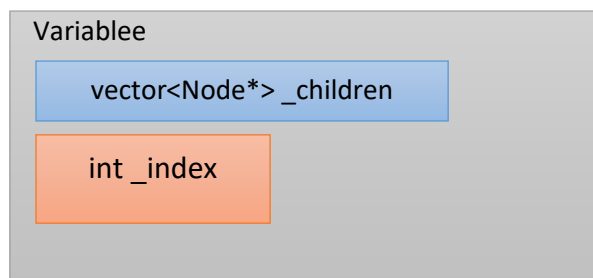


Matrice des contraintes binaires

Chaque somme est représentée par une instance de la classe *Sum*, qui contient :

- La liste des variables présentes dans la sommes, ainsi que leur coefficient ;
- La valeur ou la variable à laquelle comparer la somme ;
- Le type de comparaison à effectuer (<, ≤, =, ≠, ≥ ou >)

D. La classe *Node*



_children: permet d'accéder aux enfants de chaque nœud (et donc à l'étage suivant de l'arbre).

_index: permet de savoir à quel étage de l'arbre nous nous trouvons, et peut être associé au tableau *chosenValues* (qui stocke les valeurs actuelles de toutes les variables).

Variables *static* : utilisés pour calculer les statistiques du problème actuel, et ainsi comparer différentes méthodes de résolution.

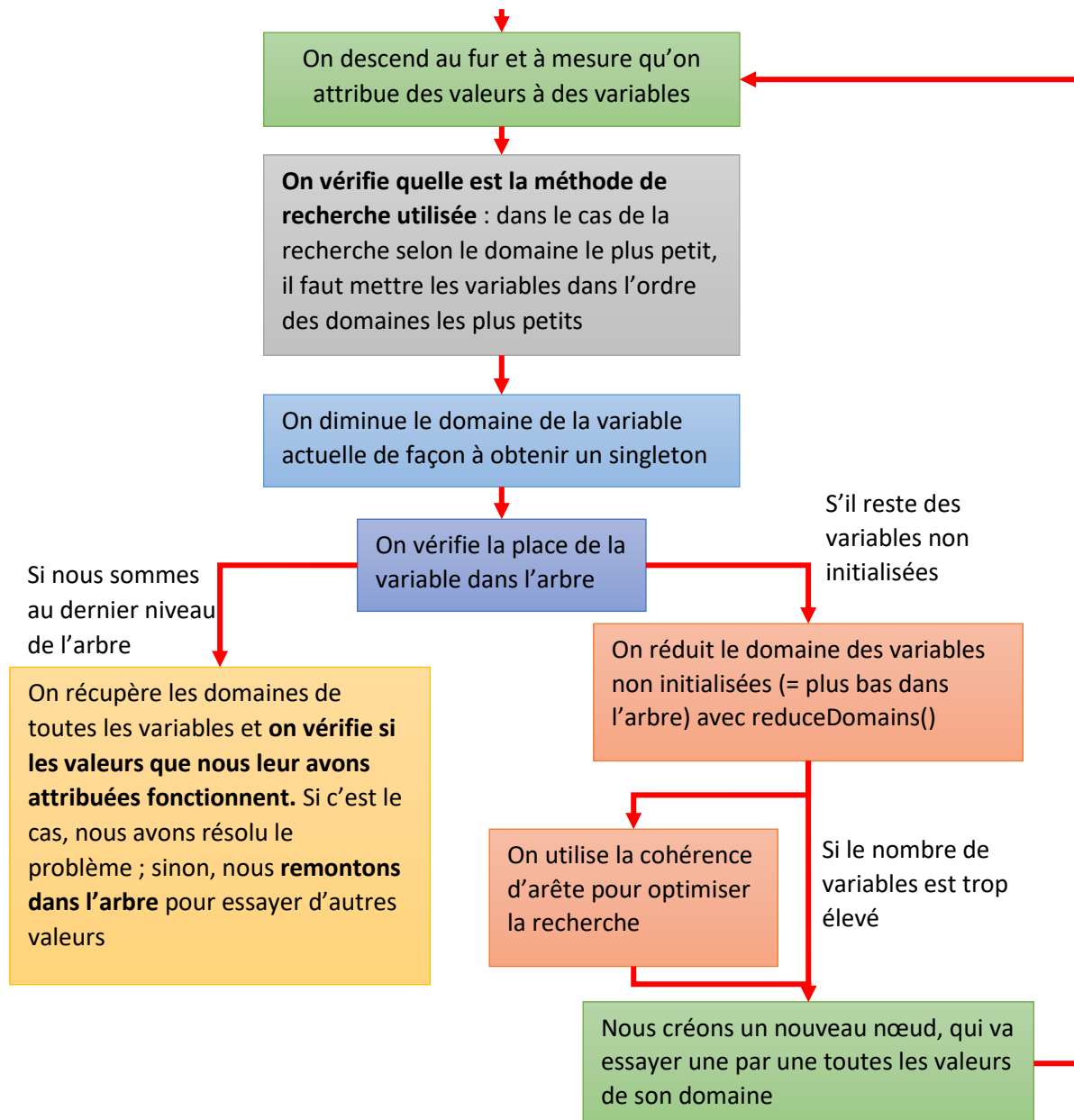
III. La recherche d'une solution au problème

