Université des Sciences et de la Technologie Houari Boumediène
Faculté d'Electronique et d'Informatique
Département d'Informatique
LMD Master 2ème Année ''Systèmes Informatiques Intelligents'' 2014/2015
Module ''Programmation par Contraintes''
Date : 17/12/2014 <11 h 20-12 h 50>

Corrigé de l'interrogation

Exercice 1 (6 points=2+2+2) :
On considère le problème d'ordonnancement de type job shop donné par la table ci-dessous, qui consiste en deux jobs J1 et J2 devant passer chacun par deux machines M1 et M2 :

	1ère tâche : <machine, durée>	2ème tâche : <machine, durée>
Job J1	<M1,2>	<M2,2>
Job J2	<M2,3>	<M1,1>

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

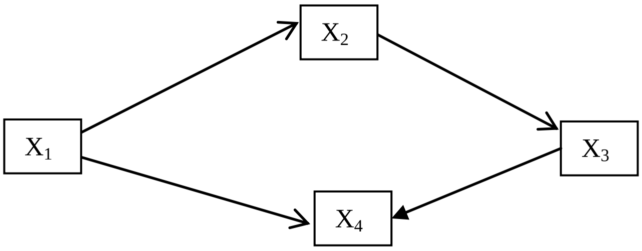
1. Modélisez le problème à l'aide d'un CSP discret P=(X,D,C)
2. Donnez la représentation graphique de P
3. Si on veut adapter un des quatre algorithmes vus en cours, de recherche récursive de solution d'un CSP discret, de telle sorte qu'il fournisse, pour un CSP modélisant un problème de job shop, une solution réalisant l'optimum, comment doit une telle adaptation initialiser l'optimum de P et la solution le réalisant ?

Solution :

1. Le problème de job shop est constitué de quatre tâches : T_1 (passage du job J1 par la machine M1), T_2 (passage du job J1 par la machine M2), T_3 (passage du job J2 par la machine M2), et T_4 (passage du job J2 par la machine M1). La somme des durées des tâches est S=2+2+3+1=8. Si on décide de placer les tâches à partir de la date 1, le problème peut être modélisé par le CSP discret P=(X,D,C) suivant :
    - $X=\{X_1,X_2,X_3,X_4\}$, la variable X_i désignant le début de la tâche T_i
    - $D(X_1)=D(X_2)=\{1,2,3,4,5,6,7\}$; $D(X_3)=\{1,2,3,4,5,6\}$;$D(X_4)=\{1,2,3,4,5,6,7,8\}$
    - $C=\{c_1 : (X_2-X_1)\in[2,+\infty[, c_2 : (X_4-X_3)\in[3,+\infty[, c_3 : (X_4-X_1)\in]-\infty,-1]\cup[2,+\infty[, c_3 : (X_3-X_2)\in]-\infty,-3]\cup[2,+\infty[\}$

    Le CSP est constitué de deux contraintes conjonctives (de précédence) et de deux contraintes disjonctives : les contraintes conjonctives c_1 et c_2 traduisent la précédence entre tâches d'un même job (la tâche T_1 du job J1 avant la tâche T_2 du même job, exprimé par c_1 ; la tâche T_3 du job J2 avant la tâche T_4 du même job, exprimé par c_2) ; les contraintes disjonctives, quant à elles, interdisent tout chevauchement entre tâches devant s'exécuter sur une même machine (c_3 interdit tout chevauchement entre T_1 et T_4 qui doivent toutes les deux passer par la machine M1 ; c_4, à son tour, interdit tout chevauchement entre les deux tâches devant passer par la machine M2, les tâches T_2 et T_3).

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



Les poids des arcs sont comme suit :

(X_i,X_j)	(X_1,X_2)
I(Xi,Xj)	

	1	2	3	4	5	6	7
1	0	0	1	1	1	1	1
2	0	0	0	1	1	1	1
3	0	0	0	0	1	1	1
4	0	0	0	0	0	1	1
5	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

(X_i,X_j)	(X_3,X_4)
I(Xi,Xj)	

	1	2	3	4	5	6	7	8
1	0	0	0	1	1	1	1	1
2	0	0	0	0	1	1	1	1
3	0	0	0	0	0	1	1	1
4	0	0	0	0	0	0	1	1
5	0	0	0	0	0	0	0	1
6	0	0	0	0	0	0	0	0

