USTHB - FEI - Département d'Informatique
Master 2 ''Systèmes Informatiques Intelligents'' 2019/2020
Module ''Programmation par Contraintes''
Date : 13/01/2020 <13h00-14h30>

Corrigé de l'interrogation

Exercice 1 (10 points)

On considère le CSP binaire discret P=(X,D,C) suivant :

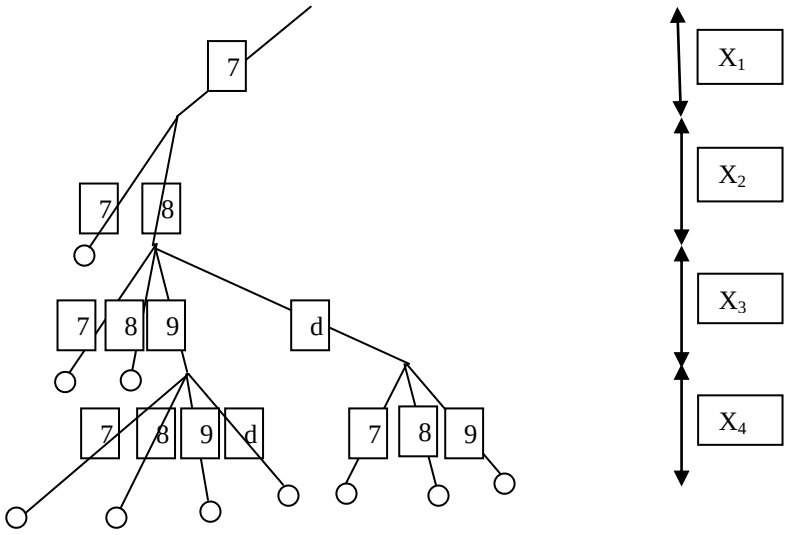
- $X=\{X_1,X_2,X_3,X_4\}$
- $D(X_i)=\{7,8,9,10\}$, pour tout $i\in\{1,2,3,4\}$
- $C=\{c_1 : X_1<X_2,\ c_2 : X_1\neq X_3,\ c_3 : X_1\neq X_4, c_4 : X_2\neq X_3,\ c_5 : X_2\neq X_4,\ c_6 : X_3>X_4\}$

1) Appliquez l'algorithme de recherche SRA (Simple Retour Arrière) au CSP P
2) Appliquez l'algorithme de recherche FC (Forward-Checking) au CSP P
3) Appliquez au CSP P l'algorithme de recherche Look-Ahead
4) Donnez la représentation graphique de P
5) Donnez la représentation matricielle de P

Pour chacun des trois algorithmes de recherche, il est demandé de respecter l'ordre statique X_1, X_2, X_3, X_4 d'instanciation des variables ; et l'ordre statique 7, 8, 9, 10 de choix des valeurs du domaine de la variable en cours d'instanciation. Il est également demandé d'expliquer tous les échecs éventuels.