Pour résoudre le problème, nous créons un arbre. Cet arbre va nous permettre de tester des combinaisons de valeurs pour les variables. Les nœuds de cet arbre représentent les association variable-valeur.

La création de nœuds intervient lors de la résolution du problème. C'est en effet en créant les nœuds un par un que nous nous allons obtenir un arbre qui contiendra une solution à notre problème et c'est grâce à cela que nous trouverons la solution au problème.

Lors de la formation de l'arbre et de la création des nœuds, des techniques plus avancées sont utilisées afin de simplifier le travail de l'ordinateur : en effet, si trop de variables sont présentes, le nombre de nœuds que le programme devra créer sera énorme ; c'est pourquoi nous avons deux méthodes pour limiter les domaines des variables et le nombre de nœuds créés : la réduction de domaine et la cohérence d'arête.

A. `createNode()`, la fonction qui crée associe une variable à une valeur

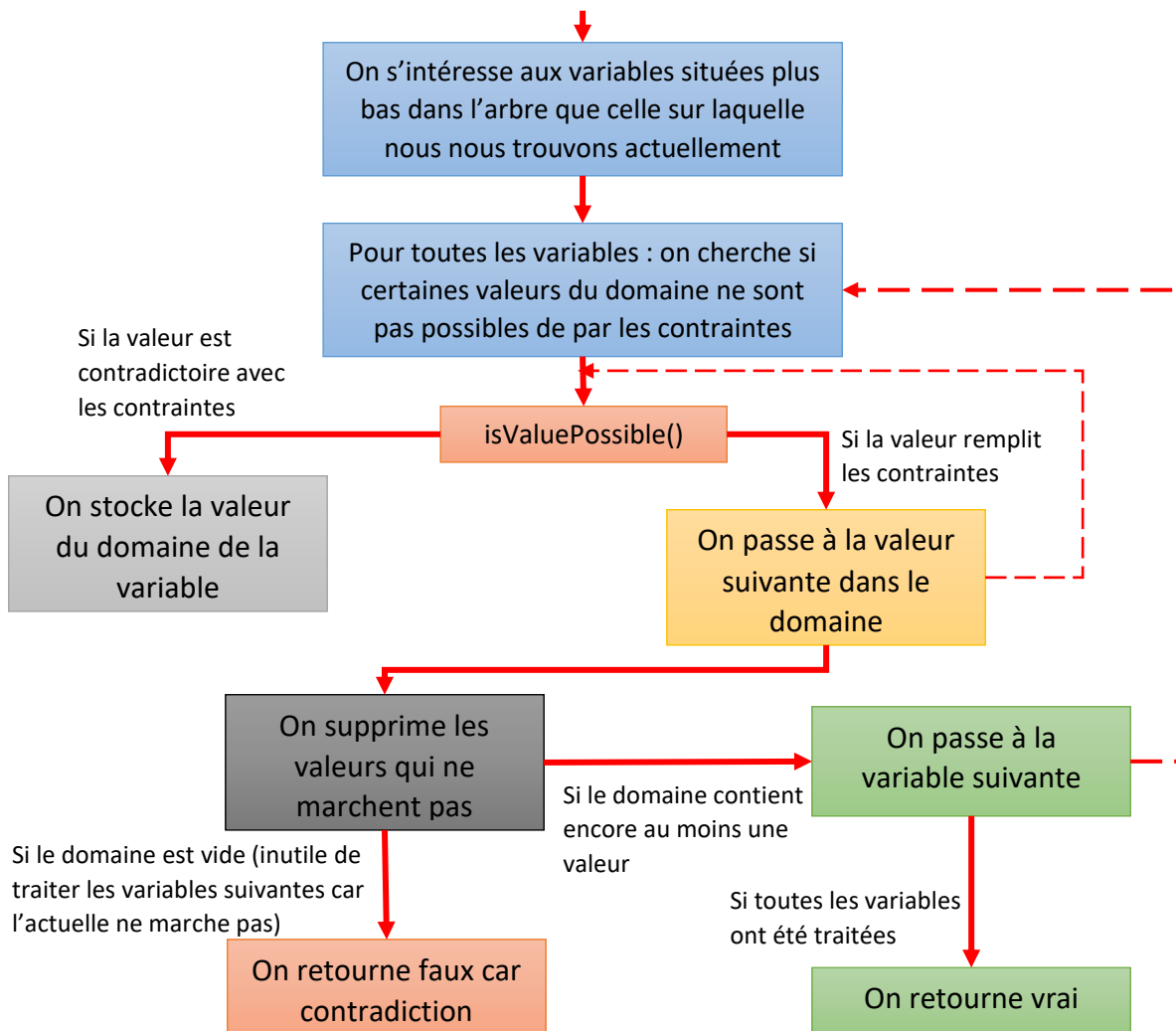


B. La réduction de domaines

La réduction de domaines nous permet d'éviter de prendre trop de temps à trouver une solution pour le problème ; en effet, pour chaque variable que l'on initialise, nous vérifions s'il est possible de

réduire les domaines des autres variables en prenant en compte cette initialisation (par exemple, si une variable X1 doit être différente d'une variable X2 et que X1 a reçu la valeur 3, alors nous pouvons directement enlever la valeur 3 du domaine de X2).