(X_i,X_j)	(X_1,X_4)
$I(X_i,X_j)$	

	1	2	3	4	5	6	7	8
1	0	0	1	1	1	1	1	1
2	1	0	0	1	1	1	1	1
3	1	1	0	0	1	1	1	1
4	1	1	1	0	0	1	1	1
5	1	1	1	1	0	0	1	1
6	1	1	1	1	1	0	0	1
7	1	1	1	1	1	1	0	0

(X_i,X_j)	(X_2,X_3)
$I(X_i,X_j)$	

	1	2	3	4	5	6
1	0	0	1	1	1	1
2	0	0	0	1	1	1
3	0	0	0	0	1	1
4	1	0	0	0	0	1
5	1	1	0	0	0	0
6	1	1	1	0	0	0
7	1	1	1	1	0	0

3. L'initialisation de l'optimum et de la solution le réalisant peut être réalisée de la façon suivante, en plaçant les tâches à partir de la date 1 :
   - On place d'abord les tâches du job J1 : la première tâche, T_1, suivie de la deuxième, T_2, sans laisser de vide entre les deux
   - On place ensuite, à la suite de la deuxième tâche T_2 de J1, les deux tâches du job J2 : la première, T_3, à la suite de T_2 sans laisser de vide entre elles ; puis la deuxième, T_4, sans laisser de vide entre elle et T_3

   Cette initialisation respecte aussi bien les contraintes conjonctives que les contraintes disjonctives du CSP modélisant l'instance du problème de job shop :
   - Opt=some des durées des tâches=2+2+3+1=8
   - Solution=(1,1+2,1+2+2,1+2+2+3)=(1,3,5,8)

Exercice 2 (5=2+1+2)

On représente le problème des quatre reines sous la forme d'un CSP binaire discret P=(X,D,C) :
- $X=\{X_1,X_2,X_3,X_4\}$
- $D(X_i)=\{1,2,3,4\}$ (la reine i se déplace sur la ligne i, et la variable X_i désigne sa position sur la ligne)

1. Exprimez M_a, la matrice booléenne représentant les contraintes entre X_i et X_{i+1} (0<i<4), puis M_b, la matrice booléenne représentant les contraintes entre X_i et X_{i+2} (0<i<3).
2. Exprimez M_c, la matrice booléenne représentant les contraintes entre X_i et X_{i+3} (0<i<2).
3. Appliquez l'algorithme de recherche Forward-Checking au CSP P. Représentez au niveau de chaque nœud de l'arbre de recherche l'évolution des domaines par une grille 4x4 dont vous marquerez les cases correspondant aux valeurs non supprimées des domaines par un point, les autres cases devant être laissées vides. Respectez l'ordre statique X_1, X_2, X_3, X_4 d'instanciation des variables ; et l'ordre statique 1, 2, 3, 4 de choix des valeurs du domaine commun {1,2,3,4}.

Solution :