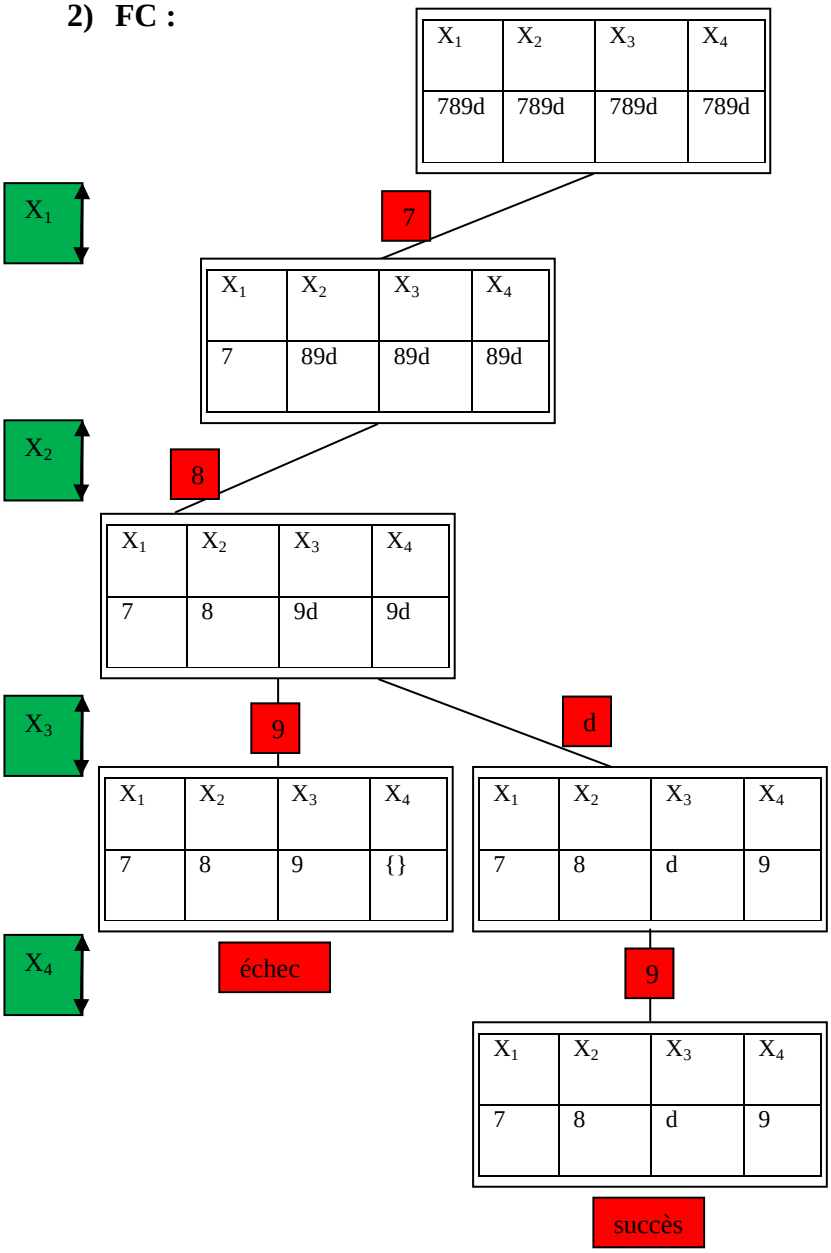
Solution :

1) SRA :

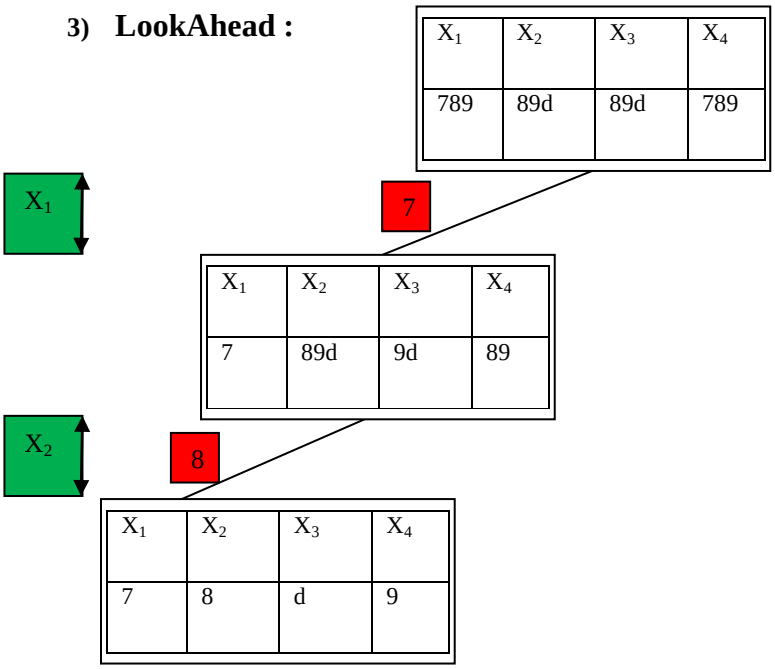


Les 9 premières feuilles de l'arbre de recherche, de gauche à droite, correspondent aux échecs de l'algorithme SRA, pour lesquels nous donnons ci-dessous, respectivement, une contrainte non satisfaite entre la variable en cours d'instanciation et une variable déjà instanciée : c_1, c_2, c_4, c_3, c_5, c_6, c_6, c_3, c_5. La feuille la plus à droite correspond au succès de SRA.