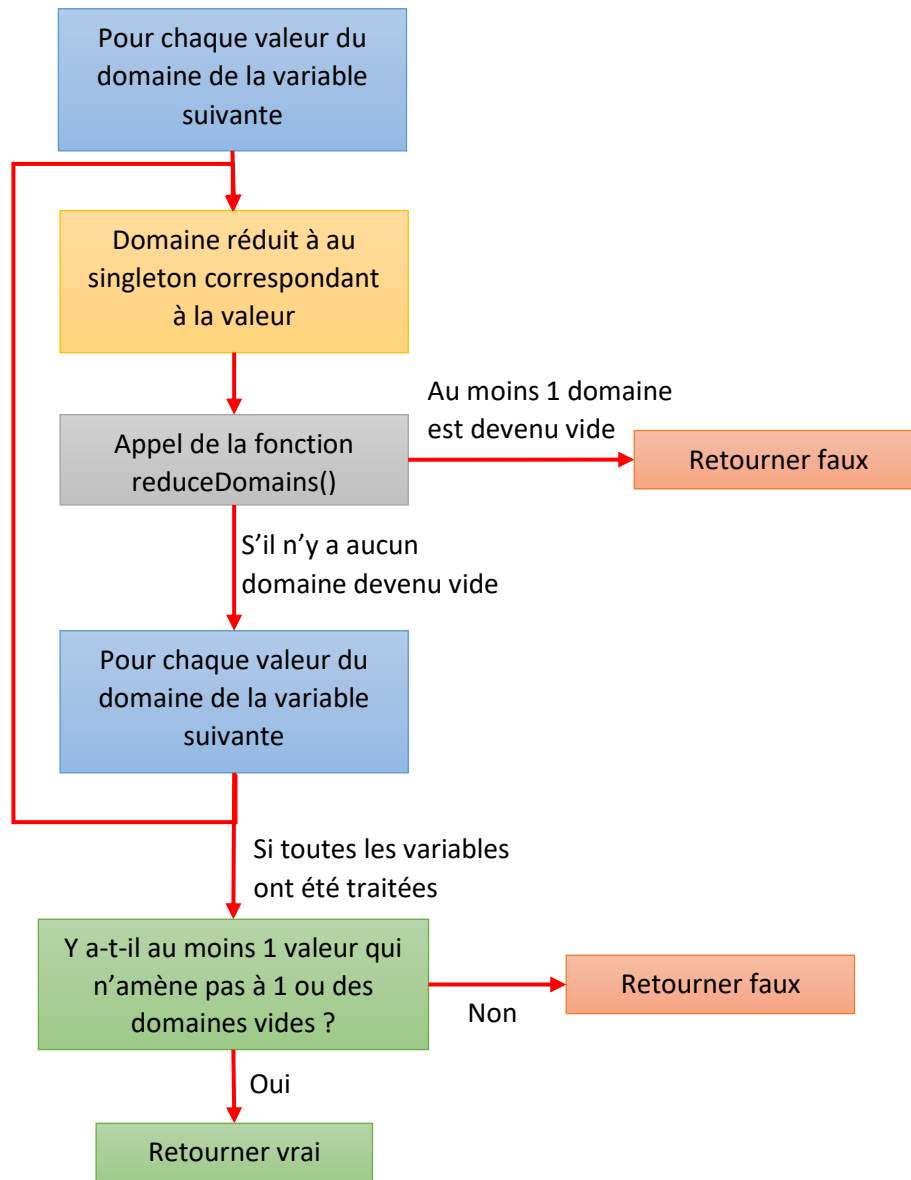
De plus, la réduction de domaine nous permet d'arrêter le traitement actuel dès que nous trouvons une contradiction : si une variable voit son domaine réduire au point d'être vide, cela signifie qu'il n'existe pas de solution avec les valeurs des variables déjà initialisées. Il n'est donc pas nécessaire de continuer l'analyse de cette branche de l'arbre, et nous pouvons remonter pour essayer d'autres combinaisons, nous faisant gagner un temps précieux.



C. La cohérence d'arête

La cohérence d'arête intervient après la réduction de domaines. Elle n'est appelée que si l'utilisateur l'a désiré et si le nombre de variables est inférieur à 20. En effet, si le nombre de variables est très grand, la cohérence d'arête sera inefficace car trop longue à effectuer de par le fait de sa récursivité.

Voici un schéma résumant les principales étapes de notre algorithme sur la cohérence d'arête.



IV. Traces d'exécution

Toutes les captures d'écran ont été effectuées **sans la cohérence d'arête**, même si elle est **tout à fait fonctionnelle**.

A. Problème 1

Methode "la variable la + contrainte d'abord" :

Statistiques :

Nombre de noeuds cree : 449

Nombre de noeuds elagues : 10786266670913138944

Profondeur moyenne des elagages : 11.2273

Profondeur maximum d'elagage : 22

Les solutions sont :

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

Methode "la variable la - contrainte d'abord" :

Statistiques :

Nombre de noeuds cree : 449

Nombre de noeuds elagues : 10786266670913138944

Profondeur moyenne des elagages : 11.2273

Profondeur maximum d'elagage : 22

Les solutions sont :

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

Methode "les variables dans leur ordre de declaration" :

Statistiques :

Nombre de noeuds cree : 3622

Nombre de noeuds elagues : 10786266670913135771

Profondeur moyenne des elagages : 17.4984

Profondeur maximum d'elagage : 32

Les solutions sont :

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

Methode "la variable ayant domaine le + petit d'abord" :

Statistiques :

Nombre de noeuds cree : 383

Nombre de noeuds elagues : 10786266670913139010

Profondeur moyenne des elagages : 45.9054

Profondeur maximum d'elagage : 60

Les solutions sont :

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

Appuyez sur une touche pour continuer...

