

Systemes dynamiques et Relaxation

Narendra JUSSIEN
École des Mines de Nantes

Systemes dynamiques et relaxation ?

- Systemes dynamiques de contraintes
 - Interactions avec l'environnement
 - ajout / retrait de contraintes au cours de la résolution
 - maintien de propriété plutôt que recherche de solution
 - Maintien de propriété
 - grande probabilité de trouver un systeme sur-contraint
 - trouver **automatiquement** la ou les contraintes à remettre en cause
- Relaxation de contraintes
 - Prise en compte des **préférences** de l'utilisateur
 - Retrait de contraintes **incrémental**

Plan de la présentation

- Inspirations et précurseurs
- Notre cadre général pour la relaxation
- Une spécialisation : le système DECORUM
- Expérimentations

I. Inspirations et précurseurs

Des problèmes statiques
aux problèmes dynamiques

Problèmes statiques

- Résolution en deux temps
 - cohérence locale (filtrage)
 - une cohérence globale est trop coûteuse
 - bien souvent arc-consistance sur les CSP
 - énumération
 - donner des valeurs aux variables
 - basée sur le retour arrière chronologique
 - les deux temps sont bien sûr panachés !

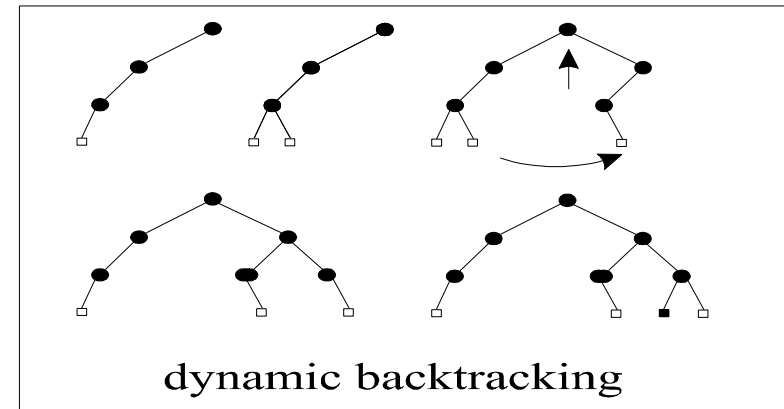
Améliorations

- Travail sur le filtrage
 - autres cohérences locales
 - algorithmes toujours améliorés (AC[4..7] ...)
 - spécialisations
- Travail sur l'énnumération
 - l'énnumération est un problème dynamique !
 - du *backtrack intelligent*
 - ... à *dynamic backtracking* (Ginsberg 1993)



Dynamic backtracking (Ginsberg 1993)

- Approche par sauts
- Utilisation des *eliminating explanations*



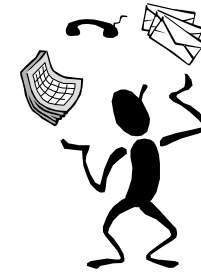
- Particularités
 - Backtrack chronologique : c'est facile
 - toutes les contraintes ont la même importance et la même relation avec l'échec courant
 - on enlève donc la dernière mise en place
 - Dynamic backtracking
 - certaines contraintes n'interviennent pas dans l'échec courant
 - on enlève la plus récente des responsables
 - on conserve ce qui n'a pas de rapport avec l'échec courant

Énumération et systèmes dynamiques

- Généraliser à tout type de contrainte
 - prendre en compte le filtrage
 - définition d'un système d'explications : double rôle
 - identification des responsabilités
 - gestion de l'incrémentalité
- En cas de conflit : toutes les contraintes ne sont pas identiques
 - prendre en compte les préférences de l'utilisateur par l'intermédiaire d'un critère

II. Un cadre général pour la relaxation

Les données



- Un domaine de calcul
 - Rationnels, Intervalles, Domaines finis ...
- Une propriété à maintenir
 - Cohérence locale, existence d'une solution, exhibition d'une solution à tout instant, ...
- Un ensemble de contraintes évolutif
 - Contraintes linéaires, contraintes quelconques ...
- Un critère reflétant les préférences
 - Minimiser le poids des contraintes relaxées, ...
 - On se donne un **comparateur**

De nouvelles notions

- Notion d'**explication de contradiction**
 - sous-ensemble contradictoire des contraintes actives
- Notion de **configuration**
 - partition des contraintes : (actives/relaxées)
 - configuration **prometteuse** : configuration ne contenant aucune explication de contradiction connue *i.e.* l'ensemble des contraintes relaxées *recouvre* les explications de contradiction
- Notion de **solution**
 - meilleure (*au sens du comparateur*) configuration satisfiable (*au sens de la propriété à vérifier*)