1. $M_a=M_P[1,2]=M_P[2,3]=M_P[3,4]=$

	1	2	3	4
1	0	0	1	1
2	0	0	0	1
3	1	0	0	0
4	1	1	0	0

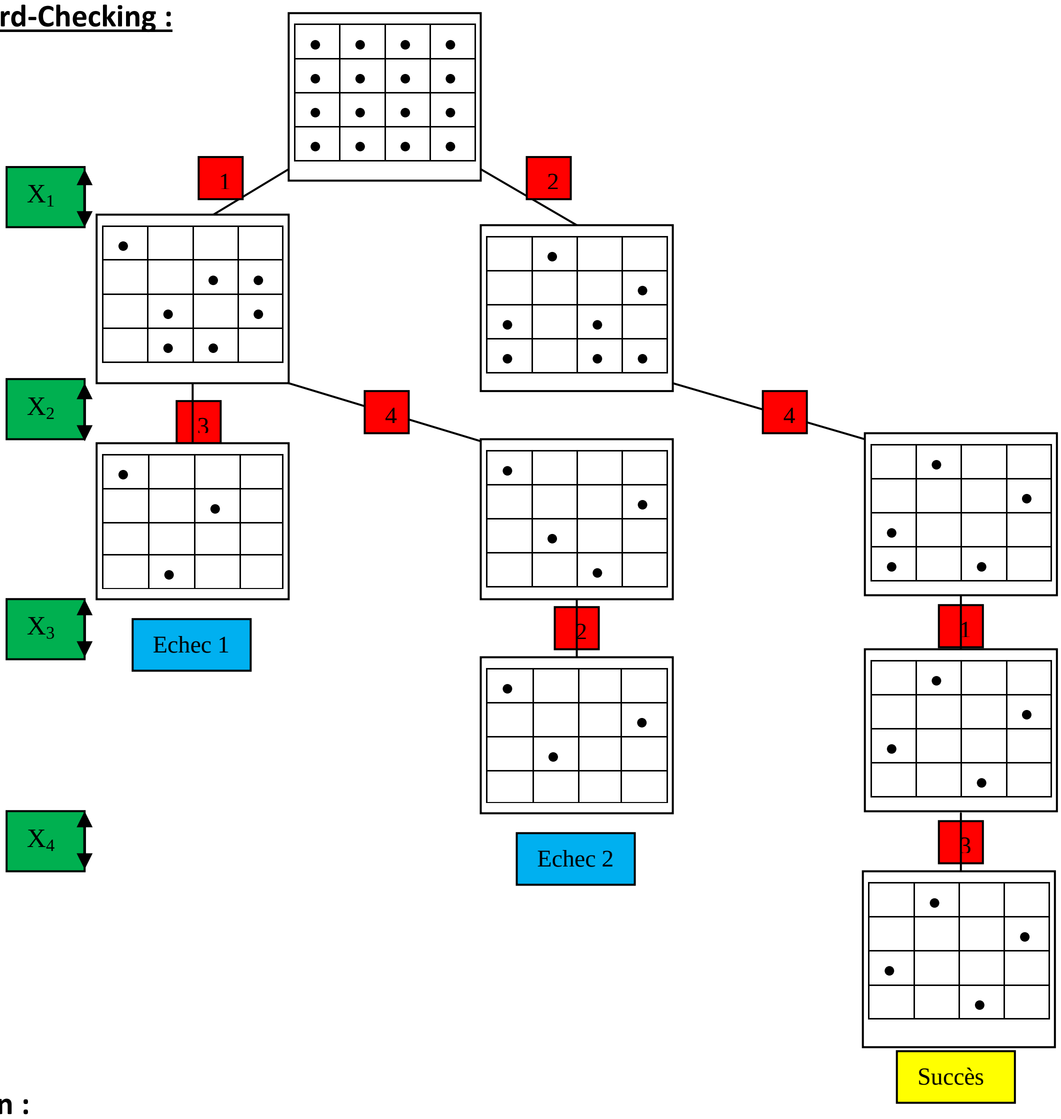
$M_b=M_P[1,3]=M_P[2,4]=$

	1	2	3	4
1	0	1	0	1
2	1	0	1	0
3	0	1	0	1
4	1	0	1	0

2. $M_c=M_P[1,4]=$

	1	2	3	4
1	0	1	1	0
2	1	0	1	1
3	1	1	0	1
4	0	1	1	0

3. <u>**Forward-Checking :**</u>



Explication :

- **Echec 1** : Instanciation de X_2 avec la valeur 3 de son domaine. On supprime des domaines des variables non encore instanciées, X_3 et X_4, les valeurs qui ne sont pas compatibles avec ce choix 3 pour l'instanciation de X_2 : ceci rend vide le domaine de X_3
- **Echec 2** : Instanciation de X_3 avec la valeur 2 de son domaine. On supprime du domaine de l'unique variable non encore instanciée, X_4, les valeurs qui ne sont pas compatibles avec ce choix 2 pour l'instanciation de X_3 : ceci rend vide le domaine de X_4

- **Succès** : Solution=(2,4,1,3)

Exercice 3 (4 points=(1+1)+2) :

1) <u>Composition de deux contraintes TCSP :</u> quelles sont les contraintes sur les variables X_1 et X_3 inférée des deux contraintes $(X_2-X_1)\in[2,5]\cup[11,14]$ et $(X_3-X_2)\in[15,20]\cup[35,37]$:
   a. Par composition ?
   b. Par composition faible ?
2) <u>Composition de deux contraintes de directions cardinales :</u> quelle est la contrainte sur les variables X_1 et X_3 inférée par composition des deux contraintes $\{NW\}(X_1,X_2)$ et $\{SE,NW\}(X_2,X_3)$?

