

Tableau récapitulatif des différents algorithmes de consistance

Name	Main ideas	Temporal Order	Space order	Comments
AC algorithms				
AC-1	recherche exhaustive... parcourt toutes les contraintes et ne s'occupent pas de savoir si il doit y avoir des changements suite à la suppression d'une valeur dans un domaine...	$O(n^3 d^3)$	-	Mauvais....;-)
AC-3	après enlèvement d'une valeur dans un domaine i (associé à la variable i), on regarde l'effet sur les domaines des variables k telles que $\exists C_{ki}$	$O(n^2 d^3)$	-	utilisation du module Queue.
AC-5	Deux boucles pour cet algorithme : – une première pour chaque contrainte C_{ij} réduisant les domaines une première sans aucune vérification – une deuxième où l'on regarde l'effet de l'enlèvement d'une valeur dans un domaine i <i>Longrightarrow</i> on utilise la fonction LocalArcCons qui se focalise sur une seule valeur enlevée d'un domaine et non pas comme AC-3 qui regardait le changement par rapport à un domaine entier	(voir comments)	-	utilisation du module Queue modifié tel qu'il permette de gérer l'insertion et la deletion des tuples de la forme : $\langle (i, j), w \rangle$. <u>Time Order</u> – LocalArcCons implémenté par ArcCons \implies AC-5 == AC-3 $\implies O(n^2 d^3)$ – LocalArcCons est en $O(d) \implies$ AC-5 == AC-4 $\implies O(n^2 d^2)$
AC-4	utilise une structure de données contenant un compteur pour chaque tuple $\langle (i, v), j \rangle$ telle que ce compteur égal le nombre d'éléments de j supportant $v \in \mathcal{D}_i$ pour le contrainte C_{ij} On va décrémenter ce compteur à chaque fois que l'on enlève une valeur supportant v . Quand le compteur est à zéro, on enlève v de \mathcal{D}_i	$O(n^2 d^2)$	-	Deux structures utilisées : – <i>counter</i> ($\langle (i, v), j \rangle$) tel que décrit avant – $S(\langle (i, v), j \rangle)$: ensemble de tous les éléments supportés par v dans $\mathcal{D}_j \implies$ permet un accès aux valeurs supportés quand v est enlevé de \mathcal{D}_i

Name	Main ideas	Temporal Order	Space order	Comments
AC-6		$O(n^2 d^2)$	-	même chose que AC-4 mais ne retient qu'un seul support (le plus petit)
AC-7		$O(n^2 d^2)$	-	même chose que AC-6 mais en exploitant la symétrie entre C_{ij} et C_{ji}
AC-2000		$O(n^2 d^3)$	-	assez proche de AC-3 mais réduit le nombre de check de contraintes.
AC-2001		$O(n^2 d^2)$	-	même que AC-4 (à mon avis) mais pas de liste de valeur supportées
DAC	Directional Arc Consistency Ordonner les variables...	$O(n^2 d^2)$	-	Un CSP est arc-directionnel consistant si C_{ij} est vrai pour tout i, j tels que $i < j$
Bound consistency	On réduit les domaines représentés ici comme des intervalles sur leurs bords.... On n'enlève que les valeurs extérieures au domaine, pas les valeurs internes \implies on ne coupera jamais un domaine en plusieurs morceaux...	-	-	moins de pruning que de l'arc consistence totale applicable a des contraintes n-aires
Path consistency	Le but : simplifier les contraintes. Soit $\langle v, w \rangle$ un couple de valeurs pour une contrainte C entre X_1 et X_2 . Si on enlève v et w de la contrainte et qu'on ne peut trouver un chemin satisfaisant pour aller de X_1 à X_2 , alors on enlèvera v et w de leur domaine respectif.	meilleur : $O(n^3 d^3)$	meilleur : $O(n^3 d^2)$	un triplet de variables (i, j, k) est chemin-consistant si $\forall v \in \mathcal{D}_i$ et $\forall w \in \mathcal{D}_j$ tels que $C_{ij}(v, w) == TRUE \implies \exists u \in \mathcal{D}_k$ tel que $C_{ik}(v, u) == TRUE$ and $C_{jk}(w, u) == TRUE$
k-consistency	Pour satisfaire la k-consistence, il faut que pour toute solution partielle de $(k-1)$ valeurs on puisse l'étendre à une solution partielle de k valeurs	$O(n^k d^k)$	$O(n^k d^k)$	nécessite de se souvenir de contraintes sur $(k-1)$ variables. Trop complexe en pratique pour le pruning

Name	Main ideas	Temporal Order	Space order	Comments
Strong k-consistency	Pour satisfaire la k-consistence, il faut que pour toute solution partielle de 1 valeurs on puisse l'étendre à une solution partielle de k valeurs	$O(nd)$	-	On parle de globally consistent si on peut passer d'une solution à 1 valeur à une solution à n valeurs. (ATTENTION : l'ordre temporel est pour le globally consistent!!!)
(k,m)-consistency	Pour satisfaire la k-consistence, il faut que pour toute solution partielle de k valeurs on puisse l'étendre à une solution partielle de $(k + m)$ valeurs			