

Réunion du 07/06

07/06/2024

Configuration d'un groupe au Musée

			L1	3-6 ans						7-12 ans						13-18 ans						Adulte					
			L2	débutant			avancé			débutant			avancé			débutant			avancé			débutant			avancé		
C1	C2	C3	L3	1p	2p	3-6p	1p	2p	3-6p	1p	2p	3-6p	1p	2p	3-6p	1p	2p	3-6p	1p	2p	3-6p	1p	2p	3-6p	1p	2p	3-6p
Apprendre	Aucun	30min		p7	p7					p2	p2					p4	p4	p6				p9c	p9c	p9f		p10	
		1h		p8	p8					p1	p1					p3	p3	p5				p9n	p9b	p9e	p12	p12	p12
		1h30																				p9	p9	p9d	p11	p11	p11
	cognitif	30min																									
		1h																									
		1h30																									
	auditif	30min																									
		1h																									
		1h30																									
	visuel	30min																									
		1h																									
		1h30																									
S'amuser	Aucun	30min																									
		1h																									
		1h30																									
	cognitif	30min																									
		1h																									
		1h30																									
	auditif	30min																									
		1h																									
		1h30																									
	visuel	30min																									
		1h																									
		1h30																									
Expérimenter	Aucun	30min																									
		1h																									
		1h30																									
	cognitif	30min																									
		1h																									
		1h30																									
	auditif	30min																									
		1h																									
		1h30																									
	visuel	30min																									
		1h																									
		1h30																									

Configuration d'un groupe au Musée

Variables

```
Nombre = model.new_int_var(1, 3, "Nombre")
Age = model.new_int_var(1, 4, "Age")
Xp = model.new_int_var(1, 2, "Xp")
Humeur = model.new_int_var(1, 3, "Humeur")
Handicap = model.new_int_var(1, 4, "Handicap")
Duree = model.new_int_var(1, 3, "Durée")
```

```
# Boolean variable
Age_is_1 = model.NewBoolVar('age_is_1')
Age_is_2 = model.NewBoolVar('age_is_2')
Age_is_4 = model.NewBoolVar('age_is_4')
Handicap_is_1 = model.NewBoolVar('Handicap_is_1')
Handicap_is_2 = model.NewBoolVar('Handicap_is_2')
```

Association booléen/variable

```
#Associate the Boolean variable with the CSP variable
model.add(Age!=1).OnlyEnforceIf(Age_is_1.Not())
model.add(Age==1).OnlyEnforceIf(Age_is_1)
model.add(Age!=2).OnlyEnforceIf(Age_is_2.Not())
model.add(Age==2).OnlyEnforceIf(Age_is_2)
model.add(Age!=4).OnlyEnforceIf(Age_is_4.Not())
model.add(Age==4).OnlyEnforceIf(Age_is_4)
```

Contraintes

```
model.add(Nombre!=1).OnlyEnforceIf(Age_is_1)
model.add(Nombre!=1).OnlyEnforceIf(Age_is_2)
```