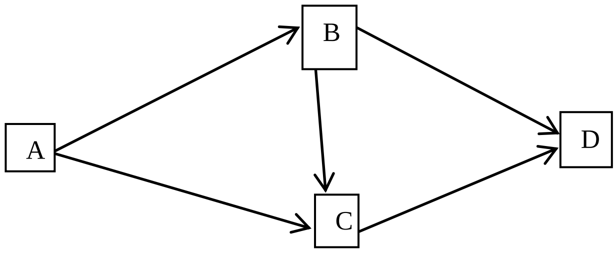
Solution :

1) TCSP : contraintes inférées par composition et par composition faible
   a. **Par composition :**
   $(X_3-X_1)\in([2,5]+[15,20])\cup([2,5]+[35,37])\cup([11,14]+[15,20])\cup([11,14]+[35,37])$
   $\Leftrightarrow(X_3-X_1)\in[17,25]\cup[37,42]\cup[26,34]\cup[46,51]$,
   $\Leftrightarrow(X_3-X_1)\in[17,25]\cup[26,34]\cup[37,42]\cup[46,51]$
   b. **Par composition faible :** la composition faible de deux contraintes est équivalente à la composition de leurs fermetures convexes ; donc à la composition des deux contraintes convexes $(X_2-X_1)\in[2,14]$ et $(X_3-X_2)\in[15,37]$:
   $(X_3-X_1)\in([2,14]+[15,37])$
   $\Leftrightarrow(X_3-X_1)\in[17,51]$
2) CSP de directions cardinales : contrainte inférée par composition
   La contrainte sur les variables X_1 et X_3 inférée par composition des deux contraintes $\{NW\}(X_1,X_2)$ et $\{SE,NW\}(X_2,X_3)$ est comme suit : $(\{NW\}°\{SE\}\cup\{NW\}°\{NW\})(X_1,X_3)$

   La composition $\{NW\}°\{SE\}$ est la relation universelle ? (l'ensemble de toutes les relations atomiques de l'algèbre des directions cardinales). En effet, comme $\{NW\}$ et $\{SE\}$ sont transposées l'une de l'autre, la conjonction $[\{NW\}(X_1,X_2)$ et $\{SE\}(X_2,X_3)]$ est équivalente à $[\{NW\}(X_1,X_2)$ et $\{NW\}(X_3,X_2)]$. Si maintenant X_1 et X_3 sont tous deux au Nord-Ouest de X_2, alors X_1 et X_3 peuvent être reliés par n'importe laquelle des neuf relations atomiques, c'est-à-dire $?(X_1,X_3)$

Exercice 4 (5 points=2,5+2,5) :

1. On s'intéresse au problème de coloriage du graphe ci-dessous avec trois couleurs, de telle sorte que deux sommets adjacents n'aient pas la même couleur. Donnez un programme Prolog modélisant le problème sous forme d'un CSP binaire discret dont il calcule les solutions :



2. Donnez un programme Prolog vérifiant :
   a. Si une liste ne contient pas de redondances.
   b. Si deux listes ne contenant pas de redondances vérifient la relation d'inclusion de l'une dans l'autre.

Solution :

```
/* Question 1. */
/* Utilisation de la bibliothèque incluant le solveur de CSP discrets */
:- use_module(library('clp/bounds')).

coloriable(Vars) :-
    /* on déclare les variables */
    Vars = [X1,X2,X3,X4],
    /* le domaine de chaque variable est {1,2,3} */
    Vars in 1..3,
    /* Deux sommets adjacents ne doivent pas avoir la même couleur */
    X1 #\= X2, X1 #\= X3,X2 #\= X3,X2 #\= X4,X3 #\= X4,
    /* on recherche une affectation des variables qui soit solution */
    label(Vars).

/* Question 2. */
sans_redondances([]).
sans_redondances([_]).
sans_redondances([X,Y|L]) :-n_appartient_pas(X,[Y|L]),
    sans_redondances([Y|L]).

n_appartient_pas(X,L) :-not(element(X,L)).

element(X,[X|_]).
element(X,[_,Y|L]) :-element(X,[Y|L]).

l_une_dans_l_autre_et_sr(L1,L2) :-sans_redondances(L1),
    sans_redondances(L2),
    l_une_dans_l_autre(L1,L2).

l_une_dans_l_autre(L1,L2) :-incluse(L1,L2).
```

```
l_une_dans_l_autre(L1,L2) :-incluse(L2,L1).

incluse([],[]).
incluse([],[_|_]).
incluse([X|L1],L2) :-element(X,L2),
    incluse(L1,L2).
```