Université des Sciences et de la Technologie Houari Boumediène
Faculté d'Electronique et d'Informatique, Département d'Informatique, Master 2 SII 2013/2014
Module ''Programmation par Contraintes''
Date : 09/01/2014 (par Mr Isli)

Corrigé de l'examen

Exercice 1 (6 points=1,5+1+1,5+2) :

On considère le problème d'ordonnancement à trois tâches T_1, T_2 et T_3 et deux machines m_1 et m_2 suivant :

- T_1 et T_2 sont des tâches d'un même job J_1, et doivent être exécutées dans l'ordre T_1, T_2
- Les tâches T_1 et T_3 doivent être exécutées sur la machine m_1
- La tâche T_2 doit être exécutée sur la machine m_2
- Les durées des tâches T_1, T_2 et T_3 sont, respectivement, 2, 3 et 4
- Toutes les tâches sont non-préemptives
- La date de début au plus tôt est t_d=2 et la date de fin au plus tard est t_f=11

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. Donnez la représentation matricielle de P
4. Comment peut-on adapter l'algorithme SRA (Simple Retour Arrière) de telle sorte qu'il fournisse, pour un CSP modélisant un problème d'ordonnancement, une solution réalisant l'optimum ? Donnez la partie de l'arbre de recherche d'une telle solution allant jusqu'à la première amélioration de l'optimum.

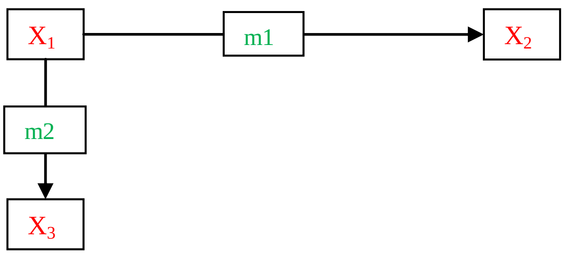
Solution :

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

X={X_1,X_2,X_3} avec X_i, i∈{1,2,3}, variable donnant le début de la tâche T_i. Les contraintes sont données par l'ensemble C={c_1,c_2} construit comme suit :

- c_1 : (X_2-X_1)∈[2,+∞[(contrainte conjonctive, ou de précédence, spécifiant que l'exécution de la tâche T_1 doit précéder celle de la tâche T_2)
- c_2 : (X_3-X_1)∈]-∞,-4]∪[2,+∞[(contrainte disjonctive spécifiant que l'exécution de la tâche T_1 ne doit pas chevaucher celle de la tâche T_3, les deux tâches devant être exécutées sur la même machine m_1)
- D(X_1)={2,…,9}
- D(X_2)={2,…,8}
- D(X_3)={2,…,7}

2) **Représentation graphique de P :**



Avec :
$m_1=M_1(X_1,X_2)=$

	2	3	4	5	6	7	8
2	0	0	1	1	1	1	1
3	0	0	0	1	1	1	1
4	0	0	0	0	1	1	1
5	0	0	0	0	0	1	1
6	0	0	0	0	0	0	1
7	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0

$m_2=M_2(X_1,X_3)=$

	2	3	4	5	6	7
2	0	0	1	1	1	1
3	0	0	0	1	1	1
4	0	0	0	0	1	1
5	0	0	0	0	0	1
6	1	0	0	0	0	0
7	1	1	0	0	0	0
8	1	1	1	0	0	0
9	1	1	1	1	0	0

3) **Représentation matricielle de P :**

$\mathbf{M_P}=$

	X_1	X_2	X_3
X_1	I_8	m_1	m_2
X_2	$(m_1)^t$	I_7	$U_{7,6}$
X_3	$(m_2)^t$	$U_{6,7}$	I_6

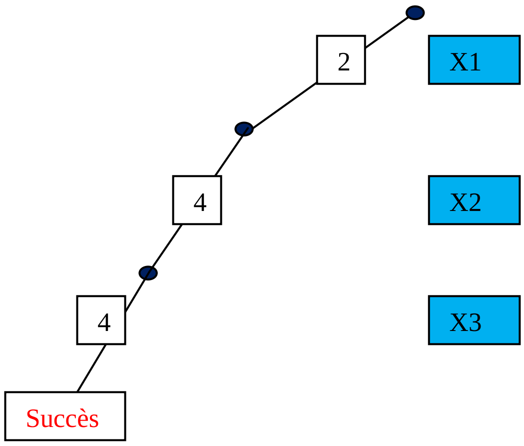
Avec $U_{i,j}$ matrice universelle (des 1 partout) à i lignes et j colonnes ; I_i matrice (carrée) identité (des 1 sur la diagonale et des 0 partout ailleurs) à i lignes et i colonnes ; $(m_i)^t$ matrice transposée de m_i.