B. Problème 2

Ici la somme des blocs/des lignes/des colonnes donne bien 45, mais nous n'avons pas trouvé de contraintes faisant appel uniquement à des sommes pour empêcher qu'il y ait plusieurs fois un même nombre sur une ligne/colonne/un bloc.

```
Methode "la variable la + contrainte d'abord" :
```

```
Statistiques :
```

```
Nombre de noeuds cree : 108
```

```
Nombre de noeuds elagues : 10786266670913139285
```

```
Profondeur moyenne des elagages : 36.9474
```

```
Profondeur maximum d'elagage : 41
```

```
Les solutions sont :
```

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

```
Methode "la variable la - contrainte d'abord" :
```

```
Statistiques :
```

```
Nombre de noeuds cree : 108
```

```
Nombre de noeuds elagues : 10786266670913139285
```

```
Profondeur moyenne des elagages : 36.9474
```

```
Profondeur maximum d'elagage : 41
```

```
Les solutions sont :
```

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

Methode "les variables dans leur ordre de declaration" :

Statistiques :

Nombre de noeuds cree : 83

Nombre de noeuds elagues : 10786266670913139310

Profondeur moyenne des elagages : 29

Profondeur maximum d'elagage : 29

Les solutions sont :

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

Methode "la variable ayant domaine le + petit d'abord" :

Statistiques :

Nombre de noeuds cree : 81

Nombre de noeuds elagues : 10786266670913139312

Profondeur moyenne des elagages : aucun elagage

Profondeur maximum d'elagage : 0

Les solutions sont :

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

Appuyez sur une touche pour continuer...

C. Problème 3

Le fichier XML du problème 3 comporte une erreur mais nous ne l'avons pas trouvée. Pourtant nous sommes certains que notre moteur est capable de trouver une solution puisque nous avons testé le même type de problème (voir trace de SEND + MORE = MONEY).

```
      Methode "la variable la + contrainte d'abord" :  
  
      Statistiques :  
Nombre de noeuds cree : 302  
Nombre de noeuds elagues : 22781249999698  
Profondeur moyenne des elagages : 3.64035  
Profondeur maximum d'elagage : 5  
  
Les solutions sont :  
  
Ret[1] => 1  
Ret[2] => 2  
Ret[3] => 1  
Ret[4] => 3  
Ret[5] => 1  
Z => 2  
E => 5  
R => 8  
O => 3  
T => 1  
I => 9  
S => 6  
D => 7  
U => 0  
N => 4
```

```
      Methode "la variable la - contrainte d'abord" :  
  
      Statistiques :  
Nombre de noeuds cree : 41466  
Nombre de noeuds elagues : 22781249958534  
Profondeur moyenne des elagages : 7.39467  
Profondeur maximum d'elagage : 11  
  
Les solutions sont :  
  
Ret[1] => 2  
Ret[2] => 3  
Ret[3] => 3  
Ret[4] => 2  
Ret[5] => 1  
Z => 7  
E => 6  
R => 5  
O => 8  
T => 2  
I => 1  
S => 4  
D => 9  
U => 3  
N => 0
```

Methode "les variables dans leur ordre de declaration" :

Statistiques :

Nombre de noeuds cree : 1993
Nombre de noeuds elagues : 22781249998007
Profondeur moyenne des elagages : 5.62194
Profondeur maximum d'elagage : 8

Les solutions sont :