2) FC :

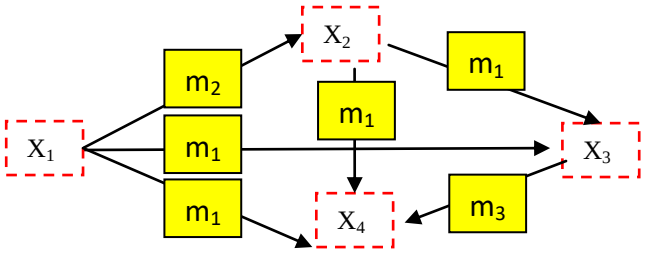

X1 X2 X3 X4
789d 789d 789d 789d
X1
7
X1 X2 X3 X4
7 89d 89d 89d
X2
8
X1 X2 X3 X4
7 8 9d 9d
X3
9
d
X1 X2 X3 X4
7 8 9 {}
échec
X1 X2 X3 X4
7 8 d 9
X4
9
X1 X2 X3 X4
7 8 d 9
succès

3) **LookAhead :**



L'arbre de recherche de LookAhead, jusqu'à l'obtention de la toute première solution, est constitué d'une seule branche de trois nœuds. LookAhead, comme FC, garde trace, au niveau de chaque nœud, de l'évolution des domaines. Les domaines au niveau de la racine sont le résultat du prétraitement : application de AC3 au CSP de départ, avec la file initialisée à toutes les paires sur lesquelles il y a une contrainte : $Q=\{(X_1,X_2), (X_2,X_1), (X_1,X_3), (X_3,X_1), (X_1,X_4), (X_4,X_1), (X_2,X_3), (X_3,X_2), (X_2,X_4), (X_4,X_2), (X_3,X_4), (X_4,X_3)\}$. Au niveau de la racine donc, après le prétraitement, aucun domaine n'est passé à vide, et il y a des domaines qui ne sont pas singletons (disjonctifs). LookAhead choisit la première variable disjonctive dans l'ordre $X_1X_2X_3X_4$, donc la variable X1, qu'il instancie avec 7 : le filtrage avec AC3 est fait au CSP résultant, avec la file initialisée comme suit : $Q=\{(X_2,X_1), (X_3,X_1), (X_4,X_1)\}$ (il convient de noter ici qu'on peut initialiser la file à tous les arcs sur lesquels il y a une contrainte ; mais pour optimiser, on a tiré profit du fait qu'au niveau du nœud père, le CSP est consistant d'arc). Enfin, le passage du deuxième nœud au troisième (la feuille) se fait par instanciation de X_2 avec 8, et filtrage avec AC3, avec la file initialisée comme suit : $Q=\{(X_1,X_2), (X_3,X_2), (X_4,X_2)\}$. Les domaines du CSP résultant sont tous singletons : LookAhead prend fin avec succès, avec une solution donnée par les domaines.

4) **Représentation graphique :** Gp=(X,E,l)



Avec $m_1=\begin{bmatrix}0&1&1&1\\1&0&1&1\\1&1&0&1\\1&1&1&0\end{bmatrix}$, $m_2=\begin{bmatrix}0&1&1&1\\0&0&1&1\\0&0&0&1\\0&0&0&0\end{bmatrix}$ et $m_3=\begin{bmatrix}0&0&0&0\\1&0&0&0\\1&1&0&0\\1&1&1&0\end{bmatrix}$ (m_3 transposée de m_2)

5) Représentation matricielle M_P :

	X_1	X_2	X_3	X_4
X_1	I_4	m_2	m_1	m_1
X_2	m_3	I_4	m_1	m_1
X_3	m_1	m_1	I_4	m_3
X_4	m_1	m_1	m_2	I_4

Exercice 2 (3 points)

On considère l'instance de la table ci-dessous du problème d'ordonnancement de type job shop, qui consiste en deux jobs J1 et J2 devant passer chacun par deux machines M1 et M2 :

	$1^{ère}$ tâche : <machine, durée>	$2^{ème}$ tâche : <machine, durée>
Job J1	<M2,8>	<M1,3>
Job J2	<M1,2>	<M2,9>

Toutes les tâches sont non-préemptives. La date de début au plus tôt est t_d=4.

On s'intéresse au problème de recherche d'une solution réalisable, c'est-à-dire satisfaisant toutes les contraintes mais ne donnant pas forcément l'optimum de l'instance.

