

Examen Final de Programmation par Contraintes

Questions de Compréhension (8 Pts)

- Quelle est la définition d'une contrainte ?
 - Une contrainte est une propriété qui doit être vérifiée entre différentes variables.
 - Une contrainte est une fonction dirigée entre les variables.
 - Une contrainte spécifie une procédure opérationnelle pour vérifier une relation.
 - Une contrainte est une égalité entre deux expressions arithmétiques.
- Que signifie établir la consistance d'arc pour une affectation partielle ?
 - Enlever du domaine de chaque variable les valeurs qui violent une contrainte binaire avec l'affectation partielle.
 - Enlever du domaine de chaque variable les valeurs telles qu'il n'existe aucune valeur pour les autres variables non affectées qui satisfasse les contraintes.
 - Filtrer le domaine de chaque variable pour ne garder que les valeurs appartenant à une solution.
 - Établir que l'affectation partielle peut être étendue en une solution.
- Quel est le but des heuristiques d'ordre d'instanciation des variables ?
 - Permettre de trouver une solution optimale au CSP.
 - Accélérer la recherche en instanciant d'abord les variables les plus critiques.
 - Filtrer les domaines des variables à chaque étape.
 - Éviter d'avoir à retourner en arrière lors de la résolution.
- Lors de l'instanciation d'une variable avec l'algorithme d'anticipation, quel est le niveau de consistance établi si on enlève du domaine des variables non instanciées les valeurs violant les contraintes binaires avec l'affectation partielle ?
 - Consistance de nœud
 - Consistance d'arc
 - Consistance de chemin
 - Consistance globale
- Qu'est-ce qu'un problème d'optimisation ? Qu'est-ce qu'une solution optimale ?

Exercice 1 (6 Pts)

Soit un sac à dos dans lequel on souhaite placer des objets à choisir parmi une liste de N objets. Chaque objet de la liste a un poids (en Kilogrammes) et une valeur (un prix). Il peut être ou non sélectionné pour être placé dans le sac. Le nombre d'objets placés dans le sac n'est pas limité, mais la somme de leurs poids ne doit pas excéder W , et la somme de leur valeur doit être au moins égale à V .

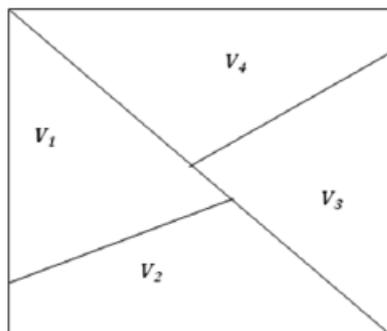
V , W ainsi que les poids et les valeurs des objets sont des données parfaitement connues lors de la résolution du problème. Donc, le solveur qui va résoudre ce problème ne doit pas décider quel poids ou quelle valeur correspond à chaque objet, ni la capacité du sac, ni la valeur minimum à transporter. Il se préoccupe de savoir quels objets mettre dans le sac.

- Modéliser cet énoncé sous forme d'un CSP.

Exercice 2 : (6 Pts)

Soit la carte suivante décrivant les frontières entre quatre villes (V_1, \dots, V_4). On voudrait colorier la carte en utilisant seulement les couleurs rouge, bleu et vert, de sorte que V_1 soit en rouge ou en vert; V_2 et V_3 soient en bleu ou en vert; et V_4 soit en vert. Toutefois, deux villes adjacentes ne peuvent avoir la même couleur.

- Donner la représentation graphique de ce CSP ?
- Donner le résultat de l'algorithme AC-3 sur ce problème. Que représente le AC3, expliquer en bref le principe derrière ce concept ? Où peut-on l'appliquer ?
- Quelle est la différence entre AC1 et AC3 ?



Examen Final de Programmation par Contraintes

Corrigé Type

Questions de Compréhension (8 Pts)

1. a (1.5 Pts)
2. b (1.5 Pts)
3. b (1.5 Pts)
4. b (1.5 Pts)
5. - Un problème d'optimisation est un type de problème où l'on cherche non seulement à satisfaire un ensemble de contraintes, mais aussi à optimiser (minimiser ou maximiser) une certaine fonction objective. (1 Pts)
 - Une solution optimale est donc une solution qui satisfait toutes les contraintes et qui atteint la valeur optimale (minimale ou maximale) de la fonction objective parmi toutes les solutions possibles du problème. (1 Pts)

Exercice 1 :

On a N objets, chaque objet i peut être modélisé par une variable X_i ayant un poids W_i et un prix V_i . Chaque objet a deux états possibles : « dans le sac » ou « pas dans le sac », donc le domaine de variation de X_i est $\{0,1\}$.