Configuration d'un groupe au Musée

Résultats:

```
Nombre=1 Age=4 Xp=2 Humeur=2 Handicap=4 Durée=2
Nombre=1 Age=4 Xp=2 Humeur=1 Handicap=4 Durée=2
Nombre=1 Age=4 Xp=2 Humeur=1 Handicap=3 Durée=2
Nombre=1 Age=4 Xp=2 Humeur=2 Handicap=3 Durée=2
Nombre=1 Age=4 Xp=2 Humeur=3 Handicap=3 Durée=2
Nombre=1 Age=4 Xp=2 Humeur=3 Handicap=4 Durée=2
Nombre=1 Age=4 Xp=2 Humeur=3 Handicap=4 Durée=1
Nombre=1 Age=4 Xp=2 Humeur=3 Handicap=3 Durée=1
Nombre=2 Age=4 Xp=2 Humeur=3 Handicap=3 Durée=1
Nombre=2 Age=4 Xp=2 Humeur=3 Handicap=4 Durée=1
Nombre=2 Age=4 Xp=2 Humeur=3 Handicap=4 Durée=2
Nombre=2 Age=4 Xp=2 Humeur=3 Handicap=3 Durée=2
Nombre=2 Age=4 Xp=2 Humeur=2 Handicap=3 Durée=2
Nombre=2 Age=4 Xp=2 Humeur=1 Handicap=3 Durée=2
Nombre=2 Age=4 Xp=2 Humeur=1 Handicap=4 Durée=2
Nombre=2 Age=4 Xp=2 Humeur=2 Handicap=4 Durée=2
Nombre=2 Age=4 Xp=2 Humeur=2 Handicap=4 Durée=1
Nombre=2 Age=4 Xp=2 Humeur=1 Handicap=4 Durée=1
Nombre=2 Age=4 Xp=2 Humeur=1 Handicap=3 Durée=1
Nombre=2 Age=4 Xp=2 Humeur=2 Handicap=3 Durée=1
Nombre=3 Age=4 Xp=2 Humeur=2 Handicap=3 Durée=1
Nombre=3 Age=4 Xp=2 Humeur=2 Handicap=4 Durée=1
Nombre=3 Age=4 Xp=2 Humeur=2 Handicap=4 Durée=2
Nombre=3 Age=4 Xp=2 Humeur=2 Handicap=3 Durée=2
Nombre=3 Age=4 Xp=2 Humeur=3 Handicap=3 Durée=2
Nombre=3 Age=4 Xp=2 Humeur=3 Handicap=4 Durée=2
Nombre=3 Age=4 Xp=2 Humeur=3 Handicap=4 Durée=1
Nombre=3 Age=4 Xp=2 Humeur=3 Handicap=3 Durée=1
Nombre=3 Age=4 Xp=2 Humeur=1 Handicap=3 Durée=1
Nombre=3 Age=4 Xp=2 Humeur=1 Handicap=3 Durée=2
Nombre=3 Age=4 Xp=2 Humeur=1 Handicap=4 Durée=2
Nombre=3 Age=4 Xp=2 Humeur=1 Handicap=4 Durée=1
Status = OPTIMAL
Number of solutions found: 258
0.01795363426208496
```

Configuration des composants

Exercice 4 (Configuration de produits). Le but est de simuler la réalisation d'un produit complexe à partir de composants. On considère ici des ordinateurs. Chaque ordinateur doit avoir : un processeur (P), de la mémoire vive (M) et un disque dur (D). On a le choix entre 3 types de P (P_1 , P_2 , P_3), 4 types de M (M_1 , M_2 , M_3 , M_4) et 3 types de D (D_1 , D_2 , D_3). L'indice indique l'année de sortie. On suppose qu'on a une fonction cout, qui associe un coût réel à chaque composant. Contraintes :

- P_1 ne marche pas avec le composant D_3
- un processeur doit avoir une mémoire au moins aussi récente que lui.
- le seul disque possible avec P_2+M_2 est le disque D_2
- le coût total du produit doit être inférieur à une valeur C.

Particularité : deux variables pour une condition (3e point)

Configuration des composants

Variables

```
P = model.new_int_var(1, 3, "Processeur")
M = model.new_int_var(1, 4, "Mémoire")
D = model.new_int_var(1, 3, "Disque")
C = 7
```

```
#Boolean variables
P_is_1 = model.NewBoolVar('P_is_1')
P_is_2 = model.NewBoolVar('P_is_2')
M_is_2 = model.NewBoolVar('M_is_2')
D_is_2 = model.NewBoolVar('D_is_2')
PM_is_2 = model.NewBoolVar('PM_is_2')
```

Association booléen/variable

```
#The only disk possible with P2 and M2 is disk D2
model.Add(P == 2).OnlyEnforceIf(P_is_2)
model.Add(M==2).OnlyEnforceIf(M_is_2)
model.Add(P != 2).OnlyEnforceIf(P_is_2.Not())
model.Add(M!=2).OnlyEnforceIf(M_is_2.Not())
model.Add(D==2).OnlyEnforceIf(D_is_2)
model.Add(D!=2).OnlyEnforceIf(D_is_2.Not())
```

Contraintes

```
model.AddBoolAnd(P_is_2,M_is_2).OnlyEnforceIf(PM_is_2)
model.Add(PM_is_2==1).OnlyEnforceIf([P_is_2,M_is_2])
model.AddImplication(PM_is_2,D_is_2)
```