Ret[1] => 1
Ret[2] => 2
Ret[3] => 1
Ret[4] => 3
Ret[5] => 1
Z => 2
E => 5
R => 8
O => 3
T => 1
I => 9
S => 6
D => 7
U => 0
N => 4

Methode "la variable ayant domaine le + petit d'abord" :

Statistiques :

Nombre de noeuds cree : 1101
Nombre de noeuds elagues : 22781249998899
Profondeur moyenne des elagages : 7.11298
Profondeur maximum d'elagage : 11

Les solutions sont :

Ret[1] => 1
Ret[2] => 2
Ret[3] => 1
Ret[4] => 3
Ret[5] => 1
Z => 2
E => 5
R => 8
O => 3
T => 1
I => 9
S => 6
D => 7
U => 0
N => 4

Appuyez sur une touche pour continuer...

D. Problème 4

```
Methode "la variable la + contrainte d'abord" :

Statistiques :
Nombre de noeuds cree : 12
Nombre de noeuds elagues : 129140151
Profondeur moyenne des elagages : 2.33333
Profondeur maximum d'elagage : 3

Les solutions sont :

Aucune solution trouvee !

Methode "la variable la - contrainte d'abord" :

Statistiques :
Nombre de noeuds cree : 31
Nombre de noeuds elagues : 129140132
Profondeur moyenne des elagages : 4.8
Profondeur maximum d'elagage : 5

Les solutions sont :

Aucune solution trouvee !

Methode "les variables dans leur ordre de declaration" :

Statistiques :
Nombre de noeuds cree : 7
Nombre de noeuds elagues : 129140156
Profondeur moyenne des elagages : 3
Profondeur maximum d'elagage : 3

Les solutions sont :

Aucune solution trouvee !
```

```
Methode "la variable ayant domaine le + petit d'abord" :

Statistiques :
Nombre de noeuds cree : 9
Nombre de noeuds elagues : 129140154
Profondeur moyenne des elagages : 5
Profondeur maximum d'elagage : 5

Les solutions sont :

Aucune solution trouvee !

Appuyez sur une touche pour continuer...
```


E. Problème 5

```
Methode "la variable la + contrainte d'abord" :

    Statistiques :
Nombre de noeuds cree : 33
Nombre de noeuds elagues : 3486784368
Profondeur moyenne des elagages : 3
Profondeur maximum d'elagage : 4

Les solutions sont :

Aucune solution trouvee !

Methode "la variable la - contrainte d'abord" :

    Statistiques :
Nombre de noeuds cree : 144
Nombre de noeuds elagues : 3486784257
Profondeur moyenne des elagages : 4.63636
Profondeur maximum d'elagage : 6

Les solutions sont :

Aucune solution trouvee !

Methode "les variables dans leur ordre de declaration" :

    Statistiques :
Nombre de noeuds cree : 21
Nombre de noeuds elagues : 3486784380
Profondeur moyenne des elagages : 3
Profondeur maximum d'elagage : 3

Les solutions sont :

Aucune solution trouvee !

Methode "la variable ayant domaine le + petit d'abord" :

    Statistiques :
Nombre de noeuds cree : 21
Nombre de noeuds elagues : 3486784380
Profondeur moyenne des elagages : 3
Profondeur maximum d'elagage : 3

Les solutions sont :

Aucune solution trouvee !

Appuyez sur une touche pour continuer...
```

F. Problème 6

Methode "la variable la + contrainte d'abord" :

Statistiques :

Nombre de noeuds cree : 69

Nombre de noeuds elagues : 13940892646525370299

Profondeur moyenne des elagages : 18.5

Profondeur maximum d'elagage : 37

Les solutions sont :

1	2	3	7	10	11	12	13	25	26
	1	2	6	7	8	9	10	24	25
		1	5	6	7	8	9	23	24
			1	2	3	4	5	22	23
				1	2	3	4	21	22
					1	2	3	20	21
						1	2	3	4
							1	2	3
								1	2
									1

Methode "la variable la - contrainte d'abord" :

Statistiques :