1) Modélisez le problème à l'aide d'un TCSP P=(X,C)
2) Donnez la représentation graphique de P

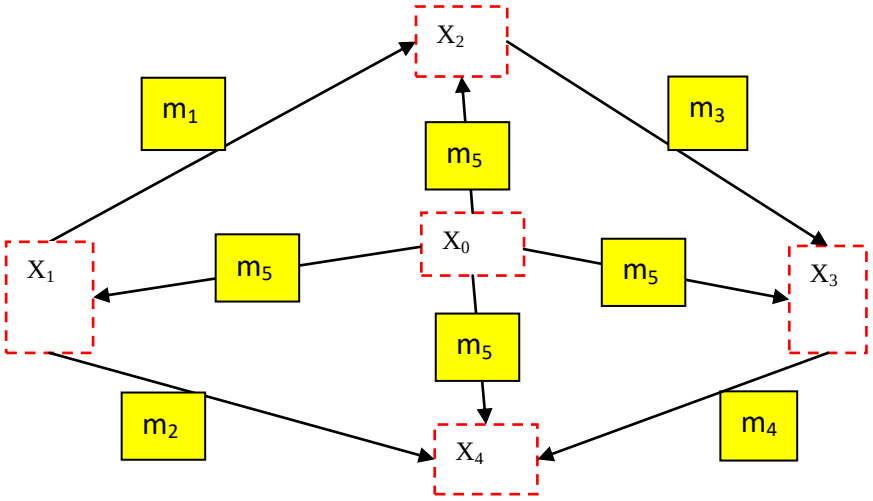
Solution :

1) **Modélisation** à l'aide d'un TCSP P=(X,C) **:**

   L'instance du problème de job shop est constituée de quatre tâches : T_1 (passage du job J1 par la machine M2), T_2 (passage du job J1 par la machine M1), T_3 (passage du job J2 par la machine M1), et T_4 (passage du job J2 par la machine M2). L'instance peut être modélisée par le TCSP P=(X,C) suivant :

   - $X=\{X_0,X_1,X_2,X_3,X_4\}$, X_0 désignant la variable ''origine du monde'', la variable X_i (i de 1 à 4) désignant la date de début de la tâche T_i
   - $C=\{c_1 : (X_2-X_1)\in[8,+\infty[, c_2 : (X_4-X_3)\in[2,+\infty[, c_3 : (X_4-X_1)\in]-\infty,-9]\cup[8,+\infty[, c_4 : (X_3-X_2)\in]-\infty,-2]\cup[3,+\infty[, c_5 : (X_1-X_0)\in[4,+\infty[, c_6 : (X_2-X_0)\in[4,+\infty[, c_7 : (X_3-X_0)\in[4,+\infty[, c_8 : (X_4-X_0)\in[4,+\infty[\}$

2) **Représentation graphique :** Gp=(X,E,l)



Avec m_1=[8,+∞[**, m_2**=]-∞,-9]∪[8,+∞[**, m_3**=]-∞,-2]∪[3,+∞[, **m_4**=[2,+∞[**et m_5**=[4,+∞[

Exercice 3 (7 points) :

On représente une matrice mxn par une liste de m listes de longueur n. Donnez un programme Prolog inversant les lignes d'indice pair d'une matrice, et ajoutant 1 aux éléments des lignes d'indice impair.

Solution :

```
%La premère ligne d'une matrice est d'indice 1%
%mystere([[10,20,30,40,50],[1,2,3,4,5],[2,3,4,5,6],[7,9,12,88,7689]],L).%
mystere([],[]):-!.
mystere([L1],[L2]):-!,
        ajout_un(L1,L2).
mystere([L1,L2|L3],[L4,L5|L6]):-ajout_un(L1,L4),
        inverse(L2,L5),
        mystere(L3,L6).
ajout_un([],[]):-!.
ajout_un([X|L1],[Y|L2]):-Y is X+1,
        ajout_un(L1,L2).
inverse([],[]):-!.
inverse([X|L1],L2):-inverse(L1,L3),
        append(L3,[X],L2).
```

Déroulement:

```
1 ?- mystere([[10,20,30,40,50],[1,2,3,4,5],[2,3,4,5,6],[7,9,12,88,7689]],L).
L = [[11, 21, 31, 41, 51], [5, 4, 3, 2, 1], [3, 4, 5, 6, 7], [7689, 88, 12, 9, 7]].

2 ?-
```