Un algorithme générique

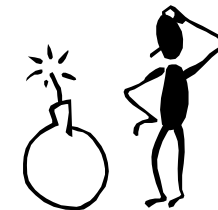
tantque C est contradictoire
calculer une nouvelle explication de contradiction
 C' est la meilleure configuration prometteuse
passer de C à C'

fintantque

- Trois fonctions paramètres
 - explique-contradiction
 - meilleure-configuration-prometteuse
 - maj-configuration

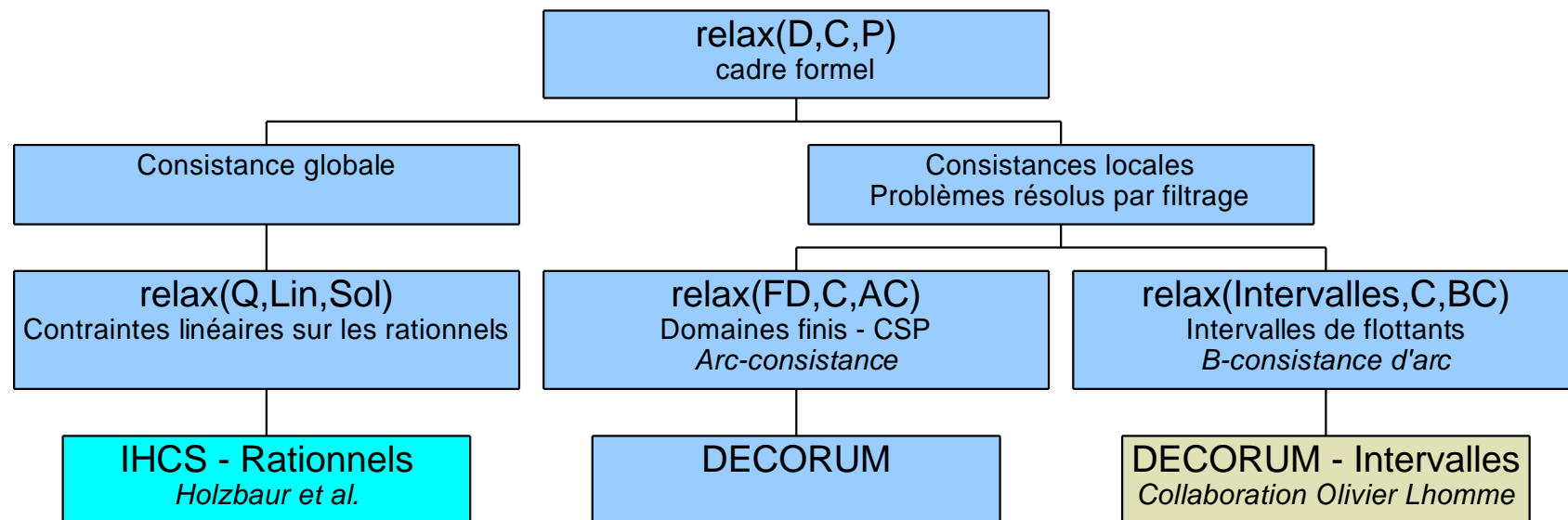
Intervention du comparateur fonction meilleure-configuration-prometteuse

- Problème de recouvrement optimal
 - NP-difficile au sens fort dans le cas général
 - Utilisation possible d'heuristiques
- Comparateur *contradiction-local*
 - non conservation des explications



*La plus importante des contraintes relaxées dans C1
est moins importante que la plus importante des
contraintes relaxées dans C2*

Déclinaisons du cadre général



III. Le système DECORUM

Le système DECORUM

- Domaine de calcul : domaines finis
- Propriété à maintenir : arc-cohérence
 - y compris bien sûr pendant l'énumération
- Contraintes : quelconques
 - en intension ou en extension
- Comparateur utilisé : contradiction-local

Un système d'explications

- Deux objectifs
 - Déterminer rapidement et précisément une explication de contradiction
 - Permettre un retrait incrémental efficace des contraintes
- Une explication pour chaque *retrait*
 - Pour le retrait de a du domaine de X :
 - Ensemble de contraintes actives devenant contradictoire si $(X=a)$ lui est adjoint
- Un algorithme d'arc-consistance avec explications

Fournir des explications

- La plus simple ... les contraintes actives
- Utilisation d'un algorithme de point fixe

tantque queue non vide

$c \leftarrow \text{pop}(\text{queue})$

réduire les domaines selon c

si un domaine est modifié **alors** push(queue, toute contrainte liée)

fantantque

- Utilisation de la sémantique des contraintes
 - exemple : $X > Y$
 - le min de X ne dépend que du min Y
 - le max de Y ne dépend que du max de X

Explication de contradiction = trace du savoir-faire du solveur

Relax(FD, C, AC)