Nombre de noeuds cree : 88548

Nombre de noeuds elagues : 13940892646525281820

Profondeur moyenne des elagages : 14.2485

Profondeur maximum d'elagage : 45

Les solutions sont :

1	2	3	7	10	11	12	13	25	26
	1	2	6	7	8	9	10	24	25
		1	5	6	7	8	9	23	24
			1	2	3	4	5	22	23
				1	2	3	4	21	22
					1	2	3	20	21
						1	2	3	4
							1	2	3
								1	2
									1

Methode "les variables dans leur ordre de declaration" :

Statistiques :

Nombre de noeuds cree : 17796

Nombre de noeuds elagues : 13940892646525352572

Profondeur moyenne des elagages : 24.8204

Profondeur maximum d'elagage : 32

Les solutions sont :

1	2	3	7	10	11	12	13	25	26
	1	2	6	7	8	9	10	24	25
		1	5	6	7	8	9	23	24
			1	2	3	4	5	22	23
				1	2	3	4	21	22
					1	2	3	20	21
						1	2	3	4
							1	2	3
								1	2
									1

Methode "la variable ayant domaine le + petit d'abord" :

Statistiques :

Nombre de noeuds cree : 159

Nombre de noeuds elagues : 13940892646525370209

Profondeur moyenne des elagages : 24.339

Profondeur maximum d'elagage : 28

Les solutions sont :

1	2	3	7	10	11	12	13	25	26
	1	2	6	7	8	9	10	24	25
		1	5	6	7	8	9	23	24
			1	2	3	4	5	22	23
				1	2	3	4	21	22
					1	2	3	20	21
						1	2	3	4
							1	2	3
								1	2
									1

Appuyez sur une touche pour continuer...

G. Problème 7

Methode "la variable la + contrainte d'abord" :

Le problème posé n'a pas de solution, donc le programme est très long pour trouver la réponse (en effet, nous sommes devant des contraintes de somme, qui sont très longues à gérer pour le programme). Ainsi, par manque d'optimisation, nous ne pouvons pas montrer l'affichage du message.

H. Problème customisé

Ce problème que nous avons inventé met en jeu les **sommes pondérées égales à une constante ou à une variable**.

M. Dupont se rend au marché acheter des fruits. Un commerçant l'interpelle et lui propose une énigme : il a sur son étalage quatre paniers de fruits ; si M. Dupont arrive à deviner le prix à l'unité de chaque fruit ainsi que le prix de chaque panier, alors il pourra repartir avec l'un deux.

Les paniers de fruits sont les suivants :

Panier 1 : 5 kiwis, 2 pastèques, 4 pommes, 8 poires

Panier 2 : 3 litchis, 2 clémentines, 2 oranges, 5 kiwis, 8 pommes

Panier 3 : 3 kiwis, 5 litchis, 2 oranges

Panier 4 : 3 pastèques, 4 poires, 8 clémentines

Pour l'aider, le commerçant lui révèle certaines informations :

*Les **paniers 1 et 3** ont le **même prix**.*

*Le **panier 2** est **plus cher** que le **panier 4**.*

*Le **panier 4** est **moins cher** que les **paniers 1 et 3**.*

*Le prix d'une **pastèque** est de **5€**.*

*Le prix d'un **litchi** est de **8€**.*

*Le prix d'une **orange** est de **3€**.*

Pouvez-vous aider M. Dupont ?

X, Y et Z => les paniers

A, B..., G => les fruits

```

Methode "la variable la + contrainte d'abord" :

Statistiques :
Nombre de noeuds cree : 11
Nombre de noeuds elagues : 159999999989
Profondeur moyenne des elagages : 1
Profondeur maximum d'elagage : 1

Les solutions sont :

A => 2
B => 5
C => 6
D => 1
E => 8
F => 1
G => 3
X => 52
Y => 90
Z => 27