Configuration des composants

Résultats pour C=6

```
Processeur=1 Mémoire=3 Disque=2  
Processeur=1 Mémoire=3 Disque=1  
Processeur=1 Mémoire=2 Disque=1  
Processeur=1 Mémoire=2 Disque=2  
Processeur=1 Mémoire=1 Disque=2  
Processeur=1 Mémoire=1 Disque=1  
Processeur=1 Mémoire=4 Disque=1  
Processeur=2 Mémoire=3 Disque=1  
Processeur=2 Mémoire=2 Disque=2  
Status = OPTIMAL  
Number of solutions found: 9  
0.0038399696350097656
```

Configuration d'une voiture Renault

Fichier XML contenant les différentes informations du problème (domaine, variable, contrainte)

Particularité :

- Beaucoup de variables
- Beaucoup de contraintes
- Des contraintes avec plus de deux variables pour la conditions

Domaine

```
<domains nbDomains="15">
<domain name="D0" nbValues="9">0..8</domain>
<domain name="D1" nbValues="5">0..4</domain>
<domain name="D2" nbValues="25">0..24</domain>
<domain name="D3" nbValues="2">0..1</domain>
<domain name="D4" nbValues="42">0..41</domain>
<domain name="D5" nbValues="1">0</domain>
<domain name="D6" nbValues="6">0..5</domain>
<domain name="D7" nbValues="3">0..2</domain>
<domain name="D8" nbValues="4">0..3</domain>
```

Variable

```
<variables nbVariables="99">
<variable domain="D0" name="1"/>
<variable domain="D1" name="2"/>
<variable domain="D2" name="3"/>
<variable domain="D3" name="4"/>
<variable domain="D4" name="5"/>
<variable domain="D3" name="6"/>
<variable domain="D3" name="7"/>
<variable domain="D3" name="8"/>
<variable domain="D3" name="9"/>
<variable domain="D5" name="10"/>
```

Contrainte

```
<constraints nbConstraints="111">
<constraint arity="9" name="C0" reference="R0" scope="1 3 81 94 95 96 99 100 101"/>
<constraint arity="3" name="C1" reference="R1" scope="3 5 6"/>
```

Relation

```
<relation arity="3" name="R1" nbTuples="1050" semantics="supports">6
0 10 1|0 11 1|0 12 0|0 13 1|0 14 1|0 15 1|0 16 1|0 17 1|0 18 1|0 19
0 28 0|0 29 1|0 30 1|0 31 1|0 32 1|0 33 1|0 34 1|0 35 1|0 36 1|0 37
```


Configuration d'une voiture Renault

Variables

```
variables = root.findall('./variable')
dico_variable={}
for variable in variables[:nbvariable] :
    dico_variable[variable.get('name')] = model.new_int_var(0, int(dico_domaine[variable.get('domain')])-1, variable.get('name'))
```

```
choix= []
for k in range(1,i+1):
    choix.append(model.NewBoolVar(f"{lists} et {j} et {k} "))
dernier=model.NewBoolVar('dernier')
```

i = nombre de variables qui est concerné par la contrainte

Association booléen/variable

```
for k in range(i):
    model.Add(dico_variable[var[k]]==int(list[k])).OnlyEnforceIf(choix[k])
    model.Add(dico_variable[var[k]]!=int(list[k])).OnlyEnforceIf(choix[k].Not())
```

var= le nom des variables
list= les valeurs à assigner
choix= les variables booléennes

Contraintes

```
model.AddBoolAnd(choix[:-1]).OnlyEnforceIf(dernier)
model.Add(dernier==1).OnlyEnforceIf(choix[:-1])
model.AddImplication(dernier,choix[i-1])
```

Configuration d'une voiture Renault

Résultats pour la voiture Mégane

10 premières variables sans contrainte : 1 209 600 combinaisons

10 premières variables avec contraintes:

```
Status = OPTIMAL  
Number of solutions found: 756000  
1014.8853089809418
```

10 premières variables avec contrainte et assignation de la variable '5' à 20 :

```
Status = OPTIMAL  
Number of solutions found: 18000  
13.856397151947021
```

OR-TOOLS: <https://developers.google.com/optimization>

Configuration d'un groupe au Musée : MuseoTUI: Apports des interactions tangibles pour la création, le choix et le suivi de parcours de visite personnalisés dans les musées par Stéphanie Rey

Configuration de composant: TD CSP d'algorithmique avancée

Configuration de voitures: Apprentissage de préférences en espace combinatoire et application à la recommandation en configuration interactive de P.F.GIMENEZ