4) **Adaptation de l'algorithme SRA :**

a. Initialisation de l'optimum et de la solution le réalisant, en faisant passer toutes les tâches du job 1 (c'est-à-dire T_1 et T_2) puis toutes celles du job 2 (c'est-à-dire T_3) sans laisser de vide entre les tâches : faire commencer T_1, T_2 et T_3 aux instants $t_d=2$, $t_d+\text{durée}(T_1)=4$ et $t_d+\text{durée}(T_1)+\text{durée}(T_2)=7$. On obtient S=(2,4,7) et Opt=9.

b. On adapte SRA de la façon suivante. On l'applique normalement, comme sur un CSP ordinaire, mais on ne s'arrête pas à la première solution éventuelle trouvée ; on continue jusqu'à l'exploration de tout l'espace de recherche. De plus, à chaque fois qu'on réussit à instancier toutes les variables sans être confronté à un échec (on est alors en présence d'une nouvelle solution), on calcule la durée globale du projet correspondant et on met éventuellement à jour la valeur de l'optimum si amélioration.

c. Si on choisit l'ordre d'instanciation statique $X_1X_2X_3$ des variables alors les instanciations $X_1=2$, $X_2=4$ et $X_3=4$ conduisent à une solution réalisable améliorant l'optimum, le faisant passer de la valeur initiale Opt=9 à la valeur Opt=6



Exercice 2 (5 points=1+1+1+2) :
On considère la description qualitative suivante des trois villes Alger, Béjaia et Oran et d'un bâteau :

- Béjaia est à l'est d'Alger
- Oran est à l'ouest d'Alger
- Le bâteau est au nord ou au nord-est de Béjaia
- Le bâteau est au sud-ouest ou au sud-est d'Oran

1. Modélisez la description à l'aide d'un CSP P de directions cardinales
2. Donnez la représentation graphique de P
3. Donnez la représentation matricielle de P
4. Montrez que l'algorithme de consistance de chemin PC2 détecte que P est inconsistant.

Solution :

1) **Modélisation à l'aide d'un CSP de directions cardinales P=(X,C) :**

$X=\{X_1,X_2,X_3,X_4\}$, avec X_1,X_2,X_3 et X_4 variables désignant, respectivement, les positions d'Alger, de Béjaia, d'Oran et du bâteau. Les contraintes sont données par l'ensemble $C=\{c_1,\ldots,c_5\}$ construit comme suit :

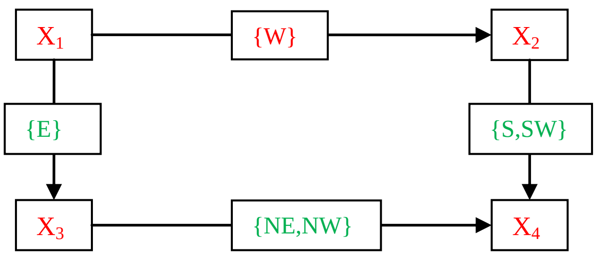
c_1 : $\{E\}(X_2,X_1)$

c_2 : $\{W\}(X_3,X_1)$

c_3 : $\{N,NE\}(X_4,X_2)$

c_4 : $\{SW,SE\}(X_4,X_3)$

2) **Représentation graphique de P :**



3) **Représentation matricielle de P :**

M_P=

	X_1	X_2	X_3	X_4
X_1	{EQ}	{W}	{E}	?
X_2	{E}	{EQ}	?	{S,SW}
X_3	{W}	?	{EQ}	{NE,NW}
X_4	?	{N,NE}	{SW,SE}	{EQ}

avec ? désignant la relation universelle de l'algèbre des directions cardinales, et qui consiste en l'ensemble des neuf relations atomiques de l'algèbre : ?={EQ,N,NE,E,SE,S,SW,W,NW}