- explique-contradiction
 - domaine vide : explication = union des explications de retrait de chacune des valeurs
- meilleure-configuration-prometteuse
 - utilisation du comparateur
- maj-configuration
 - utilisation des explications

tantque C est contradictoire **faire**

calculer explique-contradiction

C' ← meilleure-configuration-prometteuse

maj-configuration(C, C')

fintantque

Retrait incrémental de contrainte

- Deux étapes
 - déterminer les actions passées de la contrainte considérée et les défaire
 - rétablissement de l'arc-consistance
- Généralisation d 'algorithmes classiques d 'arc-consistance dynamique (DnAC*)
 - le système d'explications utilisé est plus précis mais plus coûteux en espace
 - la première étape est instantanée

Complexités

- Données
 - n variables de domaine de taille maximale d
 - e contraintes
 - un comparateur contradiction-local
- Complexité spatiale
 - stockage des explications de retraits $O((n \times d + e) \times (n + e))$
- Complexité temporelle
 - ajout/retrait de contrainte $O(d \times c_{AC})$
 - traitement de la contradiction $O(n + d + e)$

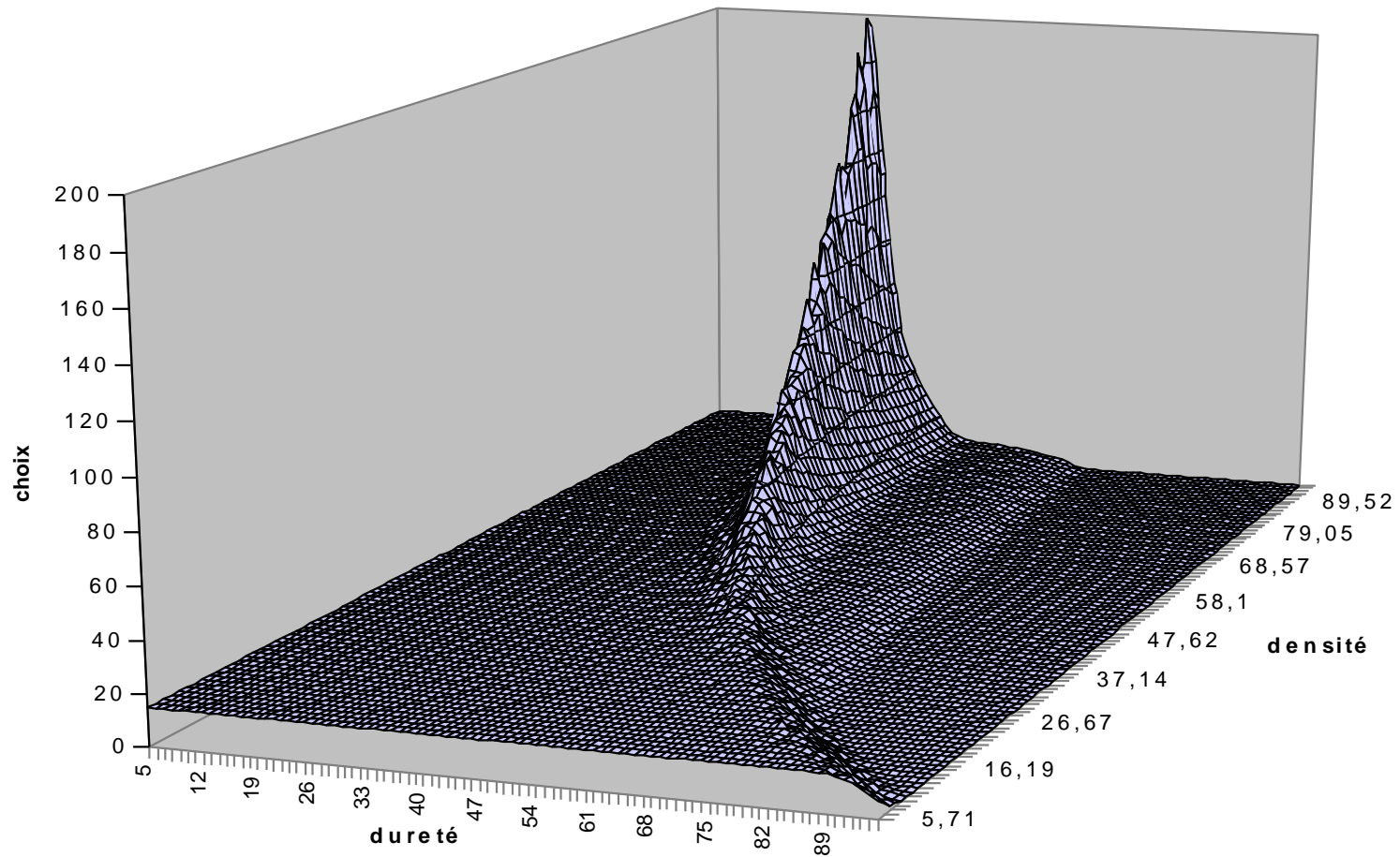
IV. Expérimentations

Résolution de CSP aléatoires

- Les contraintes sont introduites une à une
- Approche classique
 - une solution vérifiant toutes les contraintes ou preuve de non existence
- Notre approche
 - toute les contraintes ont la même importance
 - une solution vérifiant toutes les contraintes ou la meilleure configuration satisfaisable
- L'utilisation de DECORUM ne modifie pas les résultats classiques