X_i étant une variable booléenne et W, V , les W_i et les V_i sont des constantes (2 Pts).

On a deux contraintes :

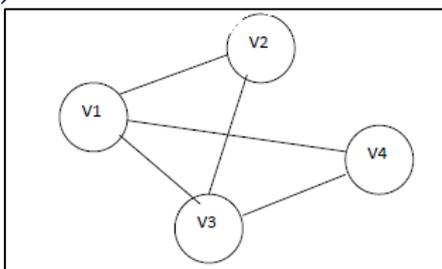
- La **somme des poids** des objets mis dans le sac ne doit pas dépasser W .
- La **somme des valeurs** des objets mis dans le sac doit être au moins égale à V .

Le CSP peut être exprimés comme suit CSP = (X,D,C) , tels que:

- $X = \{X_1, X_2, \dots, X_N\}$ (1 Pts)
- $D(X_1) = D(X_2) = \dots = D(X_N) = \{0,1\}$. (1 Pts)
- $C = \{C_1, C_2\}$ (2 Pts)
 - $C_1 : W_1 \cdot X_1 + W_2 \cdot X_2 + \dots + W_N \cdot X_N \leq W$
 - $C_2 : V_1 \cdot X_1 + V_2 \cdot X_2 + \dots + V_N \cdot X_N \geq V$

Exercice 2 :

1. Représentation graphique (1.5 Pts)



2. Variables et domaine des valeurs (1 pts)

$V_1 = \{r, v\}$, $V_2 = \{b, v\}$, $V_3 = \{b, v\}$, $V_4 = \{v\}$

Queue (ensemble des arcs de contrainte entre variables) = $\{ \langle V_1, V_1 \neq V_2 \rangle, \langle V_1, V_1 \neq V_4 \rangle, \langle V_1, V_1 \neq V_3 \rangle, \langle V_2, V_1 \neq V_2 \rangle, \langle V_2, V_2 \neq V_3 \rangle, \langle V_3, V_1 \neq V_3 \rangle, \langle V_3, V_3 \neq V_4 \rangle, \langle V_3, V_3 \neq V_2 \rangle, \langle V_4, V_1 \neq V_4 \rangle, \langle V_4, V_3 \neq V_4 \rangle \}$

Résultat final :

V1	V2	V3	V4	Contrainte
{r, v}	{b, v}	{b, v}	{v}	
{r}	{b, v}	{b, v}	{v}	V1 --- v4
{r}	{b, v}	{b}	{v}	V3 --- v4
{r}	{v}	{b}	{v}	V2 --- V3

Trace détaillée d'exécution (1 Pt)

- AC3 (Arc Consistency algorithm 3) est un algorithme permettant d'établir la consistance d'arc pour un problème de satisfaction de contraintes (CSP). (0.5 Pt)
- **Principe** : On commence par ajouter toutes les contraintes à une file d'attente, puis on boucle jusqu'à ce que la file soit vide. À chaque tour de boucle, on retire une contrainte de la file et on utilise l'algorithme Révise pour réduire les domaines des variables concernées. Si un domaine est modifié, on ajoute à la file toutes les contraintes qui impliquent cette variable, sauf la contrainte qu'on vient de traiter (0.5 Pt).
- **Application** : On peut l'appliquer pour le filtrage du domaine tel que dans l'algorithme d'anticipation. (0.5 Pt)

Examen Final de Programmation par Contraintes

3. **Différence entre AC1 et AC3** : La différence entre eux réside dans le fait que l'algorithme AC1 refait toute une passe en reconsidérant chacune des paires sur lesquelles il y a une contrainte. Tandis que, le AC3 ne réapplique la réduction qu'aux arcs dont le domaine de la variable extrémité a été modifié depuis la dernière révision. **(1 Pt)**