4) **Détection de l'inconsistance par l'algorithme PC2 :**

La contrainte sur X_1 et X_4 inférée par composition des contraintes {W}(X_1,X_2) et {S,SW}(X_2,X_4) est {SW}(X_1,X_4). La contrainte sur X_1 et X_4 inférée par composition des contraintes {E}(X_1,X_3) et {NE,NW}(X_3,X_4) est {NE,N,NW}(X_1,X_4). L'intersection des deux contraintes inférées sur la paire (X_1,X_4) est la contrainte d'inconsistance {}(X_1,X_4).

Conclusion : PC2 détecte bien l'inconsistance de P.

Remarque :

- $\{W\}°\{S,SW\}=\{W\}°\{S\}\cup\{W\}°\{SW\}=\{SW\}\cup\{SW\}=\{SW\}$
- $\{E\}°\{NE,NW\}=\{E\}°\{NE\}\cup\{E\}°\{NW\}=\{NE\}\cup\{NE,N,NW\}=\{NE,N,NW\}$

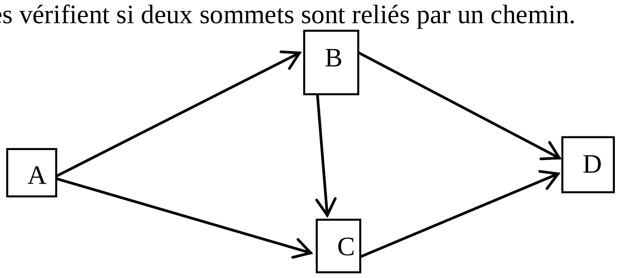
Exercice 3 (4 points=0,5+0,5+0,5+1+1,5) :
Traduire chacune des contraintes d'intervalles suivantes en une conjonction de contraintes TCSP :

a) {<}(X_1,X_2)
b) {m}(X_1,X_2)
c) {o}(X_1,X_2)
d) {<,m}(X_1,X_2)
e) {<,m,o}(X_1,X_2)

Solution :

a) {<}(X_1,X_2) $\equiv(X_{1f}-X_{1d})\in]0,+\infty[\wedge(X_{2f}-X_{2d})\in]0,+\infty[\wedge(X_{2d}-X_{1f})\in]0,+\infty[$
b) {m}(X_1,X_2) $\equiv(X_{1f}-X_{1d})\in]0,+\infty[\wedge(X_{2f}-X_{2d})\in]0,+\infty[\wedge(X_{2d}-X_{1f})\in\{0\}$
c) {o}(X_1,X_2) $\equiv(X_{1f}-X_{1d})\in]0,+\infty[\wedge(X_{2f}-X_{2d})\in]0,+\infty[\wedge(X_{2d}-X_{1d})\in]0,+\infty[\wedge(X_{1f}-X_{2d})\in]0,+\infty[\wedge(X_{2f}-X_{1f})\in]0,+\infty[$
d) {<,m}(X_1,X_2) $\equiv(X_{1f}-X_{1d})\in]0,+\infty[\wedge(X_{2f}-X_{2d})\in]0,+\infty[\wedge(X_{2d}-X_{1f})\in[0,+\infty[$
e) {<,m,o}(X_1,X_2) $\equiv(X_{1f}-X_{1d})\in]0,+\infty[\wedge(X_{2f}-X_{2d})\in]0,+\infty[\wedge(X_{2d}-X_{1d})\in]0,+\infty[\wedge(X_{2f}-X_{1f})\in]0,+\infty[$

Exercice 4 (5 points) : Ecrire un programme Prolog dont les faits représentent le graphe orienté ci-dessous, et les règles vérifient si deux sommets sont reliés par un chemin.



Solution :

```
sommet(A).
sommet(B).
sommet(C).
sommet(D).
arc(A,B).
arc(A,C).
arc(B,D).
arc(B,C).
arc(C,D).
chemin(X,Y) :-arc(X,Y).
chemin(X,Y) :-arc(X,Z),chemin(Z,Y).
```