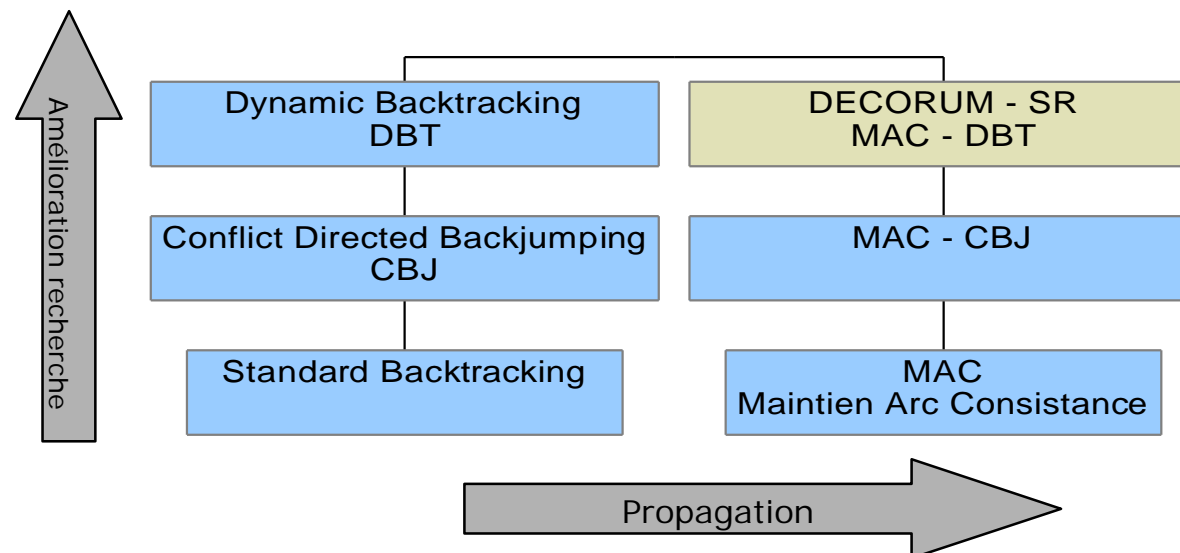
Conservation de la transition de phase

Problèmes 15x10



Un cas particulier de DECORUM

- Limitation de la relaxation aux seules contraintes d'énumération
- Une nouvelle technique d'énumération
 - *dynamic backtracking* + arc consistance



L 'algorithme

tantque il existe une variable v non instanciée

$i \leftarrow$ élément de D_v

si ajout($v=i$) contradictoire **alors**

$E \leftarrow$ explication de contradiction

relaxePlusRécente(E)

relaxePlusRécente(E) :

$c \leftarrow$ contrainte plus récente de E

si retrait($c, E \setminus \{c\}$) contradictoire **alors**

$E \leftarrow$ explication de contradiction

relaxePlusRécente(E)

Expérimentations

- Problèmes complètement aléatoires
 - léger surcoût
 - le nouvel algorithme n'apporte rien
- Problèmes *structurés*
 - immersion d'un sous-problème sur-contraint dans un grand problème sous-contraint (Bayardo 1994)
 - **bien meilleurs résultats que MAC** (Sabin et Freuder)
 - quand MAC ne donne pas de résultat au bout d'une semaine de calcul, DECORUM répond en 18 secondes !!



V. Conclusion et perspectives

Contributions

- Un cadre générique pour la résolution de problèmes dynamiques basé sur la relaxation
 - instances pour des consistances globales
 - instances pour des consistances locales
 - le système DECORUM
- Un nouvel algorithme d'énumération
 - adapté à la résolution de problèmes structurés
 - application possible aux intervalles (*cf.* ECAI 98, SCAN 98)
 - application possible pour améliorer les recherches arborescentes de la RO (*cf.* JFPLC 98 et PMS 98)

Perspectives

- Champ d'application de ces travaux
 - résolution de problèmes réels (donc dynamiques) structurés
 - allocation de ressources dans un contexte réparti
- Évolutions
 - autres comparateurs
 - relaxation incomplète ou douce
- Utilisation des systèmes d'explications
 - nouveaux algorithmes de recherche
 - débogage d'applications contraintes