```

```

Methode "la variable la - contrainte d'abord" :

Statistiques :
Nombre de noeuds cree : 1241
Nombre de noeuds elagues : 159999998759
Profondeur moyenne des elagages : 0.995
Profondeur maximum d'elagage : 2

Les solutions sont :

A => 2
B => 5
C => 2
D => 3
E => 8
F => 1
G => 3
X => 52
Y => 58
Z => 35

```

Methode "les variables dans leur ordre de declaration" :

Statistiques :
Nombre de noeuds cree : 11
Nombre de noeuds elagues : 159999999989
Profondeur moyenne des elagages : 0
Profondeur maximum d'elagage : 0

Les solutions sont :

A => 2
B => 5
C => 2
D => 3
E => 8
F => 1
G => 3
X => 52
Y => 58
Z => 35

Methode "la variable ayant domaine le + petit d'abord" :

Statistiques :
Nombre de noeuds cree : 11
Nombre de noeuds elagues : 159999999989
Profondeur moyenne des elagages : 3
Profondeur maximum d'elagage : 3

Les solutions sont :

A => 2
B => 5
C => 2
D => 3
E => 8
F => 1
G => 3
X => 52
Y => 58
Z => 35

Appuyez sur une touche pour continuer...

I. SEND + MORE = MONEY

```
    Methode "la variable la + contrainte d'abord" :  
  
    Statistiques :  
Nombre de noeuds cree : 910  
Nombre de noeuds elagues : 1295999999999999090  
Profondeur moyenne des elagages : 5.26061  
Profondeur maximum d'elagage : 6  
  
Les solutions sont :  
  
Ret[1] => 1  
Ret[2] => 0  
Ret[3] => 1  
Ret[4] => 1  
Arr[1] => 9  
Arr[2] => 5  
Arr[3] => 6  
Arr[4] => 7  
Arr[5] => 1  
Arr[6] => 0  
Arr[7] => 8  
Arr[8] => 2  
S => 9  
E => 5  
N => 6  
D => 7  
M => 1  
O => 0  
R => 8  
Y => 2
```

Methode "la variable la - contrainte d'abord" :

Statistiques :

Nombre de noeuds cree : 335

Nombre de noeuds elagues : 129599999999999665

Profondeur moyenne des elagages : 2.48182

Profondeur maximum d'elagage : 4

Les solutions sont :

Ret[1] => 1

Ret[2] => 0

Ret[3] => 1

Ret[4] => 1

Arr[1] => 9

Arr[2] => 5

Arr[3] => 6

Arr[4] => 7

Arr[5] => 1

Arr[6] => 0

Arr[7] => 8

Arr[8] => 2

S => 9

E => 5

N => 6

O => 7

M => 1

D => 0

R => 8

Y => 2


```
    Methode "les variables dans leur ordre de declaration" :

    Statistiques :
Nombre de noeuds cree : 166
Nombre de noeuds elagues : 1295999999999999834
Profondeur moyenne des elagages : 5.96667
Profondeur maximum d'elagage : 8

Les solutions sont :

Ret[1] => 1
Ret[2] => 0
Ret[3] => 1
Ret[4] => 1
Arr[1] => 9
Arr[2] => 5
Arr[3] => 6
Arr[4] => 7
Arr[5] => 1
Arr[6] => 0
Arr[7] => 8
Arr[8] => 2
S => 9
E => 5
N => 6
D => 7
M => 1
O => 0
R => 8
Y => 2
```

```
    Methode "la variable ayant domaine le + petit d'abord" :

    Statistiques :
Nombre de noeuds cree : 53
Nombre de noeuds elagues : 129599999999999947
Profondeur moyenne des elagages : 9.82609
Profondeur maximum d'elagage : 12

Les solutions sont :

Ret[1] => 1
Ret[2] => 0
Ret[3] => 1
Ret[4] => 1
Arr[1] => 9
Arr[2] => 5
Arr[3] => 6
Arr[4] => 7
Arr[5] => 1
Arr[6] => 0
Arr[7] => 8
Arr[8] => 2
S => 9
E => 5
N => 6
D => 7
M => 1
O => 0
R => 8
Y => 2
```